UNIVERSIDAD NACIONAL DE UCAYALI

FACULTAD DE INGENIERÍA DE SISTEMAS E INGENIERÍA CIVIL

CARRERA PROFESIONAL INGENIERÍA DE SISTEMAS

"SISTEMA DE GESTIÓN ACADÉMICA EN EL PROCESO DE MATRÍCULA DE LA UNIVERSIDAD NACIONAL DE UCAYALI: 2019"

TESIS PARA OPTAR POR EL TÍTULO PROFESIONAL DE

INGENIERO DE SISTEMAS

TRIGUEROS BELLIDO JORGE EDUARDO

ASESOR: Dr. JORGE LUIS HILARIO RIVAS

PUCALLPA – PERÚ
2020

UNIVERSIDAD NACIONAL DE UCAYALI

FACULTAD DE INGENIERÍA DE SISTEMAS Y DE INGENIERÍA CIVIL
ESCUELA ACADÉMICO DE INGENIERÍA DE SISTEMAS

INFORME DE ASESORÍA DE TESIS

TESISTA : TRIGUEROS BELLIDO, JORGE EDUARDO

TíTULO : "SISTEMA DE GESTIÓN ACADÉMICA EN EL PROCESO DE MATRÍCULA DE LA UNIVERSIDAD NACIONAL DE UCAYALI: 2019"

REFERENCIA: RESOLUCION DE COMITÉ DE PLANEAMIENTO N° 692/2019-CP-FISeIC-UNU del 25 de octubre de 2019

Que, el tesista TRIGUEROS BELLIDO, JORGE EDUARDO, ha cumplido con ejecutar la tesis titulada: SISTEMA DE GESTIÓN ACADÉMICA EN EL PROCESO DE MATRÍCULA DE LA UNIVERSIDAD NACIONAL DE UCAYALI: 2019, de conformidad con el Reglamento de Grados y Títulos de la Universidad Nacional de Ucayali. Siendo así, fue evaluada, el borrador de tesis de 307 páginas, en el sistema informático de antiplagio URKUN con 1% de SIMILITUD, estando dentro de los limites permitidos, por el indicado Reglamento. Por lo que:

El suscrito, declara APROBADA, en la etapa de asesoría de tesis, encontrándose apta para ser presentada y evaluada por la Comisión de Grados y Títulos de la Facultad de Ingeniería de Sistemas e Ingeniería Civil, de acuerdo a lo que establece su reglamento.

Se expide el presente documento, a solicitud del interesado, para fines exclusivamente de trámite en el proceso de sustentación de la indicada tesis.

Pucallpa, 20 de agosto de 2020.

Dr. JORGE LUIS HILARIO RIVAS
Docente ordinario Asoc/TC
FIS e IC – UNU

FIRMA DIGITAL

Firmado digitalmente por: HILARIO RIVAS Jorge Luis FIR 22481183 hard
Motivo: Soy el autor del documento
Fecha: 20/08/2020 19:01:50-0500

UNIVERSIDAD NACIONAL DE UCAYALI

FACULTAD DE INGENIERIA DE SISTEMAS E INGENIERIA CIVIL

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

COMISION DE GRADOS Y TITULOS

ACTA DE EVALUACION DE SUSTENTACION DE TESIS N° 012-2020

GRADUANDO : Bach. TRIGUEROS BELLIDO, JORGE EDUARDO

TEMA : "SISTEMA DE GESTION ACADEMICA EN LOS PROCESOS DE MATRICULA DE LA UNIVERSIDAD NACIONAL DE UCAYALI"

ASESOR : DR. JORGE LUIS HILARIO RIVAS

CRITERIOS	PUNTAJE			
	3	2	1	0
I- PRESENTACION				
* MOTIVACION	X			
* TONO DE VOZ	X			
* CALIDAD DE MATERIAL AUDIOVISUAL			X	
II- DESARROLLO DE CONTENIDO				
* SECUENCIA		X		
* DOMINIO DE VOZ		X		
* USO ADECUADO DE MATERIAL		X		
* VALIDEZ DE LOS HALLAZGOS			X	
III- ABSOLUCION DE PREGUNTAS				
* RESPONDE A LAS PREGUNTAS FORMULADAS POR EL JURADO		X		
IV- APORTE CIENTIFICO Y TECNOLOGICO				
* AL DESARROLLO DE LA COMUNIDAD		X		
* A LA PROFESION		X		

EVALUACION

- EXCELENCIA 26 A 30 PUNTOS
- UNANIMIDAD 21 A 25 PUNTOS
- MAYORIA 16 A 20 PUNTOS
- DESAPROBADO 15 A MENOS

CALIFICACION FINAL:

PUNTAJE : 20 /30

CONCLUSIONES:

APROBADO POR : MAYORIA

DESAPROBADO POR :

JURADO EVALUADOR

- ING. MG. ARTURO YUPANQUI VILLANUEVA — Presidente
- ING. MG. CESAR AUGUSTO AGURTO CHERRE — Miembro
- ING. MG. DIANA MARGARITA DIAZ ESTRADA — Miembro

SECRETARIO ACADÉMICO: ING. MSC. CLOTILDE RÍOS HIDALGO DE CERNA

PUCALLPA, 26 DE NOVIEMBRE DEL 2020

“Sistema de Gestión Académica en el Proceso de Matrícula de la Universidad Nacional de Ucayali: 2019”

POR: Bach. JORGE EDUARDO TRIGUEROS BELLIDO

APROBADO:

……

Ing. Mg. Arturo Yupanqui Villanueva

Presidente

……

Ing. Mg. Cesar Augusto Agurto Cherre

Miembro

……...

Ing. Mg. Diana Margarita Diaz Estrada

Miembro

ASESOR:

……

Dr. Jorge Luis Hilario Rivas

Asesor

UNIVERSIDAD NACIONAL DE UCAYALI
VICERRECTORADO DE INVESTIGACION
DIRECCION GENERAL DE PRODUCCION INTELECTUAL

CONSTANCIA

ORIGINALIDAD DE TRABAJO DE INVESTIGACION

SISTEMA ANTIPLAGIO URKUND

N° V/0267-2020

La **Dirección General de Producción Intelectual**, hace constar por la presente, que el Informe Final (Tesis), Titulado:

"SISTEMA DE GESTIÓN ACADÉMICA EN EL PROCESO DE MATRÍCULA DE LA UNIVERSIDAD NACIONAL DE UCAYALI: 2019"

Cuyo Autor (es)	:	**TRIGUEROS BELLIDO, JORGE EDUARDO**
Facultad	:	INGENIERIA DE SISTEMAS Y DE INGENIERIA CIVIL
Escuela profesional	:	INGENIERIA DE SISTEMAS
Asesor(a)	:	Dr. Hilario Rivas, Jorge Luis

Después de realizado el análisis correspondiente en el Sistema Antiplagio URKUND, dicho documento presenta un **porcentaje de similitud de 01%.**
En tal sentido, de acuerdo a los criterios de porcentaje establecidos en el artículo 9 de la DIRECTIVA DE USO DEL SISTEMA ANTIPLAGIO URKUND, el cual indica que no se debe superar el 10%. Se declara, que el trabajo de investigación: SI Contiene un porcentaje aceptable de similitud, por lo que SI se aprueba su originalidad.

En señal de conformidad y verificación se entrega la presente constancia.

Fecha: 11/11/2020

Dra. DINA PARI QUISPE
Dirección General de Producción Intelectual

UNIVERSIDAD NACIONAL DE UCAYALI

OEByP - REPOSITORIO INSTITUCIONAL

AUTORIZACIÓN DE PUBLICACION DE TESIS

REPOSITORIO DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Yo, JORGE EDUARDO TRIGUEROS BELLIDO

Autor(a) de la TESIS de pregrado titulada:

SISTEMA DE GESTIÓN ACADÉMICA EN LOS PROCESOS DE MATRICULA DE MATRICULA DE LA UNIVERSIDAD NACIONAL DE UCAYALI

Sustentada el año: 2020

Con la asesoría de: DR. JORGE LUIS HILARIO RIVAS

En la Facultad: DE INGENIERIA DE SISTEMAS Y DE INGENIERIA CIVIL

Escuela profesional: INGENIERIA DE SISTEMAS

Autorizo la publicación:

PARCIAL [ ] Significa que se publicará en el repositorio institucional solo La caratula, la dedicatoria y el resumen de la tesis. Esta opción solo es válida marcar **si su tesis o documento presenta material patentable**, para ello deberá presentar el trámite de CATI y/o INDECOPI cuando se lo solicite la DGPI UNU.

TOTAL [X] Significa que todo el contenido de la tesis y/o documento será publicada en el repositorio institucional.

De mi trabajo de investigación en el Repositorio Institucional de la Universidad Nacional de Ucayali (www.repositorio.unu.edu.pe), bajo los siguientes términos:

Primero: Otorgo a la Universidad Nacional de Ucayali **licencia no exclusiva** para reproducir, distribuir, comunicar, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público en general mi tesis (incluido el resumen) a través del Repositorio Institucional de la UNU, en formato digital sin modificar su contenido, en el Perú y en el extranjero; por el tiempo y las veces que considere necesario y libre de remuneraciones.

Segundo: Declaro que la **tesis es una creación de mi autoría** y exclusiva titularidad, por tanto me encuentro facultado a conceder la presente autorización, garantizando que la tesis no infringe derechos de autor de terceras personas, caso contrario, me hago único(a) responsable de investigaciones y observaciones futuras, de acuerdo a lo establecido en el estatuto de la Universidad Nacional de Ucayali y del Ministerio de Educación.

En señal de conformidad firmo la presente autorización.

Fecha: 01 / 12 / 2020

Email: JTRIGUEROS82@GMAIL.COM
Teléfono: 938131224

Firma:
DNI: 41395605

DEDICATORIA

Este trabajo es dedicado con todo mi corazón principalmente a Dios, ya que sin la gracia de él nada de esto sería posible, a mis padres que con amor, esmeros y sacrificios han guiado mi vida y me han apoyado siempre y a mis profesores que con sus enseñanzas y experiencias han aportado a mi formación profesional.

Jorge Trigueros B.

AGRADECIMIENTO

A Dios por estar conmigo en todo momento, por darme las fuerzas necesarias y las ganas de seguir adelante; a mis padres quienes con tanto amor me han apoyado siempre, a mi hijo por brindarme su sonrisa, a mi hermana y amigos por el sólo hecho de existir en mi vida.

Jorge Trigueros B.

TABLA DE CONTENIDO

Nro. **Descripción** **Pág.**

DEDICATORIA vii

AGRADECIMIENTO viii

LISTA DE ILUSTRACIONES xv

LISTA DE TABLAS xxi

RESUMEN xxv

ABSTRACT xxvi

INTRODUCCION xxvii

CAPITULO I 1

PLANTEAMIENTO DEL PROBLEMA 1

1.1. Planteamiento del problema de investigación 1

1.2. Causas del problema 5

1.3. Efectos del problema 6

1.4. Formulación del problema 8

1.4.1. Problema general: 8

1.4.2. Problemas específicos: 8

1.5. Justificaciones 9

1.5.1. Justificación practica 9

1.5.2. Justificación teórica 9

1.5.3. Justificación metodológica 10

1.6. Objetivos de la investigación 10

1.6.1. Objetivo general 10

1.6.2. Objetivos específicos 11

CAPITULO II 12

MARCO TEÓRICO 12

2.1. Antecedentes del problema 12

2.2. Planteamientos teóricos 15

2.2.1. Sistema web 15

2.2.2. Sistema de Información 16

2.2.3. Tecnologías de información en empresas 19

2.2.4. Lenguaje de programación 21

2.2.5. Programación Orientada a Objetos 23

2.2.6. Metodología SCRUM 23

2.2.7. JAVA 25

2.2.8. Primefaces 26

2.2.9. Hibernate 26

2.2.10. Servidor de Base de datos 26

2.2.11. Gestor de base de datos Oracle 27

2.2.12. Arquitectura Cliente/Servidor 28

2.2.13. PowerDesigner 28

2.2.14. Proceso de Matrícula 30

2.2.15. Requisitos en el proceso de matrícula 30

2.3. Definición de términos básicos 31

CAPITULO III 33

METODOLOGÍA DE LA INVESTIGACIÓN 33

3.1. Método de investigación 33

3.1.1. Nivel de investigación 34

3.1.2. Diseño de la investigación: Pre – Experimental 34

3.2. Población y muestra 35

3.2.1. Población 35

3.2.2. Muestra 35

3.3. Instrumentos de recolección de datos 37

3.4. Instrumentos para la recolección de datos 37

3.5. Tratamiento de datos 37

CAPITULO IV 38

METODOLOGÍA DE DESARROLLO DEL SISTEMA INFORMÁTICO 38

4.1. Normas y Roles del proyecto 38

4.2.1. Normas Internas 38

4.2.2. Roles del Proyecto 39

4.3. Historia de usuarios 41

4.3.1. Descripción de Historia de Usuarios 43

4.3.2. Lista de historias de usuario por orden de importancia (Product Backlog). 53

4.3.3. Definición de los Sprints 54

4.4. Planificación de los Sprints 57

4.5. Taskboard inicial 62

4.6. Desarrollo de los Sprints 63

4.6.1. Sprint 1 63

4.6.2. Sprint 2 81

4.6.3. Sprint 3 96

4.6.4. Sprint 4 113

4.7. Pruebas de entregables 123

4.7.1. Pruebas de autentificación de usuarios 123

4.7.2. Pruebas de gestión de unidades 123

4.7.3. Pruebas de gestión de docentes 124

4.7.4. Pruebas de gestión de estudiantes.....124

4.7.5. Pruebas de gestión de cursos 125

4.7.6. Pruebas de gestión de perfiles 125

4.7.7. Pruebas de gestión de planes académicos.....126

4.7.8. Pruebas de gestión de registrar matricula.....126

4.7.9. Pruebas de gestión pagos 128

4.8. Integración con otros sprints 128

4.8.1. Integración SPRINT 1 y SPRINT 2 128

4.8.2. Integración SPRINT 1, 2 y 3 129

4.8.3. Integración SPRINT 1, 3 y 4 130

4.9. Implantación del sistema.....131

4.9.1. Sistema operativo.....132
4.9.2. Software de aplicación.....132
4.9.3. Hardware.....132
4.9.4. Red.....132
4.9.5. Instalación de la aplicación.....133
4.10. Reportes de información para la toma de decisiones.....133
CAPITULO V.....135
RESULTADOS Y DISCUSIÓN.....135
5.1. INTERPRETACIÓN DE RESULTADOS.....135
5.1.1. PRE TEST.....135
5.1.2. POS TEST.....144
5.2. PRUEBA DE HIPÓTESIS.....153
5.2.1. Paso 1: Formulación de la hipótesis.....153
5.2.2. Paso 2: Especificar el nivel de significancia.....153
5.2.3. Paso 3: Selección de estadística a docimar.....153
5.2.4. Paso 4: Establecer criterios de decisión.....154
5.2.5. Paso 5: Realización de cálculos.....154
5.2.6. Paso 6: Decisión estadística.....156
CONCLUSIONES.....157
RECOMENDACIONES.....159
REFERENCIAS BIBLIOGRAFICAS.....160
ANEXOS.....163

Anexo 1. MATRIZ DE CONSISTENCIA.....164

Anexo 2. CUADRO DE OPERACIONALIZACION DE LAS VARIABLES 165

Anexo 3. CUESTIONARIO DE PREGUNTAS.....166

Anexo 4. INSTALACIÓN DE ORACLE V11g (SERVIDOR) 167

Anexo 5. INSTALACIÓN DE ORACLE V11g (CLIENTE).....180

Anexo 6. CÓDIGO FUENTE.....193

Anexo 7. REUNIÓN DE TRABAJOS – TASKBOARD.....275

LISTA DE ILUSTRACIONES

Ilustración 1. Ubicación Geográfica de la Universidad Nacional de Ucayali 2

Ilustración 2. Alumnos Matriculados a Nivel Nacional 3

Ilustración 3. Alumnos Matriculados en la Universidad Nacional de Ucayali 4

Ilustración 4. Diseño Conceptual de un Sistema de Información 17

Ilustración 5. Arquitectura Cliente/Servidor 28

Ilustración 6. Interfaz de PowerDesigner 29

Ilustración 7. Prototipo Autentificación de Usuario 65

Ilustración 8. Prototipo Buscar Usuario 65

Ilustración 9. Prototipo Modificar Usuario 66

Ilustración 10. Prototipo Nuevo Usuario 66

Ilustración 11. Prototipo Eliminar Usuario 67

Ilustración 12. Prototipo Nuevo Perfil 67

Ilustración 13. Prototipo Eliminar Perfil 68

Ilustración 14. Prototipo Listar Perfil 68

Ilustración 15. Prototipo Modificar Perfil 69

Ilustración 16. Prototipo Asignación Perfil a usuario 69

Ilustración 17. Prototipo Asignación Perfil a usuario 70

Ilustración 18. Modelo Físico Perfil a usuario 71

Ilustración 19. Interfaz de acceso al sistema 72

Ilustración 20. Interfaz de buscar usuario 72

Ilustración 21. Interfaz de modificar usuario 73

Ilustración 22. Interfaz de nuevo usuario 73

Ilustración 23. Interfaz de eliminar usuario 74

Ilustración 24. Interfaz de nuevo perfil 74

Ilustración 25. Interfaz de eliminar perfil 75
Ilustración 26. Interfaz de listar perfil 75
Ilustración 27. Interfaz de modificar perfil 76
Ilustración 28. Interfaz de asignación, modificación y deshabilitación de perfil a usuario 76
Ilustración 29. Interfaz de asignación, modificación y deshabilitación de perfil a usuario 77
Ilustración 30. Prototipo Nueva Facultad 82
Ilustración 31. Prototipo Modificar Facultad 82
Ilustración 32. Prototipo Listar Facultad 82
Ilustración 33. Prototipo Eliminar Facultad 83
Ilustración 34. Prototipo Nueva Carrera Profesional 83
Ilustración 35. Prototipo Modificar Carrera Profesional 83
Ilustración 36. Prototipo Listar Carrera Profesional 84
Ilustración 37. Prototipo Eliminar Carrera Profesional 84
Ilustración 38. Prototipo Nuevo Docente 85
Ilustración 39. Prototipo Modificar Docente 85
Ilustración 40. Prototipo Listar Docente 85
Ilustración 41. Modelo físico Sprint 2 86
Ilustración 42. Interfaz Nueva Facultad 87
Ilustración 43. Interfaz Modificar Facultad 87
Ilustración 44. Interfaz Listar Facultad 88
Ilustración 45. Interfaz Eliminar Facultad 88
Ilustración 46. Interfaz Nueva Carrera Profesional 89
Ilustración 47. Interfaz Modificar Carrera Profesional 89

Ilustración 48. Interfaz Listar Carrera Profesional 89
Ilustración 49. Interfaz Eliminar Carrera Profesional 90
Ilustración 50. Interfaz Nuevo Docente 90
Ilustración 51. Interfaz Modificar Docente 91
Ilustración 52. Interfaz Listar Docente 91
Ilustración 53. Interfaz Buscar Docente 91
Ilustración 54. Interfaz de asignación, modificación y deshabilitación de perfil a usuario 92
Ilustración 55. Prototipo Nuevo Plan Académico 97
Ilustración 56. Prototipo Modificar Plan Académico 97
Ilustración 57. Prototipo Listar Plan Académico 98
Ilustración 58. Prototipo Eliminar Plan Académico 98
Ilustración 59. Prototipo Nuevo Curso Académico 99
Ilustración 60. Prototipo Modificar Curso Académico 99
Ilustración 61. Prototipo Listar Cursos Académicos 100
Ilustración 62. Prototipo Eliminar Curso Académico 100
Ilustración 63. Prototipo Nuevo Sistema de Evaluación 101
Ilustración 64. Prototipo Modificar Sistema de Evaluación 101
Ilustración 65. Prototipo Listar Sistema de Evaluación 102
Ilustración 66. Prototipo Eliminar Sistema de Evaluación 102
Ilustración 67. Modelo físico Sprint 3.....103
Ilustración 68. Interfaz nuevo plan académico.....104
Ilustración 69. Interfaz modificar plan académico 104
Ilustración 70. Interfaz listar plan académico.....105
Ilustración 71. Interfaz eliminar plan académico 105

Ilustración 72. Interfaz nuevo curso académico.....106
Ilustración 73. Interfaz modificar curso académico.....106
Ilustración 74. Interfaz listar curso académico.....107
Ilustración 75. Interfaz eliminar curso académico.....107
Ilustración 76. Interfaz nuevo sistema de evaluación 108
Ilustración 77. Interfaz modificar sistema de evaluación.....108
Ilustración 78. Interfaz listar sistema de evaluación 108
Ilustración 79. Interfaz eliminar sistema de evaluación.....109
Ilustración 80. Interfaz de asignación, modificación y deshabilitación de perfil a usuario.....109
Ilustración 81. Prototipo Registro de Asignación de Cursos (matricula) 114
Ilustración 82. Prototipo Generación de tasa educacional web 114
Ilustración 83. Prototipo Generación de boleta de matrícula web.....115
Ilustración 84. Prototipo Generación Reporte Boleta de Notas Web 115
Ilustración 85. Prototipo Generación Reporte de Datos del Estudiante Web..116
Ilustración 86. Prototipo Reporte Avance Curricular Web.....116
Ilustración 87. Prototipo Creación, lectura de generación de pagos para el Banco.....117
Ilustración 88. Prototipo Recepción data de Banco para generación de pagos de estudiantes.....117
Ilustración 89. Modelo Físico Spring 4 118
Ilustración 90. Interfaz de asignación, modificación y deshabilitación de perfil a usuario.....119
Ilustración 91: Registro de alumnos matriculados – data histórica 133
Ilustración 92: % de Matriculados - Virtual vs Presencial 134

Ilustración 93: REPORTE DE MATRICULAS Y MONTOS DE DEUDA.....134
Ilustración 94. PREG_01: ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?.....135
Ilustración 95: PREG_02: ¿Cómo califica las interfaces del SIGA?.....136
Ilustración 96: PREG_03: ¿Cómo califica el acceso al SIGA? 137
Ilustración 97: PREG_04: ¿Considera importante la actualización de nuevas versiones? 138
Ilustración 98: PREG_05: ¿Considera que los datos son seguros en el SIGA? 139
Ilustración 99: PREG_06: ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali?.....140
Ilustración 100: PREG_07: ¿Considera que los procesos de matrícula son tediosos para alumnos? 141
Ilustración 101: PREG_08: ¿Qué opina del tiempo que se demoran los alumnos para matricularse? 142
Ilustración 102: PREG_09: ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula?.....143
Ilustración 103: PREG_01: ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?.....144
Ilustración 104: PREG_02: ¿Cómo califica las interfaces del SIGA?.....145
Ilustración 105: PREG_03: ¿Cómo califica el acceso al SIGA? 146
Ilustración 106: PREG_04: ¿Considera importante la actualización de nuevas versiones? 147
Ilustración 107: PREG_05: ¿Considera que los datos son seguros en el SIGA? 148

Ilustración 108: PREG_06: ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali? 149
Ilustración 109: PREG_07: ¿Considera que los procesos de matrícula son tediosos para alumnos? 150
Ilustración 110: PREG_08: ¿Qué opina del tiempo que se demoran los alumnos para matricularse? 151
Ilustración 111: PREG_09: ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula? 152
Ilustración 112: Comparación entre los resultados de pretest y postest por pregunta 154
Ilustración 113: Grafico q-q normal de pretest 155
Ilustración 114: Grafico q-q normal de postest 155

LISTA DE TABLAS

Tabla 1. Resumen de matriculados por escuela en el periodo 2018 35

Tabla 2. Muestra estratificación por escuela profesional 36

Tabla 3. Roles según SCRUM 39

Tabla 4. Historia de Usuario HUCR01 43

Tabla 5. Historia de Usuario HUCR02 43

Tabla 6. Historia de Usuario HUCR03 44

Tabla 7. Historia de Usuario HUCR04 44

Tabla 8. Historia de Usuario HUCR05 45

Tabla 9. Historia de Usuario HUCR06 45

Tabla 10. Historia de Usuario HUCR07 46

Tabla 11. Historia de Usuario HUCR08 46

Tabla 12. Historia de Usuario HUCR09 47

Tabla 13. Historia de Usuario HUCR10 47

Tabla 14. Historia de Usuario HUCR11 48

Tabla 15. Historia de Usuario HUCR12 48

Tabla 16. Historia de Usuario HUCR13 49

Tabla 17. Historia de Usuario HUCR14 49

Tabla 18. Historia de Usuario HUCR15 50

Tabla 19. Historia de Usuario HUCR16 50

Tabla 20. Historia de Usuario HUCR17 51

Tabla 21. Historia de Usuario HUCR18 51

Tabla 22. Historia de Usuario HUCR19 52

Tabla 23. Historia de Usuario HUCR20 52

Tabla 24. Tareas por orden de prioridad (product backlog) 53

Tabla 25. tabla de días de trabajo dedicado por el equipo por cada sprint 54
Tabla 26. tabla de estimación de sprint 1 55
Tabla 27. tabla de estimación de sprint 2 56
Tabla 28. tabla de estimación de sprint 3 56
Tabla 29. tabla de estimación de sprint 4 57
Tabla 30. planificación del sprint 1 58
Tabla 31. planificación del sprint 2 59
Tabla 32. planificación del sprint 3 60
Tabla 33. planificación del sprint 4 61
Tabla 34. taskboard inicial del desarrollo 62
Tabla 35. Especificación SPRINT 1 64
Tabla 36. informe de prueba funcional – sprint 1..... 78
Tabla 37. informe final de prueba funcional – sprint 1 79
Tabla 38. informe final de reunión retrospectiva – sprint 1 80
Tabla 39. Especificación SPRINT 2 81
Tabla 40. informe de prueba funcional – sprint 2..... 93
Tabla 41. informe final de prueba funcional – sprint 2 94
Tabla 42. informe final de reunión retrospectiva – sprint 2 95
Tabla 43. Especificación SPRINT 3 96
Tabla 44. informe de prueba funcional – sprint 3.....110
Tabla 45. informe final de prueba funcional – sprint 3 111
Tabla 46. informe final de reunión retrospectiva – sprint 3 112
Tabla 47. Especificación SPRINT 4 113
Tabla 48. Informe de prueba funcional – sprint 4 120
Tabla 49. informe final de prueba funcional – sprint 4 121

Tabla 50. informe final de reunión retrospectiva – sprint 4 122
Tabla 51. Prueba de autentificación de usuario.....123
Tabla 52. Prueba de gestión de unidades 123
Tabla 53. Prueba de gestión de docentes 124
Tabla 54. Prueba de gestión de estudiantes 124
Tabla 55. Prueba de gestión de cursos 125
Tabla 56. Prueba de gestión de perfiles 125
Tabla 57. Prueba de planes académicos 126
Tabla 58. Prueba de gestión registrar matricula 126
Tabla 59. Prueba de gestión de pagos de estudiantes 128
Tabla 60. Integración SPRINT 1 y 2 129
Tabla 61. Integración SPRINT 1, 2 y 3 130
Tabla 62. Integración SPRINT 1, 3 y 4 131
Tabla 63. PREG_01: ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali? 135
Tabla 64: PREG_02: ¿Cómo califica las interfaces del SIGA? 136
Tabla 65: PREG_03: ¿Cómo califica el acceso al SIGA académico? 137
Tabla 66: PREG_04: ¿Considera importante la actualización de nuevas versiones? 138
Tabla 67: PREG_05: ¿Considera que los datos son seguros en el SIGA? 139
Tabla 68: PREG_06: ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali? 140
Tabla 69: PREG_07: ¿Considera que los procesos de matrícula son tediosos para alumnos? 141

Tabla 70: PREG_08: ¿Qué opina del tiempo que se demoran los alumnos para matricularse? 142
Tabla 71: PREG_09: ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula?.....143
Tabla 72: PREG_01: ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?.....144
Tabla 73: PREG_02: ¿Cómo califica las interfaces del SIGA?.....145
Tabla 74: PREG_03: ¿Cómo califica el acceso al SIGA?.....146
Tabla 75: PREG_04: ¿Considera importante la actualización de nuevas versiones? 147
Tabla 76: PREG_05: ¿Considera que los datos son seguros en el SIGA?.....148
Tabla 77: PREG_06: ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali? 149
Tabla 78: PREG_07: ¿Considera que los procesos de matrícula son tediosos para alumnos? 150
Tabla 79: PREG_08: ¿Qué opina del tiempo que se demoran los alumnos para matricularse? 151
Tabla 80: PREG_09: ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula?.....152
Tabla 81: Estadísticos para una muestra 154
Tabla 82: Prueba para una muestra.....155

RESUMEN

La presente tesis tiene como objetivo el de determinar como el Sistema de Gestión Académica mejora el proceso de matrícula de la Universidad Nacional de Ucayali: 2019, y de esta manera poder otorgar un mejor servicio en calidad de tiempo para el estudiante, este proyecto nace a raíz a la gran congestión de alumnos que se realizaba en la Dirección General de Registros y Asuntos Académicos en cada proceso de matrículas y al crecimiento poblacional estudiantil. Además uno de los grandes problemas era que los estudiantes en su mayoría se matriculaban en fechas extemporáneas, esto debido a que se encontraban fuera de la ciudad o no contaban con el tiempo suficiente para apersonarse a las oficinas para su matrícula y cancelar el costo de la misma, o tenían ciertos inconvenientes personales la cual el estudiante tenía que pagar un adicional por matricularse en esas fechas, es así que a raíz de estos inconvenientes y siendo conocedor del problema se propone el desarrollo e implementación de un sistema de matrícula web que ayude y permita economizar el tiempo de los estudiantes y del personal administrativo, la cual es de vital importancia para la Universidad Nacional de Ucayali ya que se automatizará diferentes procesos manuales de matrícula y pago, la cual esto a su vez generará un impacto positivo entre la población estudiantil que son el principal soporte de la Universidad y del personal administrativo. Por consiguiente, este proyecto ha sido elaborado desde su análisis, diseño e implementación.

Palabras claves: Sistema informático web, matrícula web, servicios académicos.

ABSTRACT

The main objective of this thesis is to streamline the engagement process at the Universidad Nacional de Ucayali and thus be able to provide a better service in quality of time for student, this project have started as a result of the overpopulation of students that carried out in the General Directorate of Records and Academic Affairs in each enrollment process and the student population growth. In addition, one of the big problem was that the major part of students enrolled on untimely dates, due to the fact that they were out of town or did not have enough time to go to the offices to enroll and cancel the cost of the itself, or they had certain personal disadvantages which the student had to pay an additional for registering on those dates, so as a result of these inconveniences and being aware of the problem, the development and implementation of a web registration system is proposed that helps and allows to save the time of students and the management staff, which is of vital importance for Universidad Nacional de Ucayali since it will automate different manual enrollment and payment processes, which in turn will generate a positive impact among the student population They are the main support of the University and the management staff. Therefore, this project has been elaborated from its analysis, design and implementation.

Key words: Web computer system, web registration, academic services.

INTRODUCCION

El desarrollo del presente proyecto se llevó a cabo en la Universidad Nacional de Ucayali, cuenta con un portal web oficial www.unu.edu.pe desde el 2007, la población estudiantil ha ido en crecimiento durante años, a pesar de que en la misma ciudad existen otras universidades públicas y/o privadas, de las cuales muchos estudiantes de nivel secundario optan por postular e intentar su ingreso a la Universidad Nacional de Ucayali por tener más de 30 años en la región Ucayali y porque es propia de la zona, ahora último por el Licenciamiento de la Universidad por parte de la Superintendencia Nacional de Educación Superior Universitaria - SUNEDU.

Así mismo los servicios brindados por la institución deben ser accesibles, en el caso del proceso de matrícula, la Universidad desde el año 1998 recién empezó a contar con un sistema informático cliente/servidor para poder registrar toda la información de las matrículas en una base de datos que se encontraba instalado en un PC comercial y con esto empezaron a mejorar la información que en años anteriores lo tenían solo en documentos y archivados en su bóveda, en el año 2005 la universidad migró toda la información académica a un nuevo sistema informático denominado “Sistema Integrado de Gestión Académica” desarrollado en power builder y con el motor de base de datos Oracle, en esta ocasión la Base de datos se encontraba instalado ya en un Servidor Proliant, los procesos de matrículas se hicieron un poco más ágil, los primeros años los estudiantes llenaban una ficha de pre matricula conjuntamente con su asesor en su escuela profesional revisando su avance curricular, con esta ficha de pre matrícula el estudiante se acercaba a la Dirección de Coordinación Académica a realizar su cola para entregar la ficha de pre matricula firmada y sellada para que recién

pueda ser matriculado en sus cursos respectivos, unos años después el proceso cambió, ahora el estudiante se matriculaba así mismo en los laboratorios que se habilitaban para las matriculas con el apoyo de personal técnico, de esta manera evitaban las colas en la Dirección General de Registros y Asuntos Académicos, a fines del 2015 y ya con una población de más de 6 mil estudiantes, la Universidad Nacional de Ucayali se propuso la implementación de un sistema de matrícula web para estar acorde con la tecnología y brindar un buen servicio en el proceso de matrícula, a su vez tener un mejor control de los ingresos económicos por ese concepto, por tal motivo en el proceso de matrícula 2016-1 implementaron un sistema web que no tuvo un mayor impacto, la cual fue rechazado tanto por estudiantes y por autoridades vigentes, para el proceso de matrícula 2016-2 realizaron nuevamente el proceso de forma tradicional (manual), sin embargo las autoridades y en esta ocasión con la participación de la Oficina de Tecnología de Información, Sistemas y Estadística se reinicia el proyecto de análisis, desarrollo e implementación de un sistema web para mejorar el proceso de matrícula de la Universidad Nacional de Ucayali, la cual se describirá en este presente proyecto de tesis.

La estructura se encuentra dividida de la siguiente manera:

En el Capítulo I: Planteamiento del problema; Se describe la investigación realizada en el proyecto, tales como antecedentes del problema, la formulación del problema, los objetivos y la Justificación, Importancia y los Límites de la Investigación.

En el Capítulo II: Marco teórico; Se detalla un conjunto de conceptos básicos para la comprensión, se especifica las herramientas usadas en el desarrollo del

mismo, se describe el fundamento teórico de la metodología utilizada, así como conceptos claves para mejorar la interpretación de este proyecto.

En el Capítulo III: Metodología de investigación; Se detalla el método de la investigación, la población y la muestra obtenida, los procedimientos, técnicas e instrumentos usados para la recolección y procesamiento de datos.

En el Capítulo IV: Metodología de desarrollo; Se detalla el desarrollo y aplicación de la metodología SCRUM, mostrando el despliegue con las historias de usuario, SPRINTS, diseño de prototipos y código fuente.

En el Capítulo V: Resultados y discusión; Se describe los resultados de la comprobación de la Hipótesis obtenida a través de las encuestas realizadas a los usuarios.

Conclusiones y recomendaciones; Describimos todas las observaciones y recomendaciones obtenidas en el desarrollo de la presente tesis.

Finalmente se presenta; las referencias bibliográficas, y los anexos necesarios como encuestas y documentación del módulo.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. Planteamiento del problema de investigación

La Universidad fue creada mediante Decreto Ley N° 22804, de fecha 18 de diciembre de 1979, el Gobierno Militar presidido por el General Francisco Morales Bermúdez resuelve crear la Universidad Nacional de Pucallpa, conforme a lo establecido en el artículo N° 194° de la Ley General de Educación, como persona jurídica de derecho público interno, con sede en la ciudad de Pucallpa.

Que el artículo 3° dispone que el Consejo Nacional de la Universidad Peruana – CONUP - nombrará una comisión que en el término de ciento veinte (120) días, se encargue de organizar la Universidad Nacional de Pucallpa, cuyos programas serán determinados en función a las necesidades de desarrollo regional. Dicha Comisión ejercerá sus funciones hasta que se hayan constituido los órganos de gobierno en la mencionada universidad.

Que mediante Ley N° 223261 del 17 de julio de 1981, durante el gobierno democrático del Arquitecto Fernando Belaunde Terry, el Congreso de la República eleva al rango de Ley la norma de creación de la Universidad Nacional de Pucallpa.

Que el artículo 97° de la Ley N° 23733 Ley Universitaria del 09 de diciembre de 1983, cambia de denominación a la Universidad Nacional de

Pucallpa por **Universidad Nacional de Ucayali**, denominación que hasta la fecha persiste.

ILUSTRACIÓN 1. UBICACIÓN GEOGRÁFICA DE LA UNIVERSIDAD NACIONAL DE UCAYALI



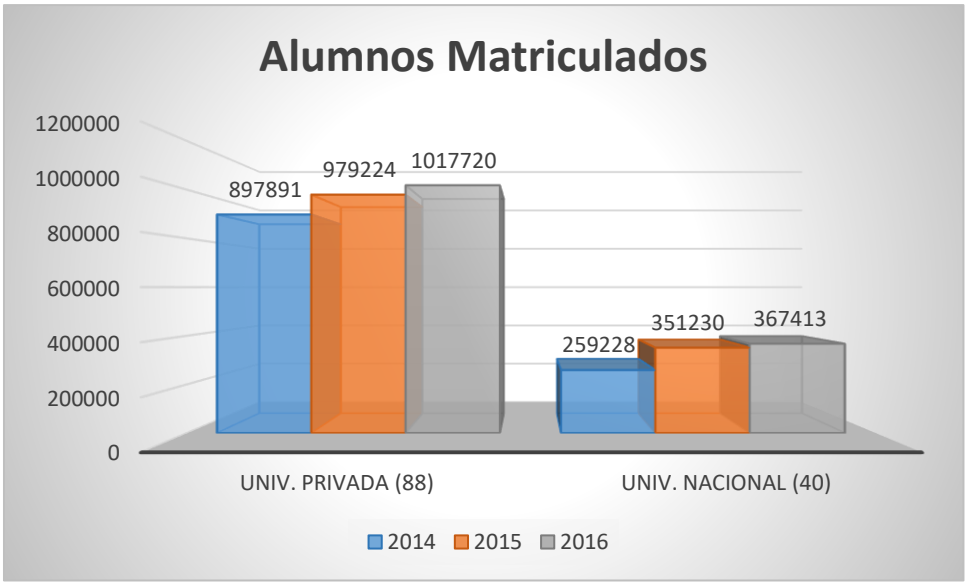
Fuente: Google Maps

La Universidad Nacional de Ucayali inició en 1981 con la Facultad de Ciencias Agropecuarias y la Facultad de Ciencias de la Salud, con sus escuelas profesionales de Ingeniería Forestal, Agronomía y Enfermería respectivamente, el 17 de marzo de 1990 se crea la facultad de Ciencias Administrativas, con su escuela profesional de Administración y el 13 de noviembre del mismo año se crea la escuela profesional de Contabilidad, el 21 de Enero de 1998 se crea la escuela profesional de Derecho, la escuela profesional de Medicina Humana y la escuela profesional de Ingeniería de Sistemas, el 29 de diciembre del 1999 se crea la escuela profesional de Educación Secundaria, el 20 de Enero del 2007 se crea la escuela profesional de Ingeniería Ambiental, el 20 de diciembre del 2008

se crea la escuela profesional de Ingeniería Agroindustrial y la escuela profesional de Ingeniería Civil, el 28 de Enero del 2009 se crea la escuela profesional de Economía y Negocios Internacionales, el 20 de septiembre se crea la escuela profesional de Educación Primaria, el 28 de Junio del 2011 se crea la escuela profesional de Educación Inicial, el 07 de diciembre del 2003 se crea la escuela profesional de Psicología, y el 27 de junio del 2014 se crea la escuela profesional de Ciencias de la Comunicación.

En el Perú tanto en las Universidades Privadas como Nacionales han tenido un crecimiento poblacional estudiantil considerable (ver gráfico N°01) de tal manera que han tenido que optimizar tecnológicamente sus procesos

ILUSTRACIÓN 2. ALUMNOS MATRICULADOS A NIVEL NACIONAL

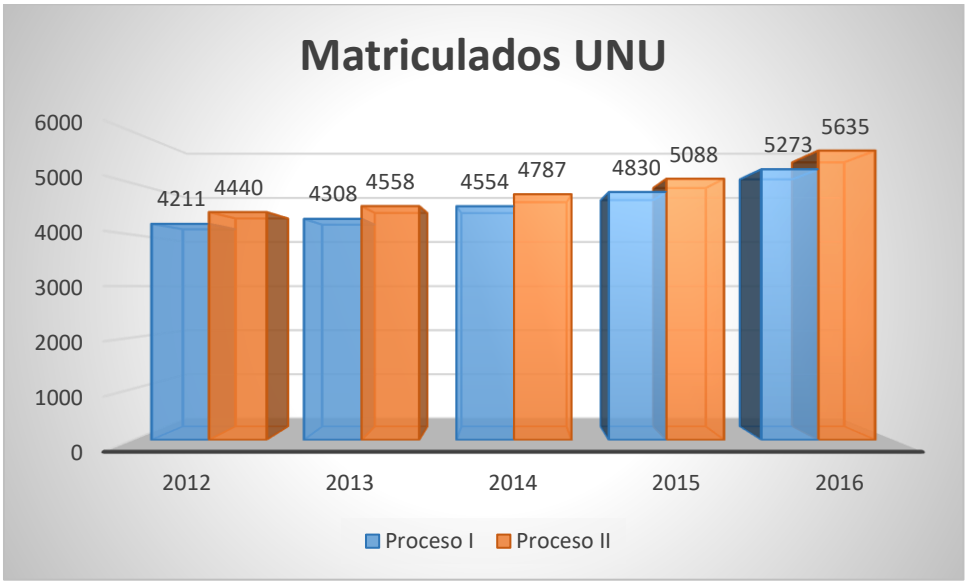


Fuente de Datos: Superintendencia Nacional de Educación Superior

La Universidad Nacional de Ucayali no ha sido ajeno al crecimiento poblacional estudiantil (ver gráfico N°02) lo cual ha generado un malestar

tanto para los estudiantes por las demoras en las matrículas (colas interminables) y como para el personal administrativo de la Dirección General de Registros y Asuntos Académicos por el aumento de la carga laboral.

ILUSTRACIÓN 3. ALUMNOS MATRICULADOS EN LA UNIVERSIDAD NACIONAL DE UCAYALI



Fuente de Datos: Oficina Ejecutiva de Estadística - UNU

El ascenso de la Universidad se debe básicamente a la alta demanda de estudiantes que desean estudiar los programas académicos ofrecidos por la institución, viendo esta oportunidad ha visto la necesidad de crear una filial en la ciudad de Aguaytia con dos carreras profesionales (Contabilidad y Agronomía), siendo la única universidad con presencia en esa ciudad.

El problema a que se abocará la presente tesis de investigación es el ineficiente proceso de matricula de alumnos de la Universidad Nacional de Ucayali, que acontinuación pasamos a describir los efectos y las causas respectivamente

1.2. Causas del problema

La Universidad Nacional de Ucayali al contar con una (01) sede y una (01) filial en la ciudad de Aguaytia, y con dieciocho (18) carreras profesionales, desde el 2016 pretende sistematizar el proceso de matrícula para brindar un mejor servicio y estar a la par con el avance tecnológico de otras universidades del Perú, además de poder llevar un mejor control de los ingresos económicos por este concepto (matrículas), desea contar con información en tiempo real de la cantidad de matriculados por carrera profesional y por día, incluyendo el ingreso económico y que por muchos años la Universidad Nacional de Ucayali ha tenido registrado la información económica en archivos Excel y no muy bien detallado, ya que el ingreso de información económica al sistema académico era de manera manual y realizado por personal de la Dirección General de Registros y Asuntos Académicos, la Oficina de Tesorería no puede llevar un control exacto de ingresos ya que los estudiantes depositaban en diferentes cuentas incluyendo en caja de la misma Universidad, toda la información es aislada e independiente, no se centraliza en un solo sistema de información, y al realizar todas estas múltiples tareas hace que toda la información no sea coherente, además que las distintas oficinas administrativas tienen que enviar la información mediante correos o documentos adjuntando CD con el archivo requerido, esto genera demoras en envío de información y conlleva a que las autoridades correspondientes puedan tomar malas decisiones, tales como la falta de pago de alumnos matriculados en fechas establecidas en el calendario académico, y esto genera un desbalance económico a la institución, o

también el ingreso de información posterior a la fecha que se indica en el calendario académico, lo cual puede generar retrasos en la matricula ordinaria de la institución, resumiéndose en:

- Ineficiente gestión de matricula.
- Todo el proceso y las tareas se realizan de manera manual, los estudiantes llenaban una ficha de pre-matricula.
- Trabajo redundante al momento de consolidar información entre las áreas involucradas.
- Carencia de una visión global en los procesos de matrícula por parte de la Dirección General de Registros y Asuntos Académicos.

1.3. Efectos del problema

La Universidad Nacional de Ucayali, al necesitar obtener los estados reales de pago de alumnos por concepto de matrícula, estudiantes que no realizan su pago por concepto de matrícula, a que cuenta realizaron los pagos, cantidad de alumnos matriculados por carrera profesional, cantidad de alumnos matriculados por día, visualizar el avance de registro de información de cupos por cursos, cantidad de alumnos matriculados por curso, o lo que aún es más importante en toda institución, tratar de hacer proyecciones para la toma de decisiones, La Universidad no ha podido obtener toda la información de la manera que realmente desearía y en el tiempo que lo requiere, ya que el encargado del área de tesorería tiene información que no coincide con lo recaudado por la Dirección General de Registros y Asuntos Académicos, y a la vez, la información de las escuelas profesionales no concuerda con lo que tiene la Dirección

General de Registros y Asuntos Académicos, prueba suficiente de que toda la información se ha venido trabajando por separado, la institución al tratar de tomar decisiones con estos tipos de resultados conllevaría al error y posteriormente a generar gastos innecesarios.

Al no poder visualizar la información de una forma general, la Dirección General de Registros y Asuntos Académicos tiende a enviar información desfasada y con errores de estudiantes matriculados, tales como cantidad de alumnos matriculados totales sin descartar los alumnos con retiro total, o que haya información personal faltante del estudiante, esto hace que se genere mala información para el posterior envío a entidades como el Ministerio de Educación - MINEDU y la Superintendencia Nacional de Educación Superior – SUNEDU, por otro lado las direcciones de las carreras profesionales al no enviar información a tiempo se puede presentar desfase en muchos procesos como la Matricula Ordinaria, registro de notas y emisión de registros, resumiéndose en:

- Pérdida de documentos que sustenta la matrícula y los pagos, documentos que no han sido registrados en el sistema académico de la Universidad Nacional de Ucayali.
- Complicados procedimientos manuales para generar reportes y consultas con los que se pueda tomar decisiones, además procesos complicados en las matrículas para los estudiantes.
- Falta de información para generar proyecciones de ingreso de dinero por concepto de matrículas en la Universidad Nacional de Ucayali.

- Constante déficit en la gestión de información precisa y oportuna para la toma de decisiones en la Universidad Nacional de Ucayali.

1.4. Formulación del problema

1.4.1. Problema general:

¿De qué manera el Sistema de Gestión Académica mejora el proceso de matrícula de la Universidad Nacional de Ucayali: 2019?

1.4.2. Problemas específicos:

1. ¿Cómo se podrá evitar la pérdida de información de los estudiantes en la Universidad Nacional de Ucayali?
2. ¿Cómo se reducirá el tiempo de atención al estudiante en el proceso de matrícula con la implementación del módulo de matrícula web en la Universidad Nacional de Ucayali?
3. ¿Cómo se podrá reducir el consumo de recursos con el sistema de matrícula web en la Universidad Nacional de Ucayali?
4. ¿Cómo se podrá obtener información relevante de los estudiantes en el proceso de matrícula en la Universidad Nacional de Ucayali?

1.5. Justificaciones

1.5.1. Justificación practica

La investigación se justifica desde el punto de vista práctica, porque el sistema informático web facilitará a gestionar el proceso de matrícula de la Dirección General de Registros y Asuntos Académicos, de la Universidad Nacional de Ucayali, encargado de registrar estudiantes, deudas, pagos, matriculas, cursos, planes curriculares, horarios, carga lectiva, cupos por curso. El desarrollo de esta aplicación web permitirá la integración y simplificación de procesos de información de manera eficaz y precisa.

1.5.2. Justificación teórica

La investigación de justifica desde el punto de vista teórico, porque un sistema informático tal cual sea su arquitectura (Cliente/Servidor o Web) como todo sistema es un conjunto de partes de hardware, software y de recursos humanos que interactúan entre sí. Un sistema informático típico emplea una computadora que usa dispositivos programables para capturar, almacenar y procesar datos como añadir, modificar, deshabilitar y ocultar. Con el uso de un sistema informático se puede tener toda la información segura en un lugar específico, donde se establecen políticas de acceso a la información por parte de la Universidad Nacional de Ucayali. Los sistemas informáticos en conjunto con la tecnología de información se han convertido en un ente imprescindible para cualquier organización que desea alcanzar un alto grado de competitividad. Java es un lenguaje de programación con el que podemos realizar cualquier tipo

de programa. Una de las principales características por las que el Lenguaje Java se ha hecho muy famoso es porque es independiente de la plataforma donde se desarrolla. Eso quiere decir que si realizamos un sistema informático en java este podrá funcionar en cualquier ordenador del mercado. Es una ventaja muy significativa para los desarrolladores de software, pues antes tenían que hacer sistemas de información para cada sistema operativo, por ejemplo, Windows, Linux, Apple, etc.

1.5.3. Justificación metodológica

La investigación se justifica metodológicamente, porque el desarrollo del sistema para la gestión de matrícula de la Universidad Nacional de Ucayali, es Scrum, una metodología de desarrollo muy simple, que requiere trabajo duro, porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. La información que le será necesaria para su proseguir en la obtención de consultas y reportes. La tecnología utilizada cuenta con herramientas y esquemas que permiten realizar sistemas robustos, seguros, prácticos y eficientes, obteniendo un gran soporte y consistencia de la información que a su vez es relevante al usuario final.

1.6. Objetivos de la investigación

1.6.1. Objetivo general

Determinar como el Sistema de Gestión Académica mejora el proceso de matrícula de la Universidad Nacional de Ucayali: 2019.

1.6.2. Objetivos específicos

1. Identificar actividades para evitar la pérdida de información de los estudiantes en la Universidad Nacional de Ucayali.
2. Identificar acciones para reducir el tiempo de atención al estudiante en el proceso de matrícula con la implementación del módulo de matrícula web en la Universidad Nacional de Ucayali.
3. Identificar acciones para reducir el consumo de recursos con el sistema de matrícula web en la Universidad Nacional de Ucayali.
4. Identificar mecanismos para obtener información relevante de los estudiantes en el proceso de matrícula en la Universidad Nacional de Ucayali.

CAPITULO II

MARCO TEÓRICO

2.1. Antecedentes del problema

1. TESIS: Ramírez (2017) nos indica que la importancia de la Gestión Académica en las Escuelas de Formación de la PNP está dada por su contribución en la formación de futuros policías altamente competitivos que contribuyan en el desarrollo nacional.

Las Escuelas de Formación de la PNP han evolucionado sustantivamente en las últimas décadas, al punto que en la actualidad cuenta con 28 Escuelas en su nivel formación, con un total de 12,983 cadetes y alumnos PNP, cubriendo la gran demanda por el Estado Peruano por requerir más policías al servicio de la ciudadanía.

Es importante destacar que, la mayoría de Institutos y Universidades hacen uso de Sistemas de Información que contribuyen en la mejora de su gestión; la presente investigación tuvo como objetico principal Implementar un Sistema Web Académico para mejorar el proceso de Gestión Académica en las Escuelas de Formación de la PNP, teniendo como hipótesis general que el Sistema Web Académico mejora el proceso de Gestión Académica en las Escuelas de Formación de la PNP

2. TESIS: Merino y Miranda (2016) Indica que gracias a los avances tecnológicos ha surgido a nivel mundial la necesidad de automatizar todo proceso en cualquier institución; ya sea de tipo gubernamental, comercial, empresarial, etc. Las nuevas aplicaciones tecnológicas hoy en día

constituyen uno de los factores clave dentro del desarrollo bien sea en el ámbito político, económico, social y cultural.

Actualmente el proceso de matrícula se realiza en hojas de cálculo y de forma manual en la Universidad Peruana Austral del Cusco, lo cual establece demasiados procesos llegando a ser hasta en algunos casos muy tediosos por parte de la secretaria académica y estudiantes, esto debido a que no cuenta con un sistema de información automatizado que permita la inscripción de matrícula de los alumnos, también podemos observar que los alumnos solo pueden consultar sus notas en la secretaría académica de la universidad.

Esta tesis tiene por finalidad presentar una solución web dirigida a la problemática que presenta actualmente en el proceso de matrículas y consulta de notas de los estudiantes universitarios y/o apoderados en la Universidad Peruana Austral del Cusco. Con esto se logrará una mejor atención de calidad en el ahorro de tiempo y eficiencia del trámite académico.

3. TESIS: Enríquez (2016) nos indica que En la actualidad, muchas universidades, institutos, centros educativos, academias preuniversitarias, y otras instituciones dedicadas al rubro de la enseñanza (tanto en sectores públicos como privados), con el fin de alcanzar la modernización y eficiencia en sus procedimientos, han optado por adquirir y/o desarrollar herramientas informáticas acorde con sus necesidades particulares (sistemas automatizados, software de oficina, redes informáticas, internet, etc.).

En el caso del centro educativo particular “Hans Kelsen” (Florencia de Mora –Trujillo), por ser una institución educativa relativamente nueva (con menos de una década en funcionamiento), hasta el momento no se ha adquirido un sistema informático que gestione sus principales procesos académicos.

Uno de los principales problemas que aquejan al personal encargado de los procesos académicos (principalmente en el área de coordinación académica) es que, debido al desorden existente en la información, se gasta más del tiempo necesario para realizar dichos procesos y origina disconformidad por parte del personal.

4. TESIS: Villacorta (2011) Indica que La Institución Microsystems es una institución privada, que ofrece servicios educativo en el área de computación e informática, cuya misión es brindar una educación de calidad orientada a la formación integral de los alumnos para enfrentar competitivamente los desafíos que impone el avance del conocimiento, tecnología y las responsabilidades cada vez mayores en la sociedad.

Las notas de los estudiantes siempre se ha manejado de una manera manual, y siempre ha sido el profesor quien lleva la labor de desempeñar dicho trabajo, pero nunca se ha llevado un control de manera automatizada y computarizada, la solución más probable era la aplicación de una correcta base de datos y un sistema con un buen diseño para que a la hora que los alumnos necesiten consultar sus notas esté lo más ordenado posible.

La mayoría de procesos de la empresa son rutinarios, llevarlos de forma manual les toma mucho tiempo, pero si estos estarían en un sistema automatizado garantizaría un mejor trato de información. La Institución Microsystems actualmente cuenta con un número 900 estudiantes aproximadamente.

2.2. Planteamientos teóricos

2.2.1. Sistema web

Los “sistemas Web” o también conocidos como “Aplicaciones Web” son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux). Sino que se aloja en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los ‘Sistemas Web’ tienen funcionalidades muy potentes que brindan respuestas a casos particulares.

Los sistemas Web se pueden utilizar en cualquier navegador Web (chrome, firefox, Internet Explorer, etc.) sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema.

Las aplicaciones Web trabajan con bases de datos que permite procesar y mostrar información de forma dinámica para el usuario.

Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy

beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema.

Un ejemplo claro de un sistema es un “panel de administración”, con él se podrán modificar y actualizar diferentes contenidos dentro de la página sin necesidad de llamar a tu proveedor para que lo haga.

Este tipo de diferencias se ven reflejada en los costos, en la rapidez de obtención de la información, en la optimización de las tareas por parte de los usuarios y en alcanzar una gestión estable. (Baez, 2013)

2.2.2. Sistema de Información

Según Cohen (2009), afirma que los sistemas de información (SI) están cambiando la forma en que operaban las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos de las empresas, proporcionan información de apoyo al proceso de toma de decisiones, lo que es más importante, facilitan el logro de ventajas competitivas a través de su implantación en las empresas.

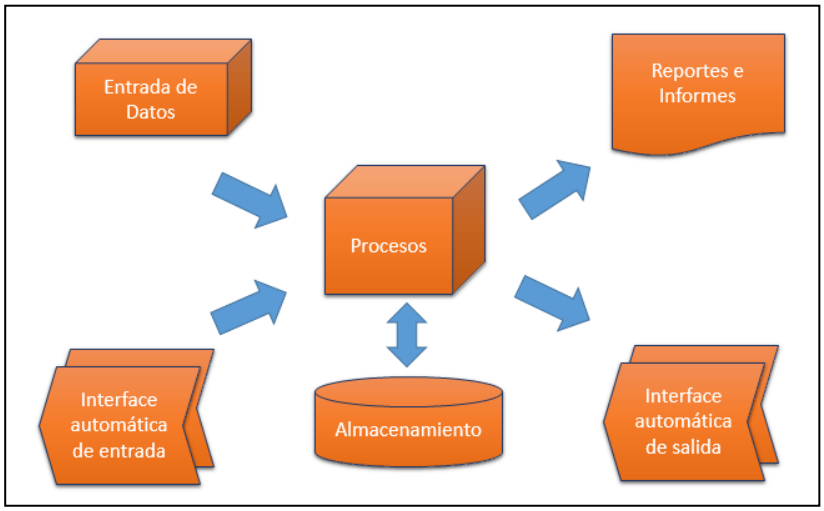
Tipos y usos de los sistemas de información

Durante los próximos años, los sistemas de información cumplirán tres objetivos básicos dentro de las organizaciones:

- Automatizar los procesos operativos.

- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr: ventajas competitivas a través de su implantación y uso.

ILUSTRACIÓN 4. DISEÑO CONCEPTUAL DE UN SISTEMA DE INFORMACIÓN



Fuente: Cohen, 2000: 9

En la figura anterior se aprecia las diferentes actividades que realiza un sistema de información, como son la entrada de proceso y de salida de datos, así como la interfaz de salida de datos.

a. Sistemas transaccionales

Sus principales características son:

- Suelen introducirse después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.

- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, genera poca información como resultado, pero puede realizar muchos cálculos durante su proceso.
- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser Sistemas de Información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivos y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un Sistema de Compra de Materiales que indique cuándo debe hacerse un pedido al proveedor o un Sistema de Simulación de Negocios que apoye la decisión de introducir un nuevo producto al mercado.

Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.

b. Sistema de apoyo a las decisiones

Sus principales funciones son:

- Permite extraer y manipular información de una manera flexible.

- Ayuda en decisiones no estructuradas.
- Permite al usuario definir interactivamente qué información necesita y cómo combinarla.
- Suele incluir herramientas de simulación, modelización, etc.
- Puede combinar información de los sistemas transaccionales internos de la empresa con los de otra empresa externa.

c. Sistemas estratégicos

Sus principales características son:

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones. Sin embargo, este tipo de sistemas puede llevar a cabo dichas funciones.
- Suelen desarrollarse in house, es decir, dentro de la organización, por lo tanto, no pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Un ejemplo de estos sistemas de información dentro de la empresa puede ser un sistema MRP (manufacturing resource planning) enfocado a reducir sustancialmente el desperdicio durante el proceso productivo, o bien, un centro de información que proporcione todo tipo de información estratégico si, y solo si, apoyan o dan forma a la estructura competitiva de la empresa

2.2.3. Tecnologías de información en empresas

Según ESAN (2009) la incorporación de la tecnología puede llevar a que empresas densas y rutinarias se transformen en ligeras,

debido a la disminución de los costos de producción y a la apertura de nuevos canales para llegar a los clientes vía internet; por ejemplo, la tecnología se convierte en una herramienta que coadyuva a cumplir las estrategias empresariales, más aún si estamos en una época de cambios, en la que se exige a las empresas ser más competitivas en el mercado local y global.

En consecuencia, la empresa peruana debe enfocar la tecnología bajo este enfoque, y adoptarla como su gran soporte. Claro que la transformación se logrará si cuenta además con gente innovadora. De esta manera, la estructura organizacional se verá reducida y los empresarios tendrán menos costos y más oportunidades para crear nuevos negocios. Las empresas se harán más reducidas y más eficientes, lo que llevará a que se especialicen mucho más y se establezcan alianzas estratégicas para que otros produzcan lo secundario. Algo interesante que ocurre en este proceso de transformación es que las empresas que han aplicado este concepto, en el que la tecnología resulta un elemento fundamental de cambio, han encontrado la posibilidad de crear nuevos negocios u optimizar los actuales, lo que les da la oportunidad de crear alianzas estratégicas. De esta manera, la tecnología permite a las empresas, por ejemplo, interconectarse con sus proveedores, y juntos reducir costos y optimizar el flujo de los insumos y de los productos. Lo mismo ocurre en el ámbito global; pero eso requiere integración, en el cual uno de sus elementos es la tecnología a través de las comunicaciones.

Así esta época, tenemos que ser más rápidos. Ya no se puede hablar de que las empresas más grandes se comen a las más chicas, sino de que las empresas más lentas desaparecerán y las más rápidas permanecerán en el mercado. Este proceso se dará en nuestra sociedad, y mientras más rápido ocurra mejor, porque la competencia no espera. Los procesos de comunicación a través de las tecnologías responden a estrategias, las que pueden ser evolutivas, de cambio, las cuales impactarán en la organización, crearán un ambiente de mínima resistencia al cambio; o pueden ser radicales, lo que implicará mucha dedicación de la empresa, a cambio de beneficios más rápidos.

2.2.4. Lenguaje de programación

Es un lenguaje formal que proporciona una serie de instrucciones que permiten a un programador escribir secuencias de órdenes y algoritmos a modo de controlar el comportamiento físico y lógico de una computadora con el objetivo de que produzca diversas clases de datos. A todo este conjunto de órdenes y datos escritos mediante un lenguaje de programación se le conoce como programa.

Algunos de estos lenguajes están formados por un conjunto de símbolos (llamado alfabeto), reglas gramaticales (léxico/morfológicas y sintácticas) y reglas semánticas, que en conjunto definen las estructuras válidas en el lenguaje y su significado. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un

programa informático se le llama programación, que es la que define la creación de software mediante la aplicación de procedimientos lógicos a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación.

Existe un error común que trata por sinónimos los términos 'lenguaje de programación' y 'lenguaje informático'. Los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como por ejemplo HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación, sino un conjunto de instrucciones que permiten estructurar el contenido de los documentos).

El lenguaje de programación permite especificar de *manera precisa* sobre qué datos debe operar un software específico, cómo deben ser almacenados o transmitidos dichos datos, y qué acciones debe dicho software tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar *relativamente* próximo al lenguaje humano o natural. Una característica relevante

de los lenguajes de programación es precisamente que más de un programador pueda usar un conjunto común de instrucciones que sean comprendidas entre ellos para realizar la construcción de un programa de forma colaborativa.

2.2.5. Programación Orientada a Objetos

La programación orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que innova la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

2.2.6. Metodología SCRUM

Según Juan Palacio (2009): Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro, porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. Como método ágil:

- Es un modo de desarrollo adaptable, antes que predictivo.
- Orientado a las personas, más que a los procesos.

- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.

Comparte los principios estructurales del desarrollo ágil: a partir del concepto o visión de la necesidad del cliente, construye el producto de forma incremental a través de iteraciones breves que comprenden fases de especulación exploración y revisión. Estas iteraciones (en Scrum llamadas Sprint) se repiten de forma continua hasta que el cliente da por cerrado el producto. Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de negocio, y que pueden llevarse a cabo en un periodo de tiempo breve (según los casos pueden tener duraciones desde una semana hasta no más de dos meses). Cada uno de estos periodos de desarrollo es una iteración que finaliza con la entrega de una parte (incremento) operativa del producto. Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución en reuniones breves diarias donde todo el equipo revisa el trabajo realizado el día anterior y el previsto para el siguiente. En Scrum se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está

especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales. Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

2.2.7. JAVA

Creado en 1991, Java es un lenguaje de programación orientado a objetos y diseñado según el modelo de otros lenguajes (C++, Eiffel, SmallTalk, Objective C, Cedar/Mesa, Ada, Perl), pero sin sus defectos.

Las ventajas de la programación orientada a objetos son: un mejor dominio de la complejidad (dividir un problema complejo en una sucesión de pequeños problemas), una reutilización más sencilla, y mayor facilidad de corrección y de evolución.

Java estándar esta provista de un conjunto de clases que permiten crear y manipular todo tipo de objetos (interfaz gráfica, acceso a la red, gestión de entradas/salidas). (Groussard, 2012)

2.2.8. Primefaces

Primefaces es una implementación JSF para aplicaciones RIA, es una librería de componentes visuales open source, sobre cualquier otra implementación JSF tiene la ventaja de ser construida sobre JQuery, es mismo que es un framework JavaScript de gran aceptación en el mercado principalmente por brindar componentes de apariencia bastante agradable y con tiempos de respuesta bastante pequeños respecto del resto de implementaciones JSF. - Las principales características de Primefaces son: - Soporte nativo de Ajax. - Kit para crear aplicaciones web para móviles. - Es compatible con otras librerías de componentes, como JBoss RichFaces. - Uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque

2.2.9. Hibernate

“Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL”.

2.2.10. Servidor de Base de datos

Servidores de Base de Datos. Son los que se utilizan para la ejecución de gestores de Base de Datos y que varios usuarios

hagan operaciones sobre ellas al mismo tiempo, situado en un lugares diferentes; se puede acceder a las BD por terminales o equipos con un programa llamado cliente que permita el acceso a ellas.

Una base de datos o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos. (Kroenke, 2007)

2.2.11. Gestor de base de datos Oracle

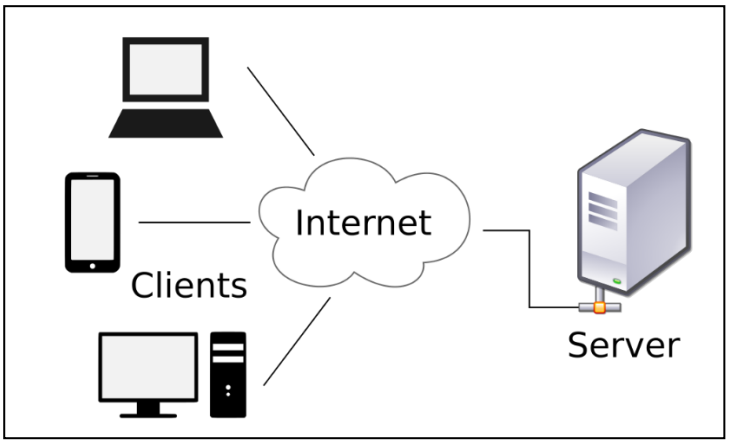
Oracle la Primera Base de Datos Diseñada para Grid Computing, es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Oracle es básicamente un herramienta cliente/servidor para la gestión de base de datos, la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general Oracle Corporation es una de las mayores compañías de software del mundo, sus productos van desde bases de datos (Oracle) hasta sistemas de gestión, cuenta además con herramientas propias de

desarrollo para realizar potentes aplicaciones, como Oracle Designer.

2.2.12. Arquitectura Cliente/Servidor

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

ILUSTRACIÓN 5. ARQUITECTURA CLIENTE/SERVIDOR



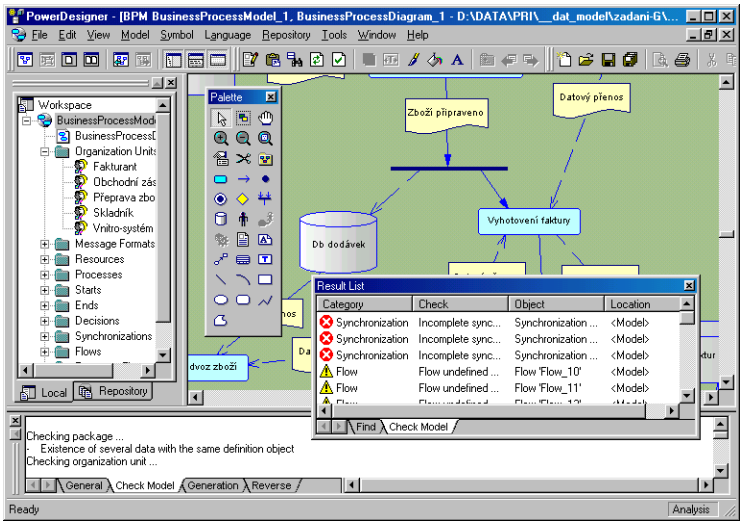
2.2.13. PowerDesigner

Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que da a los desarrolladores Cliente/Servidor la más firme base para aplicaciones de alto rendimiento.

Características de PowerDesigner

- Permite a las empresas, de manera más fácil, visualizar, analizar y manipular metadatos, logrando una efectiva arquitectura empresarial de información.
- Brinda un enfoque basado en modelos, el cual permite alinear al negocio con la tecnología de información, facilitando la implementación de arquitecturas efectivas de información empresarial.
- Brinda potentes técnicas de análisis, diseño y gestión de metadatos a la empresa.
- Combina varias técnicas estándar de modelamiento con herramientas líder de desarrollo, como .NET, Sybase WorkSpace, Sybase Powerbuilder, Java y Eclipse, para darle a las empresas soluciones de análisis de negocio y de diseño formal de base de datos.
- Trabaja con más de 60 bases de datos relacionales.

ILUSTRACIÓN 6. INTERFAZ DE POWERDESIGNER



2.2.14. Proceso de Matrícula

MINEDU (2016) es el proceso de matrícula es el conjunto de políticas, procedimientos y actividades, que permiten organizar la continuidad de los alumnos antiguos y el ingreso de alumnos nuevos.

Matrícula es el acto por medio del cual una persona se incorpora a la Universidad, adquiere o renueva su calidad de estudiante regular, queda adscrito a uno o varios programas académicos, y asume el compromiso de cumplir con todos los reglamentos y normas de la Universidad (Pontificia Universidad Javeriana, 2017).

2.2.15. Requisitos en el proceso de matrícula

La matrícula tiene requisitos y trámites académicos y administrativos. Para ser efectiva, se requiere del cumplimiento de cada uno de los siguientes procesos:

El estudiante deberá estar a paz y salvo con la Universidad.

Expedición de la orden de matrícula financiera por la Tesorería de la Universidad.

Realización del pago de los derechos de matrícula dentro de los plazos establecidos en la orden de matrícula financiera y de acuerdo con las normas aprobadas por el Consejo Directivo Universitario.

El estudiante deberá realizar la inscripción de asignaturas dentro de los plazos fijados por el calendario de la Universidad.

Declaración formal por parte del estudiante, en el momento de realizar la matrícula, que es beneficiario, de acuerdo con la ley vigente, de servicios de salud y atención médica POS. Es deber del estudiante, o de su familia, conservar vigente dichos beneficios durante el respectivo período académico.

Aceptación del director del Programa, de las asignaturas que el estudiante ha inscrito para el período académico.

El estudiante deberá actualizar la información de datos personales que le solicite la Universidad.

Los estudiantes extranjeros deberán cumplir los requisitos establecidos por la ley de acuerdo con la directriz que para tal efecto expida el Vicerrector Académico (MINEDU, 2016)

2.3. Definición de términos básicos

Base de datos: es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.

Desarrollo de Software: Es una estructura aplicada al desarrollo de un producto de software (información, análisis, diseño, implantación) bajo una metodología de trabajo y una arquitectura de desarrollo.

Desarrollo Web: Define la creación de sitios web para Internet o una intranet.

Framework: Plantillas predefinidas, que facilitan algunas tareas.

Java: Java es un entorno de computación introducido al público en 1995 por Sun Microsystems. Es un lenguaje de programación, cuya sintaxis recuerda la del C++, tiene librerías de clases y el que todos sus programas se ejecuten en una máquina virtual lo convierten en un lenguaje altamente portátil, muy apto para una red con ordenadores y sistemas operativos tan heterogéneos como Internet.

Lenguaje de Programación: Es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora.

Tecnología: Conjunto de instrumentos, recursos técnicos o procedimientos empleados en un determinado campo o sector.

Tiempo real: Deriva de procesamiento digital de señales. En una descripción muy corta un sistema de tiempo real es aquel capaz de procesar una muestra de señal antes de que ingrese al sistema la siguiente muestra.

Web: Conjunto de información que se encuentra en una dirección determinada de internet.

CAPITULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Método de investigación

Para Murillo (2008), la investigación aplicada recibe el nombre de investigación práctica o empírica", que se caracteriza porque busca la aplicación o utilización de los conocimientos adquiridos, a la vez que se adquieren otros, después de implementar y sistematizar la práctica basada en investigación. El uso del conocimiento y los resultados de investigación que da como resultado una forma rigurosa, organizada y sistemática de conocer la realidad.

Padrón (2006) al respecto, para quien la expresión se propagó durante el siglo XX para hacer referencia, en general, a aquel tipo de estudios científicos orientados a resolver problemas de la vida cotidiana o a controlar situaciones prácticas, haciendo dos distinciones:

a. La que incluye cualquier esfuerzo sistemático y socializado por resolver problemas o intervenir situaciones.

   En ese sentido, se concibe como investigación aplicada tanto la innovación técnica, artesanal e industrial como la propiamente científica.

b. La que sólo considera los estudios que explotan teorías científicas previamente validadas, para la solución de problemas prácticos y el control de situaciones de la vida cotidiana.

Sánchez (2004), la investigación aplicada, a veces llamada Investigación técnica, tiende a la resolución de problemas o al desarrollo de ideas, a

corto o media plazo, dirigidas a conseguir innovaciones, mejoras de procesos o productos, incrementos de calidad y productividad, etc.

Este proyecto está enfocado y desarrollado bajo el enfoque de **investigación aplicada**, ya que reúne las condiciones metodológicas de la misma, se utilizarán los conocimientos en ciencias informáticas a fin de aplicarlas en la gestión administrativa del proceso de matrícula de la Universidad Nacional de Ucayali.

3.1.1. Nivel de investigación

El tipo de investigación es aplicada, el enfoque es cuantitativo, el nivel es descriptivo – transversal.

3.1.2. Diseño de la investigación: Pre – Experimental

Se usará el método Pre-Test y Post-Test, que se basa en dos pruebas, antes de implementar la variable independiente y después de implementar, para medir el cambio que sufre la variable dependiente.

$$G = O_1 - - - -> X - - - -> O_2$$

Dónde:

- G: Grupo experimental.
- O_1: Medición de la variable V2, antes de la implementación de la variable V1.
- O_2: Medición de la variable V2, después de la implementación de la variable V1.

3.2. Población y muestra

3.2.1. Población

La Universidad Nacional de Ucayali Cuenta con 5082 estudiantes matriculados en la Universidad Nacional de Ucayali en el periodo 2018, tal como se aprecia en la siguiente tabla:

TABLA 1. Resumen de matriculados por escuela en el periodo 2018

ESCUELAS PROFESIONALES	ANUAL
Agronomía	367
▪ *Agronomía (P)*	*244*
▪ *Agronomía (A)*	*123*
Ing. Agroindustrial	203
Ing. Forestal	274
Ing. Ambiental	407
Enfermería	295
Medicina	348
Economía y Negocios Internacionales	335
Contabilidad	688
▪ *Contabilidad (P)*	*553*
▪ *Contabilidad (A)*	*135*
Administración	437
Ing. Sistemas	403
Ing. Civil	383
Derecho	728
Educación inicial	41
Educación primaria	38
Educación secundaria	135
▪ *Lenguaje y Literatura*	*37*
▪ *Idiomas*	*93*
▪ *Matemáticas Física e Informática*	*3*
▪ *Cs Naturales y Medio Ambiente*	*2*
TOTAL	**5082**

Fuente: (Oficina Ejecutiva de Estadística, 2018)

3.2.2. Muestra

Mediante la siguiente formula, calculamos la muestra.

$$n = \frac{Z^2_{\frac{\alpha}{2}} xPxQxN}{\in^2 x(N-1) + Z^2_{\frac{\alpha}{2}} xPxQ}$$

N = Población de estudiantes matriculados (5082).

n = Muestra suficiente para la investigación.

P = Probabilidad que el suceso ocurra (50% ≡ 0.5).

Q = Probabilidad que el suceso no ocurra (50% ≡ 0.5).

$Z_{\alpha/2}$ = 1.96.

$\mathcal{E}$ = 0.05.

Por lo tanto:

$$n = \frac{1.96^2 x 0.5 x 0.5 x 5082}{0.05^2 x (5082 - 1) + 1.96^2 x 0.5 x 0.5} = 357.23$$

Obteniendo una muestra de 358 colaboradores.

Muestra estratificada = SUBTOTAL * PORCENTAJE POBLACIONAL

TABLA 2. Muestra estratificación por escuela profesional

ESCUELAS PROFESIONALES	SUBTOTAL	PORCENTAJE POBLACIONAL	MUESTRA ESTRATIFICADA
Agronomía	367	0.07	26
Ing. Agroindustrial	203	0.04	14
Ing. Forestal	274	0.05	19
Ing. Ambiental	407	0.08	29
Enfermería	295	0.06	21
Medicina	348	0.07	25
Economía y Negocios Internacionales	335	0.07	24
Contabilidad	688	0.14	48
Administración	437	0.09	31
Ing. Sistemas	403	0.08	28
Ing. Civil	383	0.08	27
Derecho	728	0.14	50
Educación inicial	41	0.01	3
Educación primaria	38	0.01	3
Educación secundaria	135	0.03	10
TOTAL	**5082**	**1.00**	**358**

3.3. Instrumentos de recolección de datos

Encuesta: Son preguntas que conforman un cuestionario, dirigido a cierta población de nuestro estudio, buscando obtener información relevante para el informe de tesis (Marroquín, 2012).

Análisis documental: Está conformado por referencias de varios autores (Marroquín, 2012).

3.4. Instrumentos para la recolección de datos

a. Encuesta:

Son preguntas que conforman un cuestionario, dirigido a cierta población de nuestro estudio, buscando obtener información relevante para el informe de tesis (Marroquín, 2012).

b. Análisis documental:

Está conformado por referencias de varios autores (Marroquín, 2012).

3.5. Tratamiento de datos

El aspecto estadístico de la investigación es procesado en el IBM SPSS STATISTICS, con lo cual se analiza los resultados de manera descriptiva mediante gráficos y tablas, posteriormente se procede a validar el diseño pre test y post test.

CAPITULO IV

METODOLOGÍA DE DESARROLLO DEL SISTEMA INFORMÁTICO

En esta sección se describirá el desarrollo de la metodología SCRUM del presente proyecto, donde se determina que el proyecto tuvo una duración de 120 días laborables, además de acuerdo al estudio se describen 20 historias de usuarios distribuidos en 4 Sprints, donde cada historia contempla su prioridad (Alta, media o baja), importancia (de 0 a 100) y su tiempo estimado de desarrollo (días laborables), también se describe por cada Sprint su priorización, la asignación de tareas por responsable determinando los días para concluir dicha tarea, además se detalla el avance por sprint con el Burn Down Chart, se incluyen los informes de las pruebas funcionales por cada Sprint, informes de la reuniones de revisión del sprint (Sprint Review), y formatos de las reuniones retrospectivas (Restrospective Meeting), las cuales se han empleado para la construcción del sistema informático.

4.1. Normas y Roles del proyecto

4.2.1. Normas Internas

El presente proyecto pretende mejorar la atención a los estudiantes y mejorar los procesos de matrícula de la Universidad Nacional de Ucayali, está basada en una metodología ágil denominada SCRUM, la cual una de las ventajas que presenta es la de aumentarla productividad en el proyecto y potencia el compromiso del equipo, por lo cual se pone de manifiesto lo siguiente:

- Las tareas pueden afectar a otros miembros del equipo porque impactan en el desarrollo del proyecto.

- Todos los miembros del equipo pueden apoyar y ofrecer su ayuda a otros en la realización de tareas o para resolver problemas, el facilitador (SCRUM MASTER) se encargará de solucionar los impedimentos que el equipo no puede solucionar por sí solo.

- Las tareas que se realicen deben estar alineadas con el compromiso del equipo.

- Los miembros entienden las necesidades de los otros miembros del equipo respecto a su trabajo, de manera que pueden colaborar y adaptar sus trabajos para que se pueda dar un máximo valor y no realizar tareas que no aportan ningún valor o beneficio al resto del equipo.

- Si un miembro del equipo está realizando tareas por debajo del rendimiento esperado, es necesario que en las reuniones de trabajo se comunique para poner a todos los miembros del equipo en autos para buscar la mejor solución.

4.2.2. Roles del Proyecto

En la siguiente tabla se muestra los roles según la metodología SCRUM:

TABLA 3. ROLES SEGÚN SCRUM

Rol	Nombre del Encargado
Product Owner	Pedro Aparicio Campos Cabrera
Scrum Master	Jorge Eduardo Trigueros Bellido
Team	José Luis Congona Peña

- **PRODUCT OWNER:**

  Se encargará de crear la lista de funcionalidades del sistema, planificar el inicio de cada sprint y la revisión del producto al término de cada sprint para determinar si se cumplió con todas las funcionalidades.

- **SCRUM MASTER:**

  Se encargará de administrar el proceso del proyecto, su planificación, coordinación con el equipo y realizará un seguimiento e informes del progreso del proyecto, en términos de calidad, costo y plazos de entrega.

  - Realizará la planificación de todas las actividades generales del proyecto.
  - Aceptará o rechazará los resultados del trabajo del equipo.
  - Es responsable del promover los valores y normas de SCRUM.
  - Remueve impedimentos.
  - Asegurará de que el equipo es completamente funcional y productivo.
  - Permitirá la estrecha cooperación en todos los roles y funciones.

- **TEAM:**

  Las principales funciones son

  - Comprometerse al inicio de cada sprint desarrollar todas las funcionalidades en el tiempo determinado
  - Responsable de entregar el producto a cada término del Sprint.
  - Definir el desarrollo del sistema

4.3. Historia de usuarios

Las historias de usuarios que se realizaron fueron desarrolladas en conjunto con todo el equipo de trabajo involucrados en el proceso seleccionado para el desarrollo del proyecto. Los cuales se clasifican por módulos. Para la estimación de los datos se tomó el siguiente criterio:

- **Prioridades del Negocio (PN):** Se medirá en función al rango de Alta, Media y Baja, las cuales, serán asignadas por el Product Owner.

- **Importancia del Desarrollo (ID):** Se medirá en función a ponderaciones en un rango del 1 al 100 entre el Product Owner y los miembros del equipo Scrum, donde:

  a) Todos los elementos con importancia >=100 deben estar incluidos en el Sprint 1, por ser considerados de extrema importancia para el proyecto.

  b) Todos los elementos de importancia entre 99-50 deberán estar incluidos en el Sprint 2, pero eso va depender de la velocidad del Sprint.

  c) Los elementos con importancia entre 49-25 se deberá incluir en el tercer Sprint, según el avance del equipo ya que son requisitos que alteran relativamente el desarrollo del mismo o funcionalidades del mismo.

  d) Los elementos con importancia entre 24-0 se deberá incluir en el último sprint, de acuerdo a lo avanzado en el proyecto ya que no son

requisitos que alteran el desarrollo del mismo o funcionalidades del mismo.

- **Tiempo Estimado (TS):** Se medirá en función a ponderaciones entre 1 al 20 entre el Product Owner y los miembros del equipo Scrum.

Así mismo las historias de usuario se han dividido por módulos para hacer más fácil la programación de cada una de las tareas concernientes a cada uno de ellos, las cuales son:

- **Módulo de Seguridad:** Es el módulo donde se generará la creación y mantenimiento de usuarios con sus respectivos accesos.
- **Módulo de Ingresantes:** Es el módulo donde se registrará a los postulantes que lograron su ingreso a la Universidad con toda su información personal.
- **Módulo Planes:** Es el módulo en donde se registrará los planes curriculares de las diferentes carreras profesionales de la Universidad Nacional de Ucayali.
- **Módulo Escalafón:** Es el módulo en donde se registrará a los docentes de la Universidad Nacional de Ucayali, en las cuales luego se les asignará sus respectivos cursos.
- **Módulo Matrícula:** Es el módulo en donde los estudiantes podrán revisar su información personal, sus notas, su avance curricular, su ficha de pre-matricula, podrán registrar su matrícula seleccionando los cursos y posteriormente podrán visualizar su ficha de matrícula y su ficha de tasa educacional.
- **Módulo Cuenta Corriente:** Es el módulo donde se podrá registrar los pagos realizados por los estudiantes por concepto de matrícula.

4.3.1. Descripción de Historia de Usuarios

TABLA 4. HISTORIA DE USUARIO HUCR01

Historia de Usuario	
Número: HUCR01	**Usuario:** Administrador del Sistema
Nombre Historia: Autentificación de los usuarios al sistema	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Seguridad	
Descripción: Yo como usuario requiero que el sistema permita iniciar sesión y acceder a mis funciones según mi perfil	
Validación: ** Ingresar al sistema con el perfil correspondiente al usuario.* ** El sistema debe permitir el acceso siempre y cuando la información ingresada son correctas.* ** En base al perfil del usuario, el sistema le permitirá acceder a las opciones correspondientes*	

TABLA 5. HISTORIA DE USUARIO HUCR02

Historia de Usuario	
Número: HUCR02	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de usuarios	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Seguridad	
Descripción: Como Administrador requiero un mantenimiento de usuarios, lo cual pueda ingresar los siguientes datos del personal, tales nombre completo, tipo de documento, número de documento, contraseña inicial, fecha de caducidad de contraseña, una opción de activar al usuario.	
Validación: ** Son datos requeridos, nombre completo, tipo y número de documento, una opción de activar al usuario*	

TABLA 6. HISTORIA DE USUARIO HUCR03

Historia de Usuario	
Número: HUCR03	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de estudiantes	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Ingresantes	
Descripción: Como Administrador requiero un mantenimiento de estudiantes, lo cual pueda ingresar los siguientes datos, tales como periodo de ingreso, facultad de ingreso, carrera profesional de ingreso, apellido paterno, apellido materno, nombres, tipo de documento, número de documento, teléfono, fecha de nacimiento, dirección, ubigeo de dirección, correo electrónico, estado civil, sexo, lugar de nacimiento, ubigeo de nacimiento, grupo sanguíneo, nacionalidad, tipo colegio secundario, etnia, modalidad de ingreso y nota de ingreso.	
Validación: * *Todos los datos son requeridos*	

TABLA 7. HISTORIA DE USUARIO HUCR04

Historia de Usuario	
Número: HUCR04	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de perfiles	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Seguridad	
Descripción: Como administrador del sistema se encarga de registrar, asignar, actualizar y/o eliminar los perfiles de acceso al sistema, estos perfiles definen los menús que tendrá cada usuario.	
Validación: * *Son datos requeridos, nombre completo, tipo y número de documento, una opción de activar al usuario*	

TABLA 8. HISTORIA DE USUARIO HUCR05

Historia de Usuario	
Número: HUCR05	**Usuario:** User01
Nombre Historia: Asignación, modificación y deshabilitación de usuarios a los perfiles	
Prioridad en Negocio: Media	**Requisitos:** HUCR02, HUCR04
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Seguridad	
Descripción: Como administrador del sistema requiero de registrar, actualizar y/o deshabilitar los datos de los usuarios del sistema, donde se le asigna un perfil, que le darán los accesos a las funciones del sistema; se necesitan los siguientes datos: usuario, clave, nombre completo del usuario y el perfil del usuario.	
Validación: * *Todos los datos son requeridos*	

TABLA 9. HISTORIA DE USUARIO HUCR06

Historia de Usuario	
Número: HUCR06	**Usuario:** User01
Nombre Historia: Asignación, modificación y deshabilitación de estudiantes a los perfiles	
Prioridad en Negocio: Media	**Requisitos:** HUCR03, HUCR04
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Seguridad	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar los datos de los estudiantes, donde se le asigna un perfil, que le darán los accesos a las funciones del sistema; se necesitan los siguientes datos: código de estudiante, clave de acceso, nombre completo del usuario y el perfil del usuario.	
Validación: * *Todos los datos son requeridos*	

TABLA 10. HISTORIA DE USUARIO HUCR07

Historia de Usuario	
Número: HUCR07	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de Facultades	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como Administrador requiero un mantenimiento de facultades, lo cual pueda ingresar los siguientes datos, tales como Nombre completo de la Facultad, Resolución de creación de la facultad, fecha de creación de facultad y un check que me dé opción a habilitar/deshabilitar la facultad.	
Validación: * *Todos los datos son requeridos*	

TABLA 11. HISTORIA DE USUARIO HUCR08

Historia de Usuario	
Número: HUCR08	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de Carreras Profesionales	
Prioridad en Negocio: Media	**Requisitos:** HUCR07
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como Administrador requiero un mantenimiento de carreras profesionales, lo cual pueda ingresar los siguientes datos, tales como Nombre completo de la carrera profesional, una abreviatura de la carrera profesional, Resolución de creación de la carrera profesional, fecha de la resolución, fecha de creación de la carrera profesional, un check que me dé opción a habilitar/deshabilitar la facultad y un check para activar/desactivar sus planes de estudios	
Validación: * *Todos los datos son requeridos*	

TABLA 12. HISTORIA DE USUARIO HUCR09

Historia de Usuario	
Número: HUCR09	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo condición docente	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como Administrador requiero un mantenimiento de tipo condición docente, lo cual pueda ingresar los siguientes datos, tales como descripción larga del tipo de condición, descripción corta del tipo de condición, fecha de inicio del tipo de condición, fecha de término de condición y un combobox para definir el estado del tipo de condición docente.	
Validación: * *Todos los datos son requeridos*	

TABLA 13. HISTORIA DE USUARIO HUCR10

Historia de Usuario	
Número: HUCR10	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo categoría docente	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como Administrador requiero un mantenimiento de tipo condición docente, lo cual pueda ingresar los siguientes datos, tales como descripción larga del tipo de categoría, descripción corta del tipo de categoría, fecha de inicio del tipo de categoría, fecha de término de categoría y un combobox para definir el estado del tipo de categoría docente.	
Validación: * *Todos los datos son requeridos*	

TABLA 14. HISTORIA DE USUARIO HUCR11

Historia de Usuario	
Número: HUCR12	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de docentes	
Prioridad en Negocio: Media	**Requisitos:** HUCR04, HUCR09, HUCR10
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Escalafón	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar los datos de los docentes, donde se le asigna un perfil, una categoría docente, una condición docente, tipo de documento, número de documento, apellido paterno, apellido materno, nombres, sexo, estado civil, fecha de nacimiento, nacionalidad, dirección, número de teléfono fijo, número de teléfono móvil, correo personal, correo institucional, nivel educativo, fecha de vinculación.	
Validación: * *Todos los datos son requeridos*	

TABLA 15. HISTORIA DE USUARIO HUCR12

Historia de Usuario	
Número: HUCR12	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de planes académicos	
Prioridad en Negocio: Media	**Requisitos:** HUCR07, HUCR08
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar los planes académicos, donde se registrará el nombre del plan, año del plan, cantidad de ciclos, tipo de curso y cursos, el plan se asignará a una facultad y a una carrera profesional.	
Validación: * *Todos los datos son requeridos*	

TABLA 16. HISTORIA DE USUARIO HUCR13

Historia de Usuario	
Número: HUCR13	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo de curso	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar el tipo de curso creado por las diferentes carreras profesionales (Obligatorio, electivo, extracurricular, actividad extracurricular, actividad o Practica Pre Profesional)	
Validación: * *Todos los datos son requeridos*	

TABLA 17. HISTORIA DE USUARIO HUCR14

Historia de Usuario	
Número: HUCR14	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo condición curso	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar el tipo de condición del curso (Dictado, No dictado, Inactivo)	
Validación: * *Todos los datos son requeridos*	

TABLA 18. HISTORIA DE USUARIO HUCR15

Historia de Usuario	
Número: HUCR15	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo estudio curso	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar el tipo de estudio de curso (Estudio General, Estudio Específico, Estudio de Especialidad, Actividad Extracurricular)	
Validación: * *Todos los datos son requeridos*	

TABLA 19. HISTORIA DE USUARIO HUCR16

Historia de Usuario	
Número: HUCR16	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo examen	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar el tipo de examen que realizará el docente al estudiante (Practicas , Trabajos Calificados, Talleres, Intervenciones Orales)	
Validación: * *Todos los datos son requeridos*	

TABLA 20. HISTORIA DE USUARIO HUCR17

Historia de Usuario	
Número: HUCR17	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación	
Prioridad en Negocio: Media	**Requisitos:** HUCR16
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar el tipo de sistema de evaluación	
Validación: * *Todos los datos son requeridos*	

TABLA 21. HISTORIA DE USUARIO HUCR18

Historia de Usuario	
Número: HUCR18	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de cantidad de ciclos	
Prioridad en Negocio: Media	**Requisitos:** Ninguno
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar la cantidad de ciclos	
Validación: * *Todos los datos son requeridos*	

TABLA 22. HISTORIA DE USUARIO HUCR19

Historia de Usuario	
Número: HUCR19	**Usuario:** User01
Nombre Historia: Creación, lectura, modificación y deshabilitación de cursos	
Prioridad en Negocio: Alta	**Requisitos:** HUCR12, HUCR13, HUCR14, HUCR15, HUCR17, HUCR18
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Planes	
Descripción: Como administrador del sistema requiero registrar, actualizar y/o deshabilitar los cursos, donde se debe registrar el código del curso, el ciclo a la que pertenece el curso, la cantidad de créditos del curso, el tipo de curso, tipo condición del curso, tipo estudio curso, horas de teoría, horas práctica, sistema de evaluación del curso, requisito del curso, la carrera profesional a la que pertenece el curso, plan a la que pertenece el curso, facultad a la que pertenece el curso, ciclo desde que inicia a dictarse el curso, horas de laboratorio, horas de taller, horas otros	
Validación: * *Todos los datos son requeridos*	

TABLA 23. HISTORIA DE USUARIO HUCR20

Historia de Usuario	
Número: HUCR20	**Usuario:** User01
Nombre Historia: Registro de asignación de cursos por los estudiantes (Matricula)	
Prioridad en Negocio: Alta	**Requisitos:** HUCR03, HUCR19
Proceso: PC.01	**Iteración Asignada:** 1
Módulo Asignado: Matricula	
Descripción: Como estudiante requiero ingresar al sistema por medio de mi código de estudiante y una contraseña y poder visualizar mis datos personales, mi avance curricular, y los cursos que podré matricularme, al momento de seleccionar un curso requiero asignar un grupo disponible y confirmar los cursos que registre El sistema debe permitir la asignación de los cursos al estudiante de acuerdo al ciclo relativo que se encuentra, los créditos disponibles que tiene y la situación en que encuentre el estudiante	
Validación: * *Todos los datos son requeridos*	

4.3.2. Lista de historias de usuario por orden de importancia (Product Backlog).

Al tomar nota, de todos los requerimientos que requiere el PRODUCT OWNER, se ordenaron por tipo de prioridad e importancia, la cual fue asignada por el PRODUCT OWNER conjuntamente con el equipo del proyecto en reuniones, para determinar la cantidad o estimación de días para realizar el sistema solicitado, de ahora en adelante PRODUCT BACKLOG.

TABLA 24. TAREAS POR ORDEN DE PRIORIDAD (PRODUCT BACKLOG)

Módulo	Historia de Usuario	Prioridad	Importancia	Tiempo Estimado
SEG	Autentificación de los usuarios al sistema	Alta	100	5
SEG	Creación, lectura, modificación y deshabilitación de usuarios	Alta	98	5
ING	Creación, lectura, modificación y deshabilitación de estudiantes	Alta	98	5
SEG	Creación, lectura, modificación y deshabilitación de perfiles	Alta	95	5
SEG	Asignación, modificación y deshabilitación de usuarios a los perfiles	Alta	93	5
SEG	Asignación, modificación y deshabilitación de estudiantes a los perfiles	Alta	93	5
PLA	Creación, lectura, modificación y deshabilitación de Facultades	Media	83	6
PLA	Creación, lectura, modificación y deshabilitación de Carreras Profesionales	Media	80	6
PLA	Creación, lectura, modificación y deshabilitación de tipo condición docente	Media	75	6
PLA	Creación, lectura, modificación y deshabilitación de tipo categoría docente	Media	75	6
ESC	Creación, lectura, modificación y deshabilitación de docentes	Media	75	6
PLA	Creación, lectura, modificación y deshabilitación de planes académicos	Media	73	5
PLA	Creación, lectura, modificación y deshabilitación de tipo de curso	Media	72	5
PLA	Creación, lectura, modificación y deshabilitación de tipo condición curso	Media	72	5

Módulo	Historia de Usuario	Prioridad	Importancia	Tiempo Estimado
PLA	Creación, lectura, modificación y deshabilitación de tipo estudio curso	Media	72	5
PLA	Creación, lectura, modificación y deshabilitación de tipo examen	Media	72	5
PLA	Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación	Media	72	5
PLA	Creación, lectura, modificación y deshabilitación de cantidad de ciclos	Media	72	5
PLA	Creación, lectura, modificación y deshabilitación de cursos	Media	72	5
MAT	Registro de asignación de cursos por los estudiantes (Matricula)	Media	70	15
CTE	Creación, lectura, modificación y deshabilitación de pagos	Media	65	5

4.3.3. Definición de los Sprints

Se define la velocidad de desarrollo de cada Sprint según la importancia de las historias de usuario y el tiempo de trabajo del equipo scrum para el proyecto y la dedicación que se le dará al mismo.

El tiempo del equipo de trabajo está dado dentro de las jornadas laborales de 8 horas a la semana de lunes a viernes y sábados 4 horas durante 6 meses, de los cuales, se obtiene como resultado la cantidad de días de trabajo dedicados al proyecto por cada Sprint.

TABLA 25. TABLA DE DÍAS DE TRABAJO DEDICADO POR EL EQUIPO POR CADA SPRINT

Equipo Scrum	Jornada Laboral	Horas de trabajo al proyecto por día	Horas de trabajo al proyecto por semana	Semanas de trabajo por mes	Total de horas	Total de días laborables para el proyecto
Jorge Trigueros	8 horas	6 horas	34 horas	4 semanas	136 horas	17 días
José Congona	8 horas	6 horas	34 horas	4 semanas	136 horas	17 días
Total de días disponibles para el proyecto				34 días		

Debido al tiempo de dedicación que se le dará al proyecto y las horas asignadas dentro de horario de trabajo se esperan tener algunas distracciones e impedimentos pero que están dentro de las estimaciones para el proyecto, por lo cual, el Product Owner da un factor de dedicación del 90% del tiempo comprendido para el mismo.

Velocidad estimada del Sprint = Días hombre disponible x Factor de dedicación

30.6 = 34 x 90%

De acuerdo a la velocidad obtenida para la ejecución de cada sprint y tomando en cuenta el nivel de importancia definido por cada historia de usuario se procede a agrupar las mismas y determinar la cantidad de sprints para el proyecto, en donde se obtiene.

TABLA 26. TABLA DE ESTIMACIÓN DE SPRINT 1

Sprint 1				
Módulo	**Historia de Usuario**	**Prioridad**	**Importancia**	**Tiempo Estimado**
SEG	Autentificación de los usuarios al sistema	Alta	100	5 días
SEG	Creación, lectura, modificación y deshabilitación de usuarios	Alta	98	5 días
ING	Creación, lectura, modificación y deshabilitación de estudiantes	Alta	98	5 días
SEG	Creación, lectura, modificación y deshabilitación de perfiles	Alta	95	5 días

SEG	Asignación, modificación y deshabilitación de usuarios a los perfiles	Alta	93	5 días
SEG	Asignación, modificación y deshabilitación de estudiantes a los perfiles	Alta	93	5 días
Total de días del Sprint				**30 días**

TABLA 27. TABLA DE ESTIMACIÓN DE SPRINT 2

Sprint 2				
Módulo	**Historia de Usuario**	**Prioridad**	**Importancia**	**Tiempo Estimado**
PLA	Creación, lectura, modificación y deshabilitación de Facultades	Media	83	6 días
PLA	Creación, lectura, modificación y deshabilitación de Carreras Profesionales	Media	80	6 días
PLA	Creación, lectura, modificación y deshabilitación de tipo condición docente	Media	75	6 días
PLA	Creación, lectura, modificación y deshabilitación de tipo categoría docente	Media	75	6 días
ESC	Creación, lectura, modificación y deshabilitación de docentes	Media	75	6 días
Total de días del Sprint				**30 días**

TABLA 28. TABLA DE ESTIMACIÓN DE SPRINT 3

Sprint 3				
Módulo	**Historia de Usuario**	**Prioridad**	**Importancia**	**Tiempo Estimado**
PLA	Creación, lectura, modificación y deshabilitación de planes académicos	Media	73	5 días
PLA	Creación, lectura, modificación y deshabilitación de tipo de curso	Media	72	5 días

PLA	Creación, lectura, modificación y deshabilitación de tipo condición curso	Media	72	5 días
PLA	Creación, lectura, modificación y deshabilitación de tipo estudio curso	Media	72	5 días
PLA	Creación, lectura, modificación y deshabilitación de tipo examen	Media	72	5 días
PLA	Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación	Media	72	5 días
Total de días del Sprint				**30 días**

TABLA 29. TABLA DE ESTIMACIÓN DE SPRINT 4

Sprint 4				
Módulo	**Historia de Usuario**	**Prioridad**	**Importancia**	**Tiempo Estimado**
PLA	Creación, lectura, modificación y deshabilitación de cantidad de ciclos	Media	72	5 días
PLA	Creación, lectura, modificación y deshabilitación de cursos	Media	72	5 días
MAT	Registro de asignación de cursos por los estudiantes (Matricula)	Media	70	15 días
CTE	Creación, lectura, modificación y deshabilitación de pagos	Media	65	5 días
Total de días del Sprint				**30 días**

De acuerdo a la velocidad estimada de cada sprint el desarrollo del aplicativo web se ejecutará en 4 sprints, los mismos que han sido organizados por la importancia de cada una de las historias de usuario y por el tiempo de duración de cada una de las mismas.

4.4. Planificación de los Sprints

Para el desarrollo de cada SPRINT se han planificado revisiones y entregables para validar los avances obtenidos del desarrollo programado

y así generar de manera retrospectiva las acciones de mejora para los siguientes desarrollos.

Por cada Sprint se mostrarán los avances a través del Taskboard, donde se apreciaran las actividades desarrolladas, pendientes y finalizadas por cada historia de usuarios; además de mostrar el BURNDOWN para ver la velocidad en la que se está desarrollando el proyecto y determinar cuáles son las historias o actividades que están demandando mucho tiempo al desarrollo del proyecto o si las historias de usuario tiene pocas actividades de desarrollo y se están perdiendo recursos en ello.

Para validar la funcionalidad o conformidad de la elaboración de cada historia de usuario se realizarán prueba de funcionalidad por cada historia de usuario y ver los aciertos y desaciertos de los mismos, los cuales se verán reflejados en el informe de cierre de cada sprint. A continuación se procede a detallar la planificación de cada sprint indicando las fechas de revisión e historias de usuarios comprendidas.

Sprint 1

TABLA 30. PLANIFICACIÓN DEL SPRINT 1

Fecha de Inicio:	04/11/2019
Fecha Fin:	09/12/2019
Revisión de los avances:	Las revisiones se realizarán semanalmente. Las fechas de la revisión fueron las siguientes: 09/11/2019 16/11/2019 23/11/2019 30/11/2019

	09/12/2019
Tareas a desarrollar	Autentificación de los usuarios al sistema Creación, lectura, modificación y deshabilitación de usuarios Creación, lectura, modificación y deshabilitación de estudiantes Creación, lectura, modificación y deshabilitación de perfiles Asignación, modificación y deshabilitación de usuarios a los perfiles Asignación, modificación y deshabilitación de estudiantes a los perfiles

Sprint 2

TABLA 31. PLANIFICACIÓN DEL SPRINT 2

Fecha de Inicio:	09/12/2019
Fecha Fin:	14/01/2020
Revisión de los avances:	Las revisiones se realizarán semanalmente. Las fechas de la revisión fueron las siguientes: 14/12/2019 21/12/2019 28/12/2019 03/01/2020 14/01/2020
Tareas a desarrollar	Creación, lectura, modificación y deshabilitación de Facultades Creación, lectura, modificación y deshabilitación de Carreras Profesionales

	Creación, lectura, modificación y deshabilitación de tipo condición docente Creación, lectura, modificación y deshabilitación de tipo categoría docente Creación, lectura, modificación y deshabilitación de docentes

Sprint 3

TABLA 32. PLANIFICACIÓN DEL SPRINT 3

Fecha de Inicio:	15/01/2020
Fecha Fin:	19/02/2020
Revisión de los avances:	Las revisiones se realizarán semanalmente. Las fechas de la revisión fueron las siguientes: 18/01/2020 25/01/2020 01/02/2020 08/02/2020 15/02/2020 19/02/2020
Tareas a desarrollar	Creación, lectura, modificación y deshabilitación de planes académicos Creación, lectura, modificación y deshabilitación de tipo de curso

	Creación, lectura, modificación y deshabilitación de tipo condición curso Creación, lectura, modificación y deshabilitación de tipo estudio curso Creación, lectura, modificación y deshabilitación de tipo examen Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación

Sprint 4

TABLA 33. PLANIFICACIÓN DEL SPRINT 4

Fecha de Inicio:	24/02/2020
Fecha Fin:	31/03/2020
Revisión de los avances:	Las revisiones se realizarán semanalmente. Las fechas de la revisión fueron las siguientes: 29/02/2020 07/03/2020 14/03/2020 21/03/2020 26/03/2020 31/03/2020
Tareas a desarrollar	

	Creación, lectura, modificación y deshabilitación de cantidad de ciclos Creación, lectura, modificación y deshabilitación de cursos Registro de asignación de cursos por los estudiantes (Matricula) Creación, lectura, modificación y deshabilitación de pagos

4.5. Taskboard inicial

Se presenta el taskboard de desarrollo inicial del proyecto con todas las historias y la condición inicial de cada uno de los sprints.

TABLA 34. TASKBOARD INICIAL DEL DESARROLLO

FECHA DE INICIO: 04/11/2019 FECHA FIN: 31/03/2020				
	Historia de Usuario	**Pendiente**	**En Curso**	**Hecho**
Sprint Nº 1	Autentificación de los usuarios al sistema	✓		
	Creación, lectura, modificación y deshabilitación de usuarios	✓		
	Creación, lectura, modificación y deshabilitación de estudiantes	✓		
	Creación, lectura, modificación y deshabilitación de perfiles	✓		
	Asignación, modificación y deshabilitación de usuarios a los perfiles	✓		
	Asignación, modificación y deshabilitación de estudiantes a los perfiles	✓		
Sprint Nº 2	Creación, lectura, modificación y deshabilitación de Facultades	✓		
	Creación, lectura, modificación y deshabilitación de Carreras Profesionales	✓		
	Creación, lectura, modificación y deshabilitación de tipo condición docente	✓		

	Creación, lectura, modificación y deshabilitación de tipo categoría docente	✓		
	Creación, lectura, modificación y deshabilitación de docentes	✓		
Sprint Nº 3	Creación, lectura, modificación y deshabilitación de planes académicos	✓		
	Creación, lectura, modificación y deshabilitación de tipo de curso	✓		
	Creación, lectura, modificación y deshabilitación de tipo condición curso	✓		
	Creación, lectura, modificación y deshabilitación de tipo estudio curso	✓		
	Creación, lectura, modificación y deshabilitación de tipo examen	✓		
	Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación	✓		
Sprint Nº 4	Creación, lectura, modificación y deshabilitación de cantidad de ciclos	✓		
	Creación, lectura, modificación y deshabilitación de cursos	✓		
	Registro de asignación de cursos por los estudiantes (Matricula)	✓		
	Creación, lectura, modificación y deshabilitación de pagos	✓		

4.6. Desarrollo de los Sprints

Para una mejor planificación de cada uno de los SPRINTS del cuadro anterior, se ha realizado una tabla en donde nos permitirá conocer las principales tareas que correspondan a cada SPRINT, fecha en que inicia y fecha fin del SPRINT, el desarrollo y estado en que se va encontrando mientras avanza cada iteración

4.6.1. Sprint 1

En este sprint se realizan las tareas de configuración y de inicialización del sistema, los controles de acceso en base a un usuario y contraseña (credenciales), creación de perfiles, creación de usuarios, asignación de

perfil a usuarios, todos estos mantenimientos pertenecen al módulo de seguridad del sistema.

a) Priorización de Sprint 1

Las tareas asignadas para este sprint son de procesos del sistema priorizados en base a la línea de ejecución del usuario para ir avanzando en el proceso de ingreso de información, por tal motivo las primeras actividades que se han realizado son la creación de usuarios de la Universidad Nacional de Ucayali, creación de los perfiles y que serán definidos para cada usuario creado con el fin de que cada perfil tenga acceso a cierta información o módulo del sistema, por tal motivo y siguiendo la línea de acción del sistema para su desarrollo y funcionamiento lo primero fue crear estas tareas específicas.

b) Formato de Asignación del Sprint 1

TABLA 35. ESPECIFICACIÓN SPRINT 1

SPRINT 01
FECHA DE INICIO: 04/11/2019
FECHA DE CULMINACIÓN: 09/12/2019

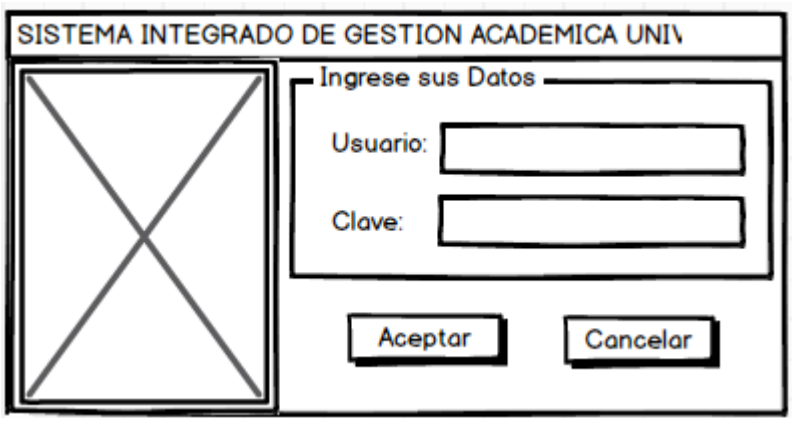
Tareas Pendientes: 0
Días Pendientes: 0

Prioridad del Negocio	Descripción de la tabla	Responsable	Días	Estado
ALTA	Autentificación de los usuarios al sistema	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de usuarios	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de estudiantes	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de perfiles	Jorge Trigueros	5	TERMINADO

ALTA	Asignación, modificación y deshabilitación de usuarios a los perfiles	Jorge Trigueros	5	TERMINADO
ALTA	Asignación, modificación y deshabilitación de estudiantes a los perfiles	Jorge Trigueros	5	TERMINADO

Prototipo Autentificación Usuario

ILUSTRACIÓN 7. PROTOTIPO AUTENTIFICACIÓN DE USUARIO



c) Prototipo CRUD Usuarios

ILUSTRACIÓN 8. PROTOTIPO BUSCAR USUARIO

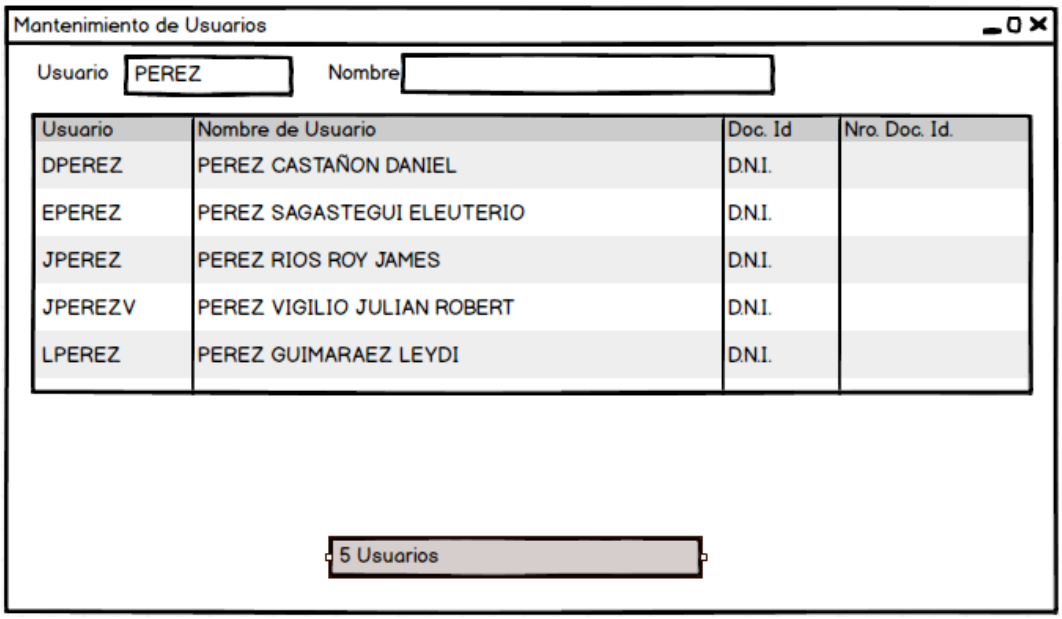


ILUSTRACIÓN 9. PROTOTIPO MODIFICAR USUARIO

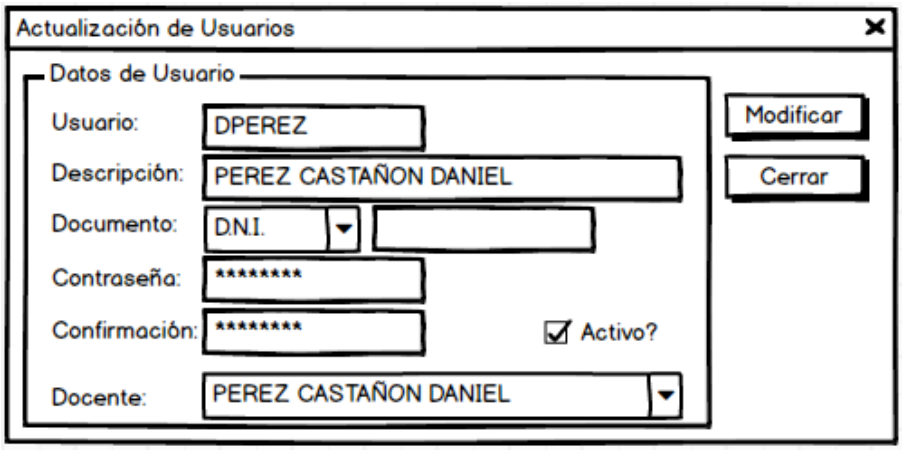


ILUSTRACIÓN 10. PROTOTIPO NUEVO USUARIO

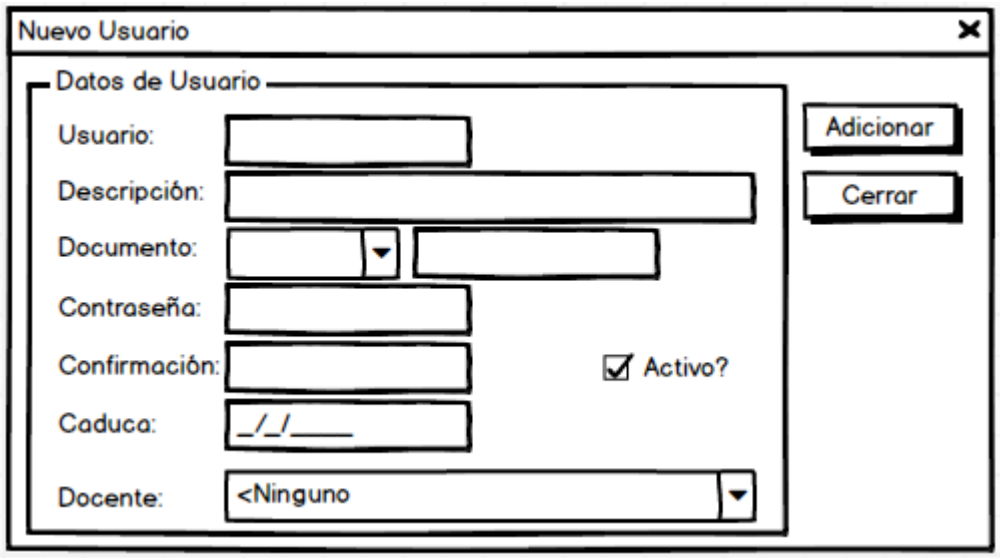
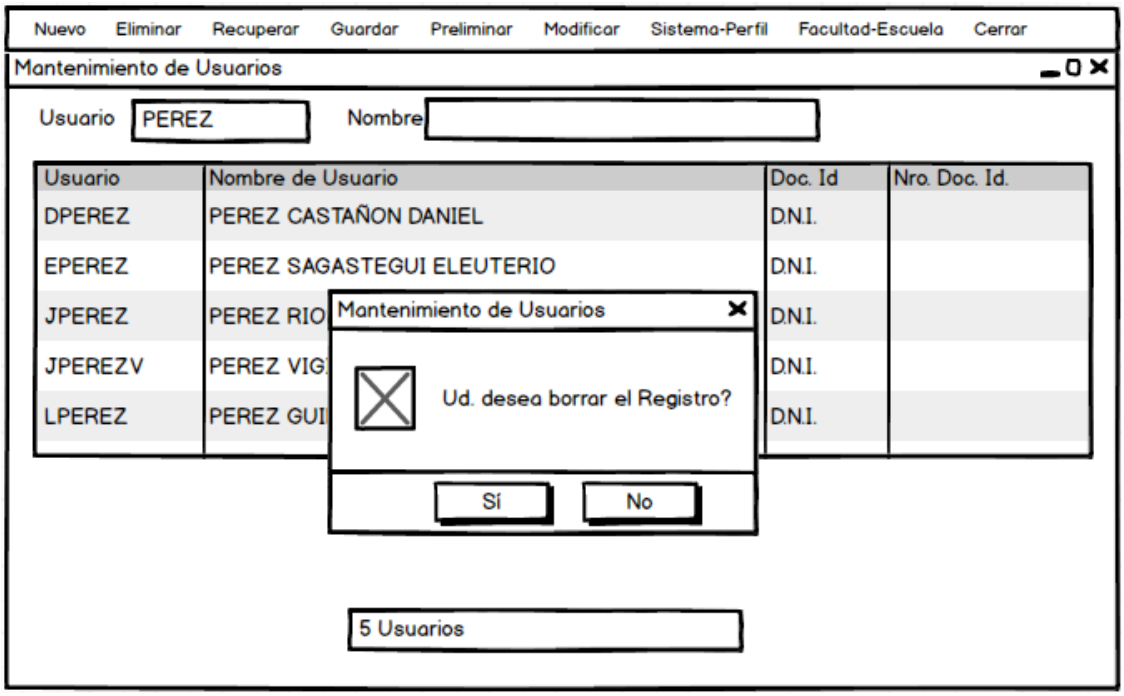


ILUSTRACIÓN 11. PROTOTIPO ELIMINAR USUARIO



d) Prototipo CRUD Perfil

ILUSTRACIÓN 12. PROTOTIPO NUEVO PERFIL

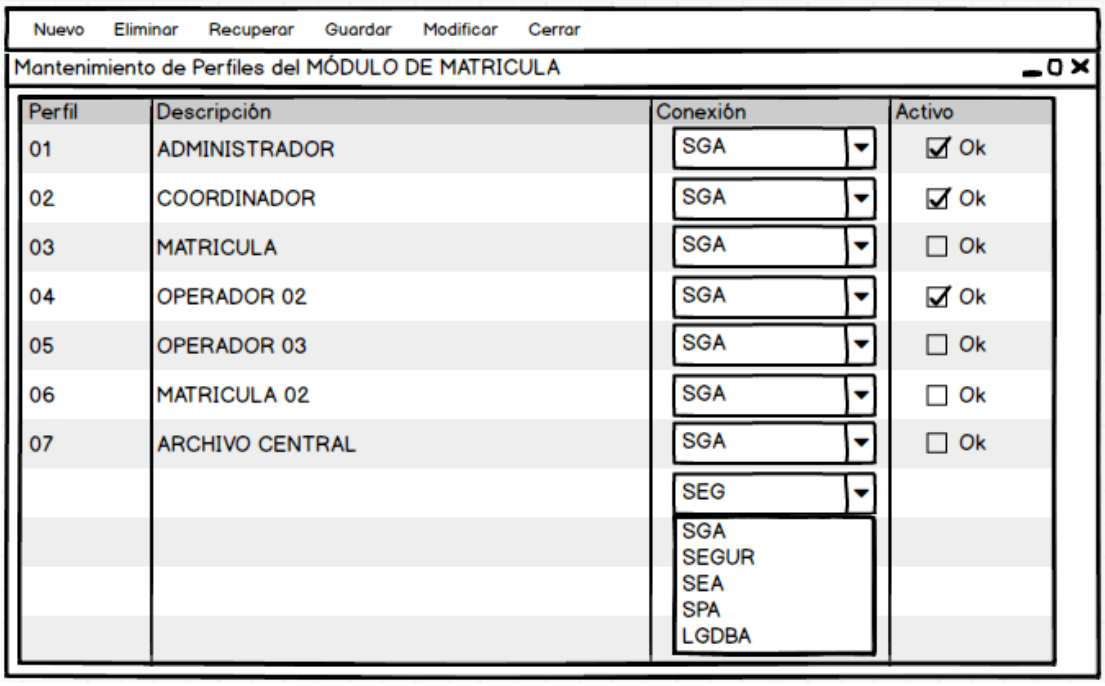


ILUSTRACIÓN 13. PROTOTIPO ELIMINAR PERFIL

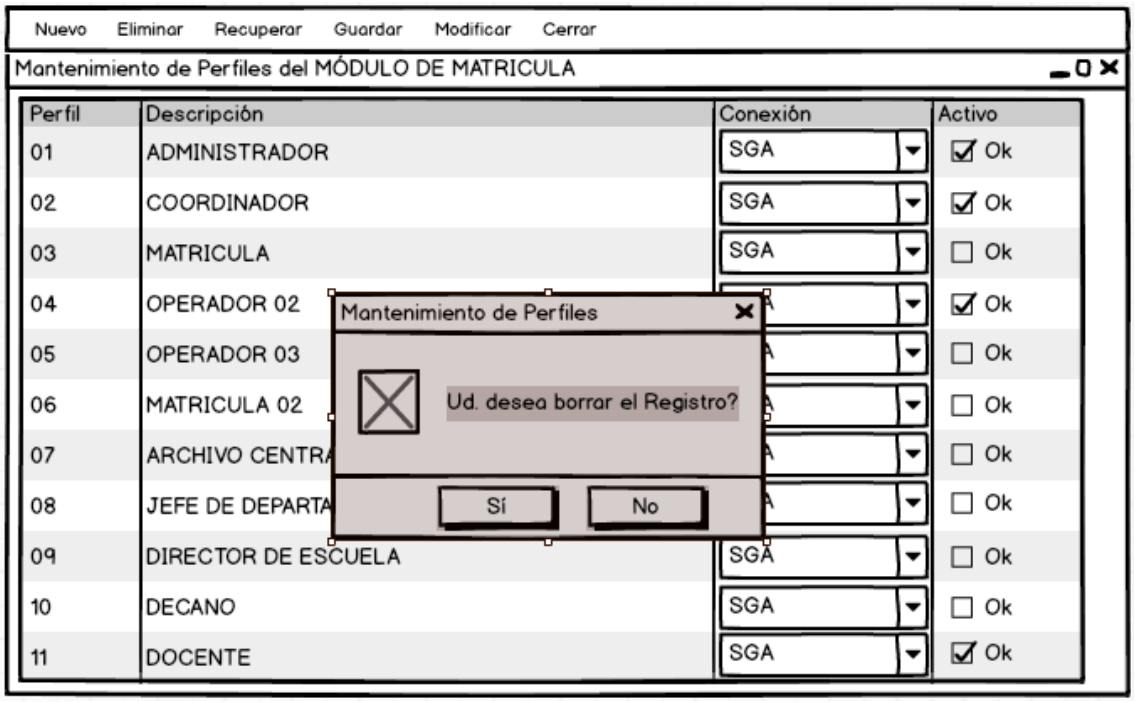


ILUSTRACIÓN 14. PROTOTIPO LISTAR PERFIL

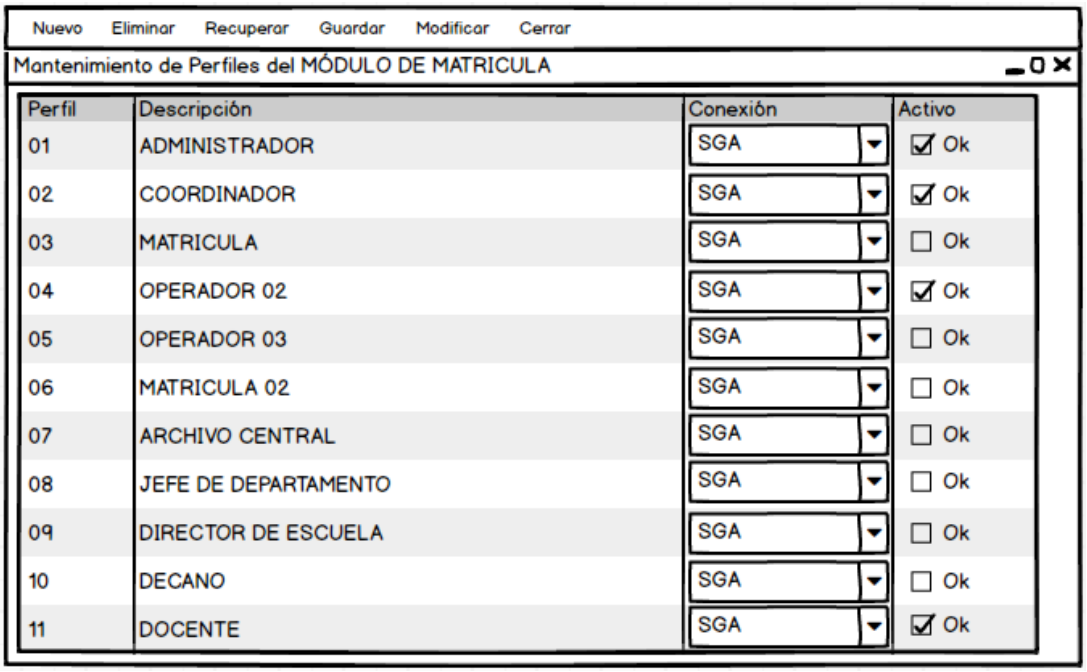
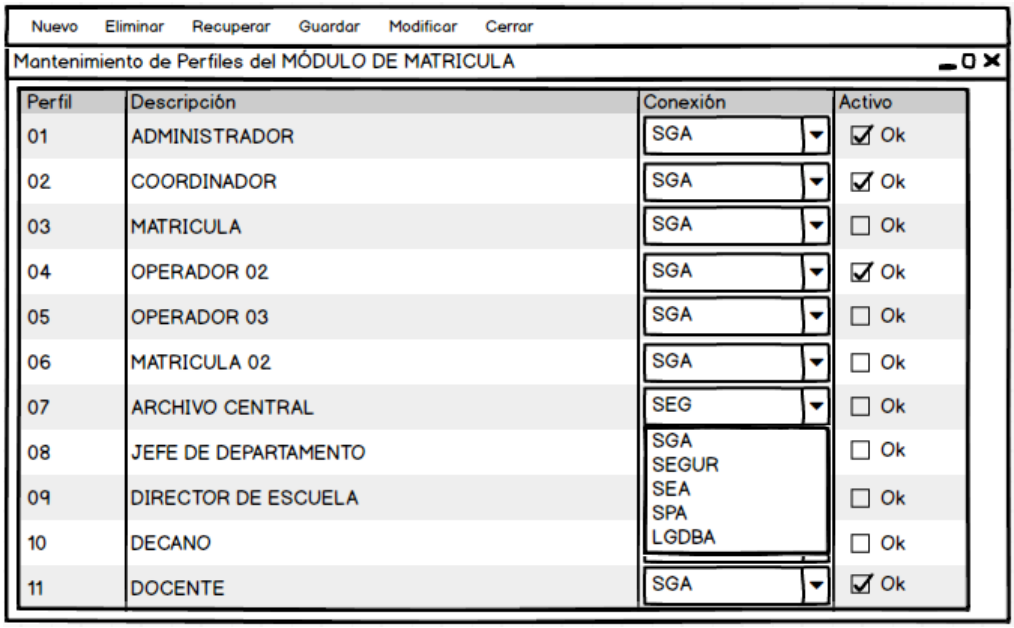
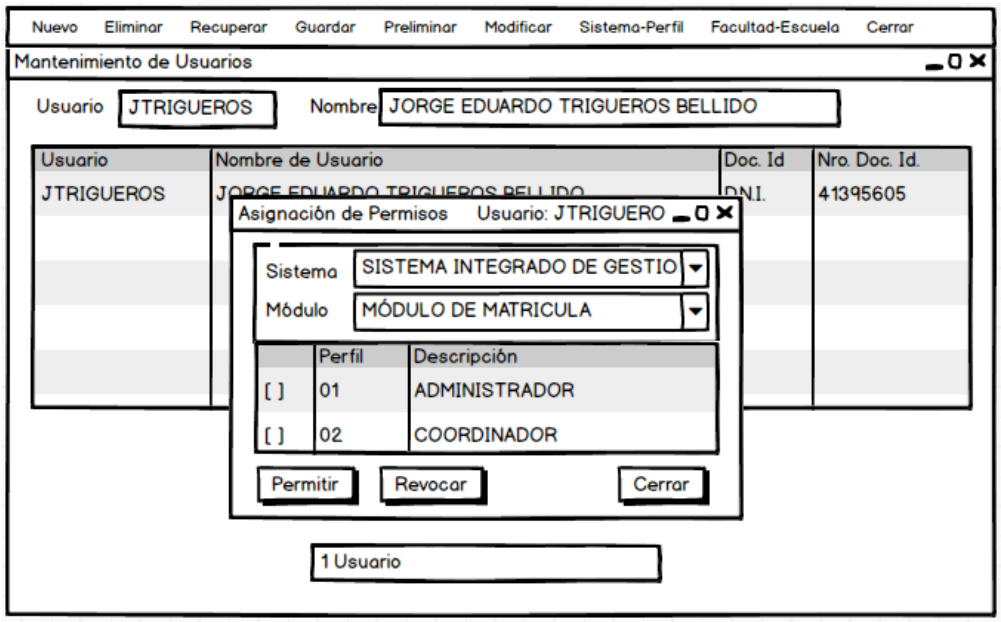


ILUSTRACIÓN 15. PROTOTIPO MODIFICAR PERFIL



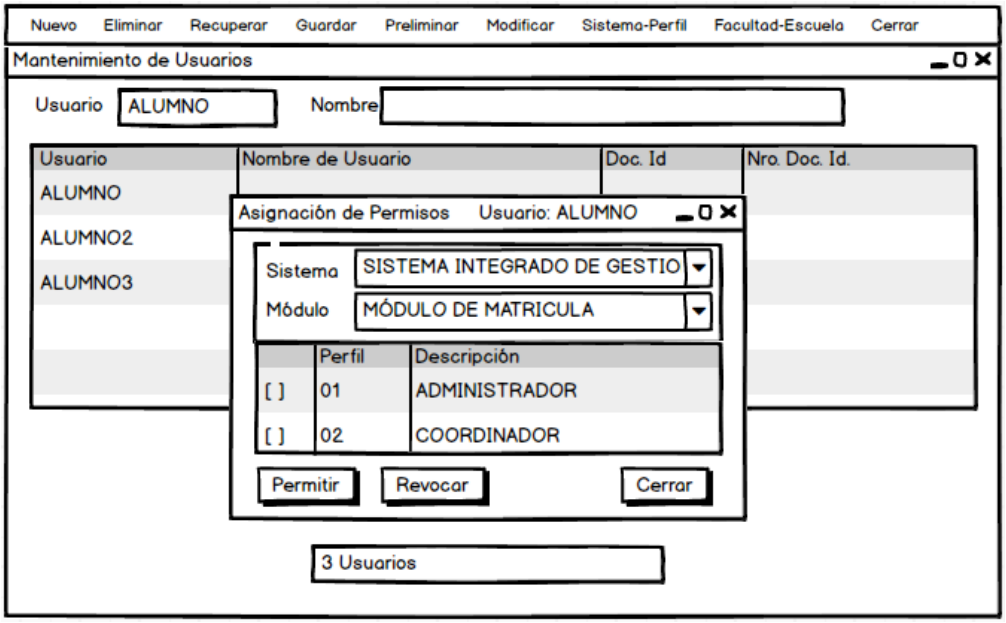
e) Prototipo Asignación, modificación y deshabilitación de usuarios a los perfiles

ILUSTRACIÓN 16. PROTOTIPO ASIGNACIÓN PERFIL A USUARIO



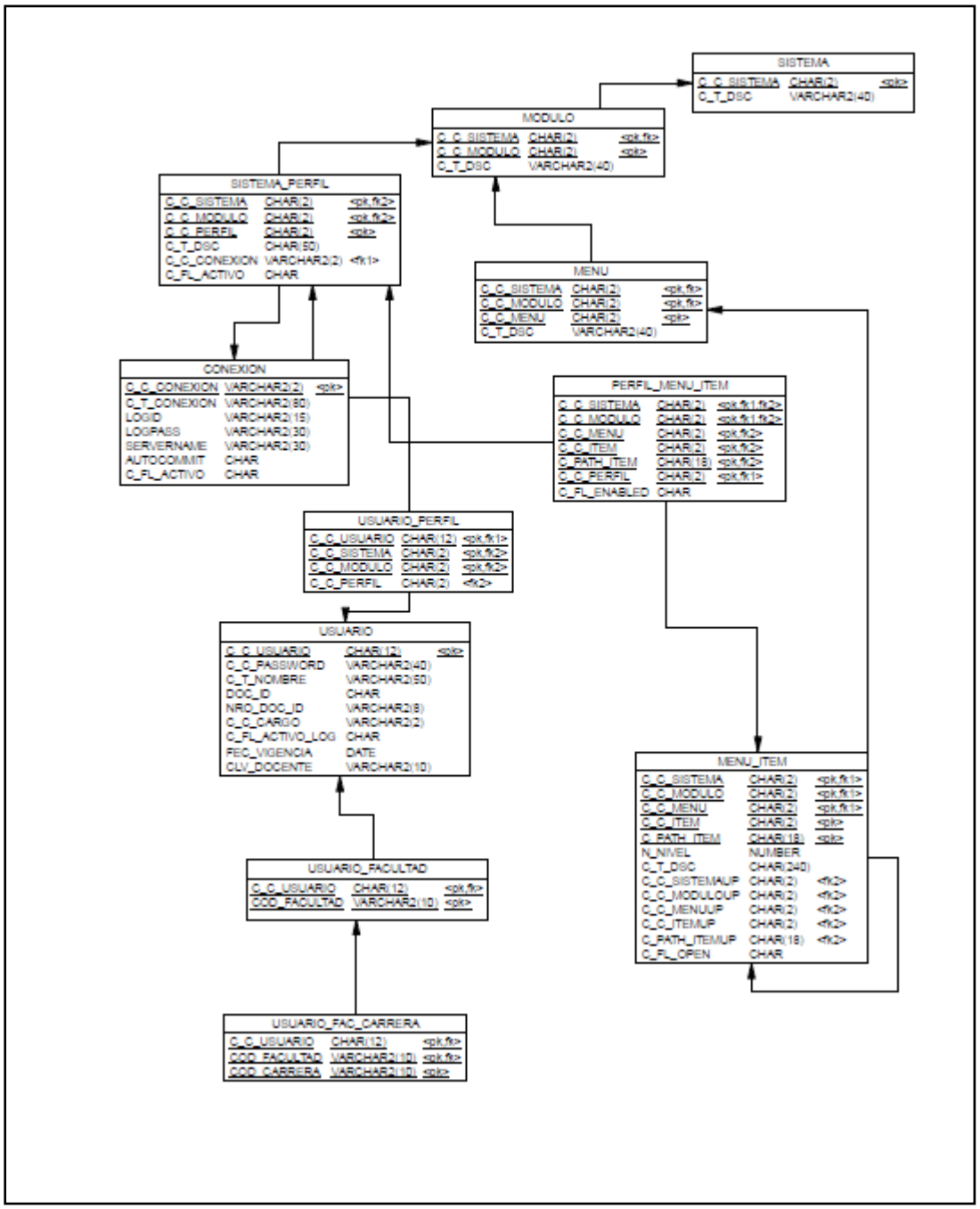
f) Prototipo Asignación, modificación y deshabilitación de alumnos a los perfiles

ILUSTRACIÓN 17. PROTOTIPO ASIGNACIÓN PERFIL A USUARIO



g) Modelo físico SPRINT 1

ILUSTRACIÓN 18. MODELO FÍSICO PERFIL A USUARIO



h) Fuente SPRINT 1

(Ver Anexo 6)

i) Interfaz autentificación de usuario en el sistema

ILUSTRACIÓN 19. INTERFAZ DE ACCESO AL SISTEMA



j) Interfaz CRUD Usuarios

ILUSTRACIÓN 20. INTERFAZ DE BUSCAR USUARIO

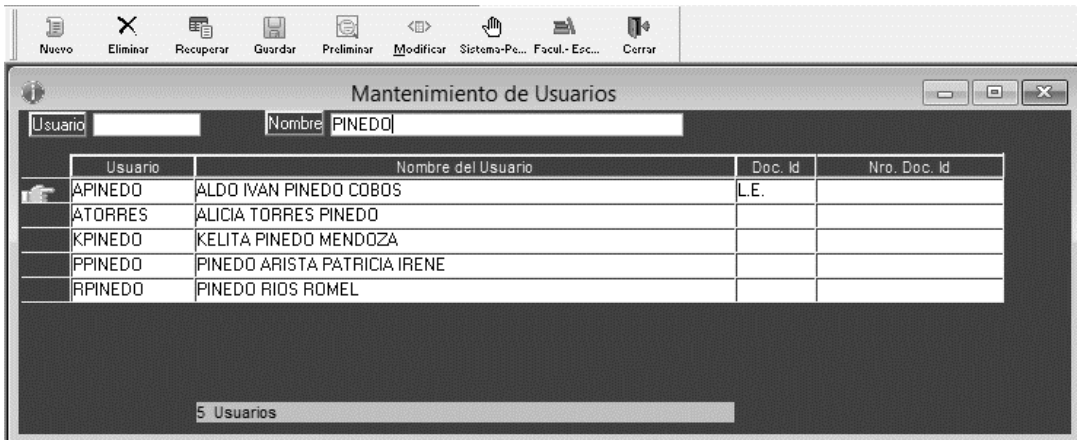


ILUSTRACIÓN 21. INTERFAZ DE MODIFICAR USUARIO

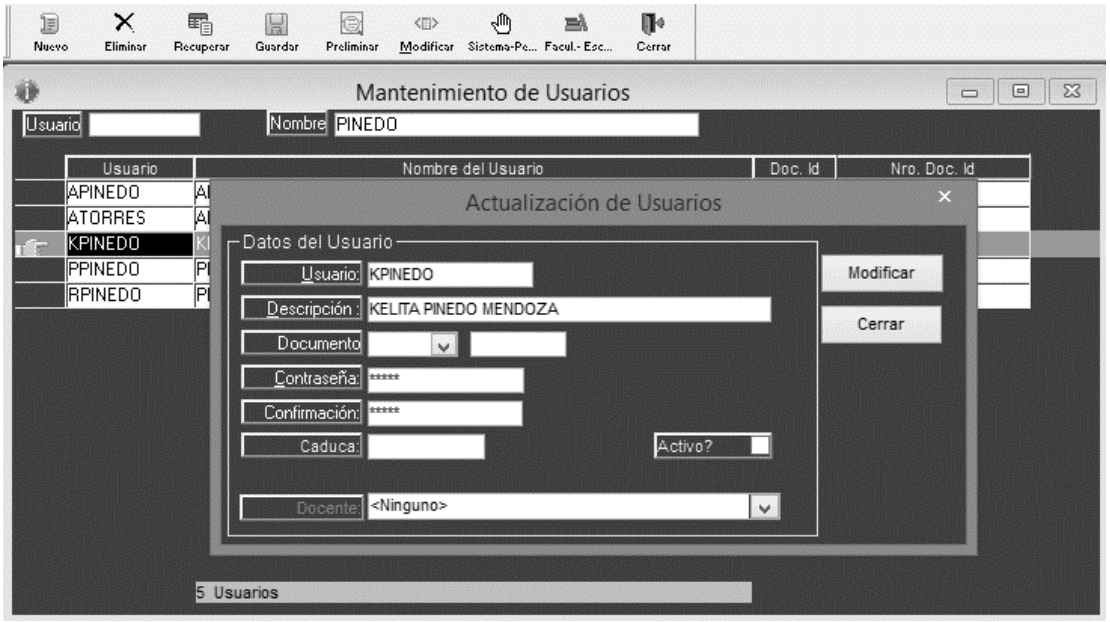


ILUSTRACIÓN 22. INTERFAZ DE NUEVO USUARIO

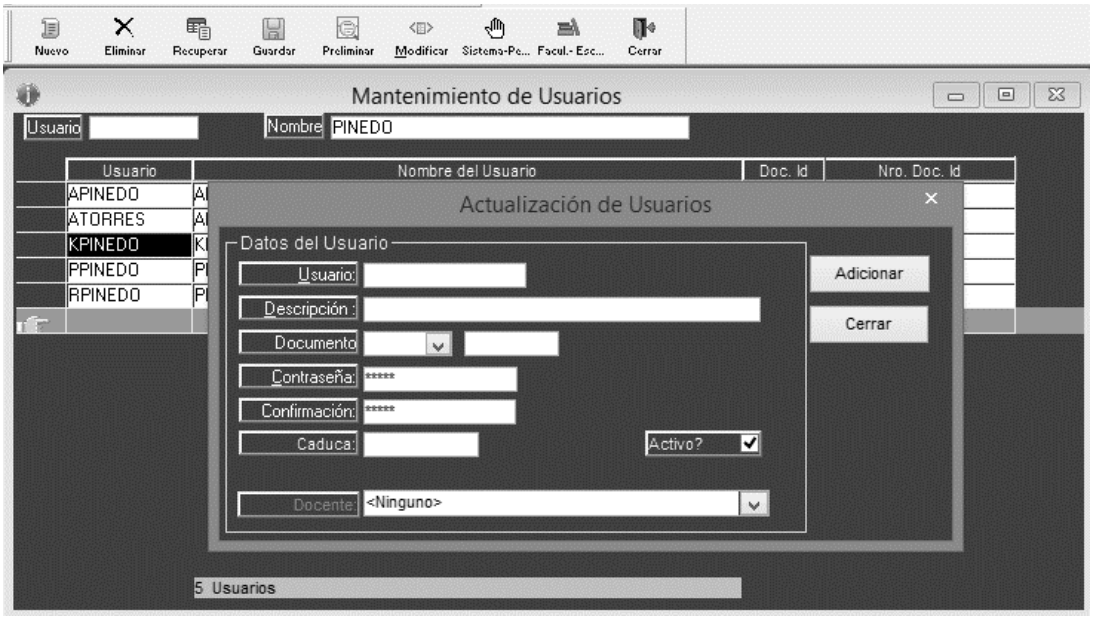
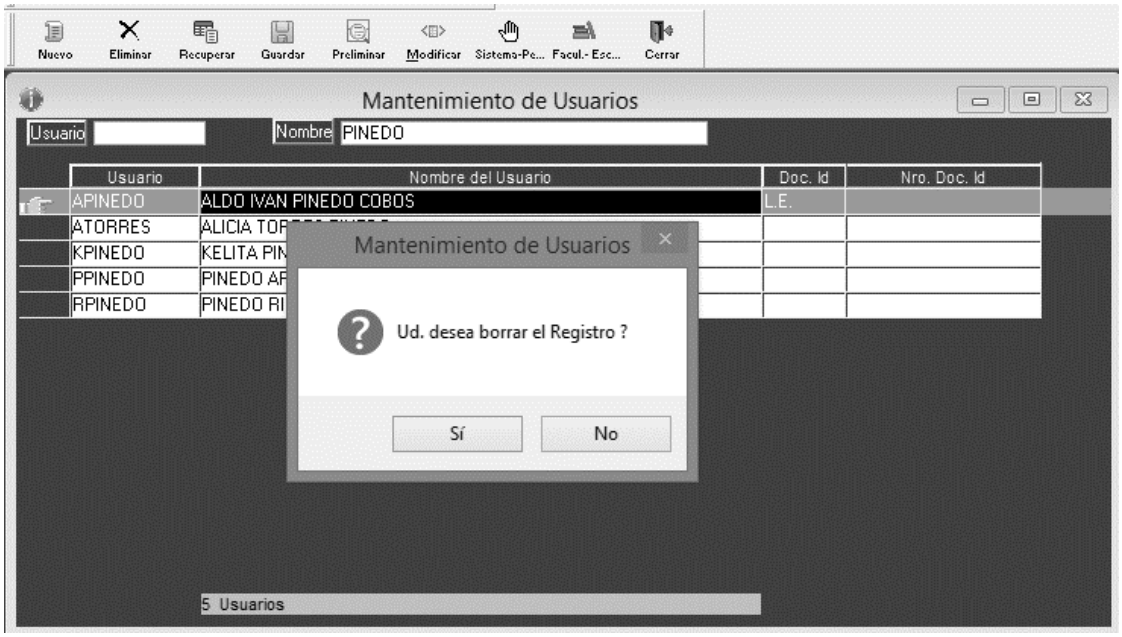


ILUSTRACIÓN 23. INTERFAZ DE ELIMINAR USUARIO



k) Interfaz CRUD Perfil

ILUSTRACIÓN 24. INTERFAZ DE NUEVO PERFIL

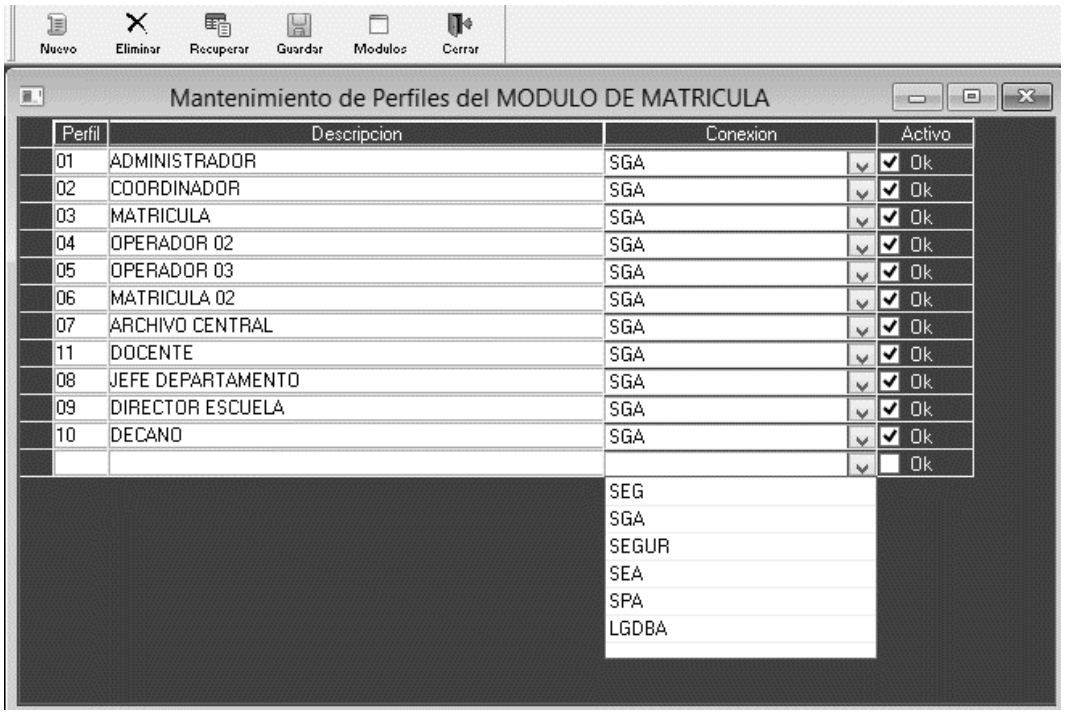


ILUSTRACIÓN 25. INTERFAZ DE ELIMINAR PERFIL

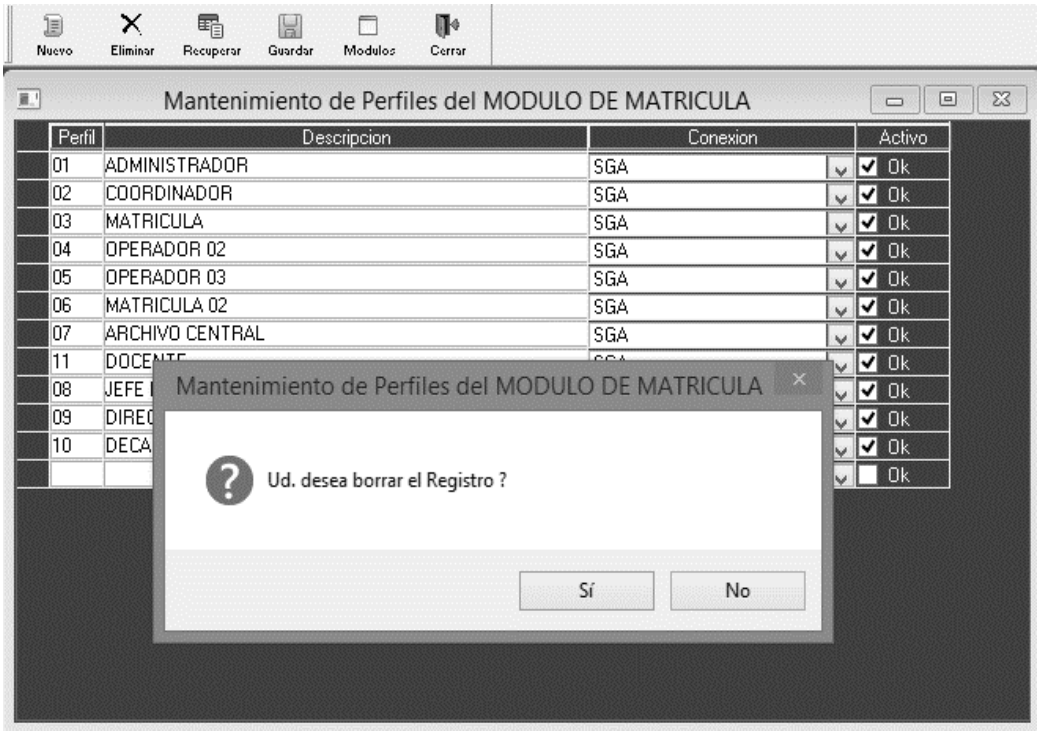


ILUSTRACIÓN 26. INTERFAZ DE LISTAR PERFIL

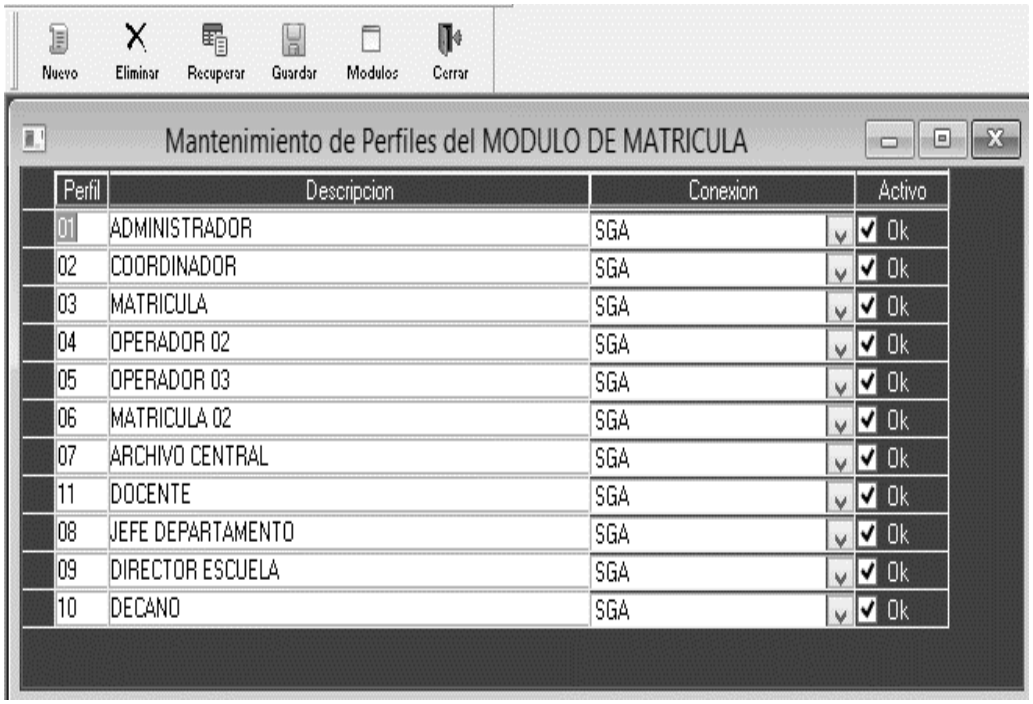
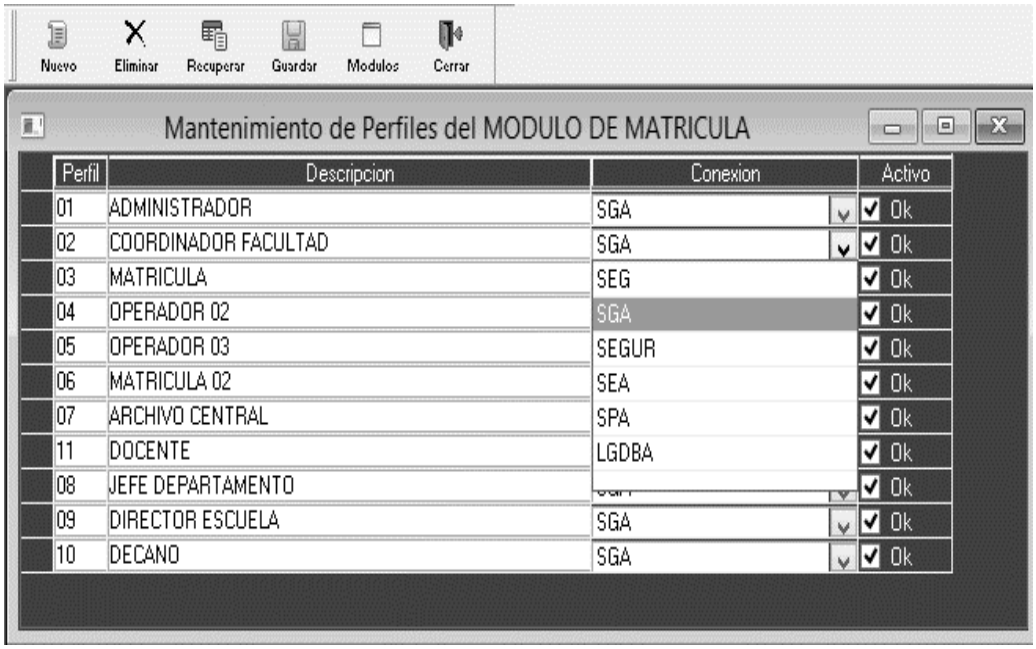
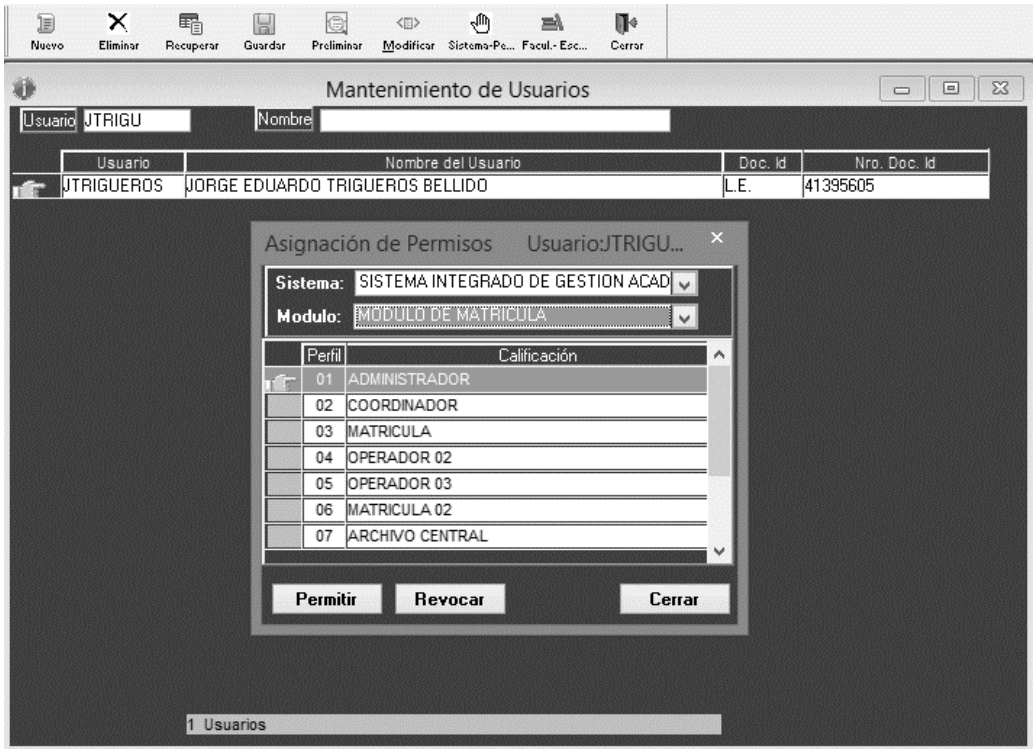


ILUSTRACIÓN 27. INTERFAZ DE MODIFICAR PERFIL



I) Interfaz Asignación Perfil Usuario

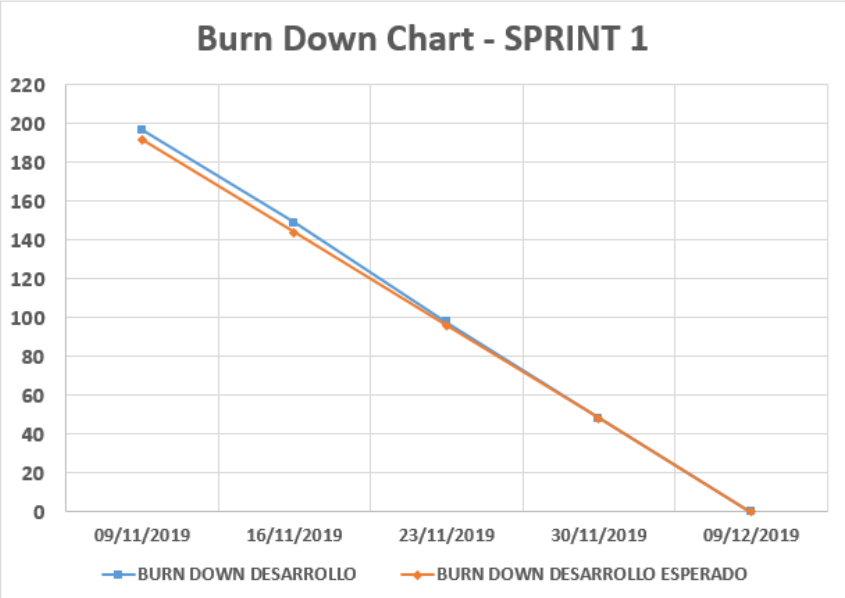
ILUSTRACIÓN 28. INTERFAZ DE ASIGNACIÓN, MODIFICACIÓN Y DESHABILITACIÓN DE PERFIL A USUARIO



m) Resultado Burn Down Chart – Sprint 1

En la siguiente ilustración se muestra el desarrollo y resultado de las 5 semanas del Sprint 1, donde se aprecia que el Burn Down de Desarrollo estaba por encima del Burn Down de desarrollo esperado, esto debido a que al iniciar el proyecto no se cumplieron con algunas tareas, sin embargo al concluir con la quinta semana se logró cumplir con el Burn Down Chart de desarrollo esperado, culminando con el SPRINT 1.

ILUSTRACIÓN 29. INTERFAZ DE ASIGNACIÓN, MODIFICACIÓN Y DESHABILITACIÓN DE PERFIL A USUARIO



n) Informe de prueba Funcional – SPRINT 1

TABLA 36. INFORME DE PRUEBA FUNCIONAL – SPRINT 1

PRUEBA FUNCIONAL – SPRINT 1			
PRUEBA Nº	Prueba de Funcionalidad Nº01	**VERSIÓN DE EJECUCIÓN**	UNU-SP01
TAREA:	Acceso al sistema (Login) Mantenimiento de Usuarios Mantenimiento de Estudiantes Mantenimiento de perfiles Asignación de perfiles a usuarios Asignación de perfiles a estudiantes	**FECHA DE EJECUCIÓN**	09/12/2019
		MÓDULO DEL SISTEMA	SEG ING
Descripción del caso de prueba:	Se procedió a realizar las pruebas de autentificación de usuarios, validando los campos cuando aparecen datos errados y los mensajes de respuesta, además se hicieron pruebas en los mantenimientos y asignaciones de usuarios, estudiantes y perfiles		

1. CASO DE PRUEBA

a. Precondiciones

No aplica

b. Pasos de prueba

- ✓ Ingresar datos no válidos para verificar los mensajes de rechazo y para validar los campos
- ✓ Validar que el acceso tenga éxito
- ✓ Realizar mantenimiento de usuarios, creando, modificando y actualizando la información.
- ✓ Realizar mantenimiento de estudiantes, creando, modificando y actualizando la información.
- ✓ Realizar mantenimiento de perfiles, creando, modificando y actualizando la información.
- ✓ Realizar asignaciones de perfiles, habilitando, modificando y deshabilitando a usuarios ya creados y registrados
- ✓ Realizar asignaciones de perfiles, habilitando, modificando y deshabilitando a ingresantes creados y registrados.

DATOS DE ENTRADA			**RESPUESTA ESPERADA DE LA APLICACIÓN**	**COINCIDE**		**RESPUESTA DEL SISTEMA**
CAMPO	**VALOR**	**TIPO ESCENARIO**		**SI**	**NO**	
Usuario	JTRIGUEROS	normal	Bienvenido al sistema	X		Acceso correcto, bienvenido al sistema
Usuario	JORGE	Prueba	Usuario o contraseña incorrectos	X		Usuario o contraseña incorrectos
	JTRIGUEROS	Normal	Los datos han sido actualizados	X		Se actualizan los datos
	0002160124	Normal	Los datos han sido actualizados	X		Se actualizan los datos, pero no muestra ningún mensaje
	ADMIN	normal	El perfil fue asignado	X		El perfil fue asignado al usuario

c. Post Condiciones

No aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones		**Veredicto**
	X	**PASO**
		FALLO

Observaciones	**Probador**
Al realizar las actualizaciones a la información del ingresante no muestra ningún mensaje de éxito.	**Firma:** **Nombre:**
	Fecha:

o) Revisión de Sprint 1 (Sprint Review Meeting)

TABLA 37. INFORME FINAL DE PRUEBA FUNCIONAL – SPRINT 1

Lugar:	Universidad Nacional de Ucayali	
Fecha	09/12/2019	
Número de Iteración / Sprint	Sprint 1	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
¿Qué salió bien en el Sprint? (aciertos)	**¿Qué no salió bien en el Sprint? (errores)**	**Lecciones aprendidas (recomendaciones)**
✓La recuperación de los datos entre el formulario login y la base de datos se ejecuta sin mayores complicaciones ✓La recuperación de datos entre los formularios de mantenimiento y la base de datos se ejecuta sin mayores complicaciones. ✓El Taskboard y el Burn down ayudan y aportan en gran medida que el equipo de trabajo se mantengan al tanto de las actividades que se realizan así como informar sobre el avance del proyecto	✓ El tiempo de ejecución de la primera historia de usuario tomo más tiempo de lo esperado, lo cual generó retrasos en el avance de las siguientes historias pero al final del sprint se logró terminar en el tiempo estimado ✓ Mensajes de error al momento de asignar dos tipos de perfiles a un usuario determinado	✓ Se sugiere siempre mantener actualizado el taskboard y el burn down para mantener informado al equipo y el mismo debe ser divulgado a todos los involucrados para no generar retrasos o se malinterpreten las necesidades del desarrollo

p) Reunión Retrospectiva – Sprint 1 (Retrospective Meeting)

Para evitar que el equipo siga cometiendo errores de retraso, se realiza una reunión de no más de dos horas al finalizar el Sprint, para detallar lo bueno, lo incómodo y dar ideas de mejora para el siguiente sprint (aportes del equipo)

TABLA 38. INFORME FINAL DE REUNIÓN RETROSPECTIVA – SPRINT 1

Lugar:	Universidad Nacional de Ucayali	
Fecha	09/12/2019	
Número de Iteración / Sprint	Sprint 1	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Lo bueno (Sprint culminado)	**Lo incomodo**	**Aportes del equipo (Próximo Sprint)**
✓ Conocer al equipo de trabajo	✓ No estar presentes todos en las reuniones establecidas	✓ Compartir conocimiento de la aplicación (Jorge Trigueros)
✓ La disciplina para la realización de las reuniones	✓ No hay puntualidad en las actividades	✓ Llegar temprano a la oficina y a las reuniones (José Congona)
✓ Integración del equipo al momento de realizar cualquier trabajo	✓ Aparatos electrónicos no necesarios en las reuniones de trabajo	✓ Compartir información de la Base de datos (Jorge Trigueros)
✓ Comunicación fluida y buen ambiente de trabajo	✓ Integración de equipo	✓ Buscar más información y organizar el conocimiento que se ha adquirido hasta el momento (Jorge Trigueros
✓ Buena experiencia con la programación	✓ Designación o delegación de otras funciones de la oficina de Tecnología de Información, Sistemas y Estadística de la Universidad Nacional de Ucayali, lo cual hacia que las tareas se presenten fuera de la fechas establecida.	✓ Implementar un desayuno los viernes (Pedro Campos)
	✓ Viajes inesperados a la filial Aguaytia.	

4.6.2. Sprint 2

El objetivo de este sprint es desarrollar las tareas para la continuidad del módulo de configuración del sistema y para ello es requerida la creación de facultades, carreras profesionales y docentes.

a) Priorización de Sprint 2

Las tareas elegidas para este sprint y el orden de priorización fueron tomadas en cuenta siguiendo el orden de ejecución del sistema y como forman parte del módulo de configuración después de que se haya creado los usuarios y perfiles, es necesario la creación de facultades, carreras profesionales y docentes.

b) Formato de asignación del Sprint 2

TABLA 39. ESPECIFICACIÓN SPRINT 2

SPRINT 02
FECHA DE INICIO: 09/12/2019
FECHA DE CULMINACIÓN: 14/01/2020

Tareas Pendientes: 0
Días Pendientes: 0

Prioridad del Negocio	Descripción de la tabla	Responsable	Días	Estado
ALTA	Creación, lectura, modificación y deshabilitación de Facultades	Jorge Trigueros	6	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de Carreras Profesionales	Jorge Trigueros	6	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo condición docente	Jorge Trigueros	6	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo categoría docente	Jorge Trigueros	6	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de docentes	Jorge Trigueros	6	TERMINADO

c) Prototipo CRUD Facultades

ILUSTRACIÓN 30. PROTOTIPO NUEVA FACULTAD

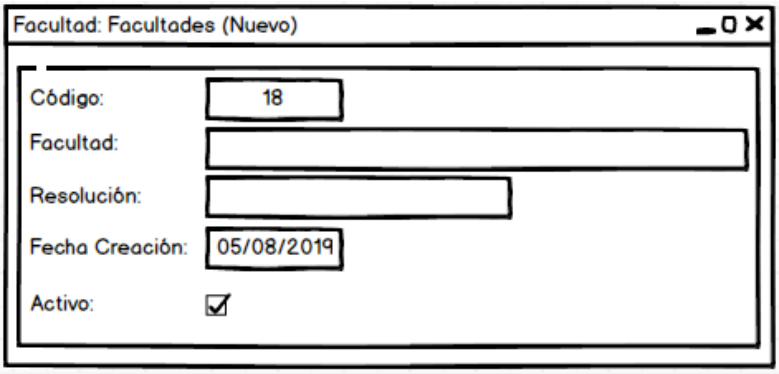


ILUSTRACIÓN 31. PROTOTIPO MODIFICAR FACULTAD

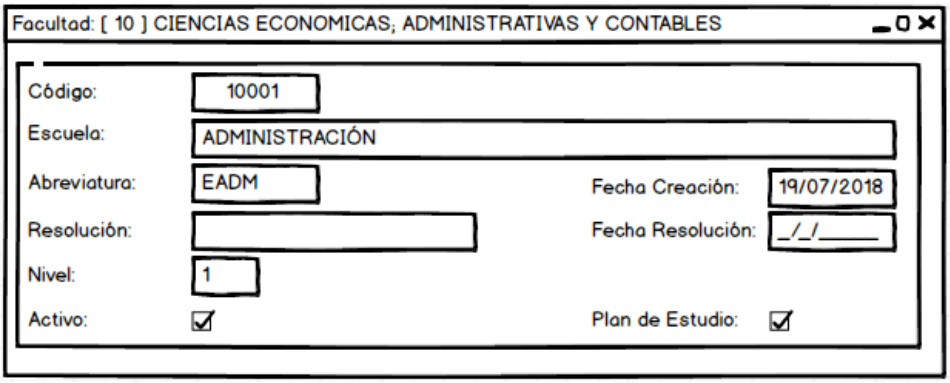


ILUSTRACIÓN 32. PROTOTIPO LISTAR FACULTAD

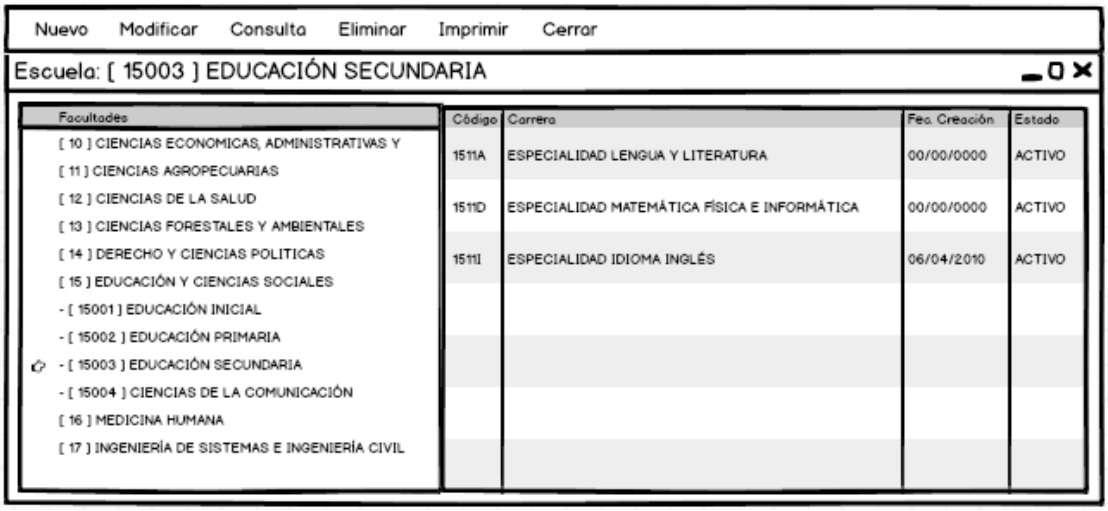
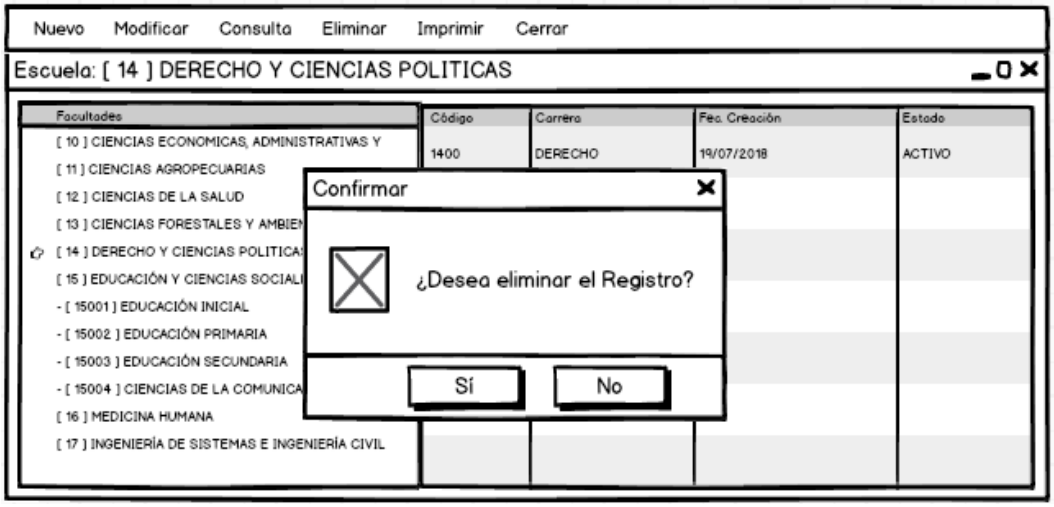


ILUSTRACIÓN 33. PROTOTIPO ELIMINAR FACULTAD



d) Prototipo CRUD Carrera Profesional

ILUSTRACIÓN 34. PROTOTIPO NUEVA CARRERA PROFESIONAL

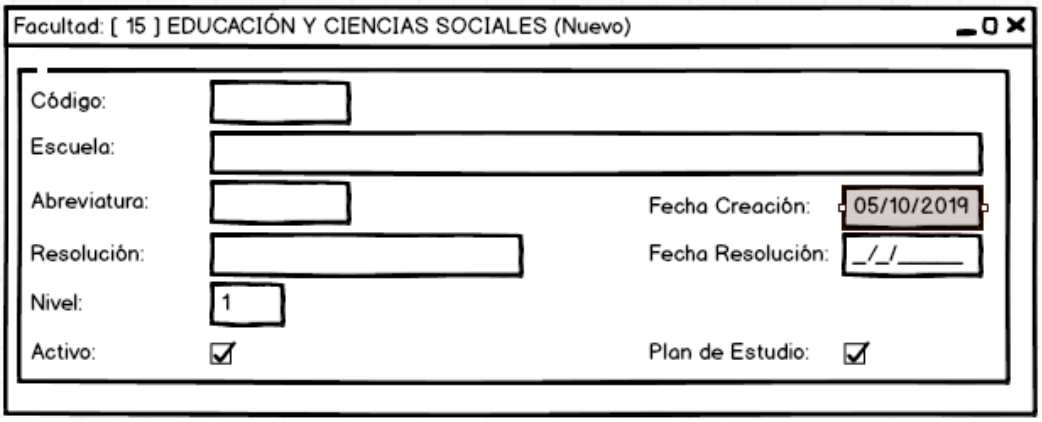


ILUSTRACIÓN 35. PROTOTIPO MODIFICAR CARRERA PROFESIONAL

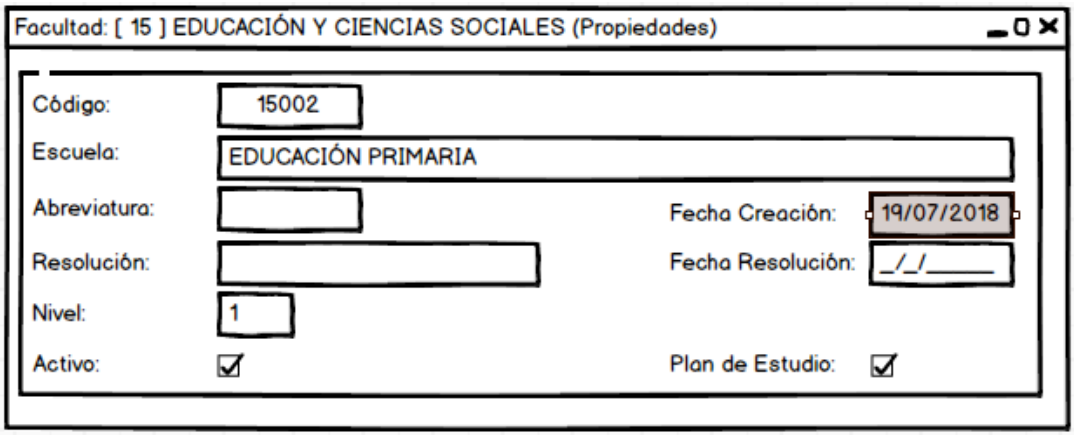


ILUSTRACIÓN 36. PROTOTIPO LISTAR CARRERA PROFESIONAL

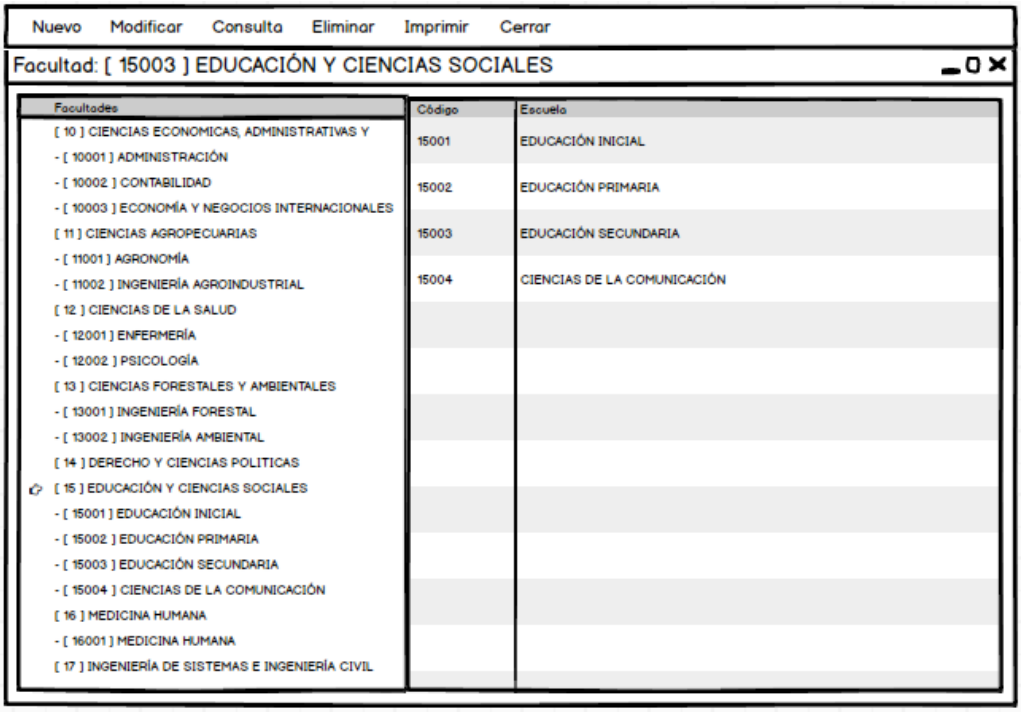
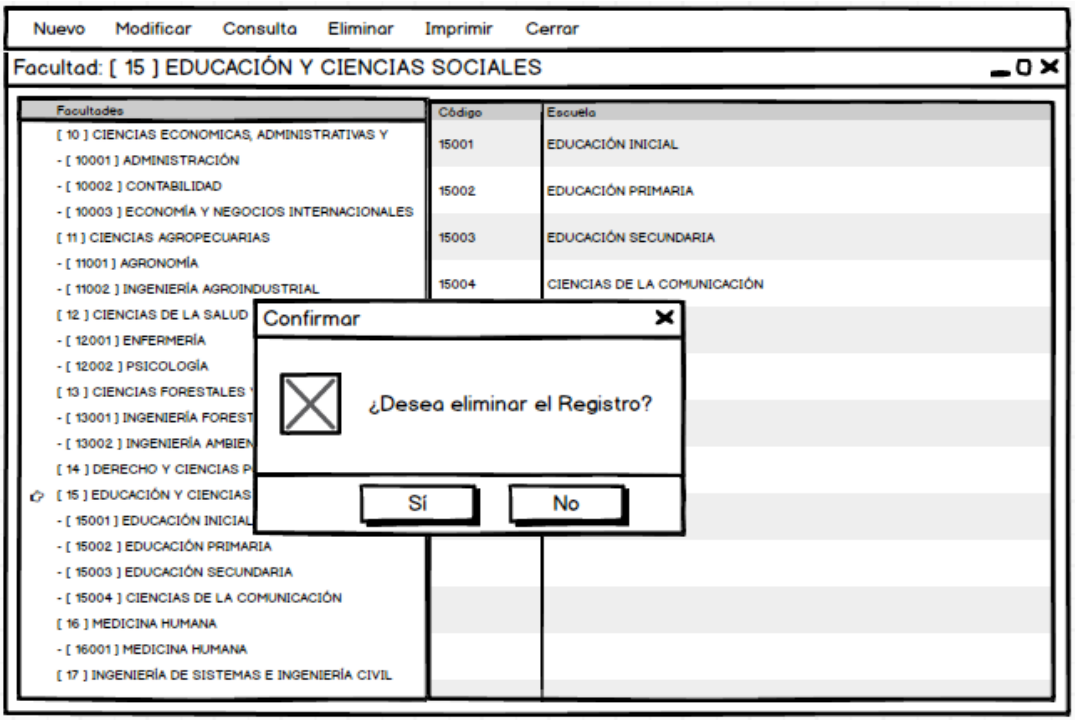


ILUSTRACIÓN 37. PROTOTIPO ELIMINAR CARRERA PROFESIONAL



e) Prototipo CRUD Docente

ILUSTRACIÓN 38. PROTOTIPO NUEVO DOCENTE

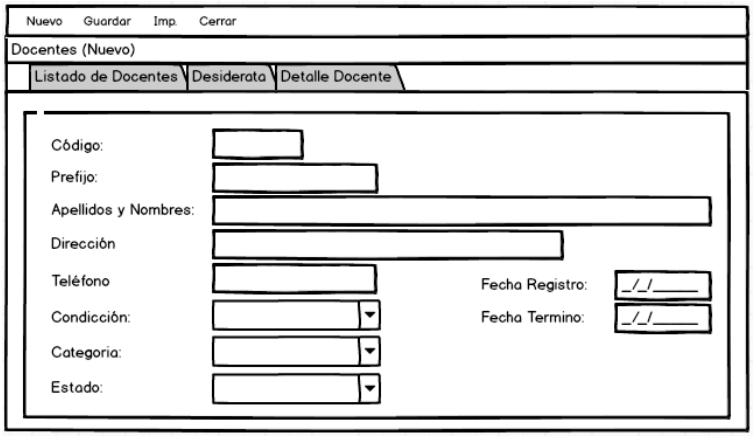


ILUSTRACIÓN 39. PROTOTIPO MODIFICAR DOCENTE

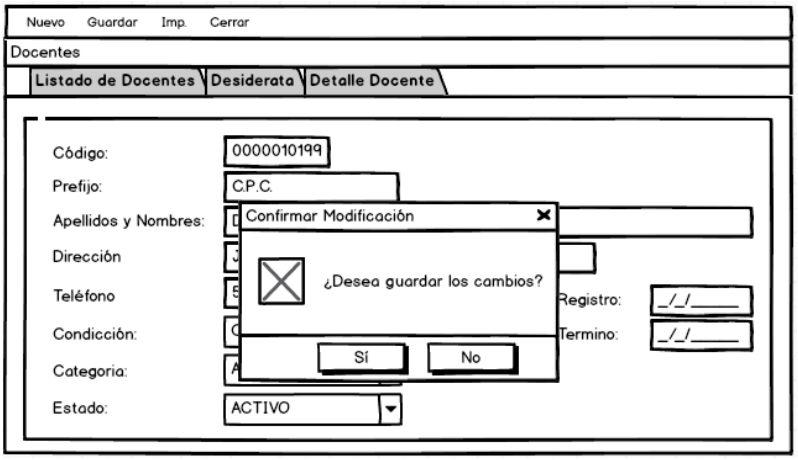
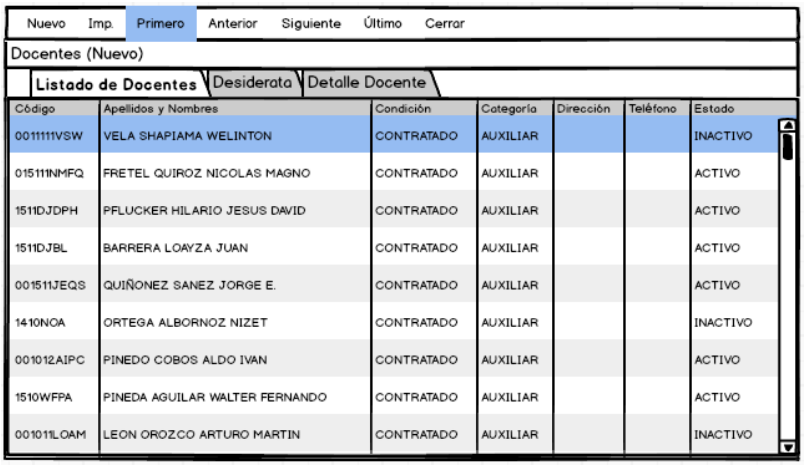
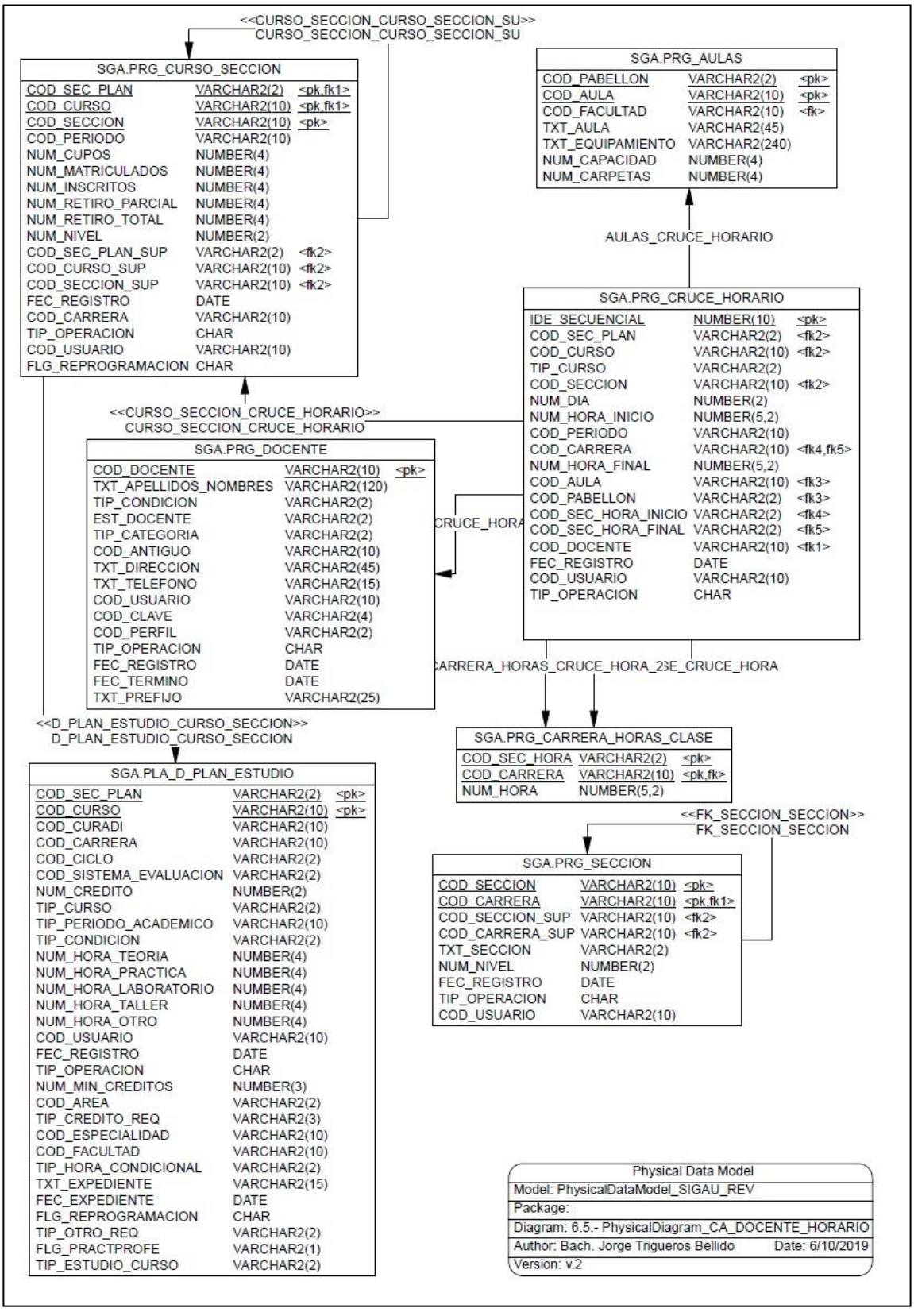


ILUSTRACIÓN 40. PROTOTIPO LISTAR DOCENTE



f) Modelo Físico Spring 2

ILUSTRACIÓN 41. MODELO FÍSICO SPRINT 2



g) Fuente SPRINT 2

(Ver Anexo 6)

h) Interfaz Gestión Facultad

ILUSTRACIÓN 42. INTERFAZ NUEVA FACULTAD

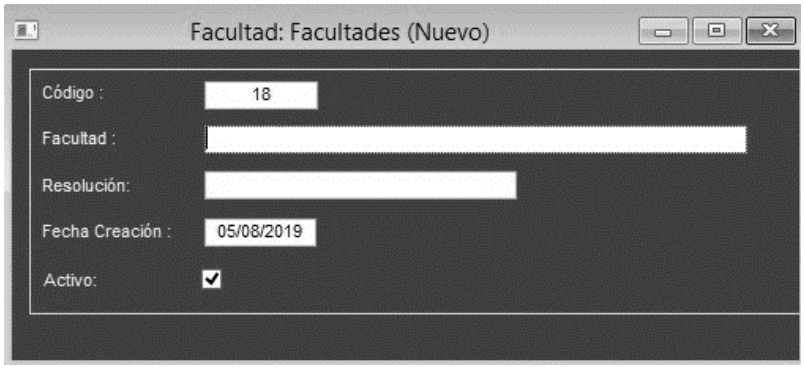


ILUSTRACIÓN 43. INTERFAZ MODIFICAR FACULTAD

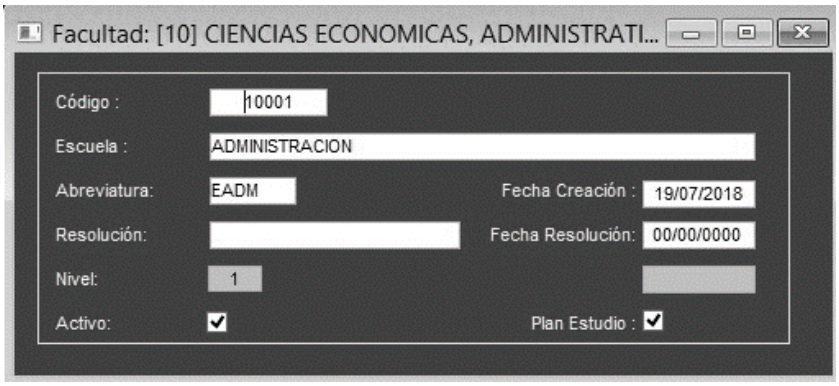


ILUSTRACIÓN 44. INTERFAZ LISTAR FACULTAD

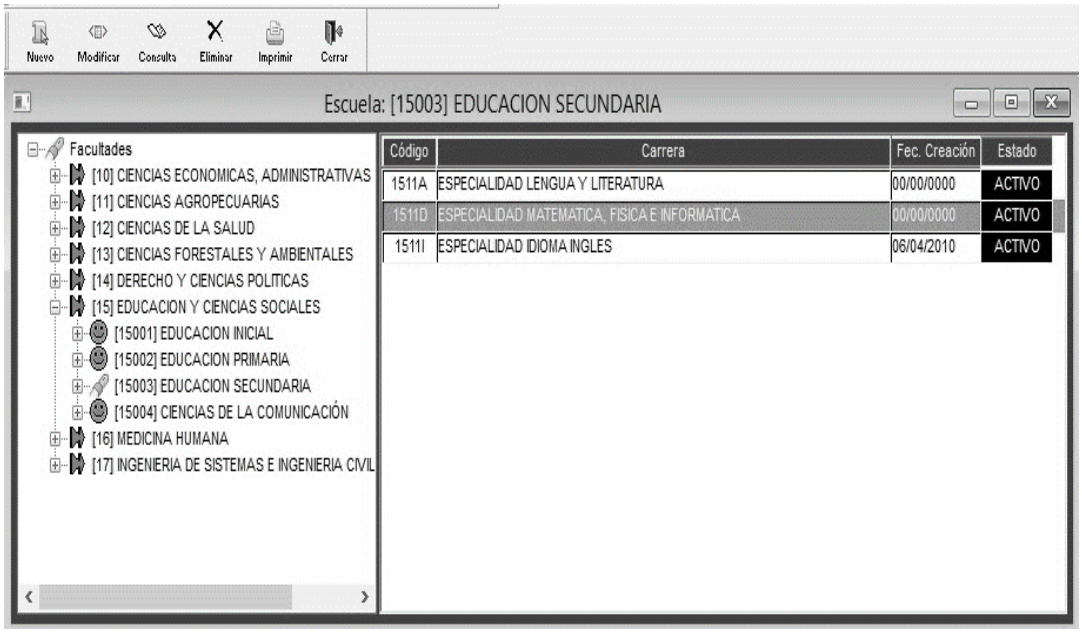
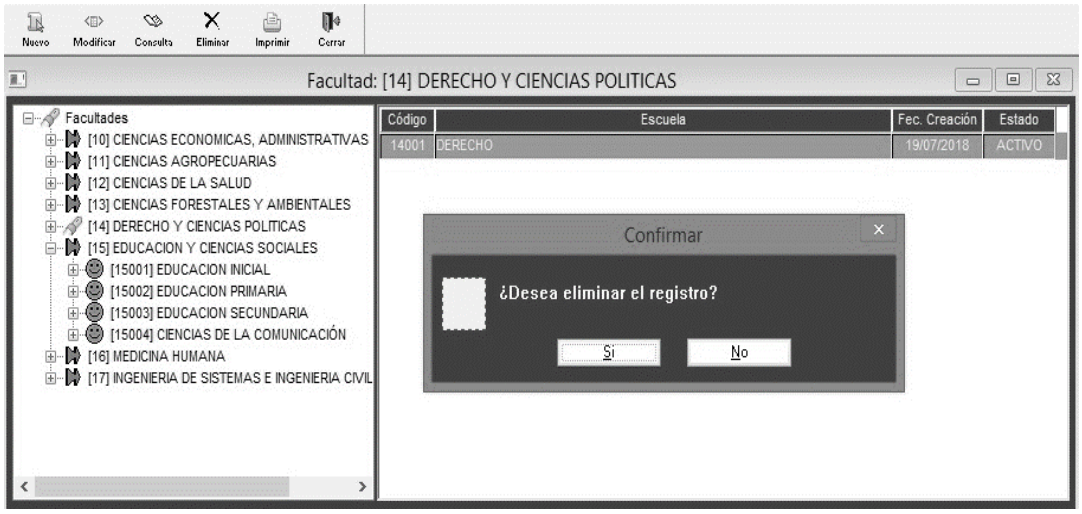


ILUSTRACIÓN 45. INTERFAZ ELIMINAR FACULTAD



i) Interfaz Gestión Carrera Profesional

ILUSTRACIÓN 46. INTERFAZ NUEVA CARRERA PROFESIONAL

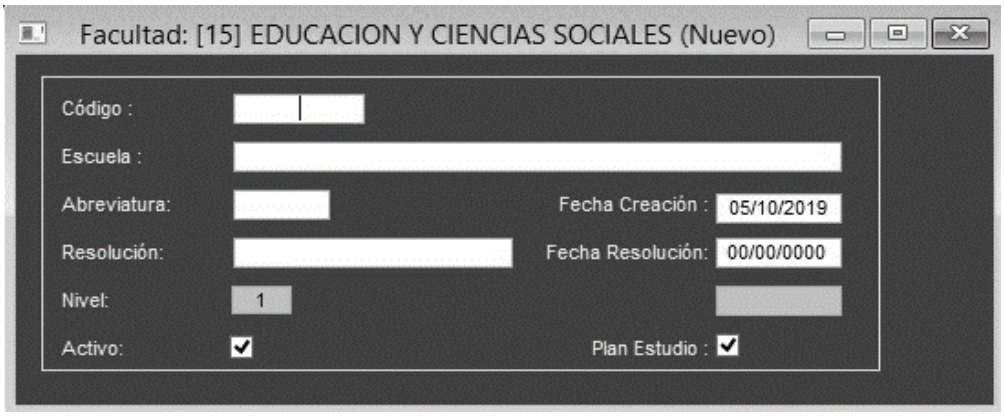


ILUSTRACIÓN 47. INTERFAZ MODIFICAR CARRERA PROFESIONAL

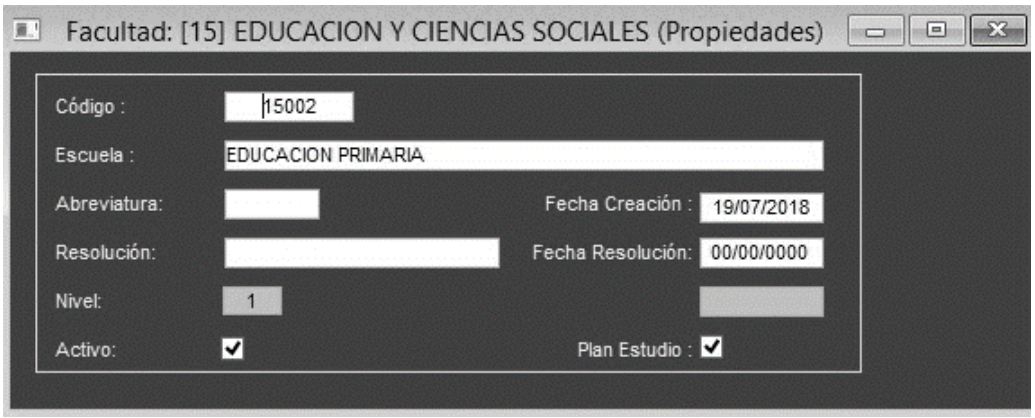


ILUSTRACIÓN 48. INTERFAZ LISTAR CARRERA PROFESIONAL

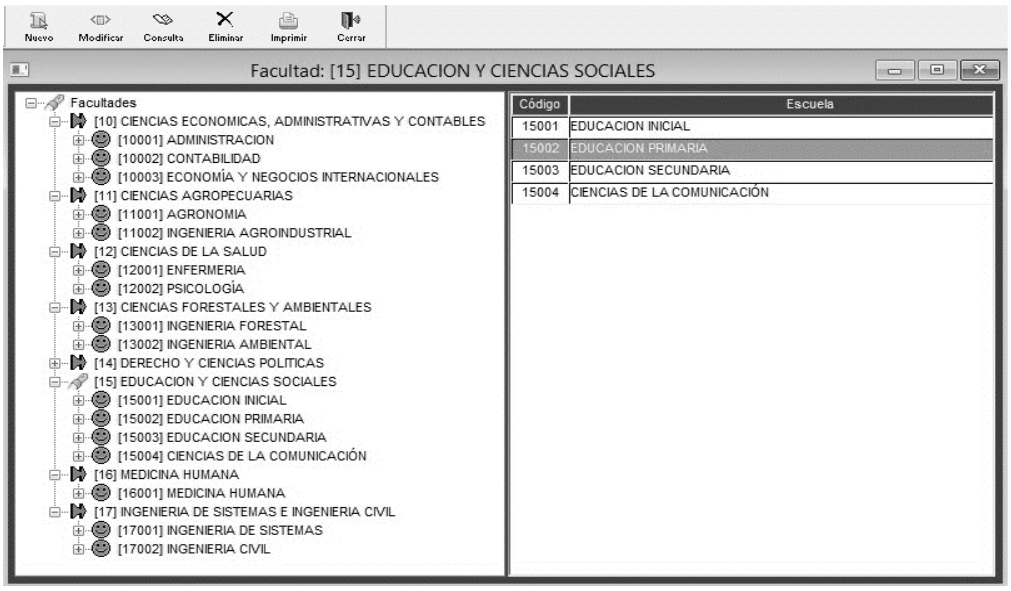
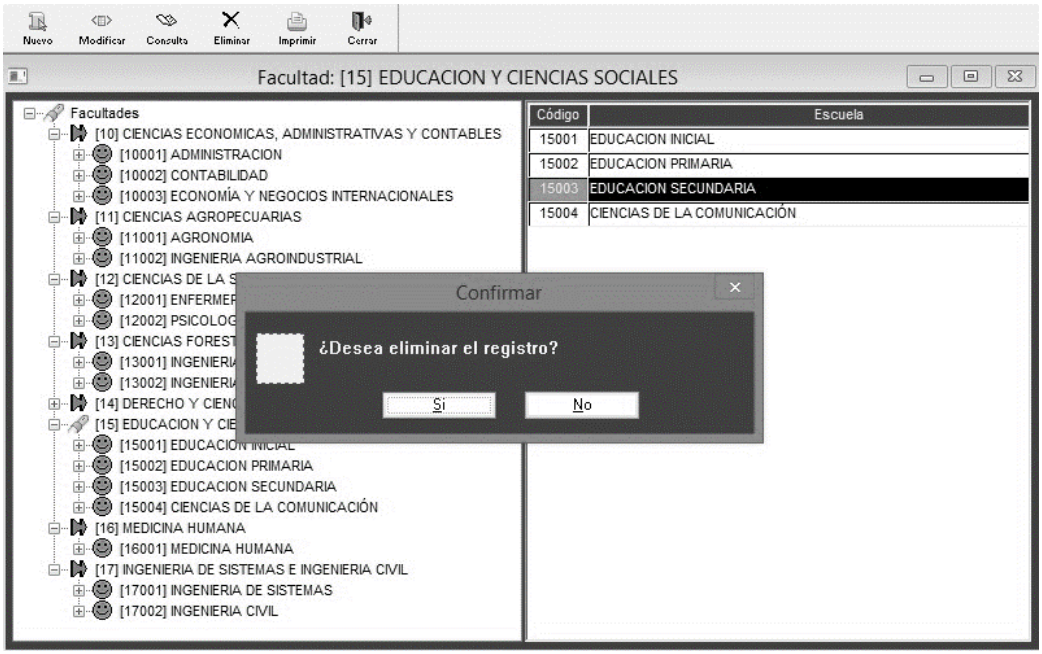


ILUSTRACIÓN 49. INTERFAZ ELIMINAR CARRERA PROFESIONAL



j) Interfaz Gestión Docente

ILUSTRACIÓN 50. INTERFAZ NUEVO DOCENTE

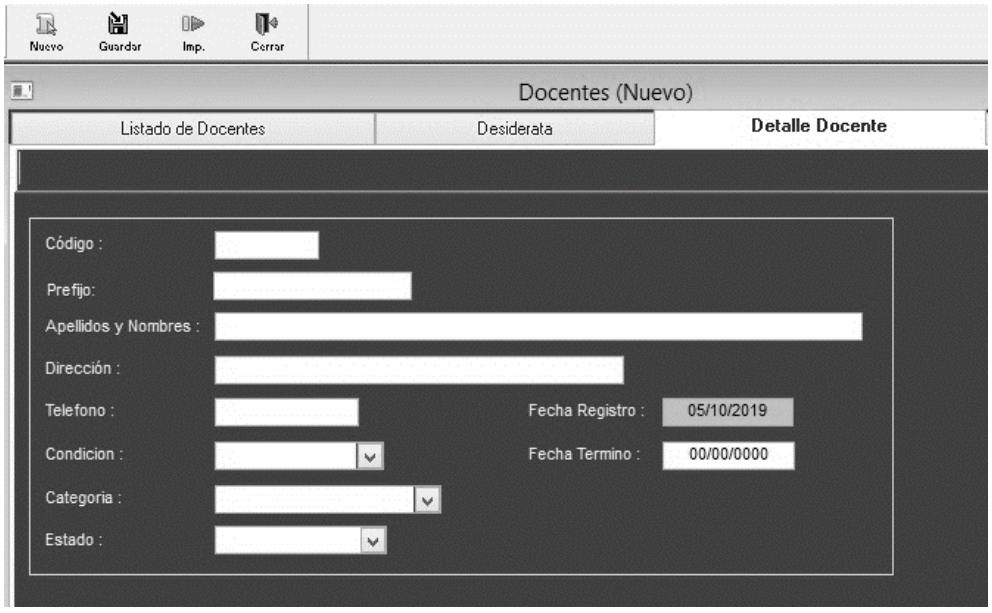


ILUSTRACIÓN 51. INTERFAZ MODIFICAR DOCENTE



ILUSTRACIÓN 52. INTERFAZ LISTAR DOCENTE

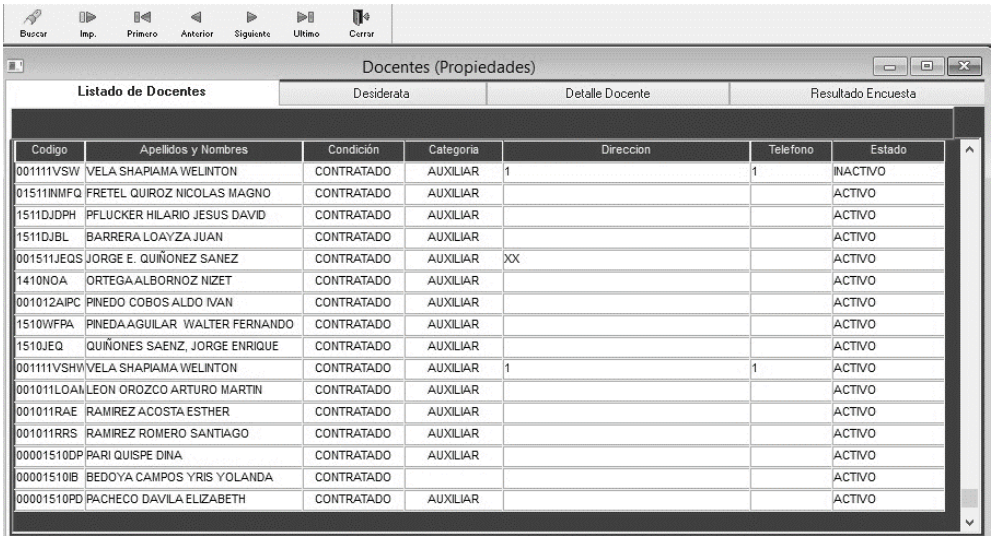
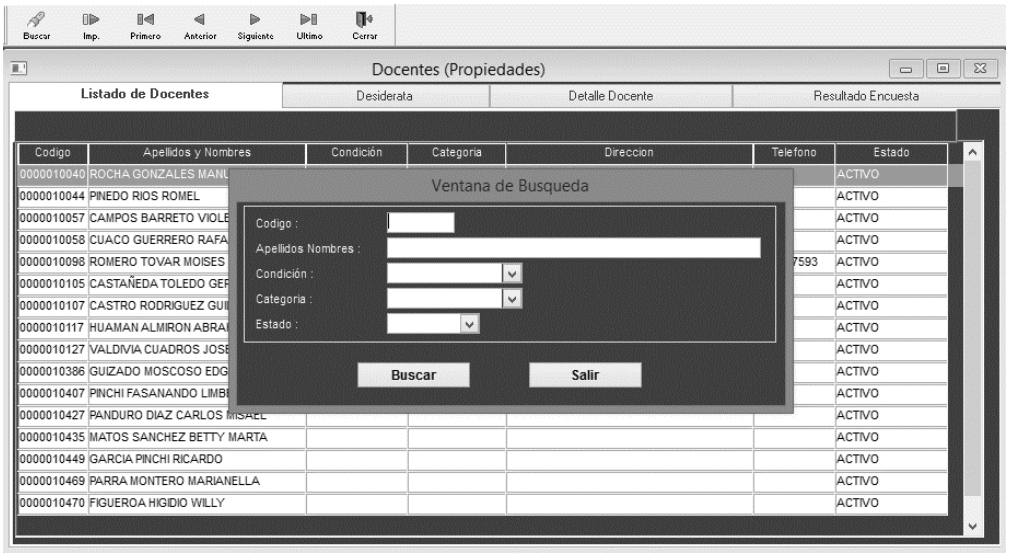


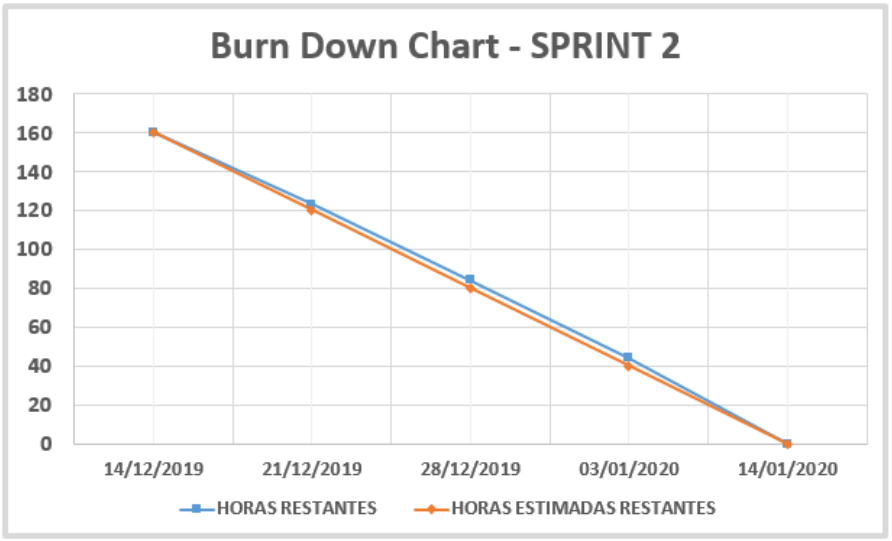
ILUSTRACIÓN 53. INTERFAZ BUSCAR DOCENTE



k) Resultado Burn Down Chart – Sprint 2

En la siguiente ilustración se muestra el desarrollo y resultado de las 5 semanas del Sprint 2, donde se puede apreciar que entre la primera y segunda semana el Burn Down de desarrollo está por encima del Burn Down del desarrollo esperado, esto debido por la tardía entrega de algunas tareas, lo cual afecto hasta mediados de la cuarta semana, sin embargo se ha podido cumplir con el SPRINT 2 en el tiempo establecido.

ILUSTRACIÓN 54. INTERFAZ DE ASIGNACIÓN, MODIFICACIÓN Y DESHABILITACIÓN DE PERFIL A USUARIO



l) Informe de Prueba funcional – SPRINT 2

TABLA 40. INFORME DE PRUEBA FUNCIONAL – SPRINT 2

<table>
<tr><td colspan="6">PRUEBA FUNCIONAL – SPRINT 2</td></tr>
<tr><td>PRUEBA Nº</td><td colspan="2">Prueba de Funcionalidad Nº02</td><td>VERSIÓN DE EJECUCIÓN</td><td colspan="2">UNU-SP02</td></tr>
<tr><td rowspan="2">TAREA:</td><td colspan="2" rowspan="2">Creación, lectura, modificación y deshabilitación de Facultades
Creación, lectura, modificación y deshabilitación de Carreras Profesionales
Creación, lectura, modificación y deshabilitación de tipo condición docente
Creación, lectura, modificación y deshabilitación de tipo categoría docente
Creación, lectura, modificación y deshabilitación de docentes</td><td>FECHA DE EJECUCIÓN</td><td colspan="2">14/01/2020</td></tr>
<tr><td>MÓDULO DEL SISTEMA</td><td colspan="2">PLA
ESC</td></tr>
<tr><td>Descripción del caso de prueba:</td><td colspan="5">Se procedió a realizar las pruebas de mantenimiento de facultades, carreras profesionales, tipo condición docente, tipo categoría docente y usuario docente, validando los campos cuando aparecen datos errados y los mensajes de respuesta.</td></tr>
<tr><td colspan="6">3. CASO DE PRUEBA</td></tr>
<tr><td colspan="6">d. Precondiciones</td></tr>
<tr><td colspan="6">Este Sprint no inicia si no se concluye al 100% el Sprint 1</td></tr>
<tr><td colspan="6">e. Pasos de prueba</td></tr>
<tr><td colspan="6">✓ Ingresar datos no válidos para verificar los mensajes de rechazo y para validar los campos
✓ Verificar que todos los datos relacionados se muestren de manera eficiente.
✓ Realizar mantenimiento de facultades, creando, modificando y actualizando la información.
✓ Realizar mantenimiento de carreras profesionales, creando, modificando y actualizando la información.
✓ Realizar mantenimiento de tipo condición docente, creando, modificando y actualizando la información de docentes ya creados y registrados.
✓ Realizar mantenimiento tipo categoría docente, habilitando, modificando y deshabilitando a usuarios docentes ya creados y registrados
✓ Realizar mantenimiento de docente, habilitando, modificando y deshabilitando a usuario docentes.</td></tr>
<tr><td colspan="3">DATOS DE ENTRADA</td><td rowspan="2">RESPUESTA ESPERADA DE LA APLICACIÓN</td><td colspan="2">COINCIDE</td><td rowspan="2">RESPUESTA DEL SISTEMA</td></tr>
<tr><td>CAMPO</td><td>VALOR</td><td>TIPO ESCENARIO</td><td>SI</td><td>NO</td></tr>
<tr><td>Categoría</td><td>Auxiliar</td><td>normal</td><td>Tipo Categoría Asignada</td><td>X</td><td></td><td>Se asignaba la información en BD pero no ejecutaba mensaje de confirmación</td></tr>
<tr><td>Condición</td><td>Contratado</td><td>normal</td><td>Tipo de condición asignada</td><td>X</td><td></td><td>Tipo de condición asignado</td></tr>
<tr><td>Facultad</td><td>0002151521</td><td>prueba</td><td>Verifique datos ingresados</td><td>X</td><td></td><td>Verifique datos ingresados</td></tr>
<tr><td>Carrera Profesional</td><td>Facultad de Ingeniería</td><td>Normal</td><td>Los datos han sido registrados</td><td>X</td><td></td><td>Los datos han sido registrados</td></tr>
<tr><td>Docente</td><td>Carlos Amasifuen1</td><td>prueba</td><td>Verifique datos ingresados</td><td>X</td><td></td><td>Verifique datos ingresados</td></tr>
<tr><td colspan="6">f. Post Condiciones</td></tr>
<tr><td colspan="6">No aplica</td></tr>
<tr><td colspan="6">4. RESULTADOS DE LA PRUEBA</td></tr>
<tr><td colspan="4">Defectos y desviaciones</td><td colspan="2">Veredicto</td></tr>
<tr><td colspan="4" rowspan="2"></td><td>X</td><td>PASO</td></tr>
<tr><td></td><td>FALLO</td></tr>
<tr><td colspan="4">Observaciones</td><td colspan="2">Probador</td></tr>
<tr><td colspan="4" rowspan="2">Al asignar el tipo categoría docente a un usuario docente no muestra ningún mensaje de éxito.</td><td colspan="2">Firma:
Nombre:</td></tr>
<tr><td colspan="2">Fecha:</td></tr>
</table>

m) Revisión de Sprint 2 (Sprint Review Meeting)

TABLA 41. INFORME FINAL DE PRUEBA FUNCIONAL – SPRINT 2

Lugar:	Universidad Nacional de Ucayali	
Fecha	14/01/2020	
Número de Iteración / Sprint	Sprint 2	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
¿Qué salió bien en el Sprint? (aciertos)	**¿Qué no salió bien en el Sprint? (errores)**	**Lecciones aprendidas (recomendaciones)**
✓La recuperación de datos entre los formularios de mantenimiento y la base de datos se ejecuta sin mayores complicaciones. ✓Las asignaciones de diferentes datos a los usuarios creados se han ejecutados sin ningún tipo de inconveniente.	✓ Dar prioridad a otras actividades no esenciales, lo cual conllevo a retrasos inesperados. ✓ No se pudo recuperar el tiempo de retraso de la presentación de las tareas hasta la última semana del Sprint, generando un incremento de horas/hombre.	✓ Se sugiere centrarse en las actividades definidas del sprint. ✓ Se debe mejorar la comunicación entre los stakeholders dueño del proceso (Dirección General de Registros y Asuntos Académicos)

n) Reunión Retrospectiva – Sprint 2 (Restrospective Meeting)

Finalizado el sprint 2, se realizó una reunión no más de dos horas con los miembros del equipo para determinar los avances del sprint concluido, además de poder detallar lo bueno, lo incómodo y aporta con ideas de mejora para el siguiente sprint (aportes del equipo)

TABLA 42. INFORME FINAL DE REUNIÓN RETROSPECTIVA – SPRINT 2

Lugar:	Universidad Nacional de Ucayali	
Fecha	14/01/2020	
Número de Iteración / Sprint	Sprint 2	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Lo bueno (Sprint culminado)	**Lo incomodo**	**Aportes del equipo (Próximo Sprint)**
✓ Mejora en la puntualidad a llamadas de reuniones diarias con el equipo de trabajo.	✓ Reuniones canceladas con el dueño del proceso, aduciendo que tenía reuniones en otras oficinas	✓ Agendar con anticipación las reuniones con el dueño del proceso (Dirección General de Registros y Asuntos Académicos) (Jorge Trigueros)
✓ La comunicación entre el equipo de trabajo ha mejorado.	✓ Interrupciones no definidas en las actividades, días no laborables y no programadas.	✓ Una mayor agilidad en la programación para culminar las tareas en el tiempo establecido (José Congona)
✓	✓	✓ Mejorar la recolección de la información para no estancar el desarrollo de las historias (Jorge Trigueros)

4.6.3. Sprint 3

El objetivo de este sprint es desarrollar las tareas para el módulo de planes del sistema y para ello es requerido la creación de cursos, planes de estudio y modelo de evaluación.

a) Priorización de Sprint 3

Las tareas elegidas para este sprint y el orden de priorización han sido tomadas en cuenta siguiendo el orden de ejecución del sistema, después de que se haya terminado con el módulo de configuración, se procede con la primera parte del módulo incluido en el sprint 3 que es información fundamental para el proceso de matrícula virtual.

b) Formato de asignación del Sprint 3

TABLA 43. ESPECIFICACIÓN SPRINT 3

SPRINT 03
FECHA DE INICIO: 15/01/2020
FECHA DE CULMINACIÓN: 19/02/2020

Tareas Pendientes: 0
Días Pendientes: 0

Prioridad del Negocio	Descripción de la tabla	Responsable	Días	Estado
ALTA	Creación, lectura, modificación y deshabilitación de planes académicos	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo de curso	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo condición curso	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo estudio curso	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo examen	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación	Jorge Trigueros	5	TERMINADO

c) Prototipo CRUD Plan Académico

ILUSTRACIÓN 55. PROTOTIPO NUEVO PLAN ACADÉMICO

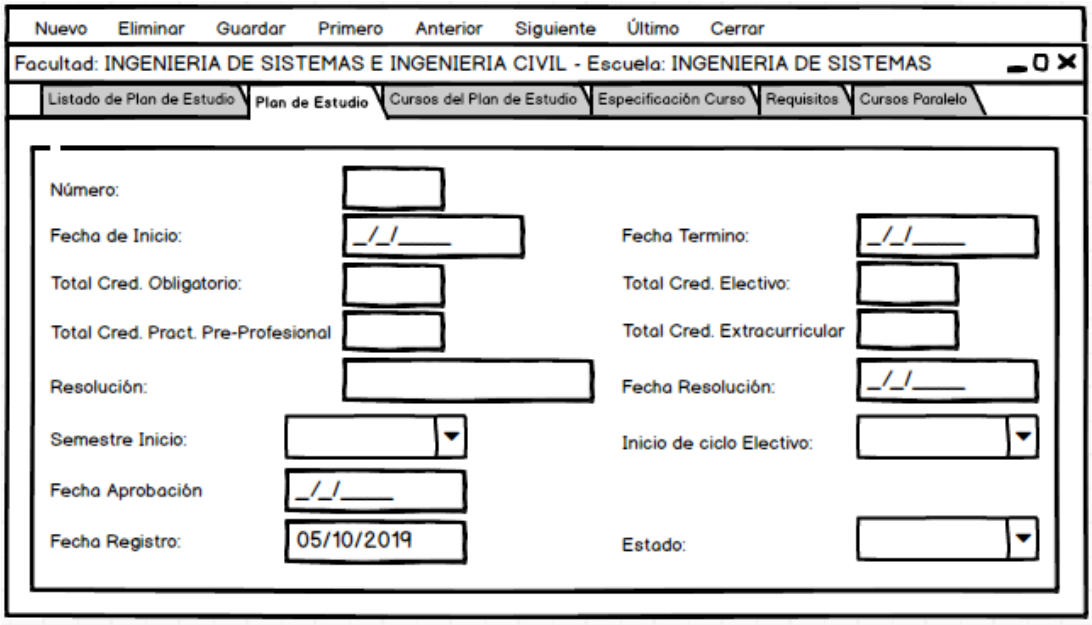


ILUSTRACIÓN 56. PROTOTIPO MODIFICAR PLAN ACADÉMICO

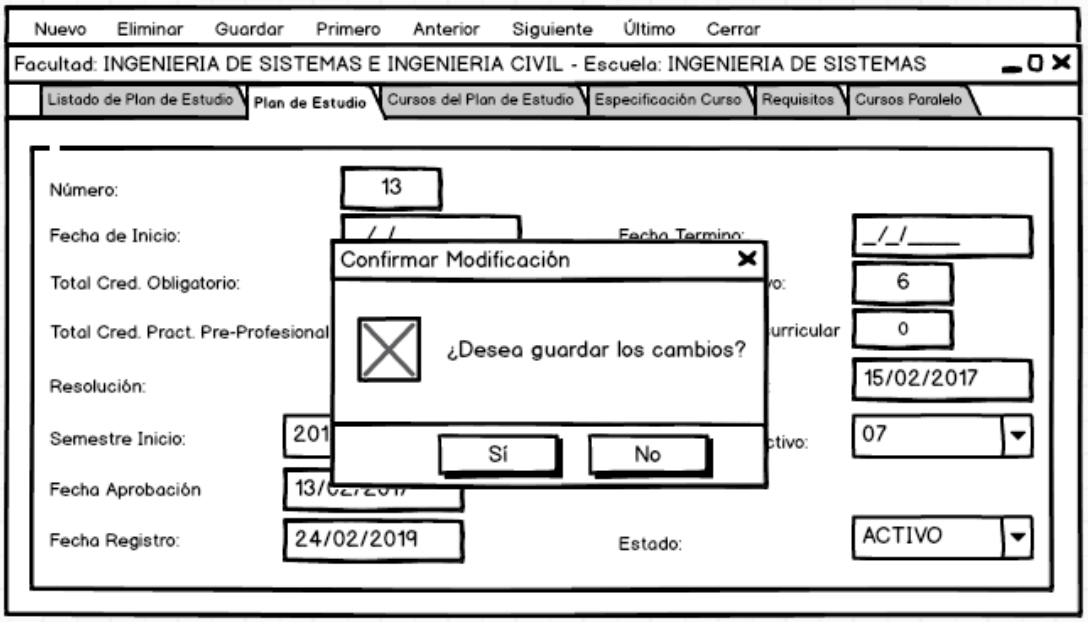


ILUSTRACIÓN 57. PROTOTIPO LISTAR PLAN ACADÉMICO

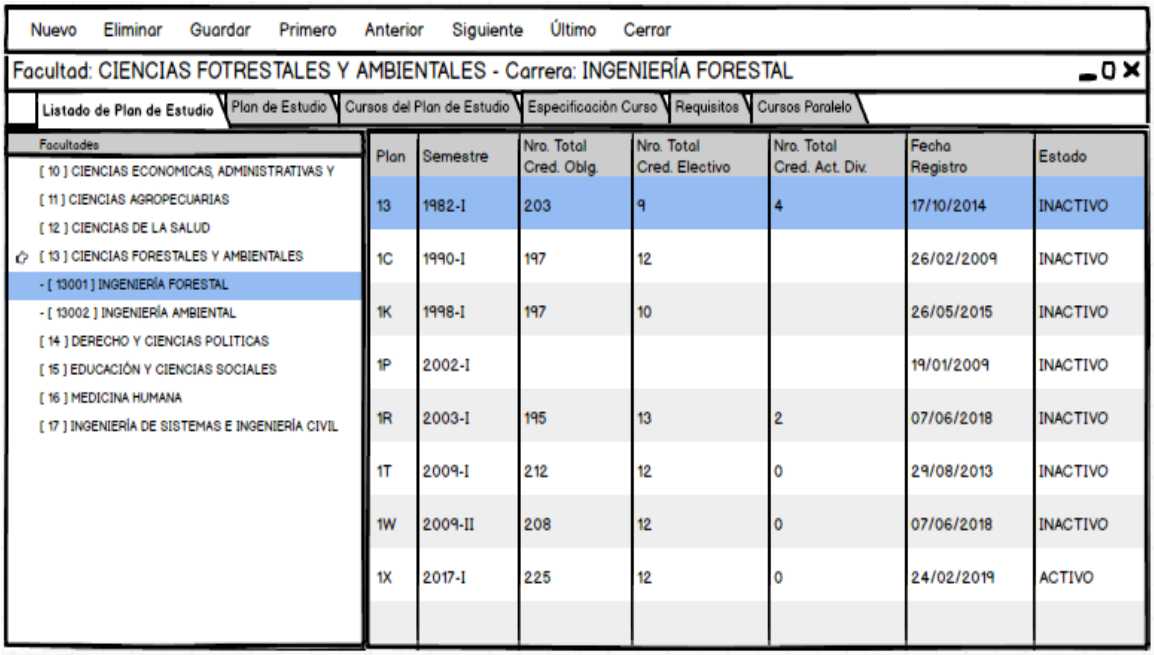
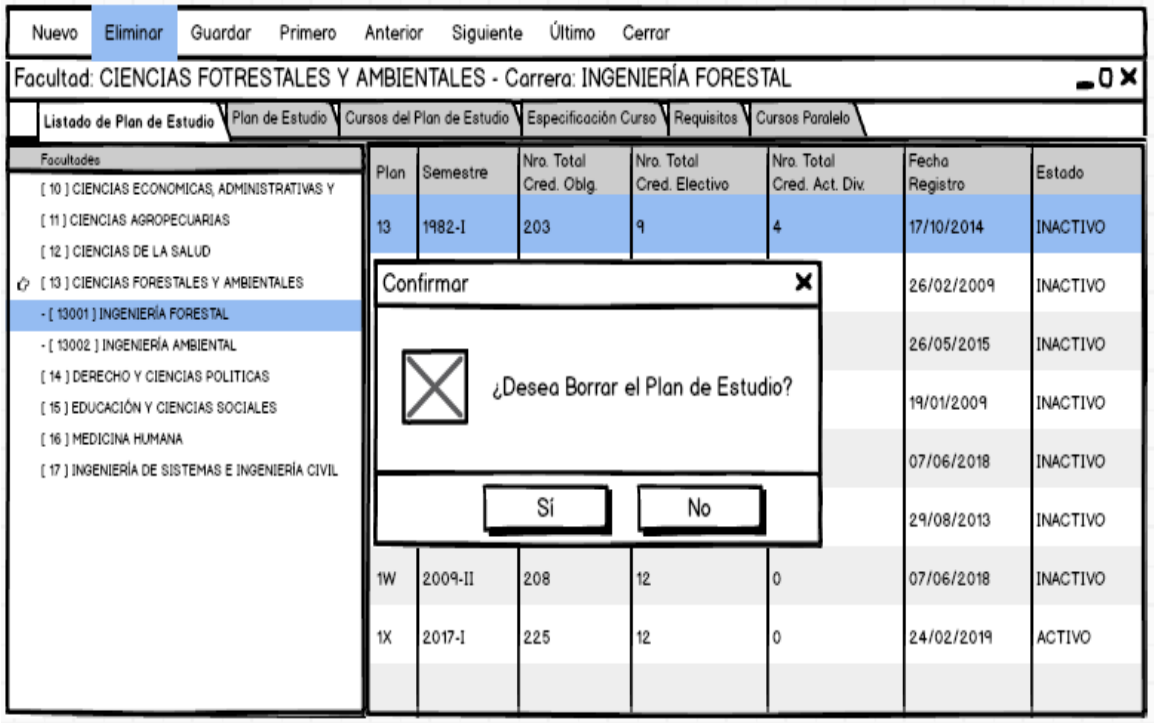


ILUSTRACIÓN 58. PROTOTIPO ELIMINAR PLAN ACADÉMICO



d) Prototipo CRUD de Cursos Académicos

ILUSTRACIÓN 59. PROTOTIPO NUEVO CURSO ACADÉMICO

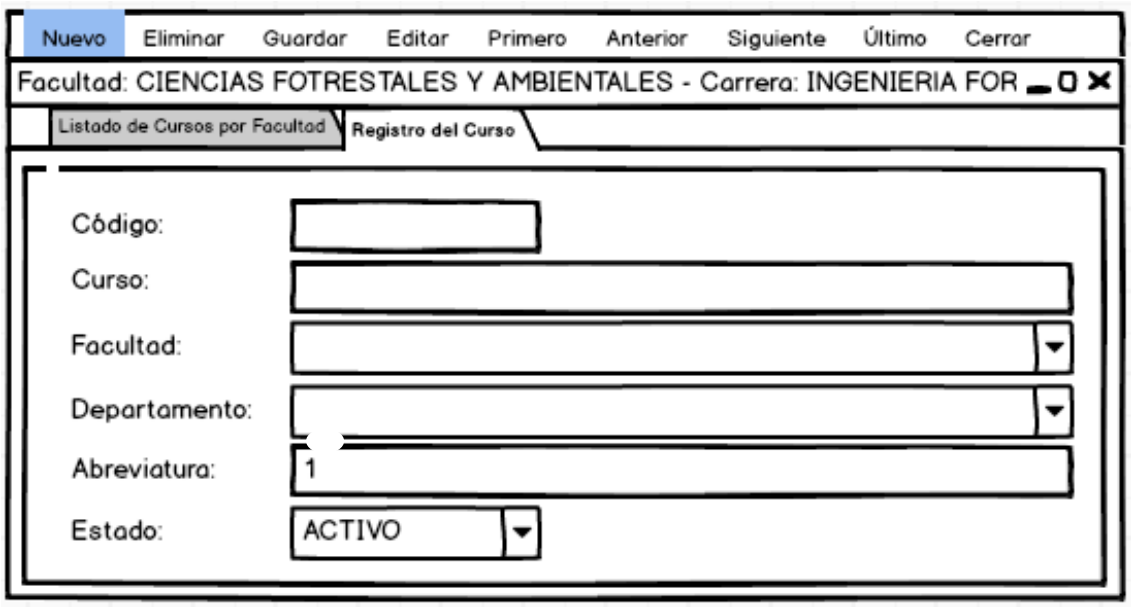


ILUSTRACIÓN 60. PROTOTIPO MODIFICAR CURSO ACADÉMICO



ILUSTRACIÓN 61. PROTOTIPO LISTAR CURSOS ACADÉMICOS

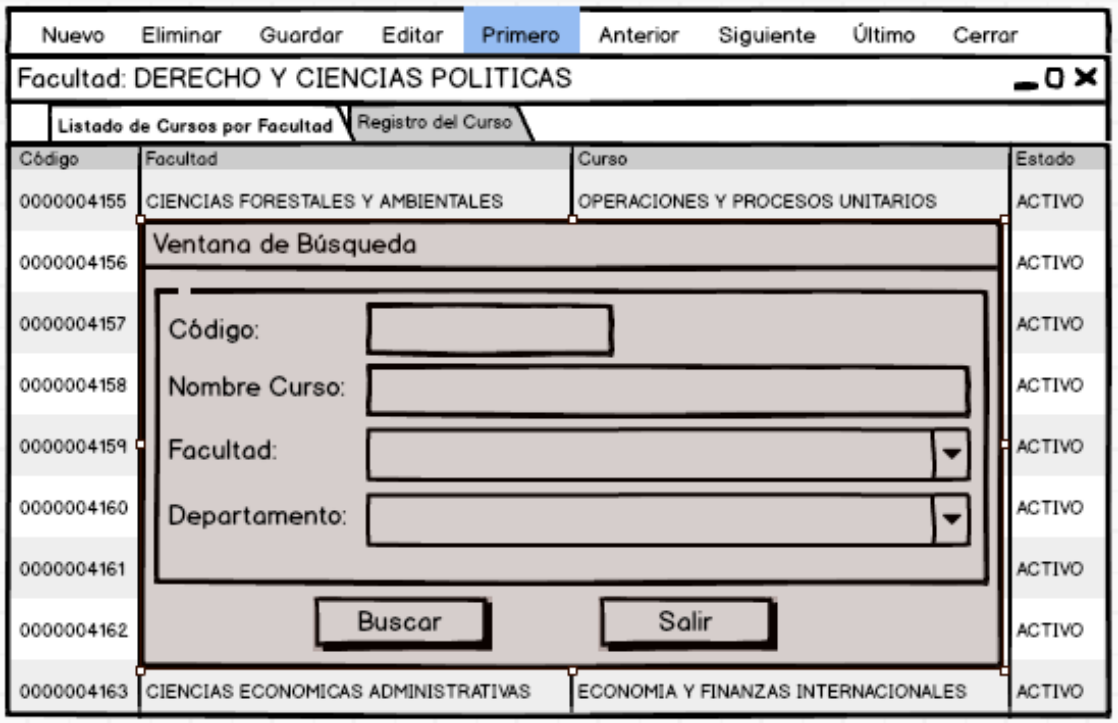
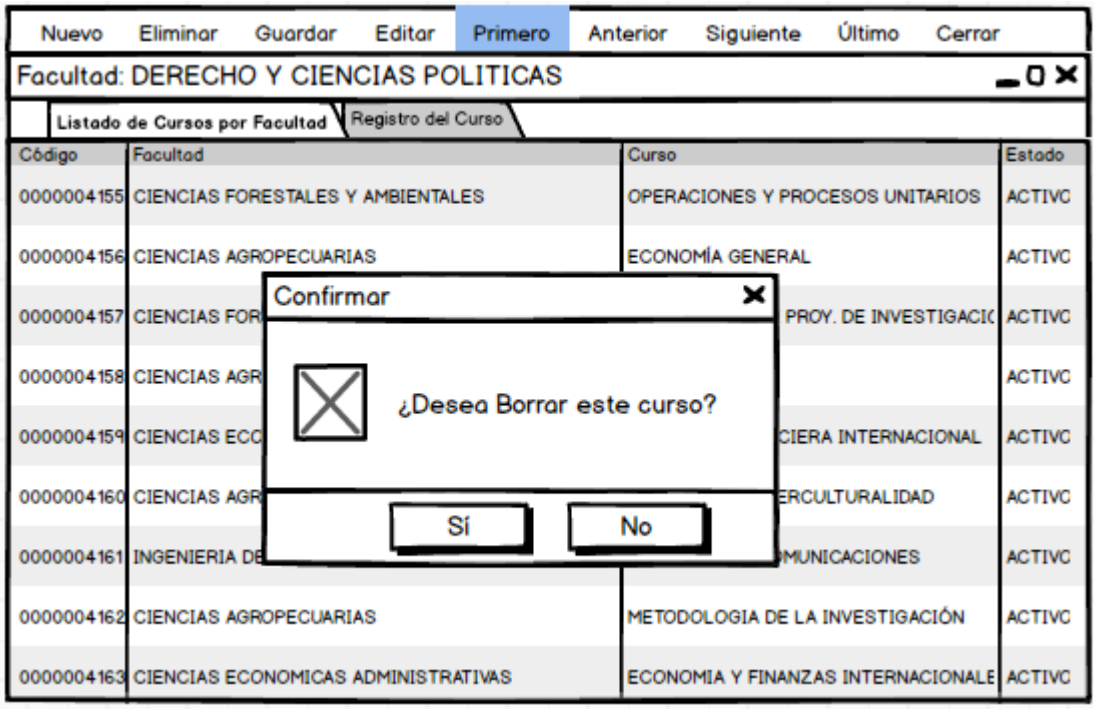


ILUSTRACIÓN 62. PROTOTIPO ELIMINAR CURSO ACADÉMICO



e) Prototipo CRUD de Sistema de Evaluación

ILUSTRACIÓN 63. PROTOTIPO NUEVO SISTEMA DE EVALUACIÓN

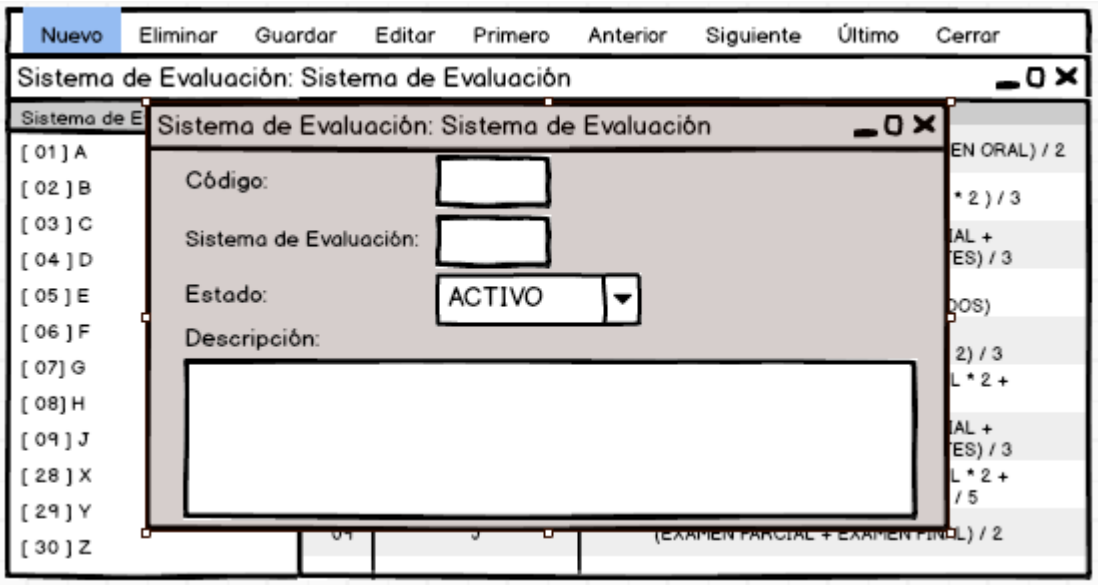


ILUSTRACIÓN 64. PROTOTIPO MODIFICAR SISTEMA DE EVALUACIÓN

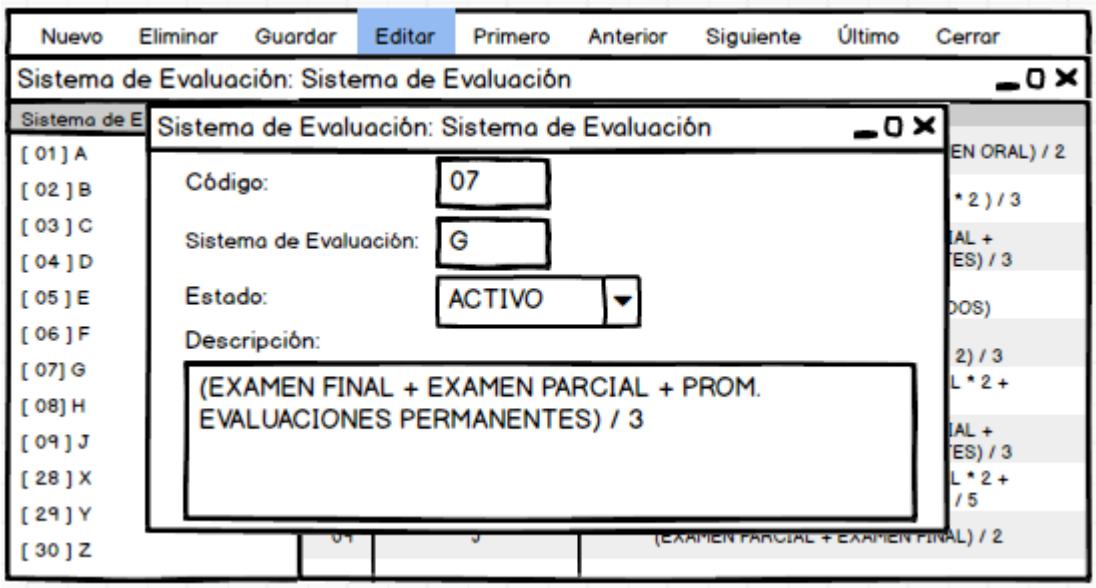


ILUSTRACIÓN 65. PROTOTIPO LISTAR SISTEMA DE EVALUACIÓN

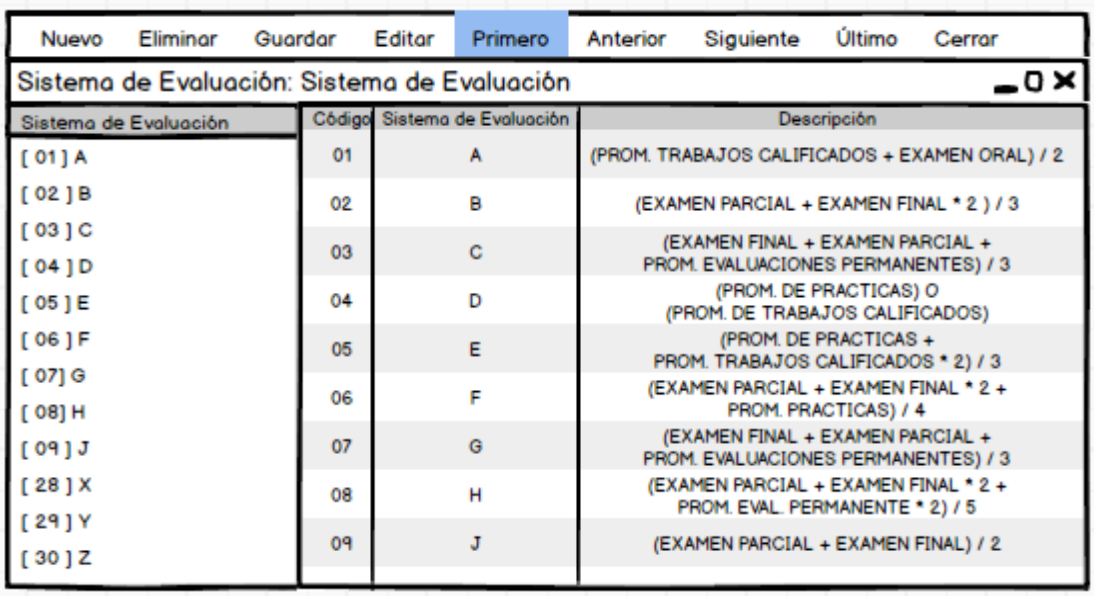
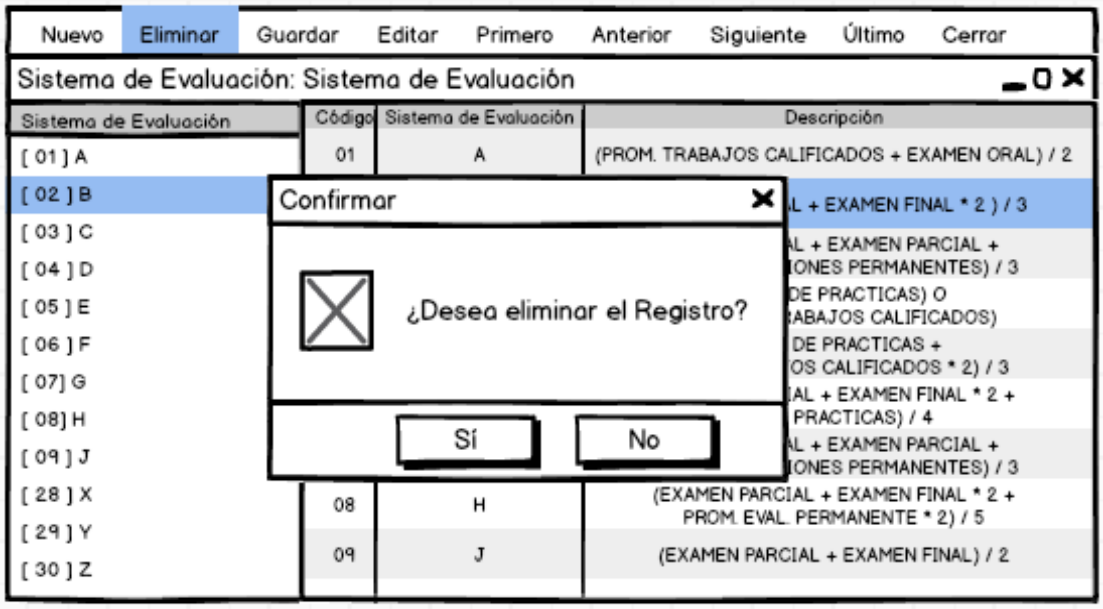
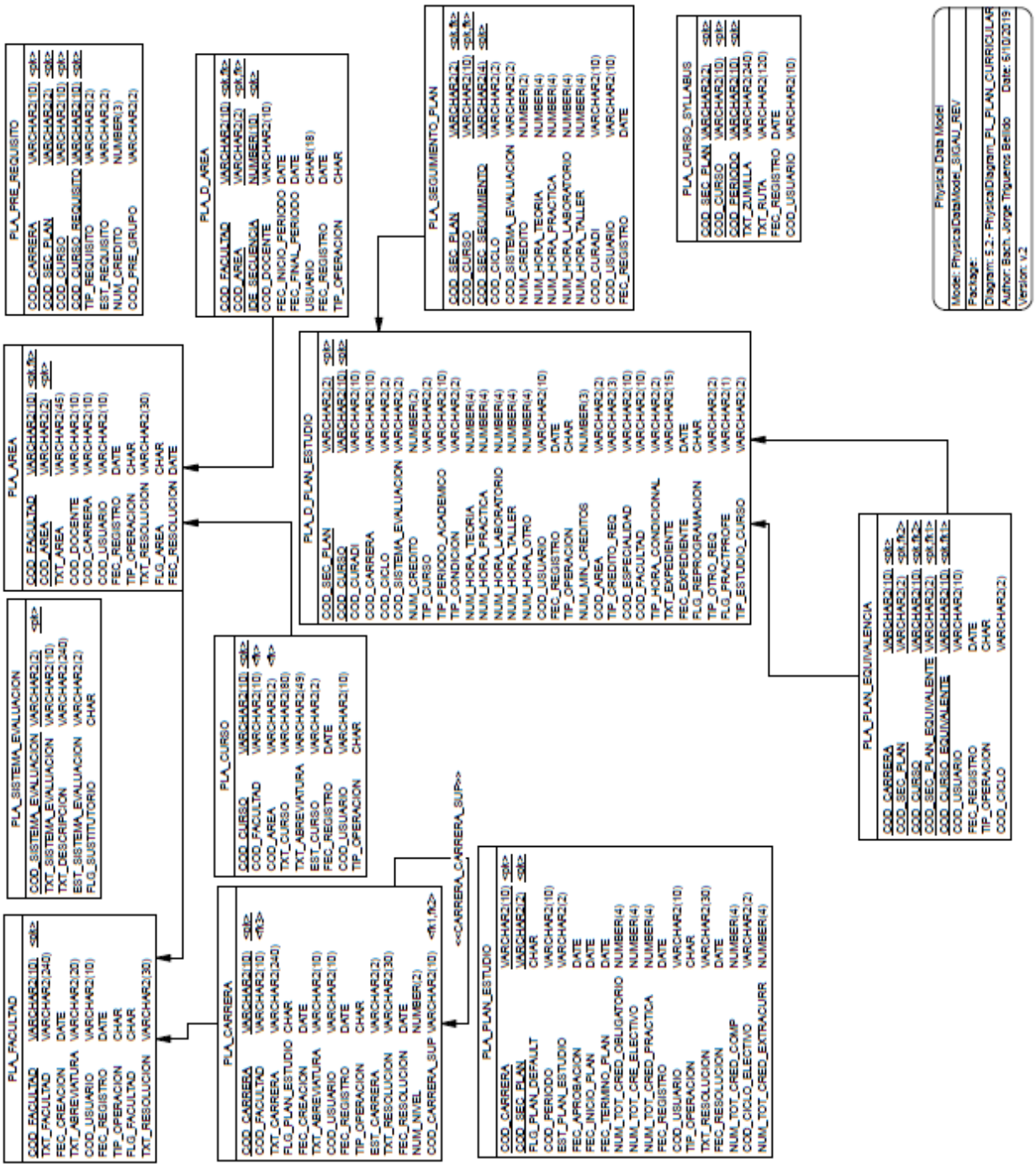


ILUSTRACIÓN 66. PROTOTIPO ELIMINAR SISTEMA DE EVALUACIÓN



f) Modelo físico Sprint 3

ILUSTRACIÓN 67. MODELO FÍSICO SPRINT 3



g) Fuente Sprint 3

(Ver Anexo 6)

h) Interfaz Gestión Plan Académico

ILUSTRACIÓN 68. INTERFAZ NUEVO PLAN ACADÉMICO

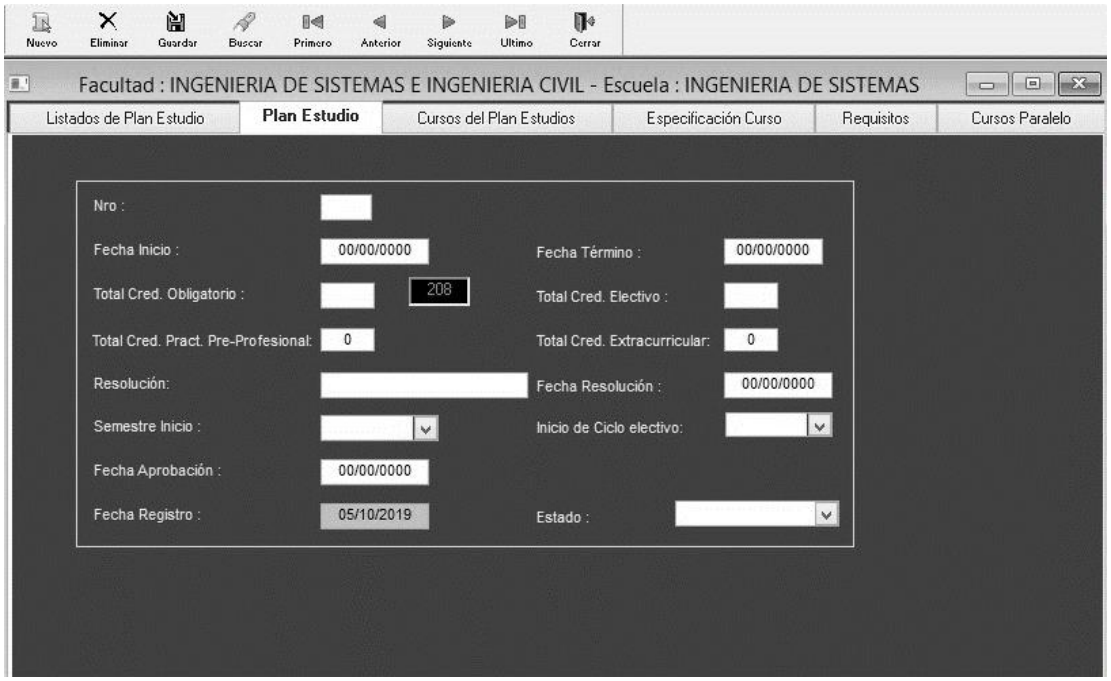


ILUSTRACIÓN 69. INTERFAZ MODIFICAR PLAN ACADÉMICO

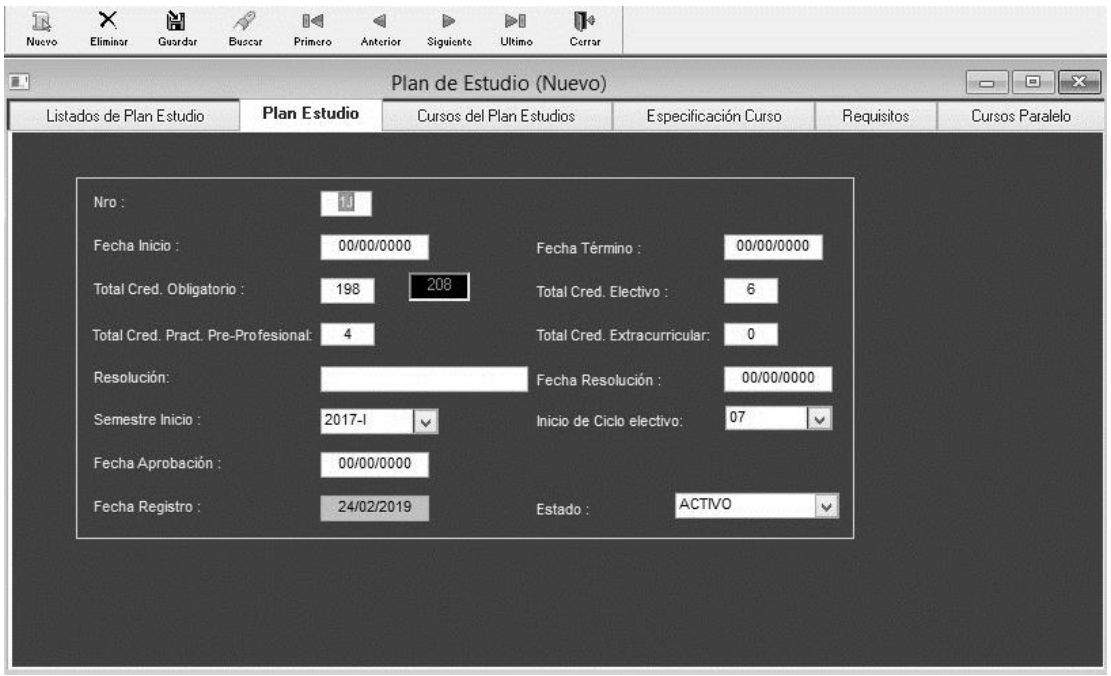


ILUSTRACIÓN 70. INTERFAZ LISTAR PLAN ACADÉMICO

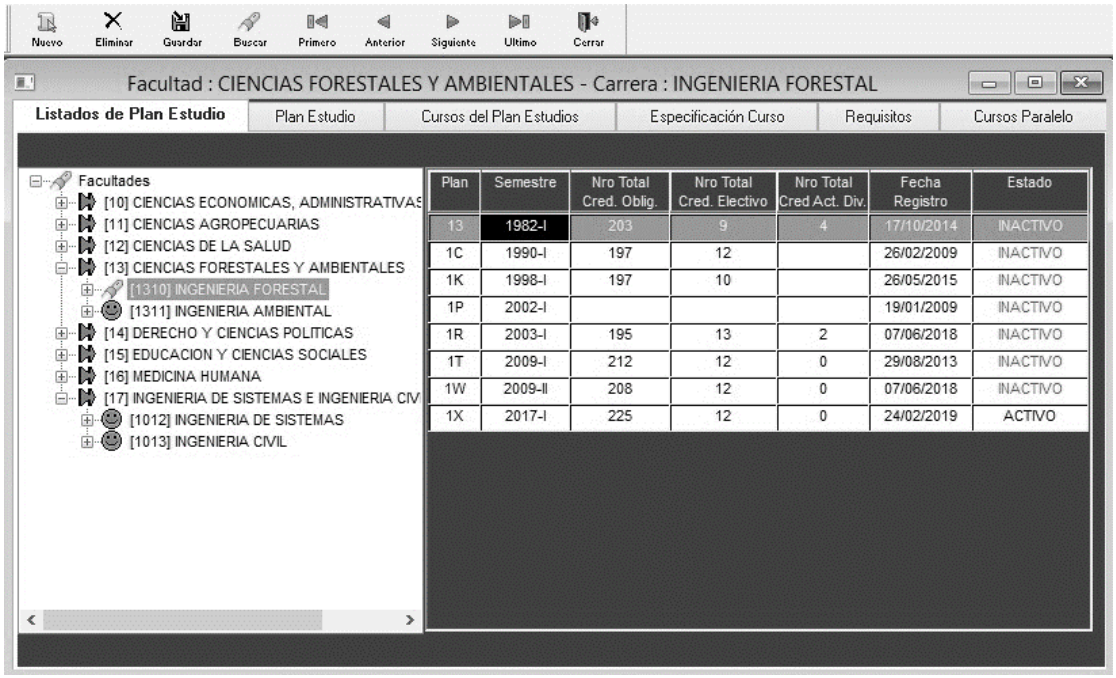
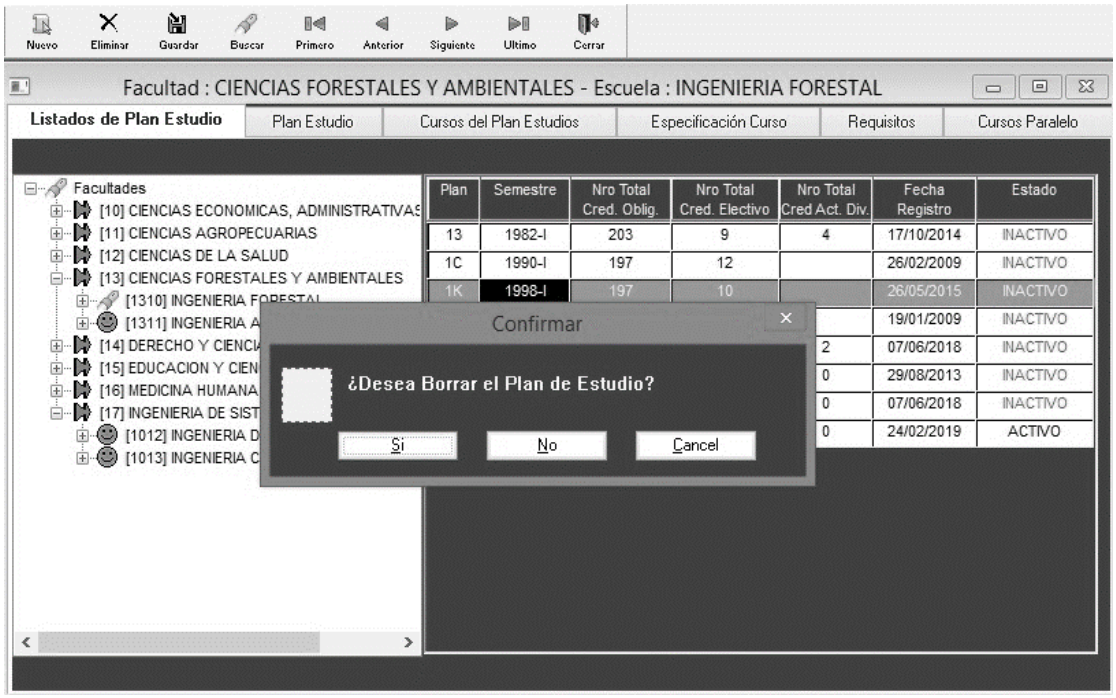


ILUSTRACIÓN 71. INTERFAZ ELIMINAR PLAN ACADÉMICO



i) Interfaz Gestión de Cursos

ILUSTRACIÓN 72. INTERFAZ NUEVO CURSO ACADÉMICO

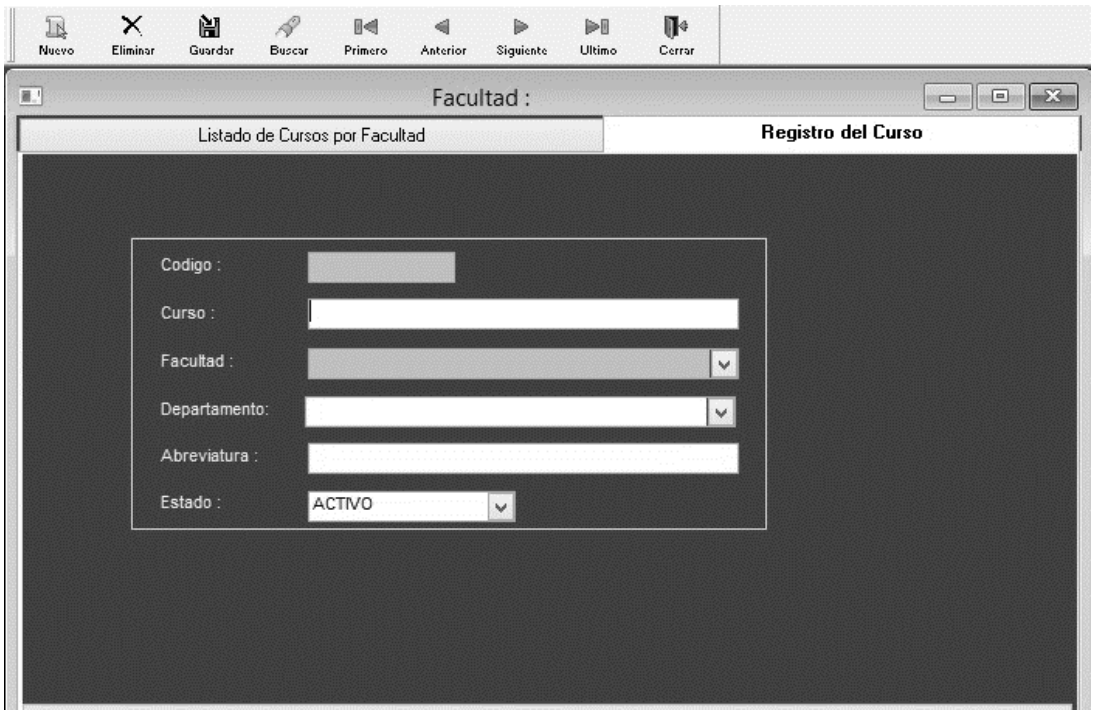


ILUSTRACIÓN 73. INTERFAZ MODIFICAR CURSO ACADÉMICO

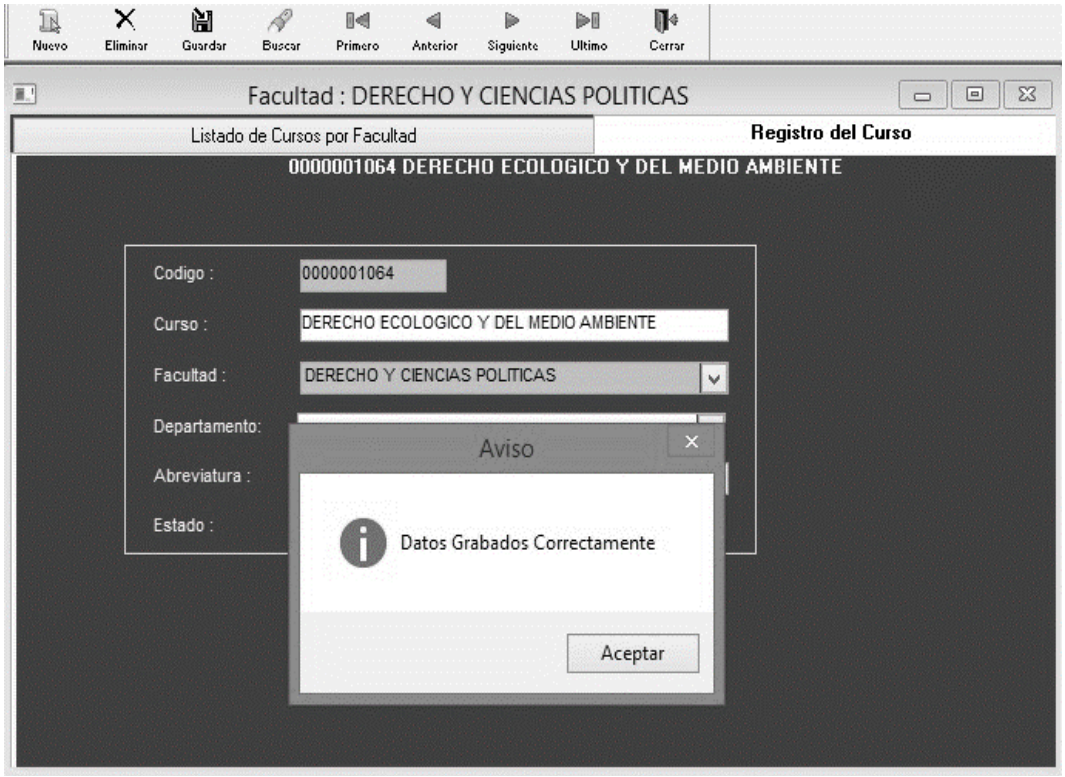


ILUSTRACIÓN 74. INTERFAZ LISTAR CURSO ACADÉMICO

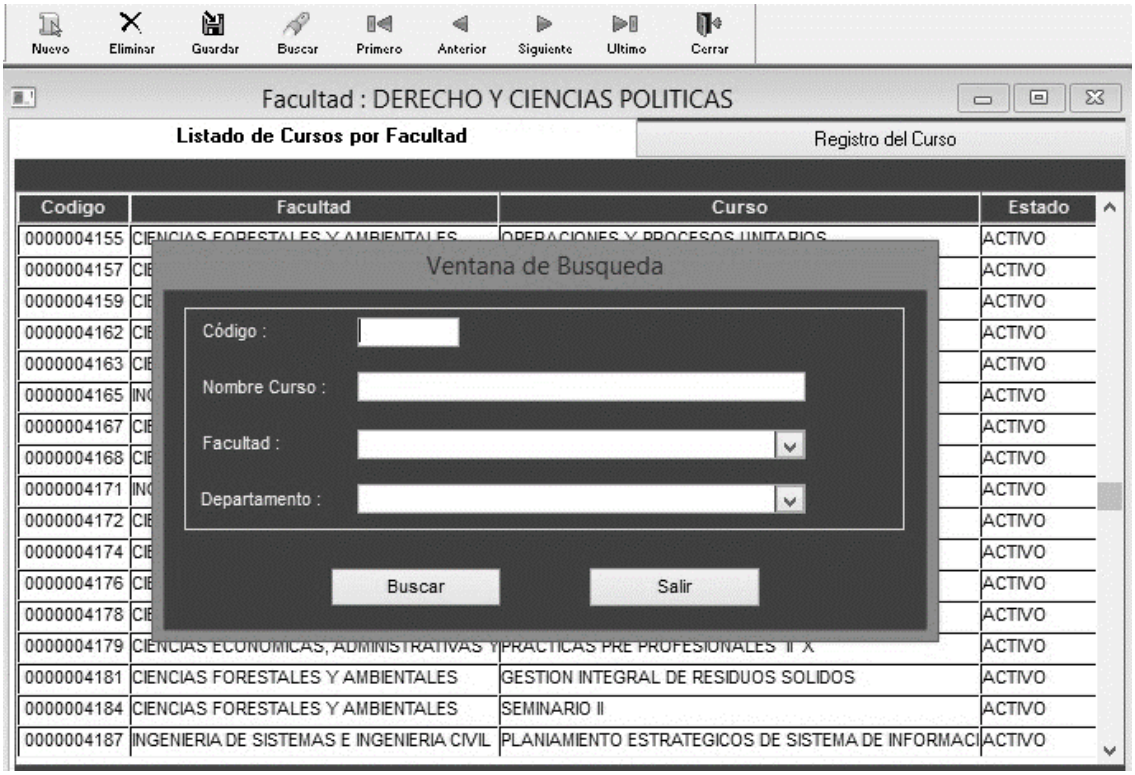
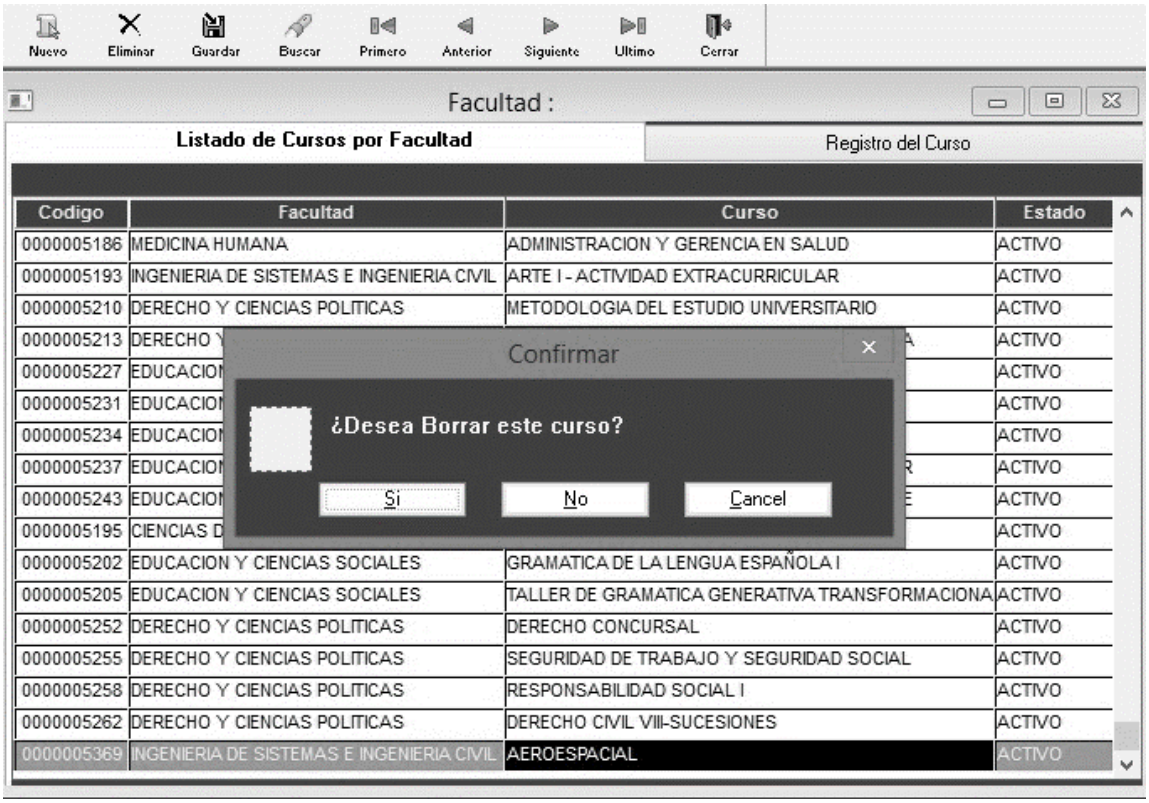


ILUSTRACIÓN 75. INTERFAZ ELIMINAR CURSO ACADÉMICO



j) Interfaz Gestión de sistema de evaluación

ILUSTRACIÓN 76. INTERFAZ NUEVO SISTEMA DE EVALUACIÓN

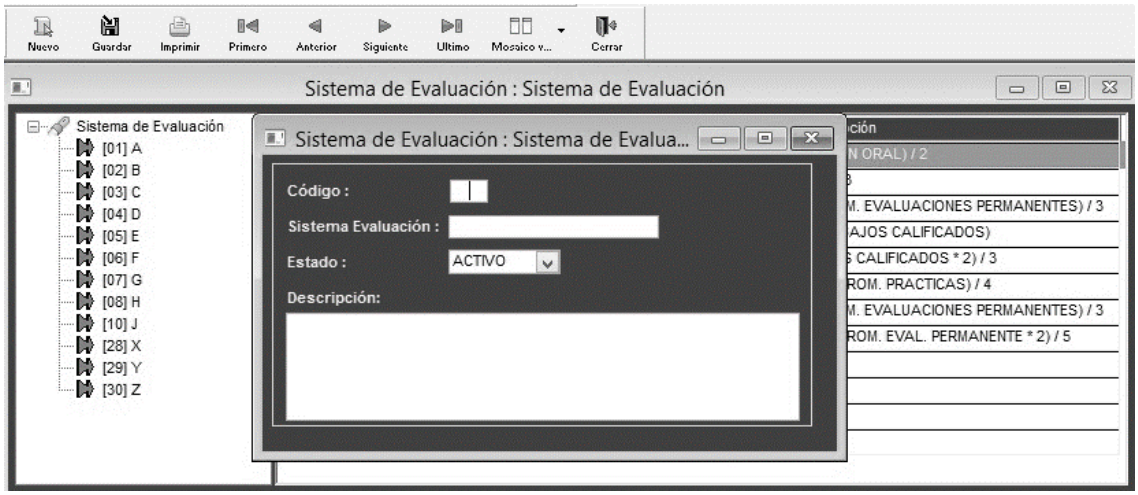


ILUSTRACIÓN 77. INTERFAZ MODIFICAR SISTEMA DE EVALUACIÓN

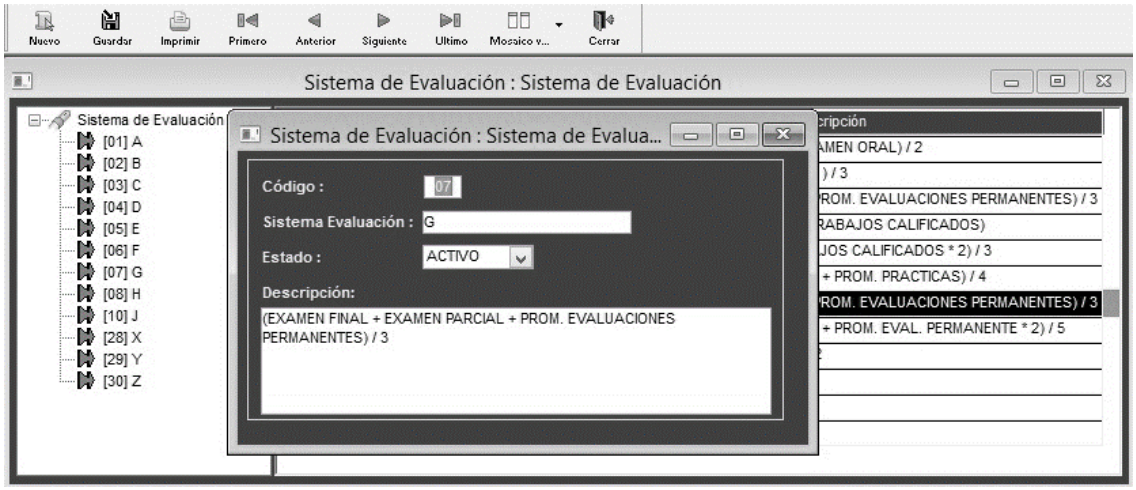


ILUSTRACIÓN 78. INTERFAZ LISTAR SISTEMA DE EVALUACIÓN

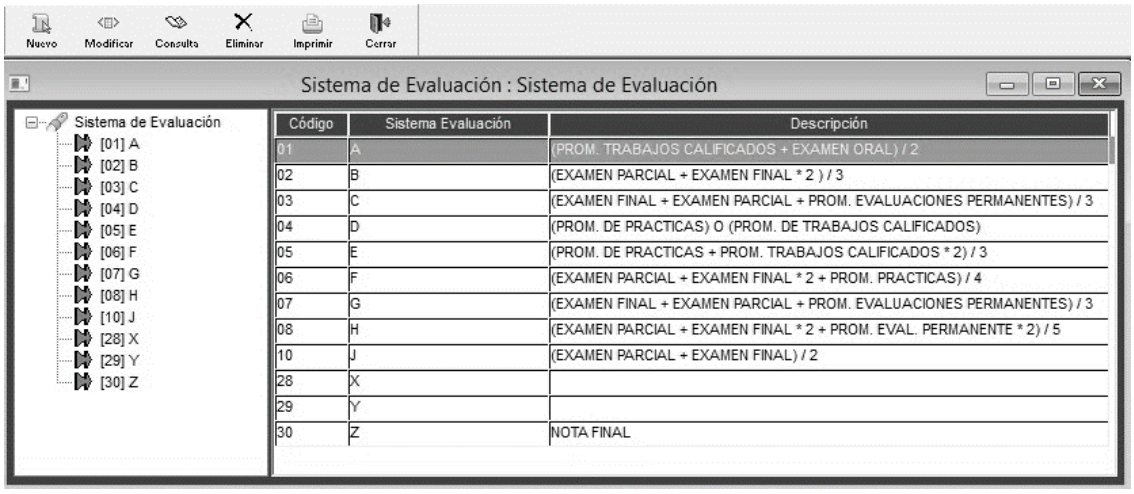
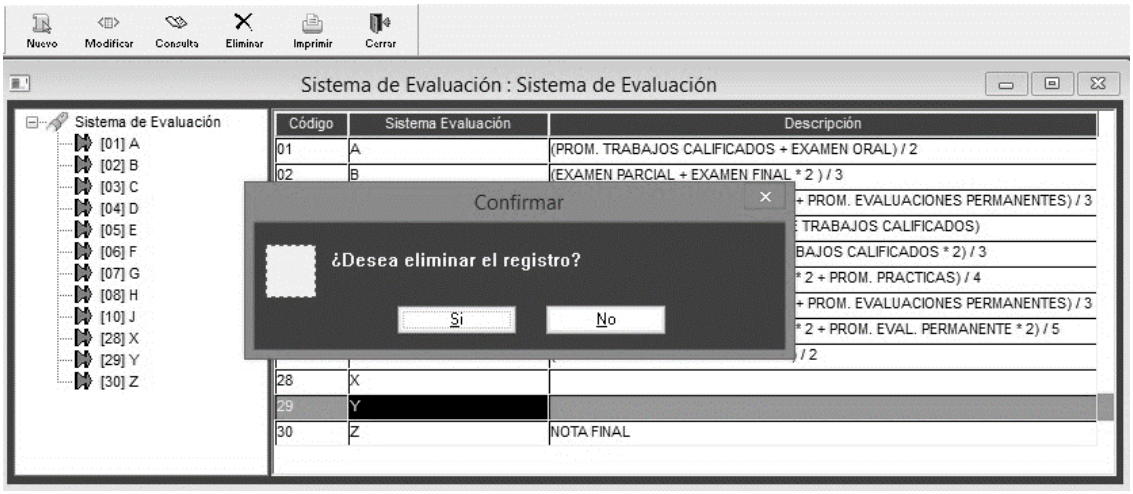


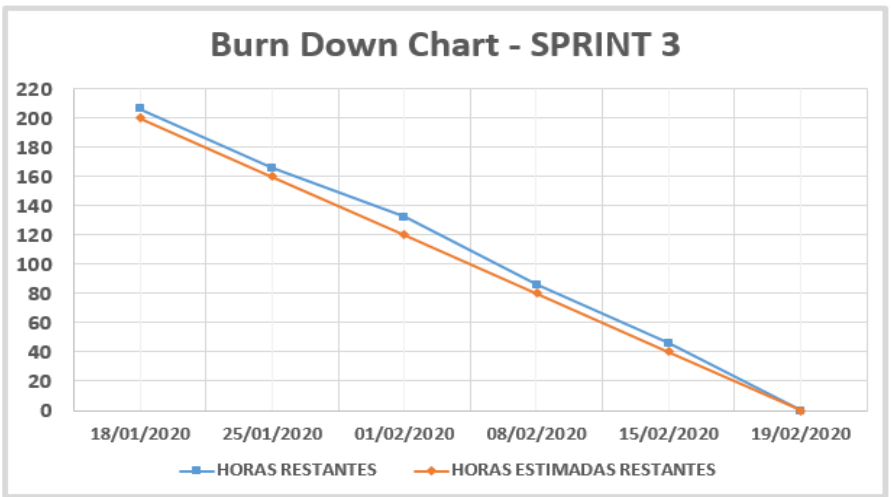
ILUSTRACIÓN 79. INTERFAZ ELIMINAR SISTEMA DE EVALUACIÓN



k) Resultado Burn Down Chart – Sprint 3

En la siguiente ilustración se muestra el desarrollo y resultado de las 6 semanas del sprint 3, donde se puede apreciar que en todo el desarrollo del sprint el Burn Down de desarrollo esta por encima del Burn Down del desarrollo esperado, esto es debido a que se han presentado las tareas fuera del tiempo de establecido, lo cual a conllevado a utilizar mas horas/hombre para poder cumplir con la finalización del sprint en el tiempo pactado.

ILUSTRACIÓN 80. INTERFAZ DE ASIGNACIÓN, MODIFICACIÓN Y DESHABILITACIÓN DE PERFIL A USUARIO



l) Informe de prueba funcional – SPRINT 3

TABLA 44. INFORME DE PRUEBA FUNCIONAL – SPRINT 3

PRUEBA FUNCIONAL – SPRINT 3			
PRUEBA Nº	Prueba de Funcionalidad Nº03	**VERSIÓN DE EJECUCIÓN**	UNU-SP03
TAREA:	Creación, lectura, modificación y deshabilitación de planes académicos Creación, lectura, modificación y deshabilitación de tipo de curso Creación, lectura, modificación y deshabilitación de tipo condición curso Creación, lectura, modificación y deshabilitación de tipo estudio curso Creación, lectura, modificación y deshabilitación de tipo examen Creación, lectura, modificación y deshabilitación de tipo de sistema de evaluación	**FECHA DE EJECUCIÓN** **MÓDULO DEL SISTEMA**	19/02/2020 PLA
Descripción del caso de prueba:	Se procedió a realizar las pruebas de mantenimiento de planes curriculares, tipo de curso, tipo condición de curso, tipo estudio curso, tipo examen y tipo del sistema de evaluación, validando los campos cuando aparecen datos errados y los mensajes de respuesta.		

5. CASO DE PRUEBA

g. Precondiciones

Este Sprint no inicia si no se concluye al 100% el Sprint 2

h. Pasos de prueba

- ✓ Ingresar datos no válidos para verificar los mensajes de rechazo y para validar los campos
- ✓ Verificar que todos los datos relacionados se muestren de manera eficiente.
- ✓ Realizar mantenimiento de planes académicos, creando, modificando y actualizando la información.
- ✓ Realizar mantenimiento de tipo de curso, creando, modificando y actualizando la información.
- ✓ Realizar mantenimiento de tipo condición curso, creando, modificando y actualizando la información de cursos ya creados y registrados.
- ✓ Realizar mantenimiento tipo estudio curso, habilitando, modificando y deshabilitando a cursos ya creados y registrados
- ✓ Realizar mantenimiento tipo examen, habilitando, modificando y deshabilitando las opciones al curso ya creado y registrado.
- ✓ Realizar mantenimiento del sistema de evaluación, habilitando, modificando y deshabilitando a cursos ya creados y registrados

DATOS DE ENTRADA			**RESPUESTA ESPERADA DE LA APLICACIÓN**	**COINCIDE**		**RESPUESTA DEL SISTEMA**
CAMPO	**VALOR**	**TIPO ESCENARIO**		**SI**	**NO**	
Planes académicos	1M	normal	Plan académico registrado	X		Plan Académico registrado
Planes académicos	1%	prueba	Verifique datos ingresados	X		Verifique datos ingresados
Sistema de Evaluación	G	normal	Sistema de evaluación registrada	X		Sistema de evaluación registrada
Sistema de Evaluación	5&	prueba	Verifique datos ingresados	X		Verifique datos ingresados

i. Post Condiciones

No aplica

6. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	**Veredicto**	
	X	**PASO**
		FALLO

Observaciones	**Probador**
	Firma: **Nombre:**
	Fecha:

m) Revisión de Sprint 3 (Sprint Review Meeting)

TABLA 45. INFORME FINAL DE PRUEBA FUNCIONAL – SPRINT 3

Lugar:	Universidad Nacional de Ucayali	
Fecha	19/02/2020	
Número de Iteración / Sprint	Sprint 3	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Jorge Trigueros Bellido José Congona Peña	
¿Qué salió bien en el Sprint? (aciertos)	**¿Qué no salió bien en el Sprint? (errores)**	**Lecciones aprendidas (recomendaciones)**
✓Un buen funcionamiento de las integraciones de código. ✓La recuperación de datos entre los formularios de mantenimiento y la base de datos se ejecuta sin mayores complicaciones. ✓Las asignaciones de diferentes datos a los usuarios creados se han ejecutados sin ningún tipo de inconveniente.	✓ Demasiados retrasos, debido a las constantes interrupciones en el desarrollo del sprint ✓ No hubo integración de historias, debido a que el equipo de trabajo, desarrollo ítems iguales, perdiendo tiempo importante. ✓ Falta de responsabilidad en la entrega de las tareas. ✓ Ejecución de código en el repositorio que dejo inoperativa la última versión del proyecto	✓ Buscar un lugar donde no haya interrupciones. ✓

n) Reunión Retrospectiva – Sprint 3 (Retrospective Meeting)

Concluido el sprint 3, se realizo una reunión no más de dos horas con los miembros del equipo de trabajo para determinar el avance del sprint concluido, donde se va a detallar el resumen del sprint 3.

TABLA 46. INFORME FINAL DE REUNIÓN RETROSPECTIVA – SPRINT 3

Lugar:	Universidad Nacional de Ucayali	
Fecha	19/02/2020	
Número de Iteración / Sprint	Sprint 3	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Jorge Trigueros Bellido José Congona Peña	
Lo bueno (Sprint culminado)	**Lo incomodo**	**Aportes del equipo (Próximo Sprint)**
✓ Se mejoró la comunicación con el dueño del proceso agendando con anticipación las reuniones programadas	✓ Huelgas convocadas por los docentes, donde no se permitió el acceso a las instalaciones de la Universidad Nacional de Ucayali.	✓ Contar con plan de contingencia para continuar con el desarrollo de las actividades (lugar físico fuera de la UNU) en caso de paralización de actividades en los ambientes de la Universidad Nacional de Ucayali (Jorge Trigueros)
✓ Al mejorar la comunicación con el dueño del proceso, también se mejoró la recolección de la información para desarrollar con mayor agilidad las actividades.	✓ Huelgas convocadas por los administrativos, donde no se pudo recabar información a su debido tiempo.	✓ Las reuniones diarias deberán ser inter diarias o dos veces por semana, para tener más tiempo a desarrollar las actividades (Jorge Trigueros)

4.6.4. Sprint 4

El objetivo de este Sprint es desarrollar las tareas para la continuidad del módulo de matrícula virtual del estudiante, y para ello era necesario realizar los sprints anteriores.

a) Priorización de Sprint 4

Las tareas elegidas para este sprint y el orden de priorización han sido tomadas en cuenta siguiendo el orden de ejecución del sistema, este sprint es la parte principal de todo el proyecto, además este sprint estará ubicado en el módulo de matrícula.

b) Formato de asignación del Sprint 4

TABLA 47. ESPECIFICACIÓN SPRINT 4

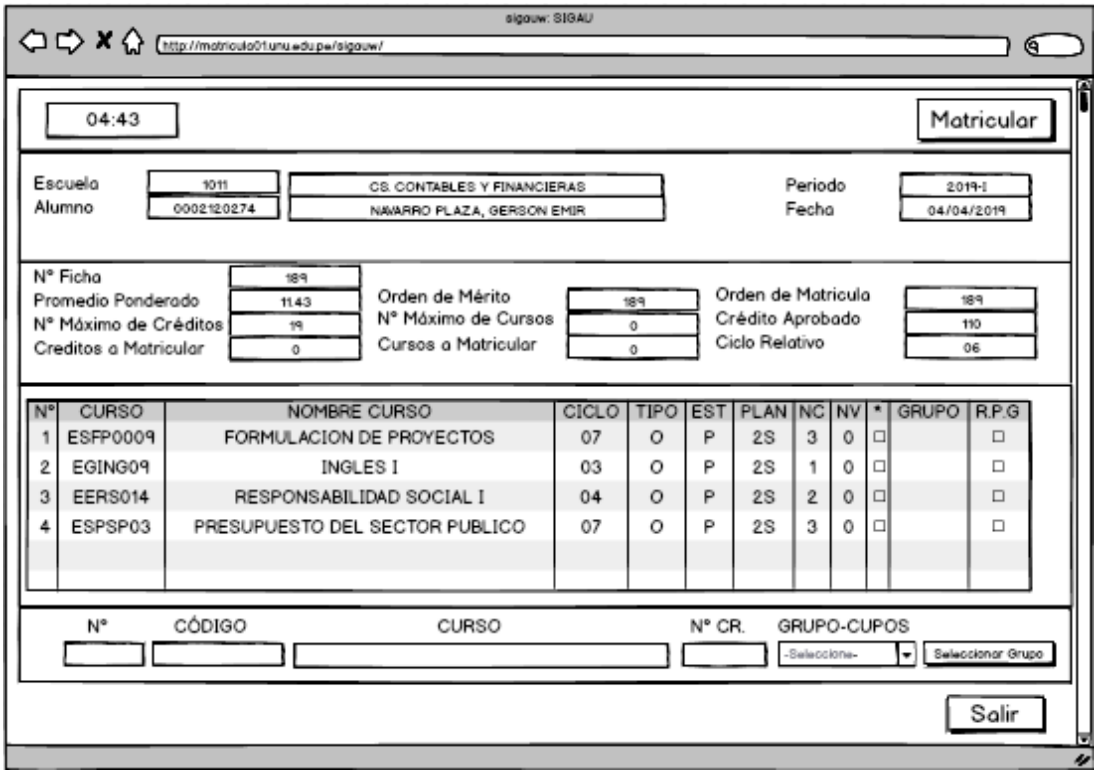
SPRINT 04
FECHA DE INICIO: 24/02/2020
FECHA DE CULMINACIÓN: 31/03/2020

Tareas Pendientes: 0
Días Pendientes: 0

Prioridad del Negocio	Descripción de la tabla	Responsable	Días	Estado
ALTA	Creación, lectura, modificación y deshabilitación de cantidad de ciclos	Jorge Trigueros	5	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de cursos	Jorge Trigueros	5	TERMINADO
ALTA	Registro de asignación de cursos (Matricula)	Jorge Trigueros	15	TERMINADO
ALTA	Creación, lectura, modificación y deshabilitación de pagos	Jorge Trigueros	5	TERMINADO

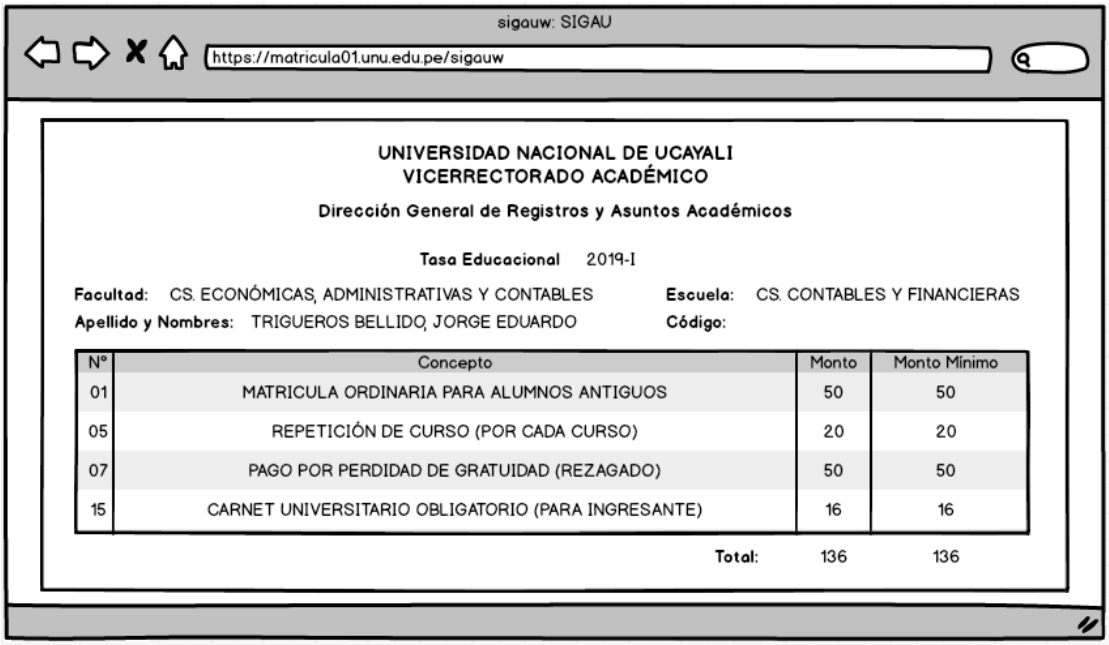
c) Prototipo registro de asignación de curso (matricula)

ILUSTRACIÓN 81. PROTOTIPO REGISTRO DE ASIGNACIÓN DE CURSOS (MATRICULA)



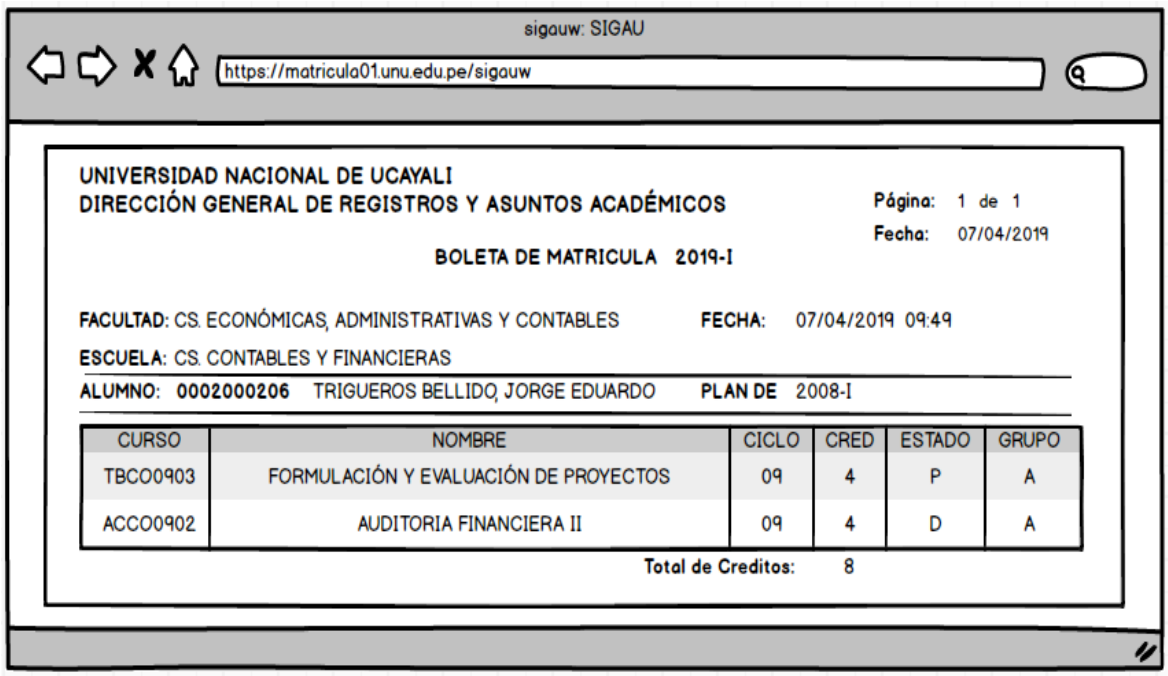
d) Prototipo de Generación de tasa educacional web

ILUSTRACIÓN 82. PROTOTIPO GENERACIÓN DE TASA EDUCACIONAL WEB



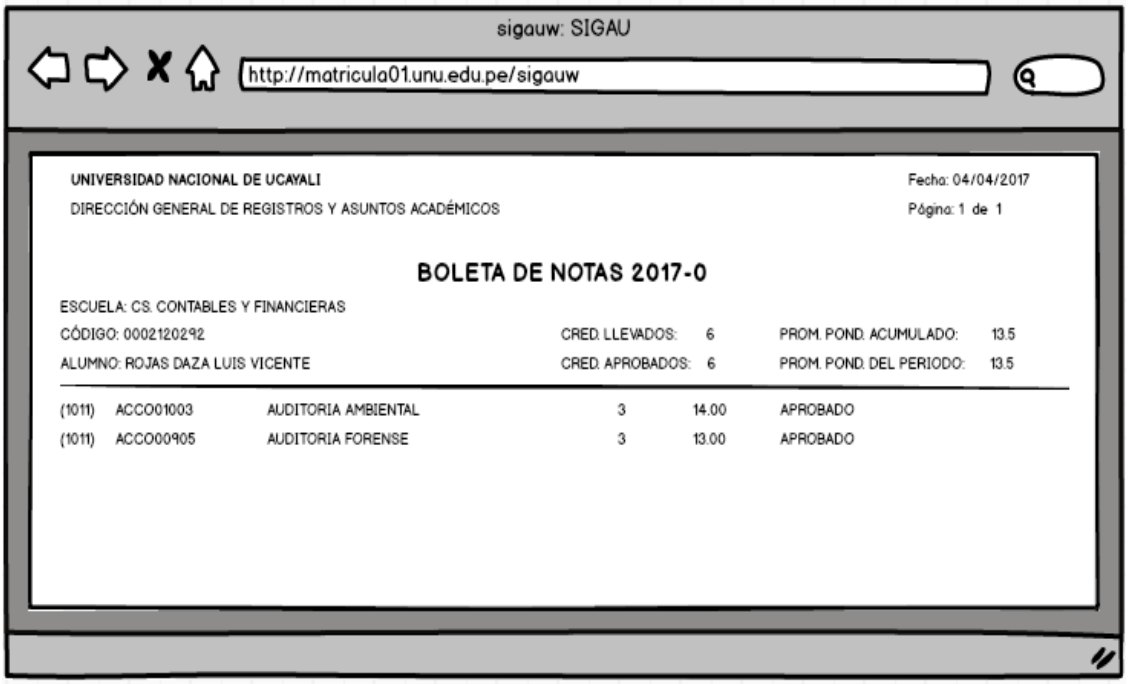
e) Prototipo de Generación de boleta de matrícula web

ILUSTRACIÓN 83. PROTOTIPO GENERACIÓN DE BOLETA DE MATRÍCULA WEB



f) Prototipo Generación Reporte Boleta de Notas Web

ILUSTRACIÓN 84. PROTOTIPO GENERACIÓN REPORTE BOLETA DE NOTAS WEB



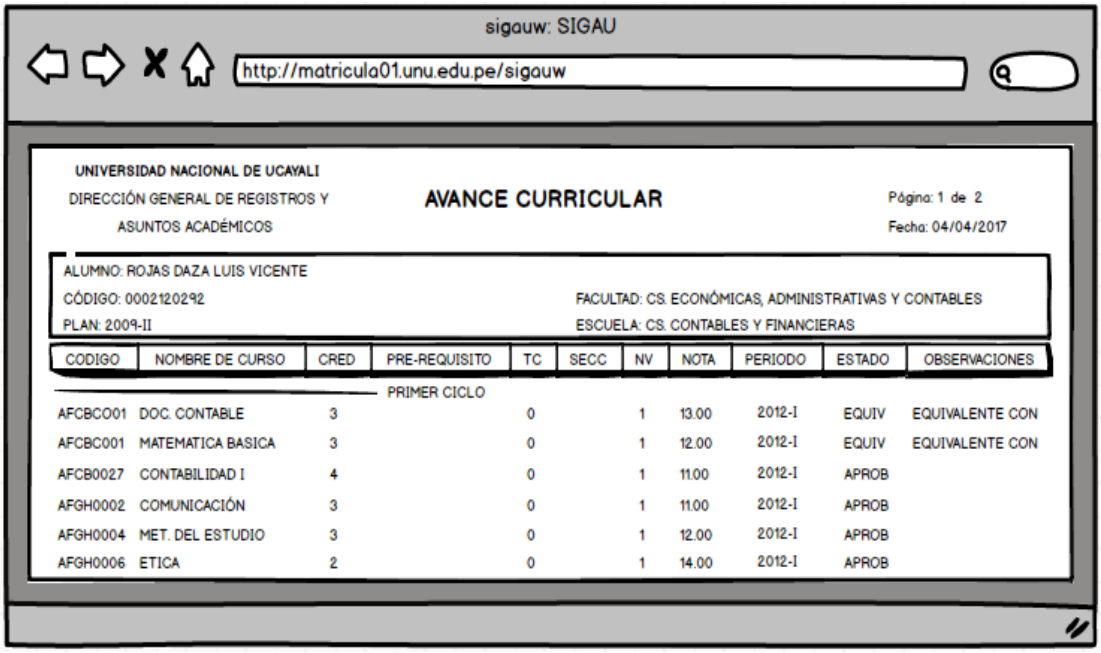
g) Prototipo Generación Reporte de Datos del Estudiante

ILUSTRACIÓN 85. PROTOTIPO GENERACIÓN REPORTE DE DATOS DEL ESTUDIANTE WEB



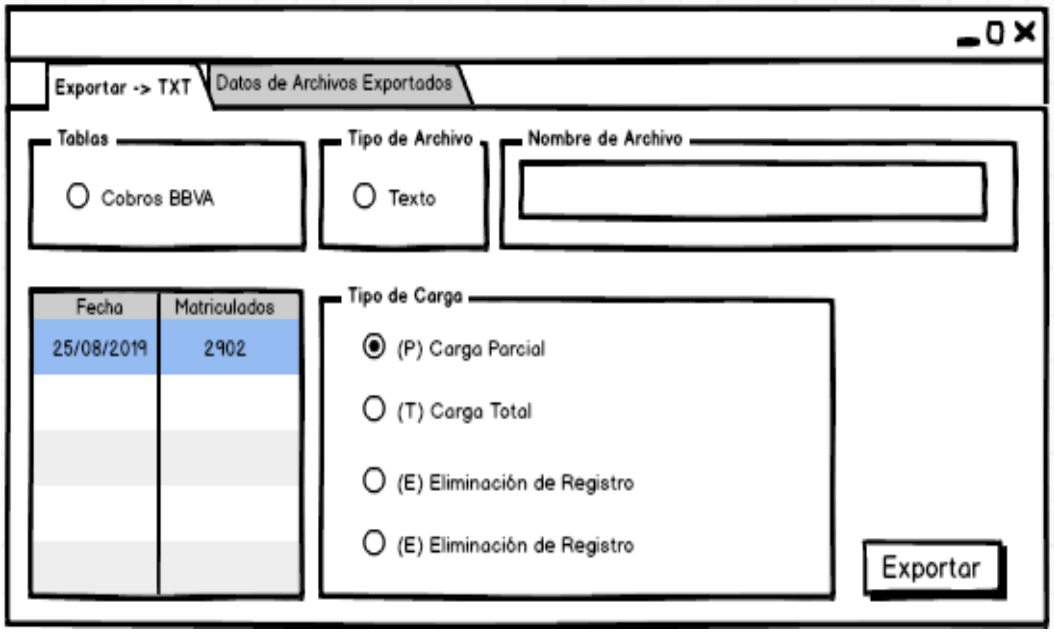
h) Prototipo de Reporte de Avance Curricular web

ILUSTRACIÓN 86. PROTOTIPO REPORTE AVANCE CURRICULAR WEB



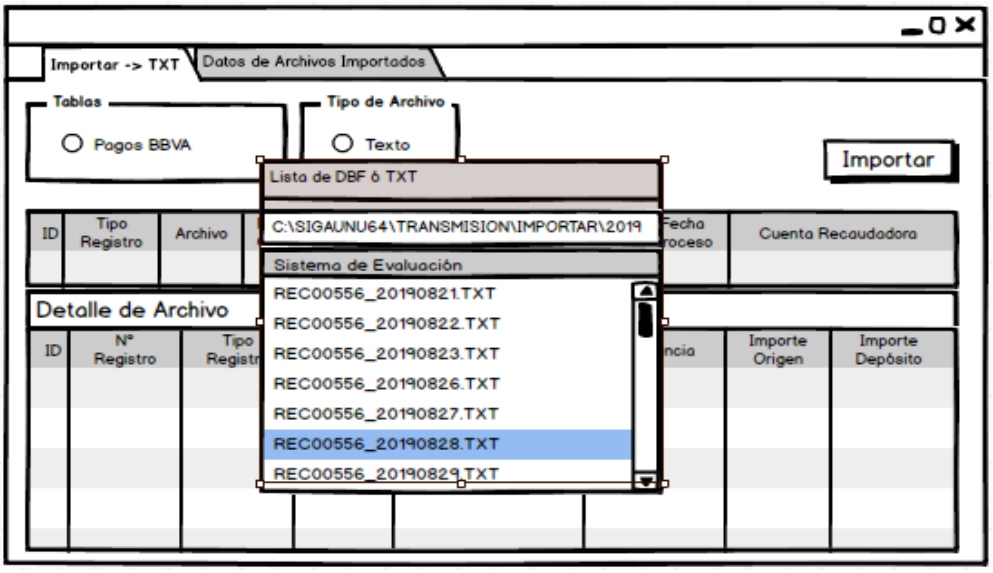
i) Creación, lectura de reporte de generación de pagos para el Banco

ILUSTRACIÓN 87. PROTOTIPO CREACIÓN, LECTURA DE GENERACIÓN DE PAGOS PARA EL BANCO



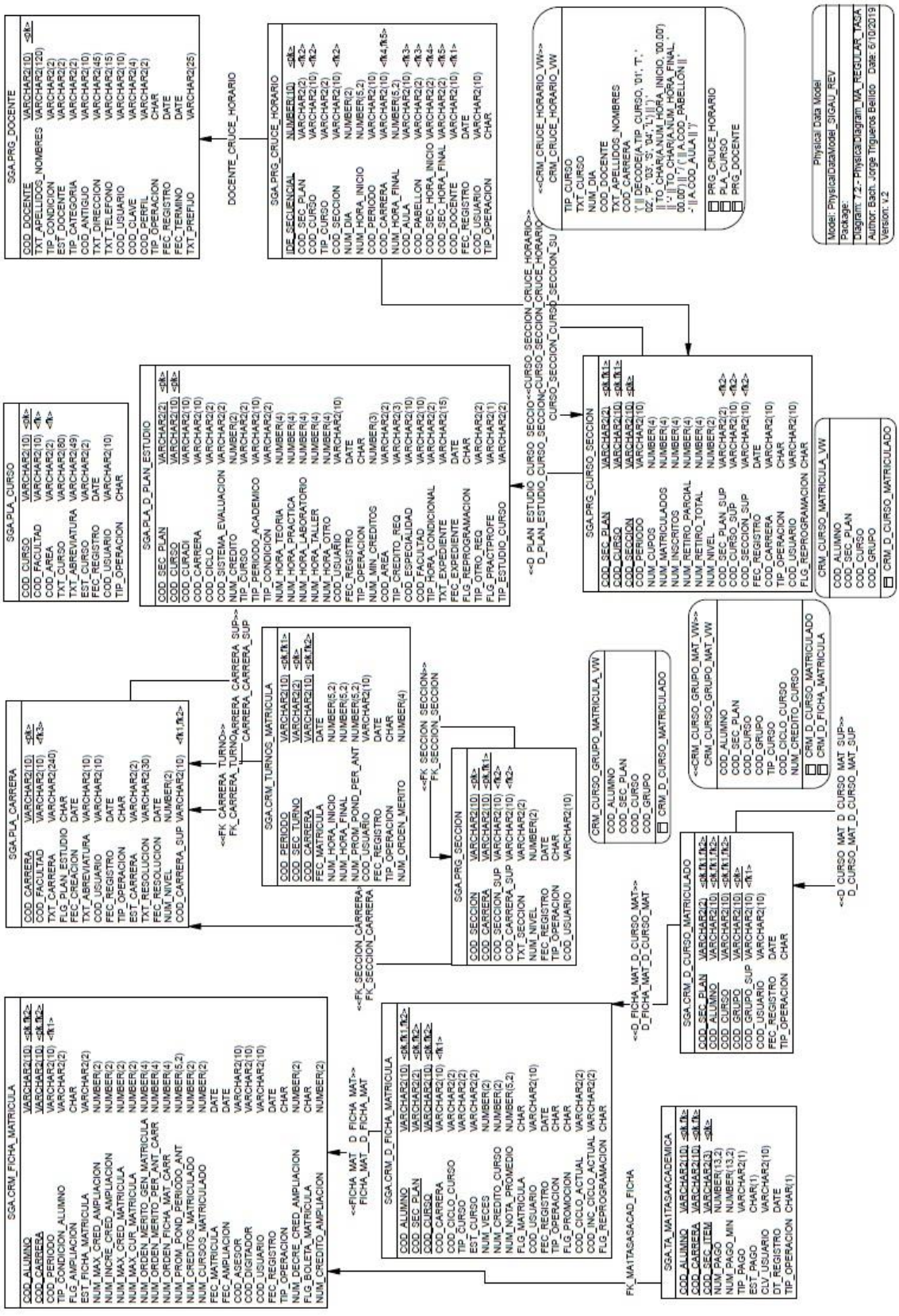
j) Recepción data del Banco para generación de pagos de estudiante

ILUSTRACIÓN 88. PROTOTIPO RECEPCIÓN DATA DE BANCO PARA GENERACIÓN DE PAGOS DE ESTUDIANTES



k) Modelo Físico Spring 4

ILUSTRACIÓN 89. MODELO FÍSICO SPRING 4



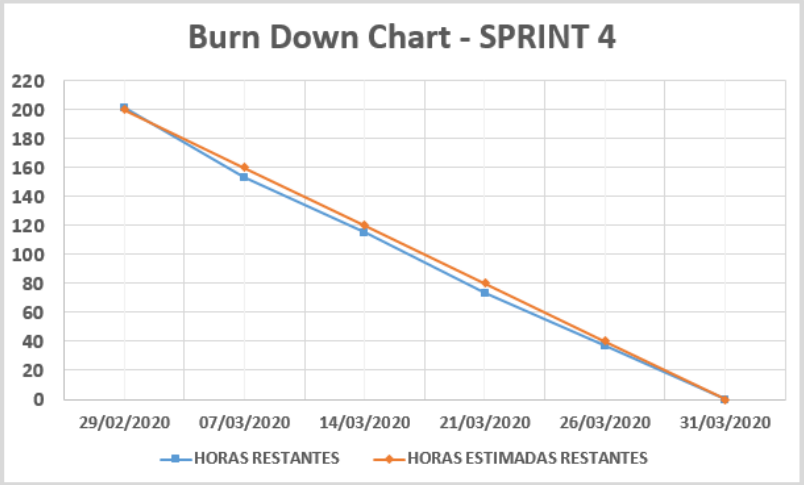
l) Fuente Spring 4

(Ver Anexo 6)

j) Resultado Burn Donw Chart – Sprint 4

En la siguiente ilustración se muestra el desarrollo y resultado de las 6 semanas del sprint 4, donde se puede apreciar que en todo el desarrollo del sprint, el Burn Down de desarrollo está por debajo del Burn Down del desarrollo esperado, donde se ve el resultado de la aplicación de las reunión retrospectiva y concluyendo con el Proyecto en el tiempo establecido.

ILUSTRACIÓN 90. INTERFAZ DE ASIGNACIÓN, MODIFICACIÓN Y DESHABILITACIÓN DE PERFIL A USUARIO



k) Informe de Prueba funcional – SPRINT 4

Tabla 48. Informe de prueba funcional – sprint 4

<table>
<tr><td colspan="6">PRUEBA FUNCIONAL – SPRINT 4</td></tr>
<tr><td>PRUEBA Nº</td><td colspan="2">Prueba de Funcionalidad Nº04</td><td>VERSIÓN DE EJECUCIÓN</td><td colspan="2">UNU-SP04</td></tr>
<tr><td rowspan="2">TAREA:</td><td colspan="2" rowspan="2">Creación, lectura, modificación y deshabilitación de cantidad de ciclos
Creación, lectura, modificación y deshabilitación de cursos.
Registro de asignación de cursos por los estudiantes (Matricula).
Creación, lectura, modificación y deshabilitación de pagos</td><td>FECHA DE EJECUCIÓN</td><td colspan="2">31/03/2020</td></tr>
<tr><td>MÓDULO DEL SISTEMA</td><td colspan="2">PLA
MAT
CTE</td></tr>
<tr><td>Descripción del caso de prueba:</td><td colspan="5">Se procedió a realizar las pruebas de mantenimiento de facultades, carreras profesionales, tipo condición docente, tipo categoría docente y usuario docente, validando los campos cuando aparecen datos errados y los mensajes de respuesta.</td></tr>
<tr><td colspan="6">7. CASO DE PRUEBA</td></tr>
<tr><td colspan="6">j. Precondiciones</td></tr>
<tr><td colspan="6">Este Sprint no inicia si no se concluye al 100% el Sprint 3</td></tr>
<tr><td colspan="6">k. Pasos de prueba</td></tr>
<tr><td colspan="6">✓ Ingresar datos no válidos para verificar los mensajes de rechazo y para validar los campos
✓ Verificar que todos los datos relacionados se muestren de manera eficiente.
✓ Realizar mantenimiento de ciclos, creando, modificando y actualizando la información.
✓ Realizar mantenimiento de asignación de cursos matriculados a estudiantes, creando, modificando y actualizando la información.
✓ Realizar mantenimiento de cursos, creando, modificando y actualizando la información
✓ Realizar mantenimiento de pagos, habilitando, modificando y deshabilitando a estudiantes matriculados registrados</td></tr>
<tr><td colspan="3">DATOS DE ENTRADA</td><td rowspan="2">RESPUESTA ESPERADA DE LA APLICACIÓN</td><td>COINCIDE</td><td rowspan="2">RESPUESTA DEL SISTEMA</td></tr>
<tr><td>CAMPO</td><td>VALOR</td><td>TIPO ESCENARIO</td><td>SI</td><td>NO</td></tr>
<tr><td>Usuario</td><td>0002000206</td><td>prueba</td><td>Ingrese contraseña</td><td>X</td><td></td><td>Ingrese contraseña</td></tr>
<tr><td>Usuario/contraseña</td><td>0002000206 / 1234</td><td>normal</td><td>Permite el acceso al sistema</td><td>X</td><td></td><td>Permite el acceso al sistema</td></tr>
<tr><td>Usuario/contraseña</td><td>0002000206 / 2468</td><td>prueba</td><td>Verifique datos ingresados</td><td>X</td><td></td><td>Verifique datos ingresados</td></tr>
<tr><td>Asignar curso</td><td>ESTET02</td><td>Normal</td><td>Curso Asignado a estudiante</td><td>X</td><td></td><td>Curso Asignado a estudiante</td></tr>
<tr><td>Código Estudiante</td><td>0002000206$</td><td>prueba</td><td>Verifique datos ingresados</td><td>x</td><td></td><td>Verifique datos ingresados</td></tr>
<tr><td colspan="6">l. Post Condiciones</td></tr>
<tr><td colspan="6">No aplica</td></tr>
<tr><td colspan="6">8. RESULTADOS DE LA PRUEBA</td></tr>
<tr><td colspan="4">Defectos y desviaciones</td><td colspan="2">Veredicto</td></tr>
<tr><td colspan="4" rowspan="2"></td><td>X</td><td>PASO</td></tr>
<tr><td></td><td>FALLO</td></tr>
<tr><td colspan="4">Observaciones</td><td colspan="2">Probador</td></tr>
<tr><td colspan="4" rowspan="3"></td><td colspan="2">Firma:</td></tr>
<tr><td colspan="2">Nombre:</td></tr>
<tr><td colspan="2">Fecha:</td></tr>
</table>

l) Revisión de Sprint 4 (Sprint Review Meeting)

TABLA 49. INFORME FINAL DE PRUEBA FUNCIONAL – SPRINT 4

Lugar:	Universidad Nacional de Ucayali	
Fecha	31/03/2020	
Número de Iteración / Sprint	Sprint 4	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
¿Qué salió bien en el Sprint? (aciertos)	**¿Qué no salió bien en el Sprint? (errores)**	**Lecciones aprendidas (recomendaciones)**
✓Los tiempos estimados para el desarrollo de cada actividad del sprint fueron los necesarios para terminar cada una de las tareas dentro de los plazos establecidos. ✓Al realizar cada módulo de trabajo de manera independiente siguiendo el orden planteado no hubo problemas al momento de generar y enlazar el código necesario para la carga de cada uno de ellos	✓ Al tener mantenimientos independientes se terminaba de ejecutar una ventana y se tenían que volver a actualizar y probar las demás debidos a que la información de unas alimentaba a otra y viceversa generando algunas complicaciones.	✓ Se recomienda hacer un análisis de todas las actividades que se puedan presentar dentro del desarrollo del proyecto para que las fechas y plazos dados no se prolonguen demasiado.

m) Reunión Retrospectiva – Sprint 4 (Retrospective Meeting)

TABLA 50. INFORME FINAL DE REUNIÓN RETROSPECTIVA – SPRINT 4

Lugar:	Universidad Nacional de Ucayali	
Fecha	31/03/2020	
Número de Iteración / Sprint	Sprint 4	
Personas convocadas a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Personas que asistieron a la reunión	Pedro Campos Cabrera Jorge Trigueros Bellido José Congona Peña	
Lo bueno (Sprint culminado)	**Lo incomodo**	**Aportes del equipo (Próximo Sprint)**
✓ No se presentaron paralizaciones durante el desarrollo del sprint, lo cual se concluyó en el tiempo establecido	✓ Presión por parte de las autoridades universitarias para culminar el desarrollo del proyecto antes del tiempo previsto.	✓ Una mejor planificación de los tiempos de desarrollo para no incumplir con los entregables durante todo el proyecto (Jorge Trigueros)
✓Se mejoraron los tiempos de desarrollo del sprint.		✓ Realizar un análisis de riesgos antes de cada sprint para que el desarrollo del proyecto no tenga retrasos en los entregables.
✓Se pudo entregar el proyecto final en el tiempo establecido, quedando la Universidad Nacional de Ucayali conforme.		

4.7. Pruebas de entregables

4.7.1. Pruebas de autentificación de usuarios

TABLA 51. PRUEBA DE AUTENTIFICACIÓN DE USUARIO

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Usuario Incorrecto	El sistema debe mostrar un mensaje de error de ingreso al sistema	El sistema mostró el mensaje de error de acceso
2	Clave incorrecta	El sistema debe mostrar un mensaje de error de ingreso al sistema	El sistema mostró el mensaje de error de acceso
3	Usuario y clave correctos	El sistema debe permitir el acceso al sistema identificando el perfil del usuario.	El sistema permitió el acceso al sistema, identificando el perfil.

4.7.2. Pruebas de gestión de unidades

TABLA 52. PRUEBA DE GESTIÓN DE UNIDADES

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Usuario y contraseña del administrador del Sistema	El Sistema deberá permitir el acceso al sistema con el perfil de administrador del Sistema	El sistema permitió el acceso al sistema, identificando el perfil.
2	Datos requeridos para la creación de facultades	El sistema deberá solicitar los datos obligatorios para la creación de facultades y carreras profesionales	El sistema emitió un mensaje crítico indicando que falta datos para terminar de registrar.
3	Datos duplicados	El sistema debe mostrar un mensaje de error de ingreso de los datos indicando que dicha facultad o carrera ya se encuentra registrada	El sistema emitió un mensaje de error indicando que facultad y/o carrera ya existe
4	Datos para guardar las facultades	El sistema deberá guardar la facultad y/o carrera registrada	El sistema registró la facultad y/o carrera
5	Visualización de las facultades	Una vez registrada la facultad y/o carrera, el sistema permitirá la visualización de la misma.	El sistema permitió la visualización de la facultad y/o carrera registrada.

4.7.3. Pruebas de gestión de docentes

TABLA 53. PRUEBA DE GESTIÓN DE DOCENTES

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Datos para guardar un docente	Ingresar la información solicitada del docente, el sistema deberá guardar en la base de datos y emitirá un mensaje de “Docente Registrado”	Se guardó la información registrada del docente y el sistema mostró el mensaje de “Docente Registrado”
2	Datos para actualizar un docente	Se modificará los datos del docente y se actualizará en el sistema, este cambio se confirmará con un mensaje de “Cambio Exitoso”	Se actualizó la información del docente y el sistema mostró un mensaje de “Cambio Exitoso”
3	Visualización de Docentes	Una vez registrada la actualización, el sistema deberá visualizar los cambios realizados.	El sistema permitió la visualización de las modificaciones registradas del Docente

4.7.4. Pruebas de gestión de estudiantes

TABLA 54. PRUEBA DE GESTIÓN DE ESTUDIANTES

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Datos para guardar un estudiante	Llenar la información solicitada y el estudiante se guardará en la base de datos mientras el sistema nos muestra un mensaje de éxito	El estudiante se guardó en la base de datos y el sistema nos mostró el mensaje de éxito
2	Dato DNI duplicado	Al intentar grabar un estudiante con un DNI ya registrado, el sistema nos mostrará un mensaje de advertencia, indicando que dicha persona ya se encuentra en la base de datos y nos dará la opción de grabar y cambiar su tipo de pago	El sistema mostró un mensaje de advertencia indicando que estudiante ya se encuentra en la base de datos y nos dio la opción de modificar su tipo de pago
3	Datos para actualizar un estudiante	Modificaremos el campo a actualizar en el sistema y este cambio se guardará dándonos un mensaje de éxito en la interfaz	El estudiante se actualizó en la BD y el sistema nos mostró un mensaje de éxito
4	Visualización de estudiantes	Una vez registrado el estudiante el sistema permitirá la visualización del mismo	El sistema permitió la visualización de los estudiantes en la interfaz

4.7.5. Pruebas de gestión de cursos

TABLA 55. PRUEBA DE GESTIÓN DE CURSOS

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Datos para guardar cursos	Llenar la información solicitada y el curso se guardará en la base de datos mientras el sistema nos muestra un mensaje de éxito	El curso se guardó en la Base de datos y el sistema nos mostró el mensaje de éxito
2	Dato código curso duplicado	Al intentar guardar un curso con el mismo código registrado, el sistema mostrará un mensaje de error, indicando que el código de curso ya existe	El sistema mostró un mensaje de error indicando que el código de curso ya existe en la Base de datos.
3	Datos para actualizar curso	Modificaremos el campo a actualizar en el sistema y este cambio se guardará en la base de datos dándonos un mensaje de éxito en la interfaz	El curso se actualizó en la base de datos y el sistema nos mostró el mensaje de éxito.
4	Visualización de cursos	Una vez guardado el curso el sistema permitirá la visualización del mismo	El sistema permitió la visualización del cliente en la interfaz

4.7.6. Pruebas de gestión de perfiles

TABLA 56. PRUEBA DE GESTIÓN DE PERFILES

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Datos para crear un perfil	Llenar la información solicitada y el perfil se creará en la Base de datos mientras el sistema nos muestra un mensaje de éxito	El perfil se creó en la BD y el sistema nos mostró el mensaje de éxito
2	Datos para asignar un perfil a un usuario	Asignar el perfil a un determinado usuario y se guardará en la Base de datos mientras el sistema nos emitirá un mensaje de éxito	La asignación se guardó en la BD y el sistema nos mostró el mensaje de éxito
3	Datos actualizar asignación de perfiles	Se agregará o se revocará perfiles en los usuarios y este cambio se guardará en la base de datos mientras el sistema nos mostrará un mensaje de éxito	La revocación y reasignación de perfiles se guardó en la BD y el sistema nos mostró el mensaje de éxito
4	Visualización de perfiles	Una vez guardado los perfiles el sistema permitirá la visualización del mismo	El sistema permitió la visualización de los perfiles en la interfaz

4.7.7. Pruebas de gestión de planes académicos

TABLA 57. PRUEBA DE PLANES ACADÉMICOS

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Datos para guardar un plan académico	Llenar la información solicitada y el plan académico se guardará en la base de datos mientras el sistema nos muestra un mensaje de éxito.	El plan académico se guardó en la base de datos y el sistema nos mostró el mensaje de éxito
2	Datos código plan académico duplicado	Al intentar grabar un plan académico con un código registrado, el sistema mostrará un mensaje de error, indicando que el código de plan académico ya existe	El sistema mostró un mensaje de error indicando que el código de plan académico ya existe en la Base de Datos.
3	Datos para actualizar el plan académico	Modificaremos el campo a actualizar en el sistema y este cambio se guardará dándonos un mensaje de éxito en la interfaz	El plan académico se actualizó en la base de datos y el sistema nos mostró un mensaje de éxito
4	Visualización de planes académicos	Una vez guardado el plan académico el sistema permitirá la visualización del mismo	El sistema permitió la visualización del plan académico en la interfaz

4.7.8. Pruebas de gestión de registrar matricula

TABLA 58. PRUEBA DE GESTIÓN REGISTRAR MATRICULA

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Validación de acceso al sistema	Al dar clic en Ingresar el sistema debe validar los datos ingresados y constatar con la información en la base de datos, en caso de no existir el sistema enviará un mensaje de error.	El sistema validó información ingresada y en caso de datos errados el sistema envió mensaje de error
2	Seleccionar cursos errados	Al seleccionar un curso que no corresponde, el sistema debe emitir un mensaje de “Debe matricularse primero en cursos obligatorios”	Al seleccionar el curso el sistema emitió el mensaje de error

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
3	Seleccionar cursos obligatorios	Al seleccionar el curso correspondiente, el sistema debe habilitar la opción grupo	Al seleccionar el curso el sistema habilito la opción grupo
4	Seleccionar grupo-curso	Al seleccionar el grupo del curso el sistema debe asignar el grupo al curso y cambiar de estado seleccionado (color amarillo)	Al asignar grupo al curso el sistema cambio estado de curso y cambio color
5	Matricularse en más créditos permitidos	Al tratar de seleccionar más cursos con un total de créditos de los permitidos el sistema deberá emitir un mensaje de error	Al seleccionar cursos adicionales con créditos por demás de lo permitido el sistema emitió el mensaje de error
6	Datos para actualizar matricula	Modificaremos el campo a actualizar en el sistema y este cambio se guardará dándonos un mensaje de éxito en la interfaz	La matrícula se actualizó en la base de datos y el sistema nos mostró un mensaje de éxito
7	Datos para generar matricula	Al generar Matricula con los datos correspondientes el Sistema debe mostrar un mensaje de éxito	Se generó matrícula y el sistema emitió mensaje de éxito
8	Datos para generar tasa educacional	Al registrar matricula el sistema automáticamente debe generar la tasa educacional	Se generó matrícula y el sistema automáticamente generó la tasa educacional
9	Datos para generar ficha de matricula	Al registrar matricula el sistema automáticamente debe generar la ficha de matrícula	Se generó matrícula y el sistema automáticamente generó la ficha de matrícula

4.7.9. Pruebas de gestión pagos

TABLA 59. PRUEBA DE GESTIÓN DE PAGOS DE ESTUDIANTES

N° PRUEBA	VARIABLES	SE ESPERA	SE OBTUVO
1	Datos para generar reporte de tasa educacional del día	Al final del día el sistema debe generar reporte de estudiantes matriculados y que han generado su tasa educacional	El sistema emitió reporte de tasa educacional del día por estudiante matriculado
2	Datos para registrar pagos	Al migrar la data enviada por el banco el sistema debe reconocer el archivo e identificar la información para su posterior registro de pago por estudiante	El sistema reconoció el archivo, identificó la información y registró pagos de cada estudiante
3	Datos para actualizar el pago	Modificaremos el campo a actualizar en el sistema y este cambio se guardará dándonos un mensaje de éxito en la interfaz	El pago se actualizó en la base de datos y el sistema nos mostró un mensaje de éxito
4	Visualización pago	Una vez registrado el pago el sistema permitirá la visualización del mismo	El sistema permitió la visualización del pago en la interfaz

4.8. Integración con otros sprints

4.8.1. Integración SPRINT 1 y SPRINT 2

Esta integración de SPRINT integra historias de usuario 1, 2, 3, 4, 5, 6, 7 correspondiente al SPRINT 1, y 8, 9, 10, 11 correspondiente del SPRINT 2, esta integración proporciona la información e inicialización del sistema para el módulo de seguridad y permite la asignación de perfiles a usuarios.

TABLA 60. INTEGRACIÓN SPRINT 1 Y 2

N° PRUEBA	INTEGRACION	DESCRIPCIÓN	SE ESPERA	SE OBTUVO
1	Información básica para el módulo de seguridad e información del sistema	La integración de estos dos SPRINT, se basa en la información básica y fundamental para el correcto funcionamiento del sistema, como autentificación de usuarios, datos de usuarios datos de estudiantes, datos docentes, datos facultad	Con la correcta carga de información se espera continuar con el proceso de matricula	Se obtuvo una correcta carga e inicialización del sistema para el módulo de seguridad
2	Asignación de perfiles a los usuarios	Al dar clic sobre crear usuario o docente, en la interfaz se nos debe permitir asignar un perfil	El sistema solo me permitirá escoger entre los usuarios que pertenezcan a la parte administrativa y docente	El sistema se comportó de la forma esperada

4.8.2. Integración SPRINT 1, 2 y 3

Esta integración de SPRINT integra historias de usuario 1, 2, 3, 4, 5, 6, 7 correspondiente al SPRINT 1, historias de usuario 8, 9, 10, 11 correspondiente del SPRINT 2 e historias de usuario 12, 13, 14, 15, 16, 17, 18 y 19 correspondiente del SPRINT 3 esta integración proporciona la información e inicialización del sistema para el módulo de planes y permite la asignación de cursos a los planes.

TABLA 61. INTEGRACIÓN SPRINT 1, 2 Y 3

N° PRUEBA	INTEGRACION	DESCRIPCIÓN	SE ESPERA	SE OBTUVO
1	Asignación de facultades a planes académicos	Al crear un plan académico es necesario primero ingresar la facultad al que pertenecerá, con esta información se puede seguir con la creación y asignación de planes de estudio	El sistema permite asignar solo una facultad por plan de estudio y esto depende solo y únicamente si la facultad ha sido registrado previamente	El sistema se comportó de la forma esperada
	Asignación de carrera profesional a planes académicos	Al crear un plan académico es necesario primero ingresar la facultad y la carrera profesional al que pertenecerá con esta información se pueden crear y asignar planes de estudio		El sistema se comportó de la forma esperada

4.8.3. Integración SPRINT 1, 3 y 4

Esta integración de SPRINT integra historias de usuario 1, 2, 3, 4, 5, 6, 7 correspondiente al SPRINT 1, historias de usuario 12, 13, 14, 15, 16, 17, 18 y 19 correspondiente del SPRINT 3 e historias de usuario 20 y 21 correspondiente del SPRINT 4, esta integración proporciona la información e inicialización del sistema para el módulo de matrícula y permite la asignación de cursos a los estudiantes.

TABLA 62. INTEGRACIÓN SPRINT 1, 3 Y 4

N° PRUEBA	INTEGRACION	DESCRIPCIÓN	SE ESPERA	SE OBTUVO
1	Asignación de cursos a los estudiantes (Matricula)	Al generar la asignación de los cursos, primero se debe asignar los grupos según disponibilidad y p	El sistema permite asignar solo una facultad por plan de estudio y esto depende solo y únicamente si la facultad ha sido registrado previamente	El sistema se comportó de la forma esperada
2	Asignación de carrera profesional a planes académicos	Al crear un plan académico es necesario primero ingresar la facultad y la carrera profesional al que pertenecerá con esta información se pueden crear y asignar planes de estudio		El sistema se comportó de la forma esperada

4.9. Implantación del sistema

Para la construcción del sistema web de matrícula, se usó el lenguaje de programación Java, el gestor de base de datos Oracle en su versión 11g, un servidor de aplicaciones, un servidor para la Base de Datos Cliente/Servidor, un servidor para la base de datos web y herramientas de terceros.

De la misma manera para ejecutar el sistema de matrícula web se ha utilizado dos servidores de aplicaciones y estructura de redes con conexión a internet.

4.9.1. Sistema operativo

Para el desarrollo del sistema se usó el sistema operativo Windows 8 Ultimate Edition en su versión en español, para que se ejecute el sistema web de matrícula se utilizó Windows Server.

4.9.2. Software de aplicación

El Software que se necesita para que el sistema web de matrícula se ejecute correctamente es:

- Oracle versión 11g
- Java JDK 8.xx
- Apache Tomcat 8.x

4.9.3. Hardware

Para la ejecución del sistema de una manera correcta, se requirió la instalación de un servidor con las siguientes características:

- Procesador Core i7
- RAM 32Gb en adelante
- Disco Duro 1Tb como mínimo
- Tarjeta de red 100/1000 Mb

4.9.4. Red

El sistema es una aplicación que puede ser ejecutada desde una intranet, como también desde internet, y por la gran cantidad de conexiones activas que tendrá, se debe contar con el siguiente equipo y servicio:

- Switch 100/1000Mb
- Conexión a internet mínima de 30Mb

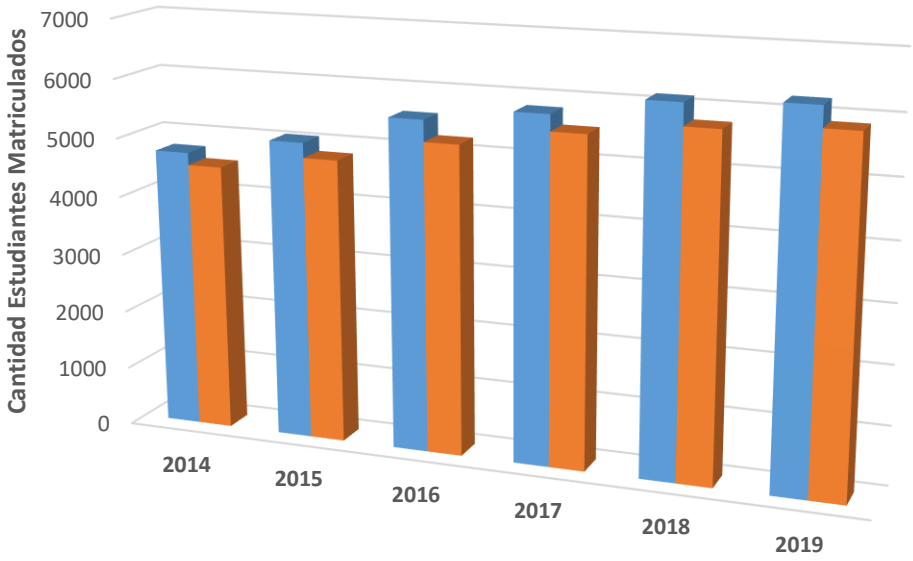
4.9.5. Instalación de la aplicación

Una vez implementado el hardware y el software mínimo necesario, se procedió con el despliegue del sistema.

Se instaló el gestor de base de datos Oracle en su versión 11g. Ver Anexos 3 y 4.

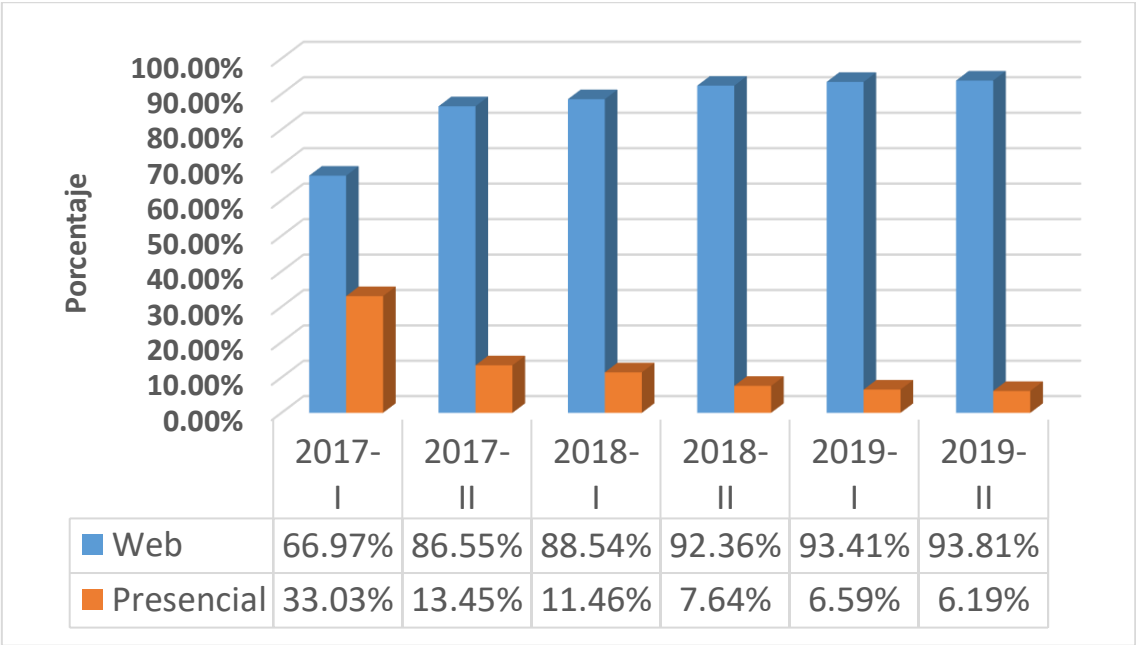
4.10. Reportes de información para la toma de decisiones

ILUSTRACIÓN 91: REGISTRO DE ALUMNOS MATRICULADOS – DATA HISTÓRICA



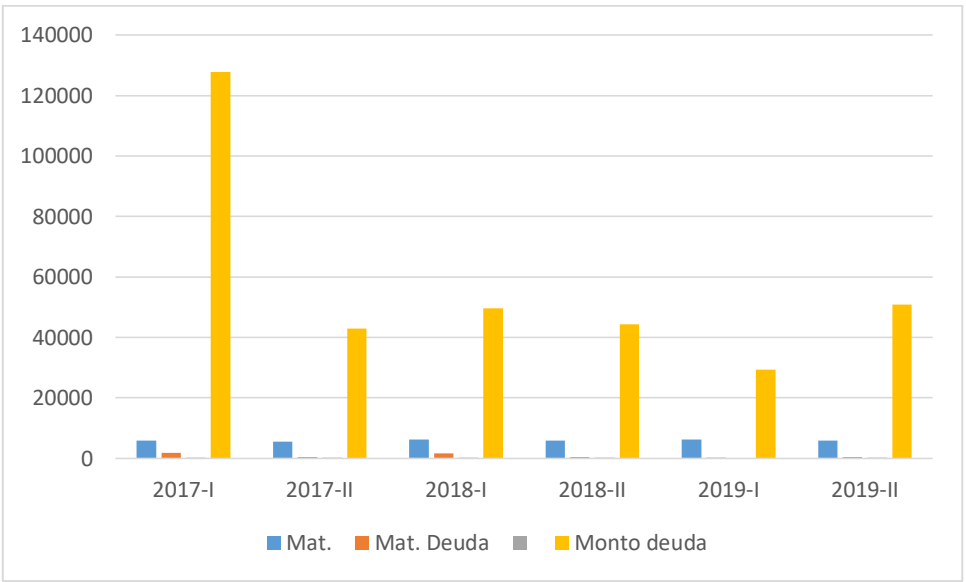
	2014	2015	2016	2017	2018	2019
Ciclo Impar	4739	5059	5583	5813	6133	6225
Ciclo Par	4535	4809	5222	5537	5760	5876

ILUSTRACIÓN 92: % DE MATRICULADOS - VIRTUAL VS PRESENCIAL



	2017-I	2017-II	2018-I	2018-II	2019-I	2019-II
Web	66.97%	86.55%	88.54%	92.36%	93.41%	93.81%
Presencial	33.03%	13.45%	11.46%	7.64%	6.59%	6.19%

ILUSTRACIÓN 93: REPORTE DE MATRICULAS Y MONTOS DE DEUDA



CAPITULO V

RESULTADOS Y DISCUSIÓN

5.1. INTERPRETACIÓN DE RESULTADOS

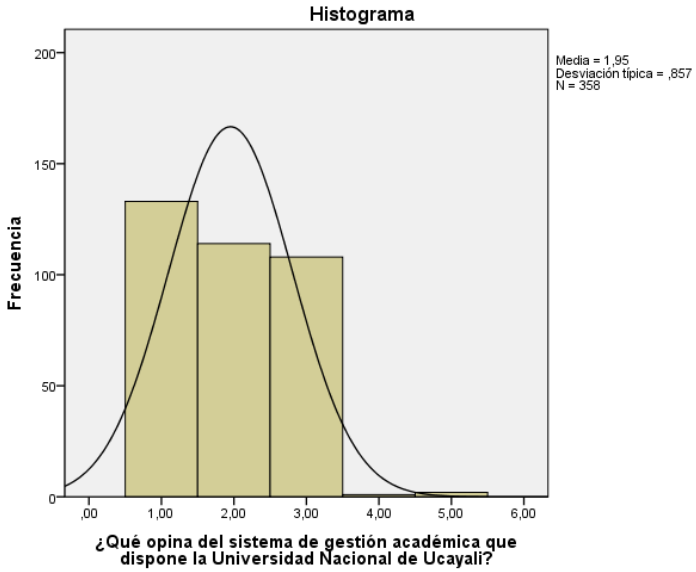
5.1.1. PRE TEST

PREG_01: ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?

TABLA 63. PREG_01: ¿QUÉ OPINA DEL SISTEMA DE GESTIÓN ACADÉMICA QUE DISPONE LA UNIVERSIDAD NACIONAL DE UCAYALI?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	133	37,2	37,2
	Malo	114	31,8	69,0
	Regular	108	30,2	99,2
	Bueno	1	,3	99,4
	Muy bueno	2	,6	100,0
	Total	358	100,0	

ILUSTRACIÓN 94. PREG_01: ¿QUÉ OPINA DEL SISTEMA DE GESTIÓN ACADÉMICA QUE DISPONE LA UNIVERSIDAD NACIONAL DE UCAYALI?



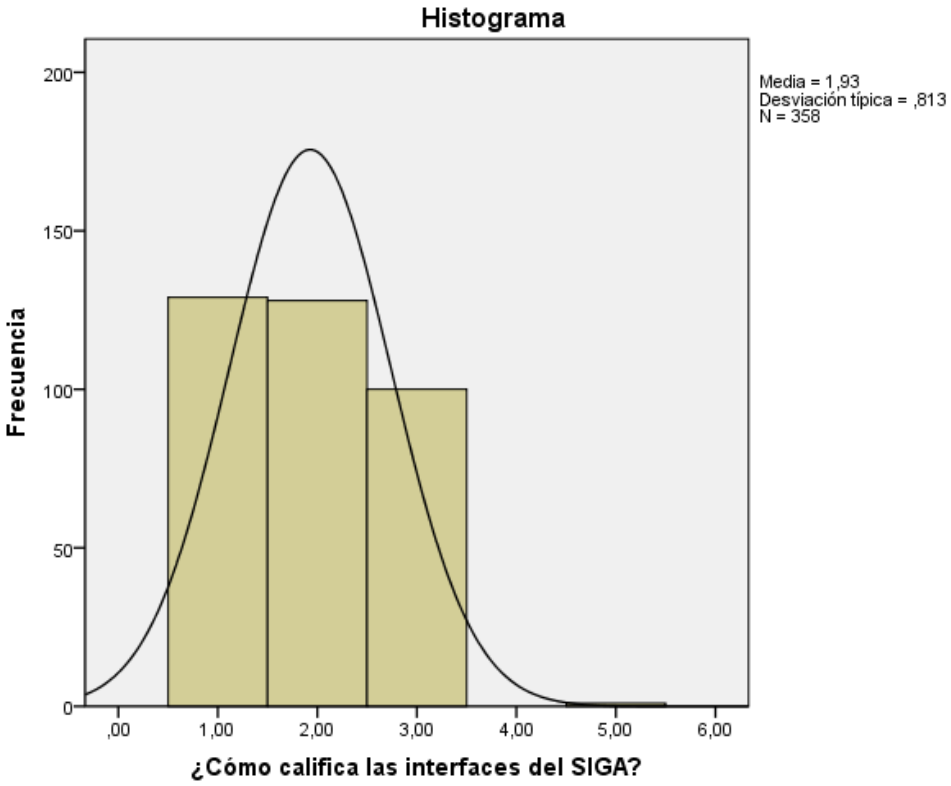
LECTURA. – En el Pretest, del total de encuestados (358), a la pregunta ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?, el 37,2% manifestaron Muy malo, el 31,8% Malo, el 30,2% Regular y un menor número bueno y muy bueno.

PREG_02: ¿Cómo califica las interfaces del SIGA?

TABLA 64: PREG_02: ¿CÓMO CALIFICA LAS INTERFACES DEL SIGA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	129	36,0	36,0
	Malo	128	35,8	71,8
	Regular	100	27,9	99,7
	Muy bueno	1	,3	100,0
	Total	358	100,0	

ILUSTRACIÓN 95: PREG_02: ¿CÓMO CALIFICA LAS INTERFACES DEL SIGA?



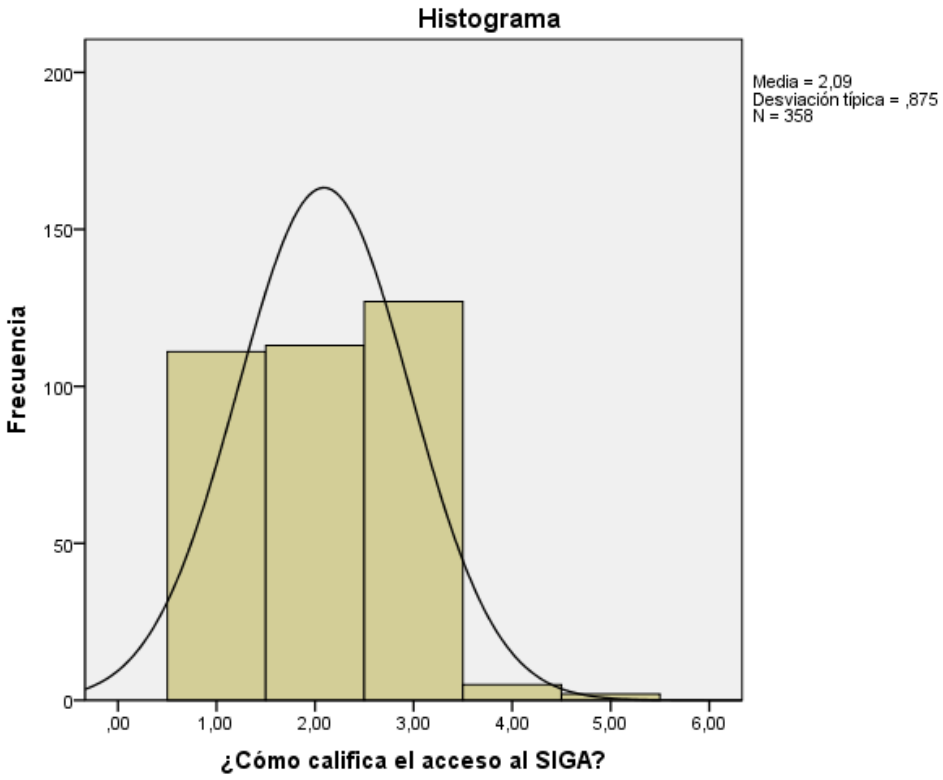
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Cómo califica las interfaces del SIGA?, el 36% manifestaron Muy malo, el 35,8% Malo, el 27,9% Regular y un menor número bueno y muy bueno.

PREG_03: ¿Cómo califica el acceso al SIGA Académico?

TABLA 65: PREG_03: ¿CÓMO CALIFICA EL ACCESO AL SIGA ACADÉMICO?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	111	31,0	31,0
	Malo	113	31,6	62,6
	Regular	127	35,5	98,0
	Bueno	5	1,4	99,4
	Muy bueno	2	,6	100,0
	Total	358	100,0	

ILUSTRACIÓN 96: PREG_03: ¿CÓMO CALIFICA EL ACCESO AL SIGA?



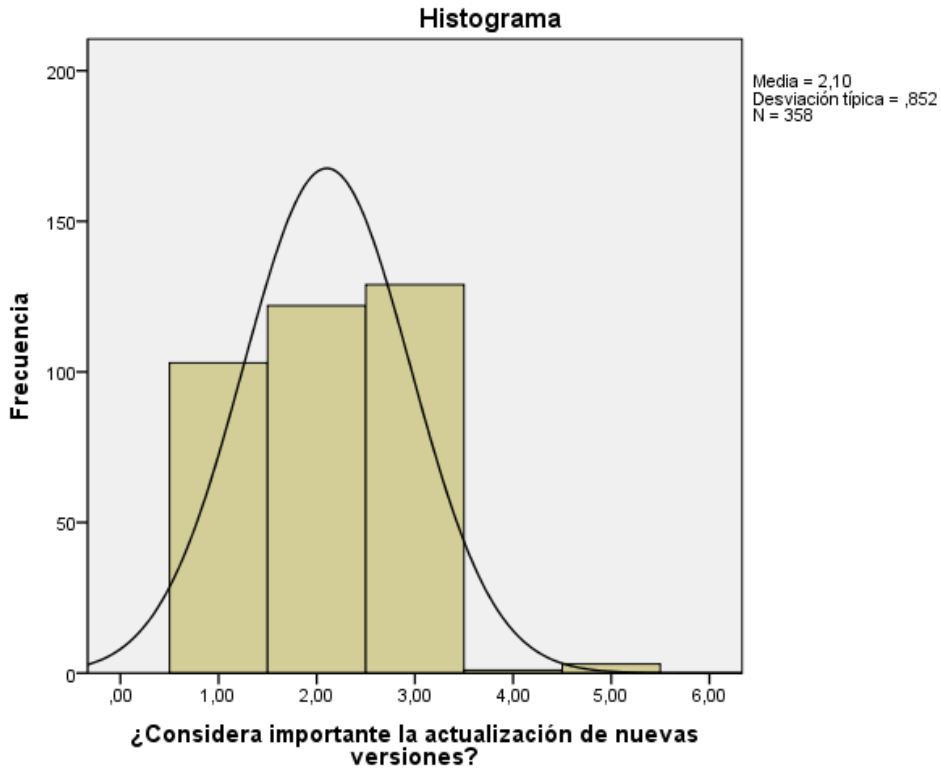
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Cómo califica el acceso al SIGA?, el 35,5% manifestaron Regular, el 31,6% Malo, el 31% Muy Malo y un menor número bueno y muy bueno.

PREG_04: ¿Considera importante la actualización de nuevas versiones?

TABLA 66: PREG_04: ¿CONSIDERA IMPORTANTE LA ACTUALIZACIÓN DE NUEVAS VERSIONES?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	103	28,8	28,8
	Malo	122	34,1	62,8
	Regular	129	36,0	98,9
	Bueno	1	,3	99,2
	Muy bueno	3	,8	100,0
	Total	358	100,0	

ILUSTRACIÓN 97: PREG_04: ¿CONSIDERA IMPORTANTE LA ACTUALIZACIÓN DE NUEVAS VERSIONES?



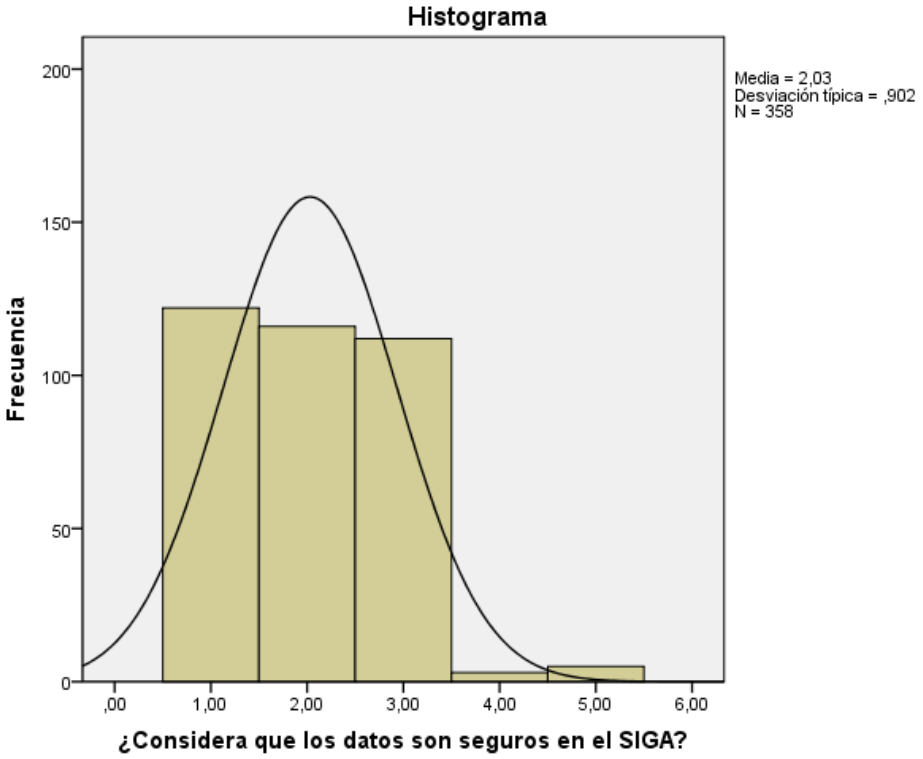
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Considera importante la actualización de nuevas versiones?, el 36% manifestaron Regular, el 34,1% Malo, el 28,8% Muy Malo y un menor número bueno y muy bueno.

PREG_05: ¿Considera que los datos son seguros en el SIGA?

TABLA 67: PREG_05: ¿CONSIDERA QUE LOS DATOS SON SEGUROS EN EL SIGA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	122	34,1	34,1
	Malo	116	32,4	66,5
	Regular	112	31,3	97,8
	Bueno	3	,8	98,6
	Muy bueno	5	1,4	100,0
	Total	358	100,0	

ILUSTRACIÓN 98: PREG_05: ¿CONSIDERA QUE LOS DATOS SON SEGUROS EN EL SIGA?



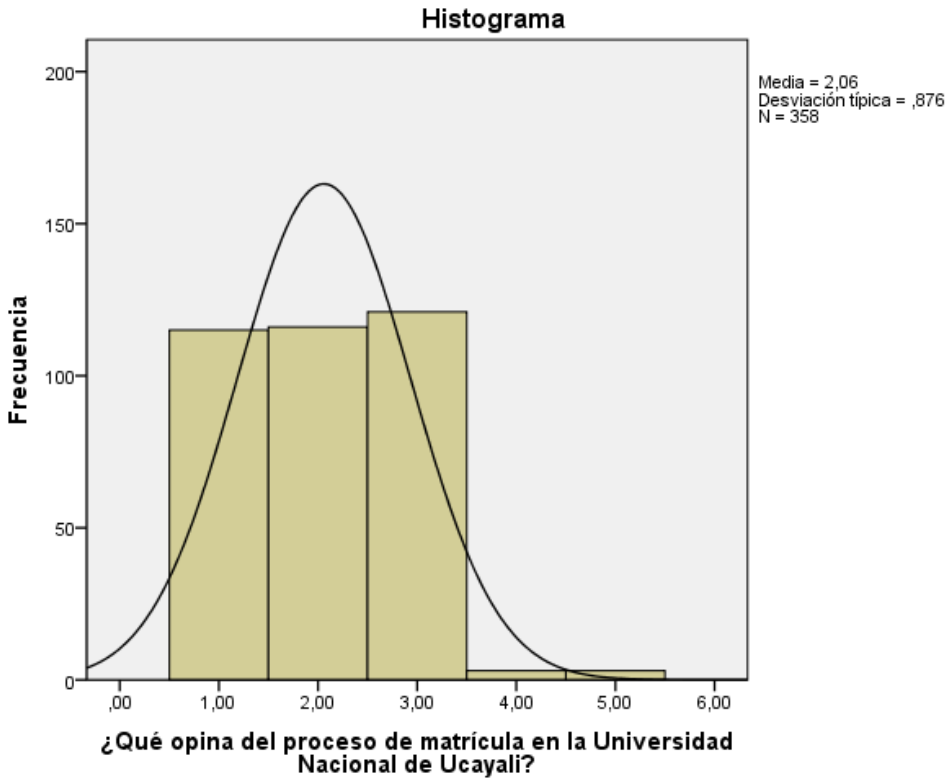
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Considera que los datos son seguros en el SIGA?, el 34,1% manifestaron Muy malo, el 32,4% Malo, el 31,3% Regular y un menor número bueno y muy bueno.

PREG_06: ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali?

TABLA 68: PREG_06: ¿QUÉ OPINA DEL PROCESO DE MATRÍCULA EN LA UNIVERSIDAD NACIONAL DE UCAYALI?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	115	32,1	32,1
	Malo	116	32,4	64,5
	Regular	121	33,8	98,3
	Bueno	3	,8	99,2
	Muy bueno	3	,8	100,0
	Total	358	100,0	

ILUSTRACIÓN 99: PREG_06: ¿QUÉ OPINA DEL PROCESO DE MATRÍCULA EN LA UNIVERSIDAD NACIONAL DE UCAYALI?



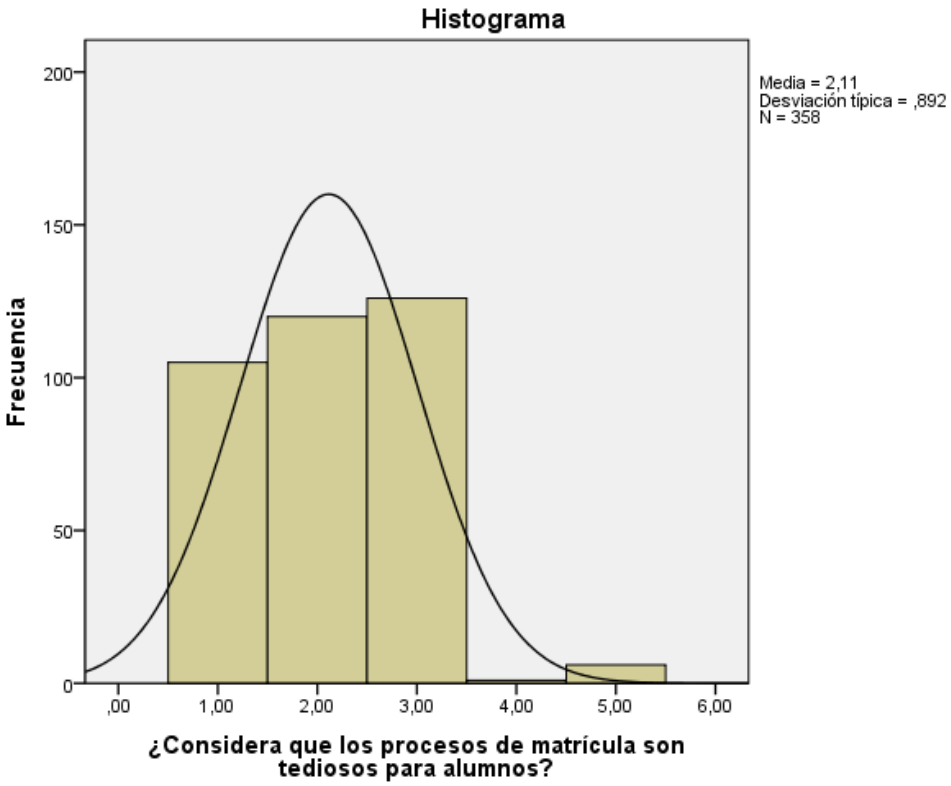
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali?, el 33,8% manifestaron Regular, el 32,4% Malo, el 32,1% Muy Malo y un menor número bueno y muy bueno.

PREG_07: ¿Considera que los procesos de matrícula son tediosos para alumnos?

TABLA 69: PREG_07: ¿CONSIDERA QUE LOS PROCESOS DE MATRÍCULA SON TEDIOSOS PARA ALUMNOS?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	105	29,3	29,3
	Malo	120	33,5	62,8
	Regular	126	35,2	98,0
	Bueno	1	,3	98,3
	Muy bueno	6	1,7	100,0
	Total	358	100,0	

ILUSTRACIÓN 100: PREG_07: ¿CONSIDERA QUE LOS PROCESOS DE MATRÍCULA SON TEDIOSOS PARA ALUMNOS?



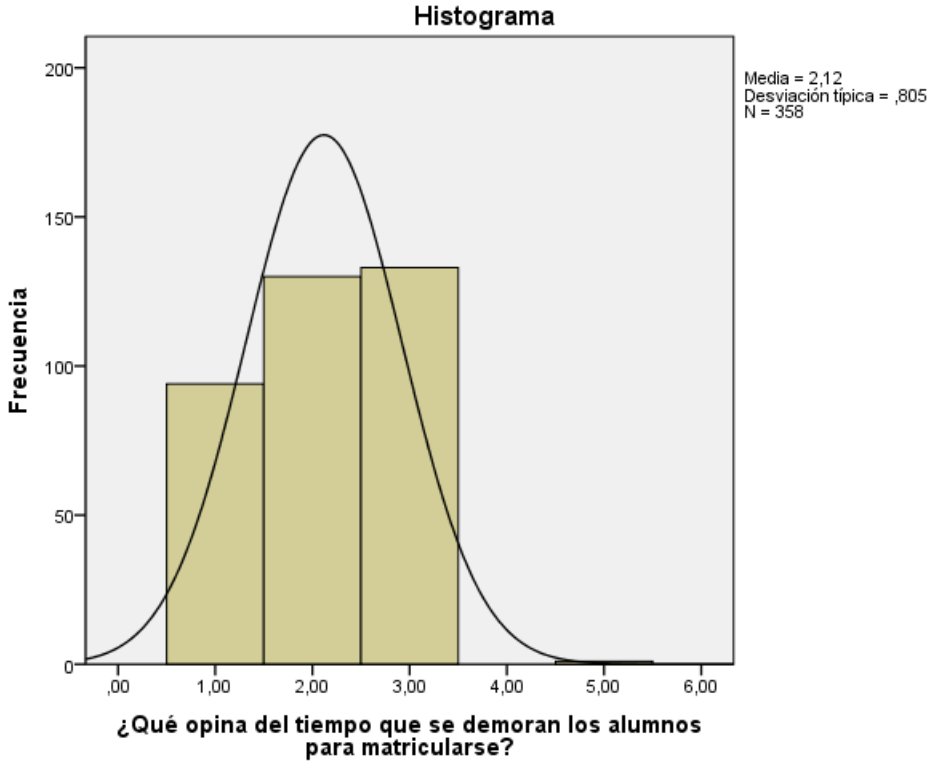
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Considera que los procesos de matrícula son tediosos para alumnos?, el 35,2% manifestaron Regular, el 33,5% Malo, el 29,3% Muy Malo y un menor número bueno y muy bueno.

PREG_08: ¿Qué opina del tiempo que se demoran los alumnos para matricularse?

TABLA 70: PREG_08: ¿QUÉ OPINA DEL TIEMPO QUE SE DEMORAN LOS ALUMNOS PARA MATRICULARSE?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	94	26,3	26,3
	Malo	130	36,3	62,6
	Regular	133	37,2	99,7
	Muy bueno	1	,3	100,0
	Total	358	100,0	

ILUSTRACIÓN 101: PREG_08: ¿QUÉ OPINA DEL TIEMPO QUE SE DEMORAN LOS ALUMNOS PARA MATRICULARSE?



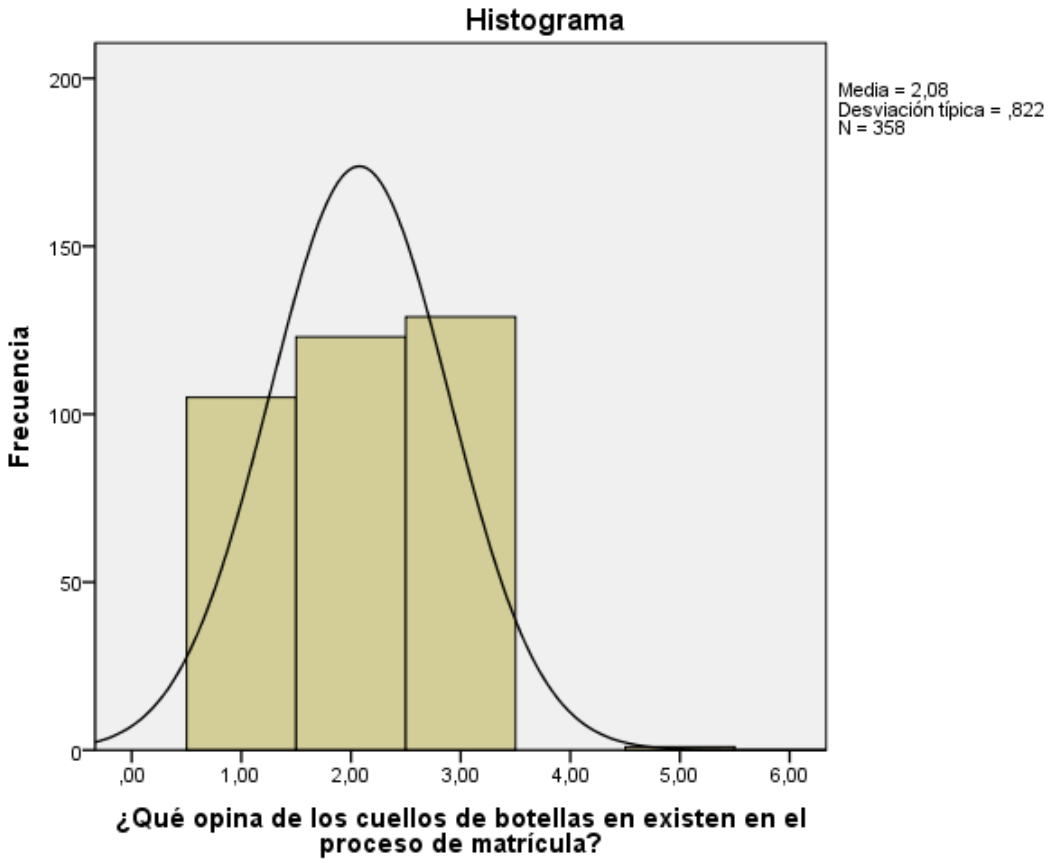
LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Qué opina del tiempo que se demoran los alumnos para matricularse? el 37,2% manifestaron Regular, el 36,3% Malo, el 26,3% Muy Malo y un menor número bueno y muy bueno.

PREG_09: ¿Qué opina de los cuellos de botellas que existen en el proceso de matrícula?

TABLA 71: PREG_09: ¿QUÉ OPINA DE LOS CUELLOS DE BOTELLAS EN EXISTEN EN EL PROCESO DE MATRÍCULA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	105	29,3	29,3
	Malo	123	34,4	63,7
	Regular	129	36,0	99,7
	Muy bueno	1	,3	100,0
	Total	358	100,0	

ILUSTRACIÓN 102: PREG_09: ¿QUÉ OPINA DE LOS CUELLOS DE BOTELLAS EN EXISTEN EN EL PROCESO DE MATRÍCULA?



LECTURA. - En el Pretest, del total de encuestados (358), a la pregunta ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula? el 36,0% manifestaron Regular, el 34,4% Malo, el 29,3% Muy Malo y un menor número bueno y muy bueno.

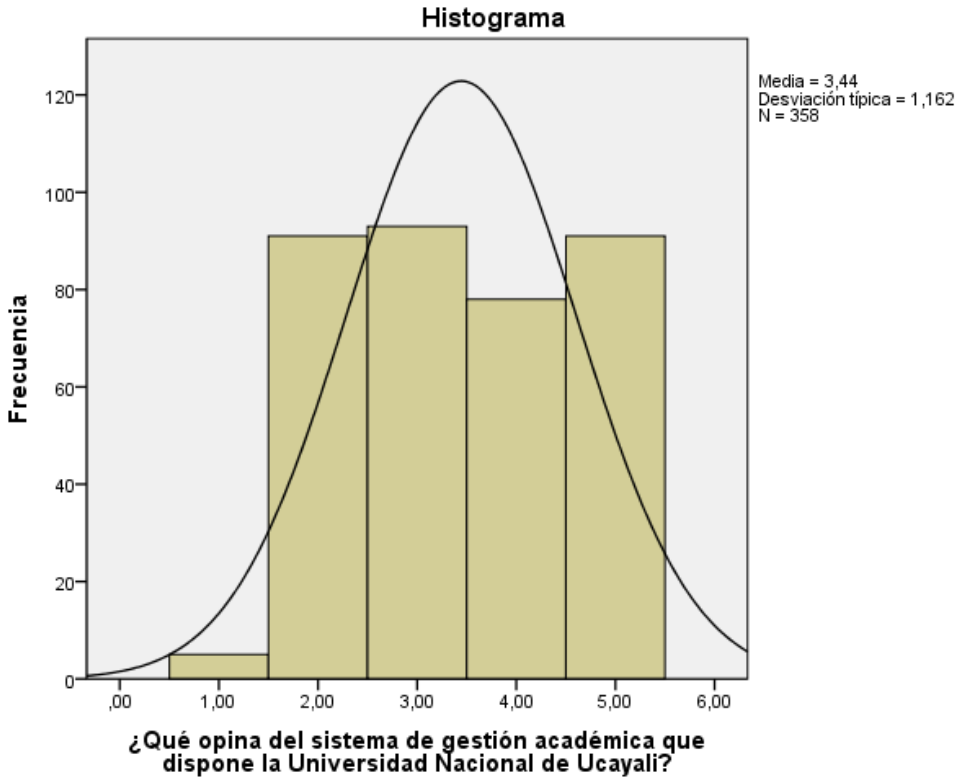
5.1.2. POS TEST

PREG_01: ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?

TABLA 72: PREG_01: ¿QUÉ OPINA DEL SISTEMA DE GESTIÓN ACADÉMICA QUE DISPONE LA UNIVERSIDAD NACIONAL DE UCAYALI?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	5	1,4	1,4
	Malo	91	25,4	26,8
	Regular	93	26,0	52,8
	Bueno	78	21,8	74,6
	Muy bueno	91	25,4	100,0
	Total	358	100,0	

ILUSTRACIÓN 103: PREG_01: ¿QUÉ OPINA DEL SISTEMA DE GESTIÓN ACADÉMICA QUE DISPONE LA UNIVERSIDAD NACIONAL DE UCAYALI?



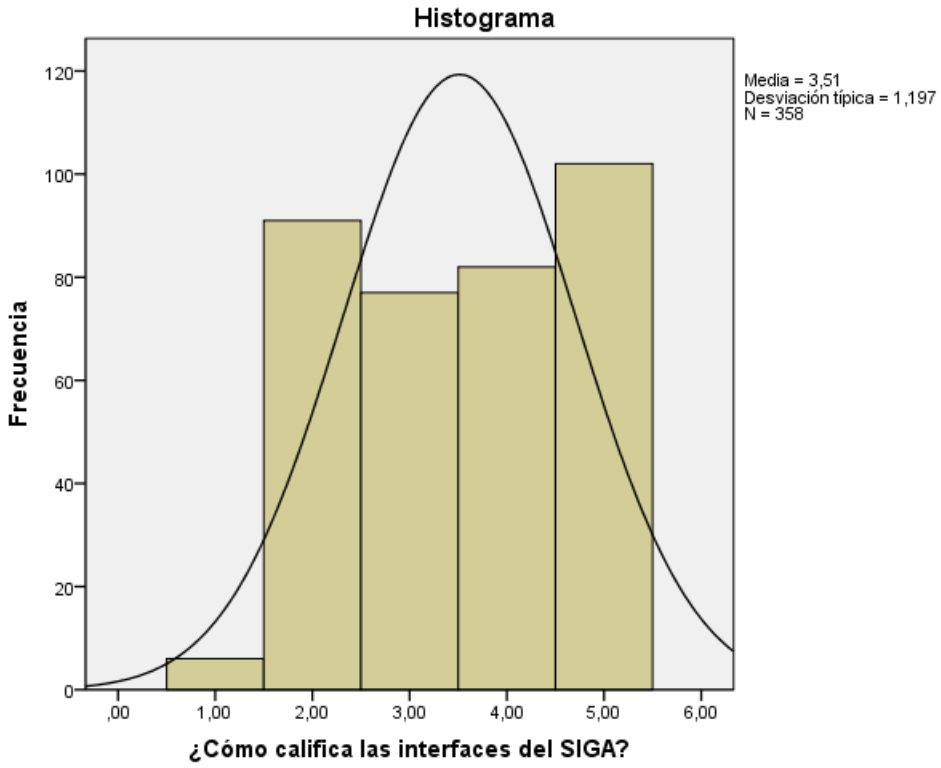
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali? el 26,0% manifestaron Regular, el 25,4% Malo y otro 25,4% Muy bueno, el 21,8% Bueno.

PREG_02: ¿Cómo califica las interfaces del SIGA?

TABLA 73: PREG_02: ¿CÓMO CALIFICA LAS INTERFACES DEL SIGA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	6	1,7	1,7
	Malo	91	25,4	27,1
	Regular	77	21,5	48,6
	Bueno	82	22,9	71,5
	Muy bueno	102	28,5	100,0
	Total	358	100,0	

ILUSTRACIÓN 104: PREG_02: ¿CÓMO CALIFICA LAS INTERFACES DEL SIGA?



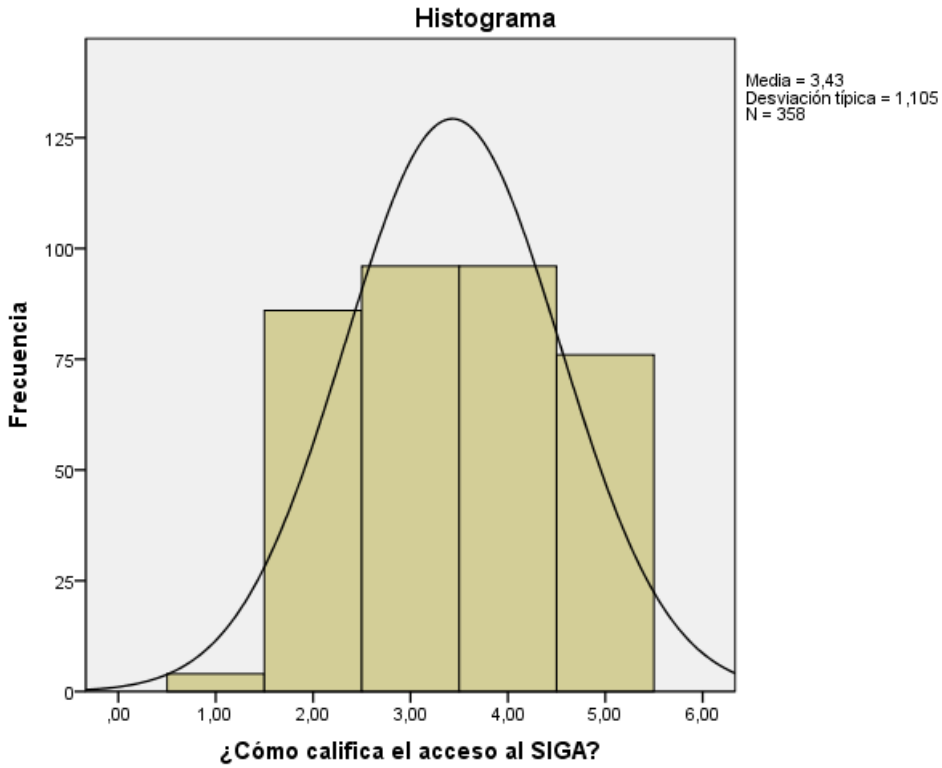
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Cómo califica las interfaces del SIGA? el 28,5% manifestaron Muy bueno, el 25,4% Malo, 22,9% Bueno, 21,5% Regular y otro muy pequeño % Muy malo.

PREG_03: ¿Cómo califica el acceso al SIGA?

TABLA 74: PREG_03: ¿CÓMO CALIFICA EL ACCESO AL SIGA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	4	1,1	1,1
	Malo	86	24,0	25,1
	Regular	96	26,8	52,0
	Bueno	96	26,8	78,8
	Muy bueno	76	21,2	100,0
	Total	358	100,0	

ILUSTRACIÓN 105: PREG_03: ¿CÓMO CALIFICA EL ACCESO AL SIGA?



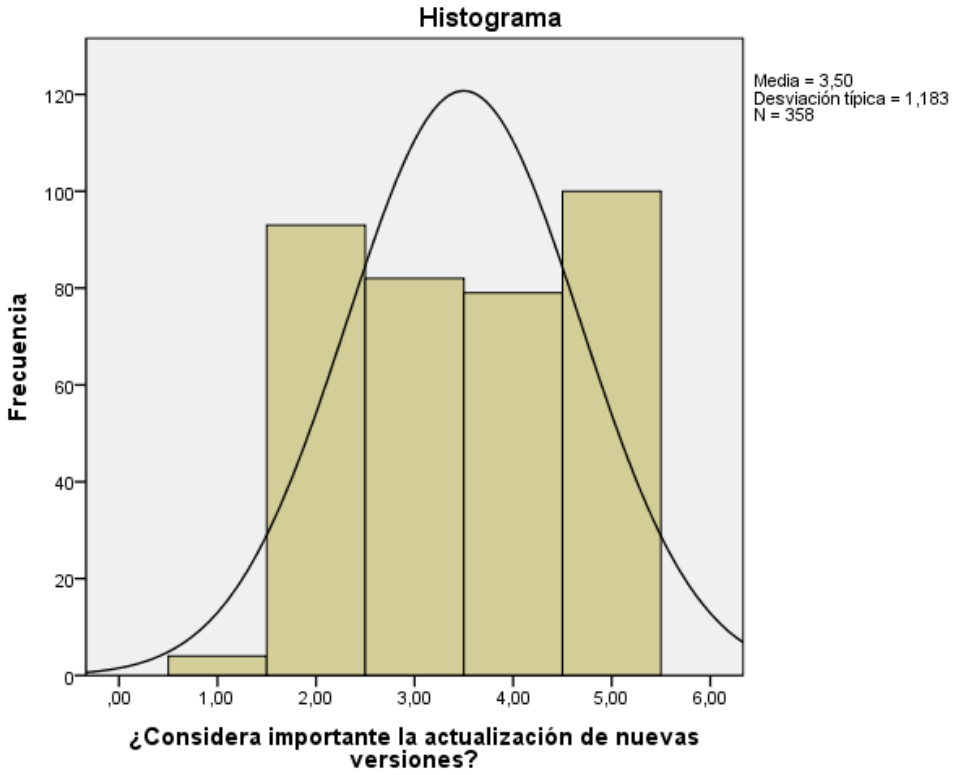
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Cómo califica el acceso al SIGA? el 26,8% manifestaron Bueno, 26,8% Regular, 24% Malo, 21,2% Muy bueno y otro muy pequeño % Muy malo.

PREG_04: ¿Considera importante la actualización de nuevas versiones?

TABLA 75: PREG_04: ¿CONSIDERA IMPORTANTE LA ACTUALIZACIÓN DE NUEVAS VERSIONES?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	4	1,1	1,1
	Malo	93	26,0	27,1
	Regular	82	22,9	50,0
	Bueno	79	22,1	72,1
	Muy bueno	100	27,9	100,0
	Total	358	100,0	

ILUSTRACIÓN 106: PREG_04: ¿CONSIDERA IMPORTANTE LA ACTUALIZACIÓN DE NUEVAS VERSIONES?



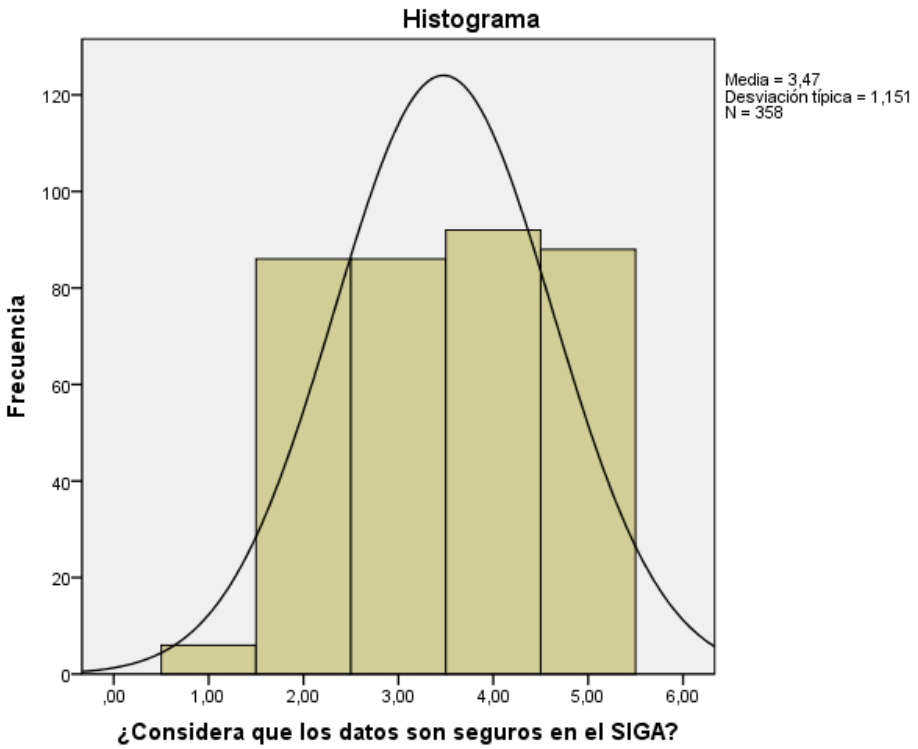
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Considera importante la actualización de nuevas versiones? el 27,9% manifestaron Muy bueno, 26% Malo, 22,9% Regular, 22,1% Bueno y otro muy pequeño % Muy malo.

PREG_05: ¿Considera que los datos son seguros en el SIGA?

TABLA 76: PREG_05: ¿CONSIDERA QUE LOS DATOS SON SEGUROS EN EL SIGA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	6	1,7	1,7
	Malo	86	24,0	25,7
	Regular	86	24,0	49,7
	Bueno	92	25,7	75,4
	Muy bueno	88	24,6	100,0
	Total	358	100,0	

ILUSTRACIÓN 107: PREG_05: ¿CONSIDERA QUE LOS DATOS SON SEGUROS EN EL SIGA?



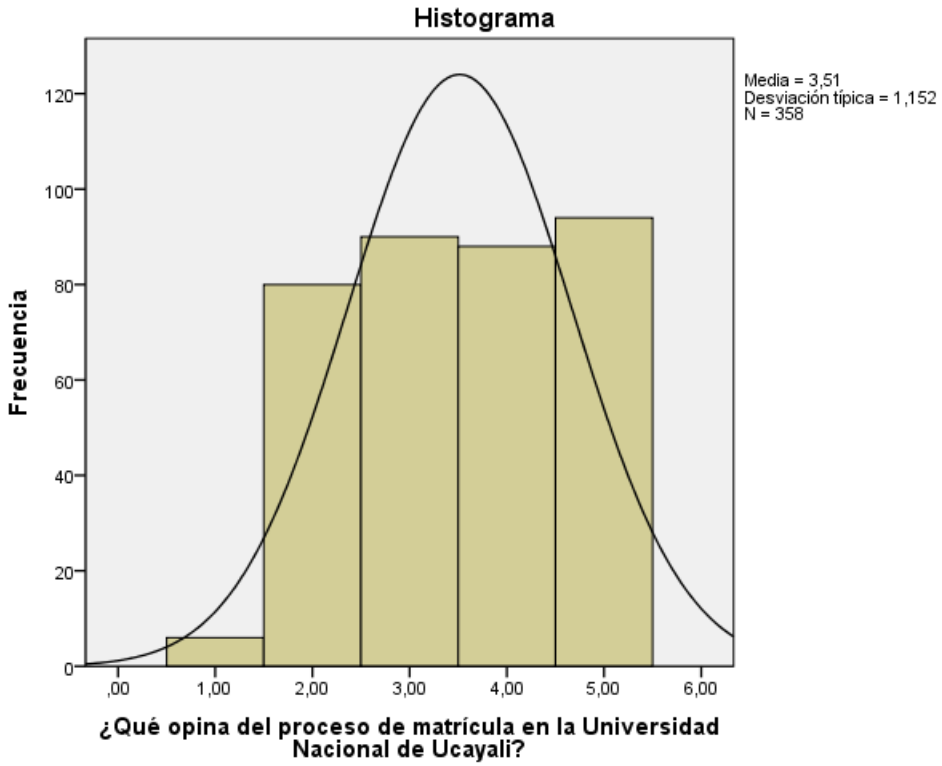
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Considera que los datos son seguros en el SIGA? el 25,7% manifestaron Bueno, 24,6% Muy bueno, 24% Regular, 24,0% Malo y otro muy pequeño % Muy malo.

PREG_06: ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali?

TABLA 77: PREG_06: ¿QUÉ OPINA DEL PROCESO DE MATRÍCULA EN LA UNIVERSIDAD NACIONAL DE UCAYALI?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	6	1,7	1,7
	Malo	80	22,3	24,0
	Regular	90	25,1	49,2
	Bueno	88	24,6	73,7
	Muy bueno	94	26,3	100,0
	Total	358	100,0	

ILUSTRACIÓN 108: PREG_06: ¿QUÉ OPINA DEL PROCESO DE MATRÍCULA EN LA UNIVERSIDAD NACIONAL DE UCAYALI?



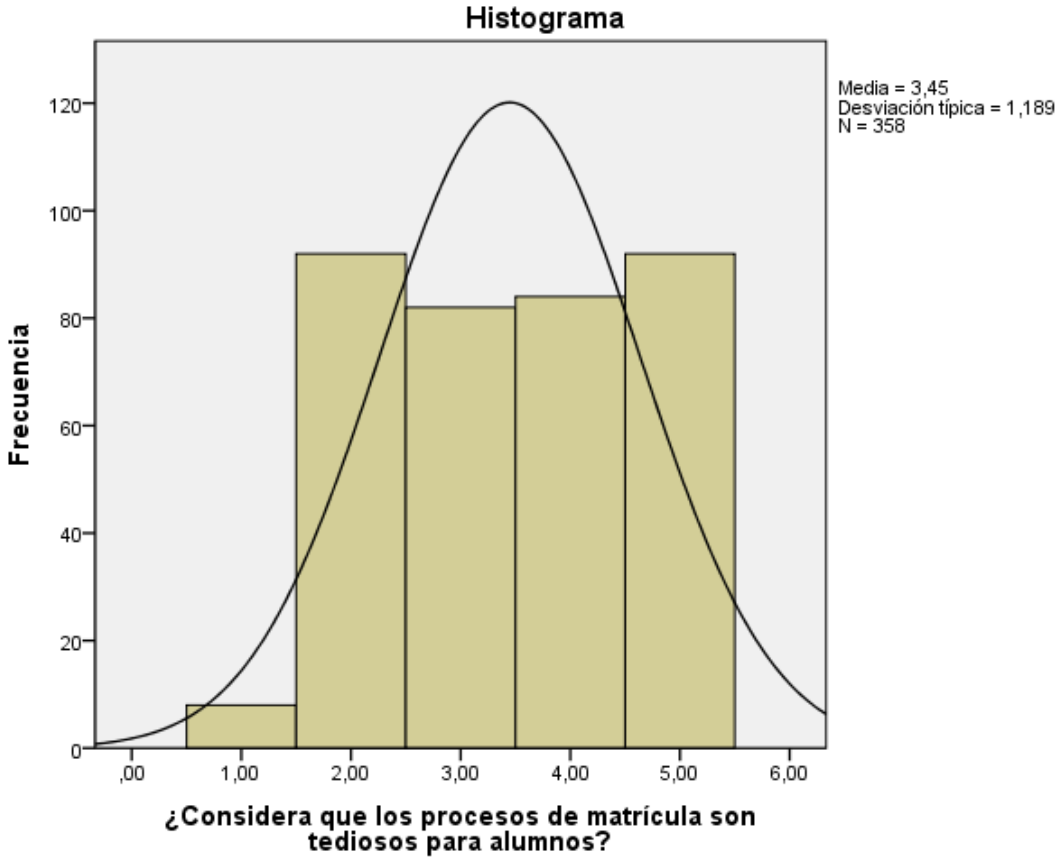
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali? el 26,3% manifestaron Muy bueno, 25,1% Regular, 24,6% Bueno, 22,3% Malo y otro muy pequeño % Muy malo.

PREG_07: ¿Considera que los procesos de matrícula son tediosos para alumnos?

TABLA 78: PREG_07: ¿CONSIDERA QUE LOS PROCESOS DE MATRÍCULA SON TEDIOSOS PARA ALUMNOS?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	8	2,2	2,2
	Malo	92	25,7	27,9
	Regular	82	22,9	50,8
	Bueno	84	23,5	74,3
	Muy bueno	92	25,7	100,0
	Total	358	100,0	

ILUSTRACIÓN 109: PREG_07: ¿CONSIDERA QUE LOS PROCESOS DE MATRÍCULA SON TEDIOSOS PARA ALUMNOS?



LECTURA. - - En el Postest, del total de encuestados (358), a la pregunta ¿Considera que los procesos de matrícula son tediosos para alumnos? el 25,7% manifestaron Muy bueno, 25,7% Malo, 23,5% Bueno, 22,9% Regular y otro muy pequeño % Muy malo.

PREG_08: ¿Qué opina del tiempo que se demoran los alumnos para matricularse?

TABLA 79: PREG_08: ¿QUÉ OPINA DEL TIEMPO QUE SE DEMORAN LOS ALUMNOS PARA MATRICULARSE?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	3	,8	,8
	Malo	103	28,8	29,6
	Regular	85	23,7	53,4
	Bueno	85	23,7	77,1
	Muy bueno	82	22,9	100,0
	Total	358	100,0	

ILUSTRACIÓN 110: PREG_08: ¿QUÉ OPINA DEL TIEMPO QUE SE DEMORAN LOS ALUMNOS PARA MATRICULARSE?



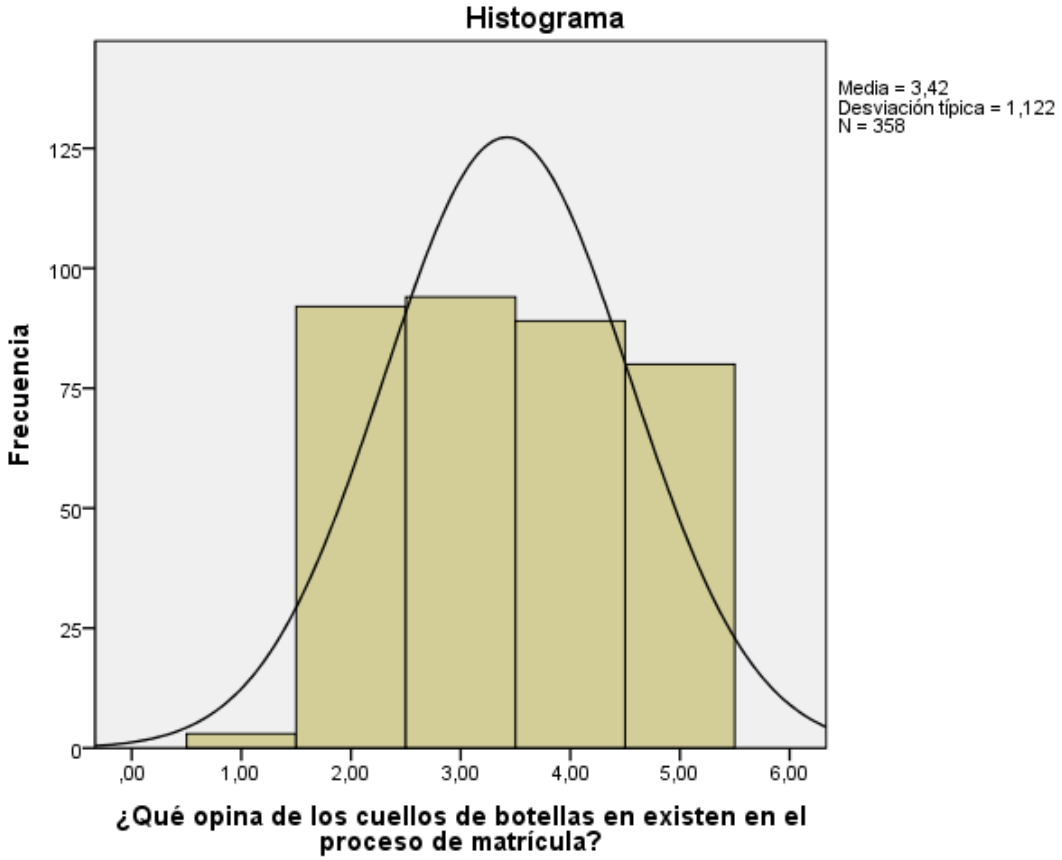
LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Qué opina del tiempo que se demoran los alumnos para matricularse? el 28,8% manifestaron Malo, 23,7% Regular, 23,7% Bueno, 22,9% Muy Bueno y otro muy pequeño % Muy malo.

PREG_09: ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula?

TABLA 80: PREG_09: ¿QUÉ OPINA DE LOS CUELLOS DE BOTELLAS EN EXISTEN EN EL PROCESO DE MATRÍCULA?

		Frecuencia	Porcentaje	Porcentaje acumulado
Válidos	Muy malo	3	,8	,8
	Malo	92	25,7	26,5
	Regular	94	26,3	52,8
	Bueno	89	24,9	77,7
	Muy bueno	80	22,3	100,0
	Total	358	100,0	

ILUSTRACIÓN 111: PREG_09: ¿QUÉ OPINA DE LOS CUELLOS DE BOTELLAS EN EXISTEN EN EL PROCESO DE MATRÍCULA?



LECTURA. - En el Postest, del total de encuestados (358), a la pregunta ¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula? el 26,3% manifestaron Regular, 25,7% Malo, 24,9% Bueno, 22,3% Muy Bueno y otro muy pequeño % Muy malo.

5.2. PRUEBA DE HIPÓTESIS

5.2.1. Paso 1: Formulación de la hipótesis

$\boldsymbol{H_0}$ = “El Sistema de Gestión Académica no mejora el proceso de matrícula de la Universidad Nacional de Ucayali: 2019”

$\boldsymbol{H_1}$ = “El Sistema de Gestión Académica mejora el proceso de matrícula de la Universidad Nacional de Ucayali: 2019”

Donde:

H_0 : Hipótesis nula.

H_1 : Hipótesis alterna.

5.2.2. Paso 2: Especificar el nivel de significancia

- Se utilizó 5% del nivel de significancia y el 95% del nivel de confiabilidad.

5.2.3. Paso 3: Selección de estadística a docimar

$\boldsymbol{n}$ **: Muestra**

Si: $n > 30 \rightarrow t_c = \frac{|\mu_1|}{\frac{\sigma_1}{\sqrt{n_1}}}$

Donde:

$t_c =$ t calculada

μ_1= Media de la muestra

σ_1= Desviación estándar

5.2.4. Paso 4: Establecer criterios de decisión

Para aceptar la hipótesis general de la investigación, el nivel de significancia (∝) que representa el 5%, debe ser mayor al Sig. (Bilateral), caso contrario se rechaza la hipótesis nula.

$$\propto \geq \text{Sig. (Bilateral)} \quad => \quad \text{R}\,(H_0) \quad \& \quad A(H_a)$$

5.2.5. Paso 5: Realización de cálculos

ILUSTRACIÓN 112: COMPARACIÓN ENTRE LOS RESULTADOS DE PRETEST Y POSTEST POR PREGUNTA



TABLA 81: ESTADÍSTICOS PARA UNA MUESTRA

	N	Media	Desviación típ.	Error típ. de la media
PRETEST	9	2,0522	,06924	,02308
POSTEST	9	3,4578	,04265	,01422

TABLA 82: PRUEBA PARA UNA MUESTRA

	Valor de prueba = 0					
					95% Intervalo de confianza para la diferencia	
	t	gl	Sig. (bilateral)	Diferencia de medias	Inferior	Superior
PRETEST	88,915	8	,000	2,05222	1,9990	2,1054
POSTEST	243,192	8	,000	3,45778	3,4250	3,4906

ILUSTRACIÓN 113: GRAFICO Q-Q NORMAL DE PRETEST

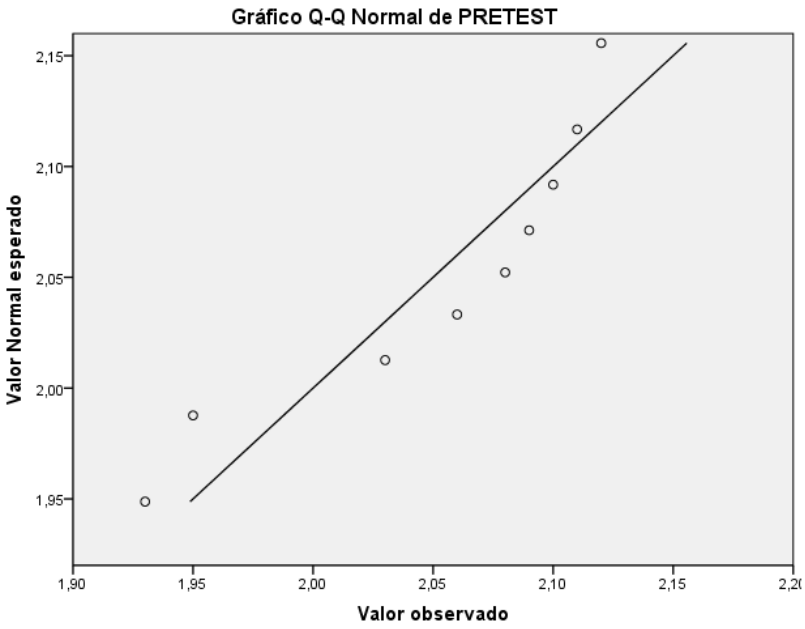
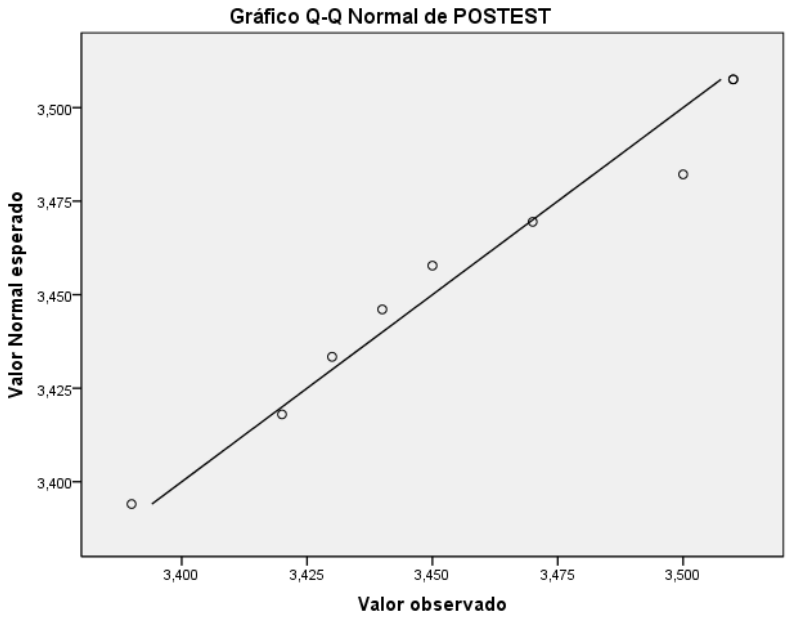


ILUSTRACIÓN 114: GRAFICO Q-Q NORMAL DE POSTEST



5.2.6. Paso 6: Decisión estadística

Como $0.05 \geq 0.00$ se rechaza la Hipótesis nula y se acepta la hipótesis alterna: "***El Sistema de Gestión Académica mejora el proceso de matrícula de la Universidad Nacional de Ucayali: 2019***"

CONCLUSIONES

1) La actividad principal que evita la pérdida de información de los estudiantes, es la seguridad de los datos, en la Universidad Nacional de Ucayali.
2) Mediante la implementación del módulo de matrícula se ha logrado reducir el tiempo de atención al estudiante en el proceso de matrícula con la implementación del módulo de matrícula web en la Universidad Nacional de Ucayali.
3) Las acciones para reducir el consumo de recursos con el sistema de matrícula web en la Universidad Nacional de Ucayali, estuvieron establecidas en tres aspectos importantes: a) La integración de SPRINT integra historias de usuario 1, 2, 3, 4, 5, 6, 7 correspondiente al SPRINT 1, y 8, 9, 10, 11 correspondiente del SPRINT 2, esta integración proporciona la información e inicialización del sistema para el módulo de seguridad y permite la asignación de perfiles a usuarios. b) La integración de SPRINT integra historias de usuario 1, 2, 3, 4, 5, 6, 7 correspondiente al SPRINT 1, historias de usuario 8, 9, 10, 11 correspondiente del SPRINT 2 e historias de usuario 12, 13, 14, 15, 16, 17, 18 y 19 correspondiente del SPRINT 3 esta integración proporciona la información e inicialización del sistema para el módulo de planes y permite la asignación de cursos a los planes. Y c) La integración de SPRINT integra historias de usuario 1, 2, 3, 4, 5, 6, 7 correspondiente al SPRINT 1, historias de usuario 12, 13, 14, 15, 16, 17, 18 y 19 correspondiente del SPRINT 3 e historias de usuario 20 y 21 correspondiente del SPRINT 4, esta integración proporciona la

información e inicialización del sistema para el módulo de matrícula y permite la asignación de cursos a los estudiantes.

4) Los mecanismos para obtener información relevante de los estudiantes en el proceso de matrícula en la Universidad Nacional de Ucayali, como: a) Registro de alumnos matriculados – data histórica, y b) % de Matriculados - Virtual vs Presencial. Ayudan a la alta dirección de la Universidad a tomar decisiones con mayor certeza posible.

RECOMENDACIONES

1) El responsable de informática, debe proceder a realizar mantenimientos periódicos al Sistema de Gestión Académica, simplificando los procesos de matrícula de la Universidad Nacional de Ucayali.
2) El responsable de informática deberá proceder a tomar backup de los datos a fin de evitar la pérdida de información de los estudiantes en la Universidad Nacional de Ucayali.
3) La implementación y mantenimiento del Sistema de Gestión Académica, permitirá reducir el consumo de recursos con el sistema de matrícula web en la Universidad Nacional de Ucayali
4) El responsable de informática, deberá entrevistar a cada uno de los usuarios en medida que van implementándose el módulo de matrícula web en la Universidad Nacional de Ucayali, periódicamente.

REFERENCIAS BIBLIOGRAFICAS

Baez, S. (20 de Junio de 2013). *Fraktal Web*. Obtenido de http://fraktalweb.com/blog/sistemas-web-para-que-sirven/

Cohen, Daniel (2009), *Tecnologías de la Información en los Negocios, 5ta edición.* México: Mc Graw Hill Educación

Enriquez. E. I. (2016). *Sistema de Información Web y su mejora en la Gestión Académica del Colegio Privado Hans Kelsen del Distrito de Florencia de Mora Trujillo. Perú: (tesis de grado) para optar el titulo profesional de Ingeniero de sistemas – Universidad Nacional de Trujillo.*

ESAN. (2009). *Las tecnologías de la información en las empresas* obtenido de https://www.esan.edu.pe/conexion/actualidad/2009/10/10/las-tecnologias-de-la-informacion-en-las-empresas/

Groussard, T. (2012). *JAVA 7: Los fundamentos del lenguaje Java.* Ediciones ENI.

Kroenke, D. (2007). *Procesamiento de Base de Datos: Fundamentos, Diseño e Implementación.* Madrid: Pearson Educación.

Marroquín, R. (2012). *Metodología de la investigación.* Lima - Perú: Universidad nacional de educación Enrique Guzmán y Valle.

Merino y Miranda (2016). *Sistema de Matricula y Consulta de notas para la Universidad Peruana Austral del Cusco (SMCN-UPAC). Perú: (tesis de grado) para obtener el título de Ingeniero de Sistemas e Informática. – Universidad Peruana Austral del Cusco.*

MINEDU. (2016). *Proceso de matricula*. Obtenido de https://www.mineducacion.gov.co/sistemasdeinformacion/1735/w3-article-297837.html?_noredirect=1

Murillo, W. (2008). La investigación científica. España: Ed Semphis

Oficina Ejecutiva de Estadística. (2018). *Boletín Estadístico.* Pucallpa - Perú: Universidad Nacional de Ucayali.

Padrón, J. (2006). *Investigar, reflexionar y actuar en la práctica docente.* Recuperado de http//padron.entretemas.com/InvAplicada/index.htm

Palacio, J. (2009). *Flexibilidad con Scrum, Principios de diseño e implantación de campos de Scrum.* España: Safe Creative

Pontificia Universidad Javeriana. (2017). *Normas y procedimientos generales para los aspectos administrativos de la matrícula de los programas de pregrado de la Pontificia Universidad Javeriana.*

Ramirez, J. A. (2017). *Implementación de un Sistema Web para mejorar el proceso de Gestión Académica en las Escuelas de la PNP. Perú: (tesis de grado) para obtener el título de ingeniero en computación y sistemas – Universidad Peruana de las Américas*

Sánchez, J. C. (2004). Metodología de la investigación científica y tecnológica. En J. C. Sánchez, *Metodología de la investigación científica y tecnológica*. Ediciones Díaz de Santos.

Villacorta, C. M. (2011). *Desarrollo e Implementación de un Sistema Web de Matrícula y Notas para ls Institución Microsystems. Perú: (Informe práctico*

de suficiencia) para optar el título profesional de Ingeniero de Sistemas e Informática – Universidad Nacional de la Amazonía Peruana.

ANEXOS

Anexo 1. MATRIZ DE CONSISTENCIA

<table>
<tr><th>FORMULACIÓN</th><th>OBJETIVOS</th><th>HIPÓTESIS</th><th>METODOLOGÍA DE LA INVESTIGACIÓN</th></tr>
<tr><td>¿De qué manera el sistema de gestión académica mejora los procesos de matrícula de la Universidad Nacional de Ucayali: 2019?</td><td>Determinar de qué manera el sistema de gestión académica mejora los procesos de matrícula de la Universidad Nacional de Ucayali: 2019.</td><td rowspan="5">El sistema de gestión académica mejora los procesos de matrícula de la Universidad Nacional de Ucayali: 2019.</td><td rowspan="5"><u>TIPO DE INVESTIGACIÓN:</u>
▪ APLICADA
<u>NIVEL DE INVESTIGACIÓN:</u>
▪ DESCRIPTIVA
<u>DISEÑO DE LA INVESTIGACIÓN:</u>
▪ PRE – EXPERIMENTAL: PRE-TEST Y POST-TEST
$G = O_1 - - - -> X - - - -> O_2$
<u>POBLACIÓN:</u>
▪ Cuenta con 5082 estudiantes matriculados en la Universidad Nacional de Ucayali en el periodo 2018.
<u>MUESTRA:</u>
▪ 357.
<u>SW DE PROC. DATOS:</u>
▪ SPSS
<table><tr><th>Fuentes</th><th>Técnicas</th></tr><tr><td>Primaria</td><td>Cuestionario de preguntas</td></tr></table></td></tr>
<tr><td>1. ¿Cómo se podrá evitar la pérdida de información de los estudiantes en la Universidad Nacional de Ucayali?</td><td>1. Identificar actividades para evitar la pérdida de información de los estudiantes en la Universidad Nacional de Ucayali.</td></tr>
<tr><td>2. ¿Cómo se reducirá el tiempo de atención al estudiante en el proceso de matrícula con la implementación del módulo de matrícula web en la Universidad Nacional de Ucayali?</td><td>2. Identificar acciones para reducir el tiempo de atención al estudiante en el proceso de matrícula con la implementación del módulo de matrícula web en la Universidad Nacional de Ucayali.</td></tr>
<tr><td>3. ¿Cómo se podrá reducir el consumo de recursos con el sistema de matrícula web en la Universidad Nacional de Ucayali?</td><td>3. Identificar acciones para reducir el consumo de recursos con el sistema de matrícula web en la Universidad Nacional de Ucayali.</td></tr>
<tr><td>4. ¿Cómo se podrá obtener información relevante de los estudiantes en el proceso de matrícula en la Universidad Nacional de Ucayali?</td><td>4. Identificar mecanismos para obtener información relevante de los estudiantes en el proceso de matrícula en la Universidad Nacional de Ucayali.</td></tr>
</table>

Anexo 2. CUADRO DE OPERACIONALIZACION DE LAS VARIABLES

VARIABLES	DEFINICION OPERACIONAL	INDICADORES	1	2	3	4	5
SISTEMA DE GESTIÓN ACADÉMICA	Elementos compuestos por el sistema de gestión académica disponible, las interfaces, el acceso del SIGA, la actualización de nuevas versiones y la seguridad de los datos.	Disponibilidad del SIGA	1	2	3	4	5
		Interfaces del SIGA	1	2	3	4	5
		Acceso al SIGA	1	2	3	4	5
		Actualización de nuevas versiones del SIGA	1	2	3	4	5
		Seguridad de los datos en el SIGA	1	2	3	4	5
PROCESOS DE MATRÍCULA	Conjunto de actividades, compuestas por el proceso de matrícula, y el acceso por los usuarios, tiempo de demora y los cuellos de botella.	Proceso de matrícula	1	2	3	4	5
		Procesos de matrícula son tediosos para alumnos	1	2	3	4	5
		Tiempo que se demoran los alumnos para matricularse	1	2	3	4	5
		Cuellos de botellas existen en el proceso de matrícula	1	2	3	4	5

CUADRO DE CALIFICACIONES CUANTITATIVO - CUALITATIVO

1	2	3	4	5
Muy malo	Malo	Regular	Bueno	Muy bueno

Anexo 3. CUESTIONARIO DE PREGUNTAS

CUESTIONARIO DE PREGUNTAS DIRIGIDO A LOS ALUMNOS MATRICULADOS DE LA UNIVERSIDAD NACIONAL DE UCAYALI

No de cuestionario: _____
Fecha: _____

El cuestionario que se presenta a continuación, recoge información de manera confidencial acerca de las variables de estudio, por lo tanto, debe marcar (X) aquella respuesta que se aproxime más a su criterio. Cuya escala es el siguiente:

1	2	3	4	5
Muy malo	**Malo**	**Regular**	**Bueno**	**Muy bueno**

PREGUNTAS ESPECÍFICAS

N°	PREGUNTAS	1	2	3	4	5
SISTEMA DE GESTIÓN ACADÉMICA						
PREG 01	¿Qué opina del sistema de gestión académica que dispone la Universidad Nacional de Ucayali?					
PREG 02	¿Cómo califica las interfaces del SIGA?					
PREG 03	¿Cómo califica el acceso al SIGA?					
PREG 04	¿Considera importante la actualización de nuevas versiones?					
PREG 05	¿Considera que los datos son seguros en el SIGA?					
PROCESOS DE MATRÍCULA						
PREG 06	¿Qué opina del proceso de matrícula en la Universidad Nacional de Ucayali?					
PREG 07	¿Considera que los procesos de matrícula son tediosos para alumnos?					
PREG 08	¿Qué opina del tiempo que se demoran los alumnos para matricularse?					
PREG 09	¿Qué opina de los cuellos de botellas en existen en el proceso de matrícula?					

Gracias por su respuesta.

Anexo 4. INSTALACIÓN DE ORACLE V11g (SERVIDOR)

MANUAL DE INSTALACION DE ORACLE 11g – SERVIDOR de base de datos

1. Insertar CD de instalación y hacer clic en el archivo ejecutable setup.exe, saldrá la siguiente ventana, Seleccionar la opción “Base de Datos Oracle 11g” y hacer clic en “Siguiente”



2. Seleccionar la opción “Instalación Básica”, Ingrese la Ruta del directorio base donde se va instalar el Oracle, ubicación del directorio raíz de la base de datos que se creará, seleccionar en “tipo de instalación” la opción “Enterprise Edition”, escribir el nombre de la Base de datos y se recomienda poner una contraseña



3. En la siguiente imagen hacer Clic en el Botón **"Siguiente"**.



Instalación de paquetes Oracle



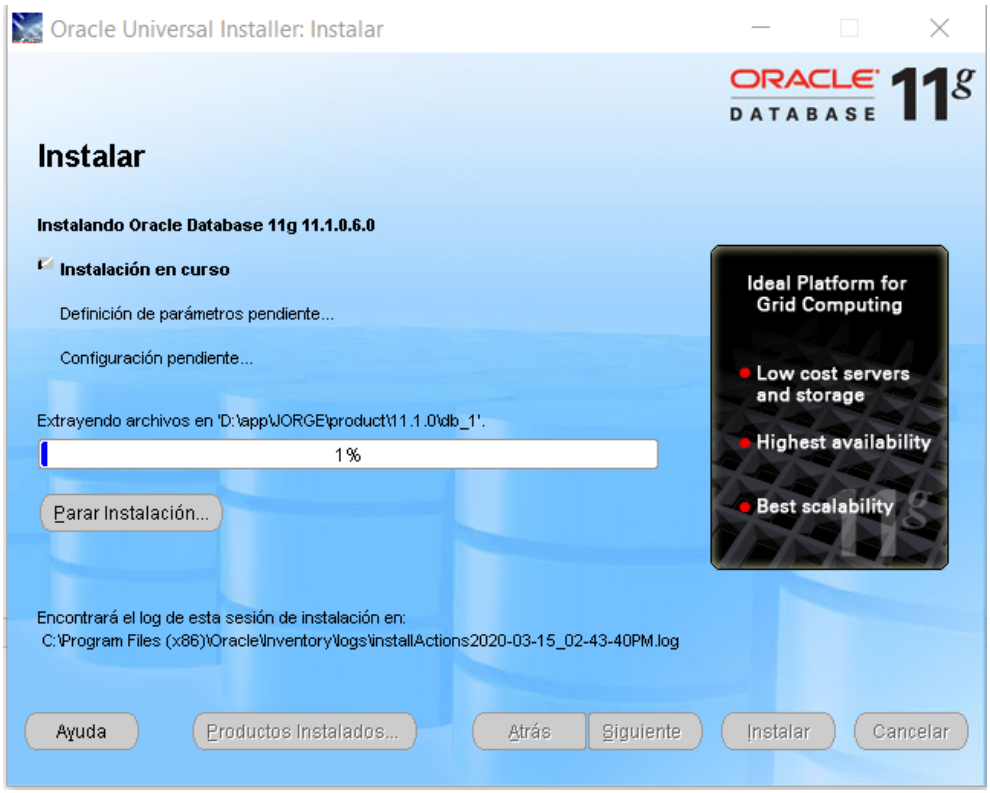
4. En la siguiente imagen hacer Clic en el botón “Siguiente”

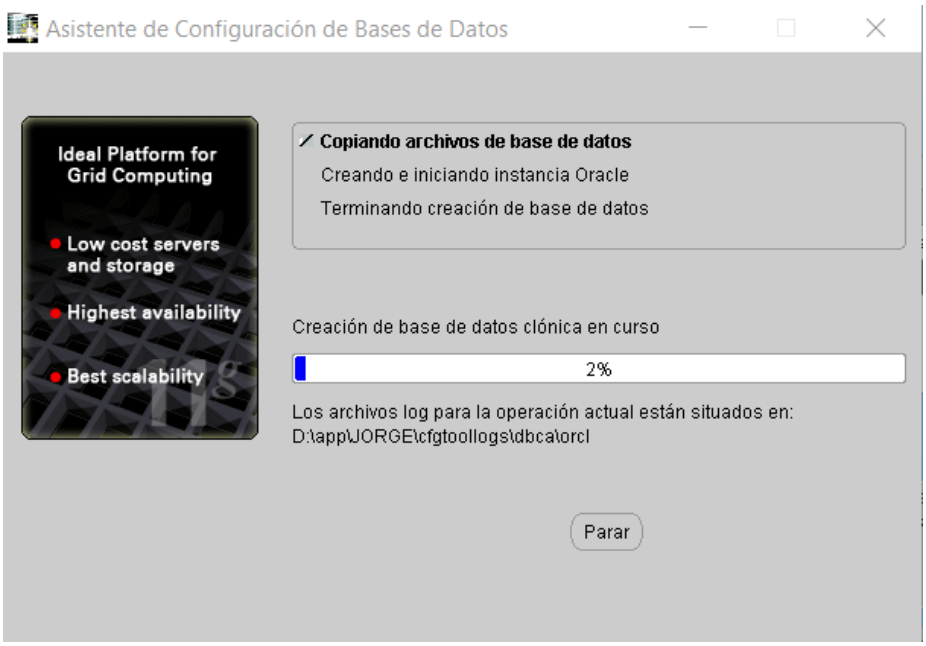


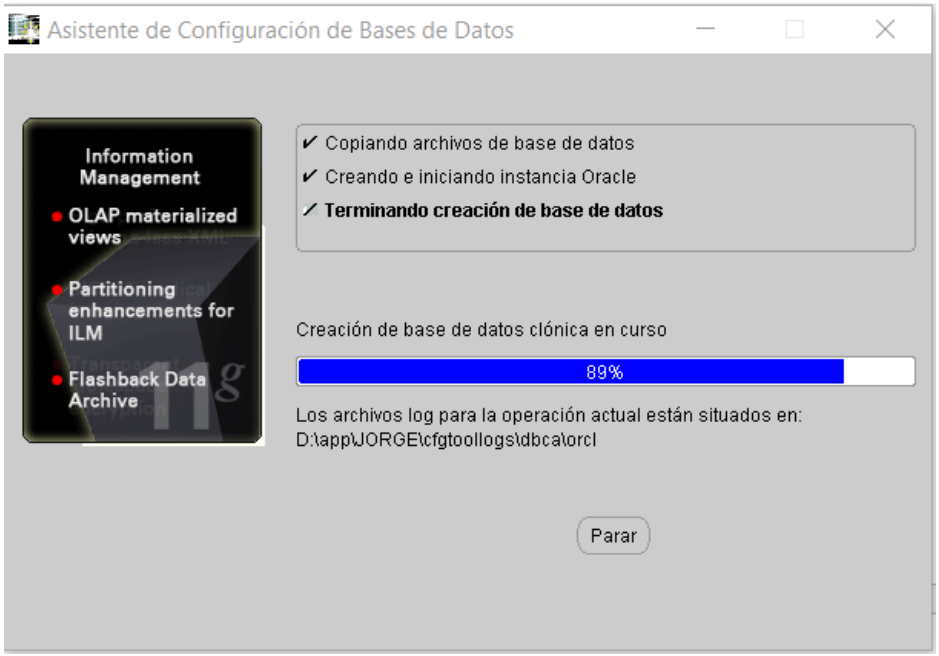
5. En la siguiente imagen se detalla el resumen de lo que se instalará y requisitos mínimos que se requiere, hacer clic en el botón “Instalar”



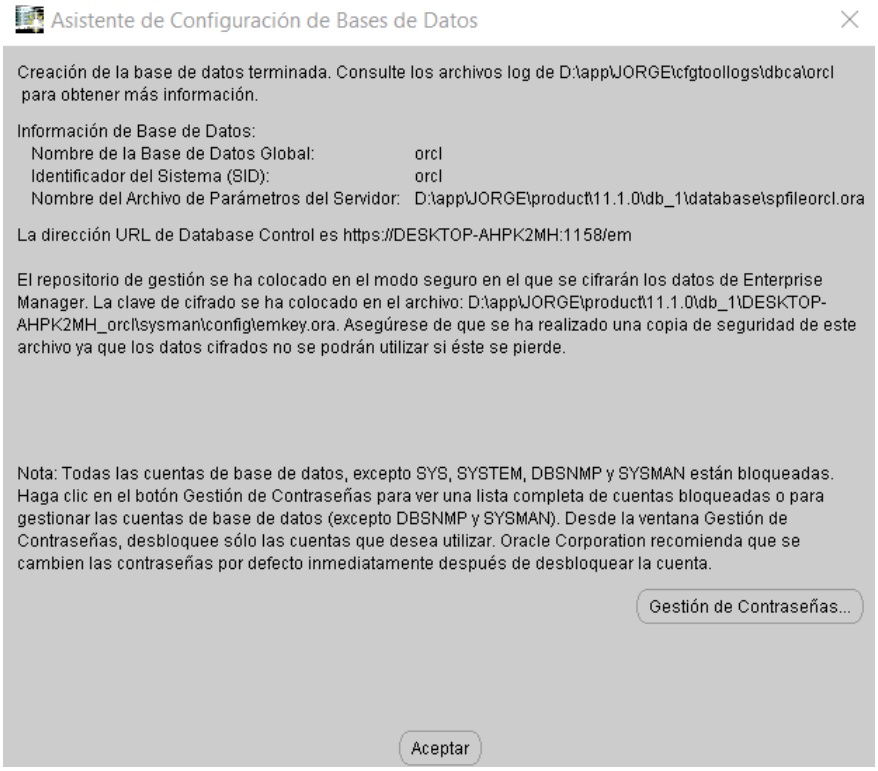
6. Proceso de intalación




Asistente de Configuración de Bases de Datos
Ideal Platform for Grid Computing
Low cost servers and storage
Highest availability
Best scalability
Copiando archivos de base de datos
Creando e iniciando instancia Oracle
Terminando creación de base de datos
Creación de base de datos clónica en curso
2%
Los archivos log para la operación actual están situados en:
D:\app\JORGE\cfgtoollogs\dbca\orcl
Parar


Asistente de Configuración de Bases de Datos
Information Management
OLAP materialized views
Partitioning enhancements for ILM
Flashback Data Archive
Copiando archivos de base de datos
Creando e iniciando instancia Oracle
Terminando creación de base de datos
Creación de base de datos clónica en curso
89%
Los archivos log para la operación actual están situados en:
D:\app\JORGE\cfgtoollogs\dbca\orcl
Parar

8. Hacer Clic en el Botón “**Aceptar”.**

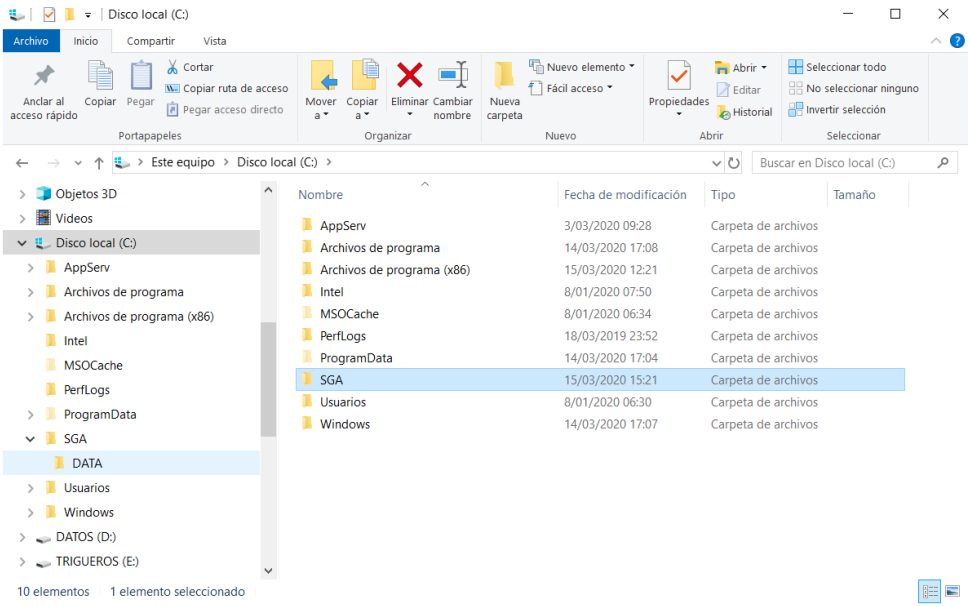


9. En la siguiente imagen hacer clic en “Salir” para concluir la instalación



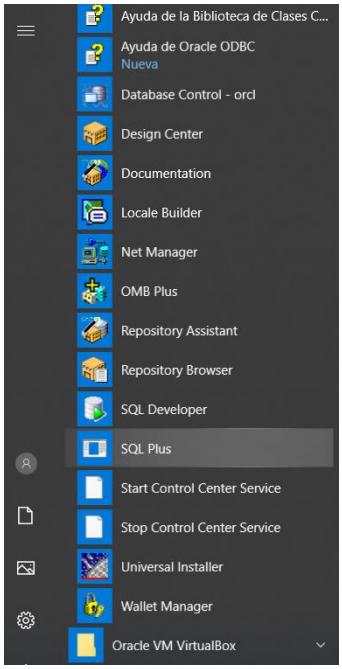
CREACIÓN DE LA BASE DE DATOS

1. Creación del Directorio C:\SGA y el Subdirectorio C:\SGA\DATA

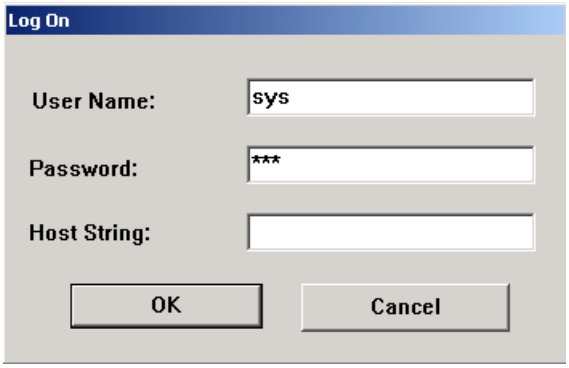


En este subdirectorio C:\SGA\DATA, se va a crear los archivos DATAFILE de Oracle.

2. Ingresar a inicio y ejecutar SQL PLUS**:**



3. Ingresar el **User Name: SYS y Password: SYS** y luego **OK.**



4. Creación de los **Datafiles y TableSpace** para la Data, Index, Temporales, Rollback Segment. Tal como se muestra en el Gráfico.

4.1. Creación del Datafile y TableSpace para la Data:

Create tablespace sga_data datafile 'c:\sga\data\sga_data.ora' size 2024 m default storage (initial 50 m next 50 m); (ENTER)

4.2. Creación del Datafile y TableSpace para los Índices:

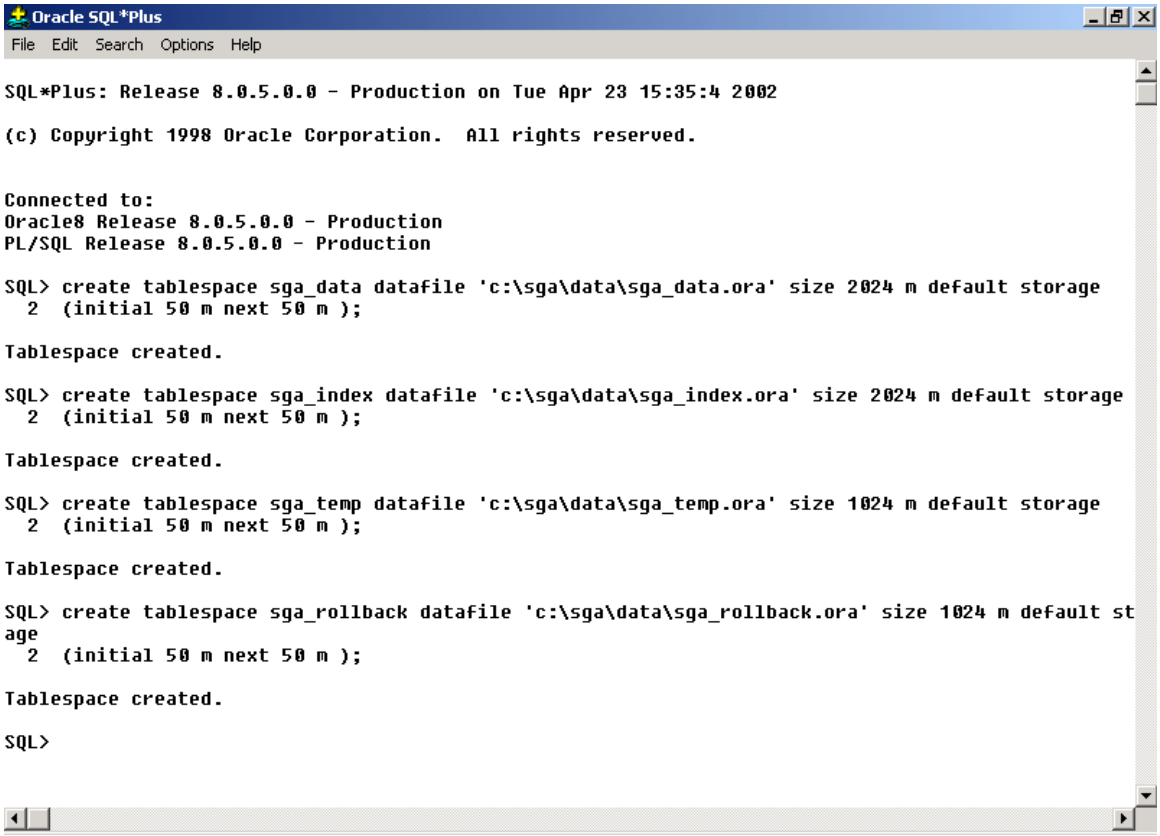
Create tablespace sga_index datafile c:\sga\data\sga_index.ora' size 2024 m default storage (initial 50 m next 50 m); **(ENTER)**

4.3. Creación del Datafile y TableSpace para los Temporales:

create tablespace sga_temp datafile 'c:\sga\data\sga_temp.ora' size 1024 m default storage (initial 50 m next 50 m); (ENTER)

4.4. Creación del Datafile y TableSpace para los Rollback Segment:

Create tablespace sga_rollback datafile 'c:\sga\data\sga_rollback.ora' size 1024 m default storage (initial 50 m next 50 m); (ENTER)



5. Creación de los Rollback Segment: **escribir las siguientes sentencias tal como se indica en el gráfico:**

Create rollback segment sga_rb1 tablespace sga_rollback

Storage (initial 50m next 50m); **(ENTER)**

Create rollback segment sga_rb2 tablespace sga_rollback

Storage (initial 50m next 50m); **(ENTER)**

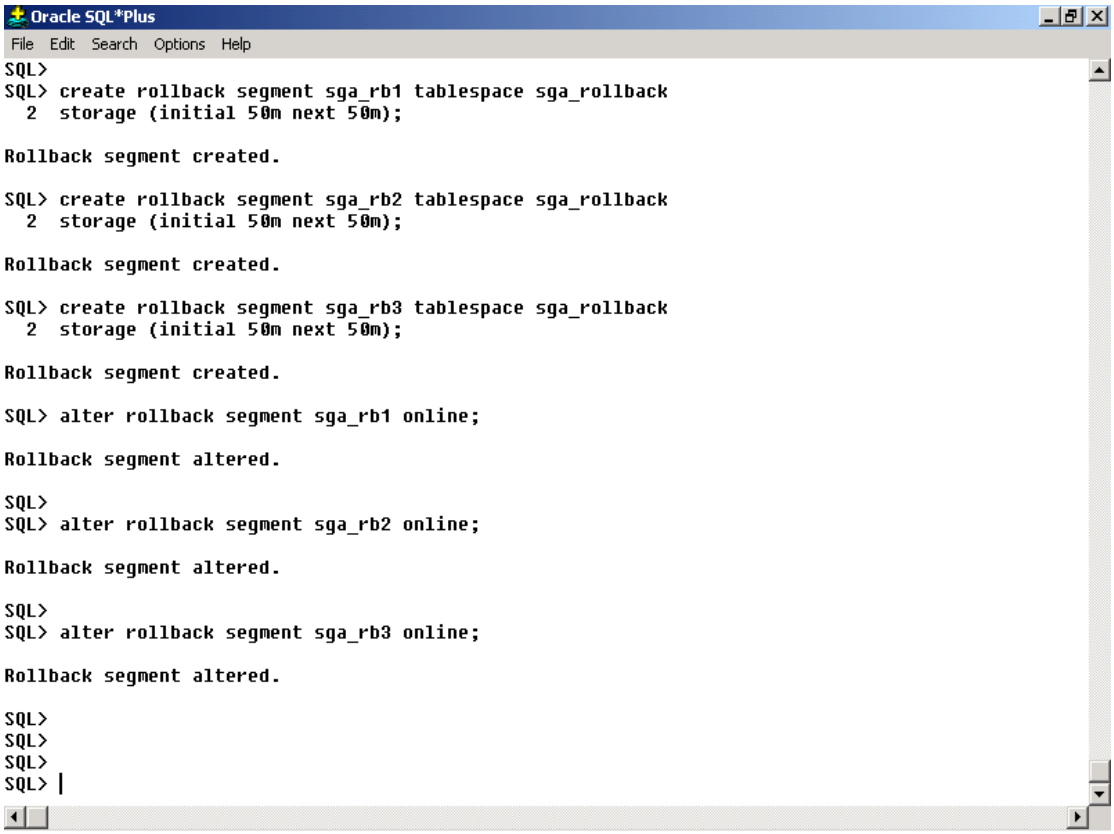
Create rollback segment sga_rb3 tablespace sga_rollback

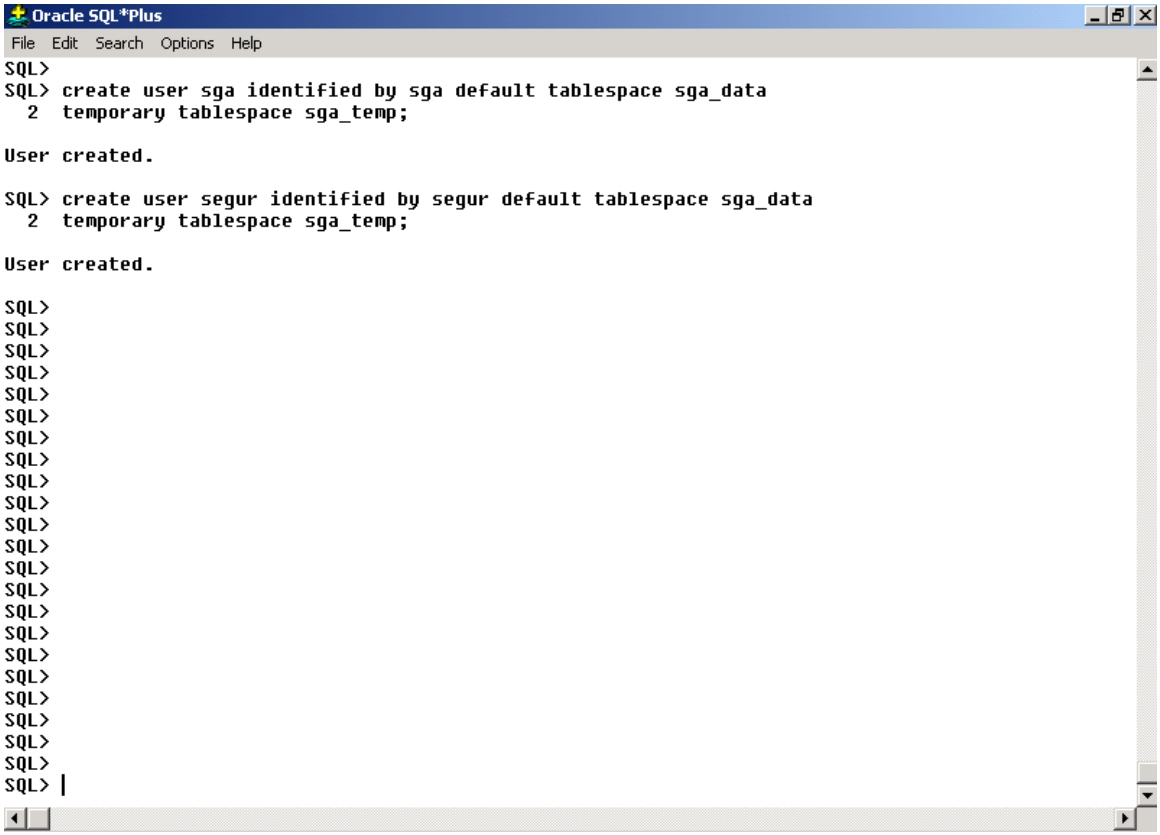
Storage (initial 50m next 50m); **(ENTER)**

Alter rollback segment sga_rb1 online; **(ENTER)**

Alter rollback segment sga_rb2 online; **(ENTER)**

Alter rollback segment sga_rb3 online; **(ENTER)**


Oracle SQL*Plus
File Edit Search Options Help
SQL>
SQL> create rollback segment sga_rb1 tablespace sga_rollback
2 storage (initial 50m next 50m);
Rollback segment created.
SQL> create rollback segment sga_rb2 tablespace sga_rollback
2 storage (initial 50m next 50m);
Rollback segment created.
SQL> create rollback segment sga_rb3 tablespace sga_rollback
2 storage (initial 50m next 50m);
Rollback segment created.
SQL> alter rollback segment sga_rb1 online;
Rollback segment altered.
SQL>
SQL> alter rollback segment sga_rb2 online;
Rollback segment altered.
SQL>
SQL> alter rollback segment sga_rb3 online;
Rollback segment altered.
SQL>
SQL>
SQL>


Oracle SQL*Plus
File Edit Search Options Help
SQL>
SQL> create user sga identified by sga default tablespace sga_data
2 temporary tablespace sga_temp;
User created.
SQL> create user segur identified by segur default tablespace sga_data
2 temporary tablespace sga_temp;
User created.
SQL>
SQL>
SQL>

, CREATE ROLE

, CREATE TRIGGER

, CREATE DATABASE LINK

, CREATE SNAPSHOT

, SELECT ANY TABLE

, INSERT ANY TABLE

, UPDATE ANY TABLE

, SELECT ANY SEQUENCE

, EXECUTE ANY PROCEDURE

, EXECUTE ANY LIBRARY

, DROP ANY SYNONYM

, DROP ANY TABLE

, DROP ANY VIEW

, DROP ANY PROCEDURE

, DROP ANY SEQUENCE

, DROP ANY TRIGGER

TO role_sga; **(ENTER)**

GRANT EXECUTE ON standard TO role_sga; **(ENTER)**

Grant role_sga to sga; **(ENTER)**

Grant dba to sga; **(ENTER)**

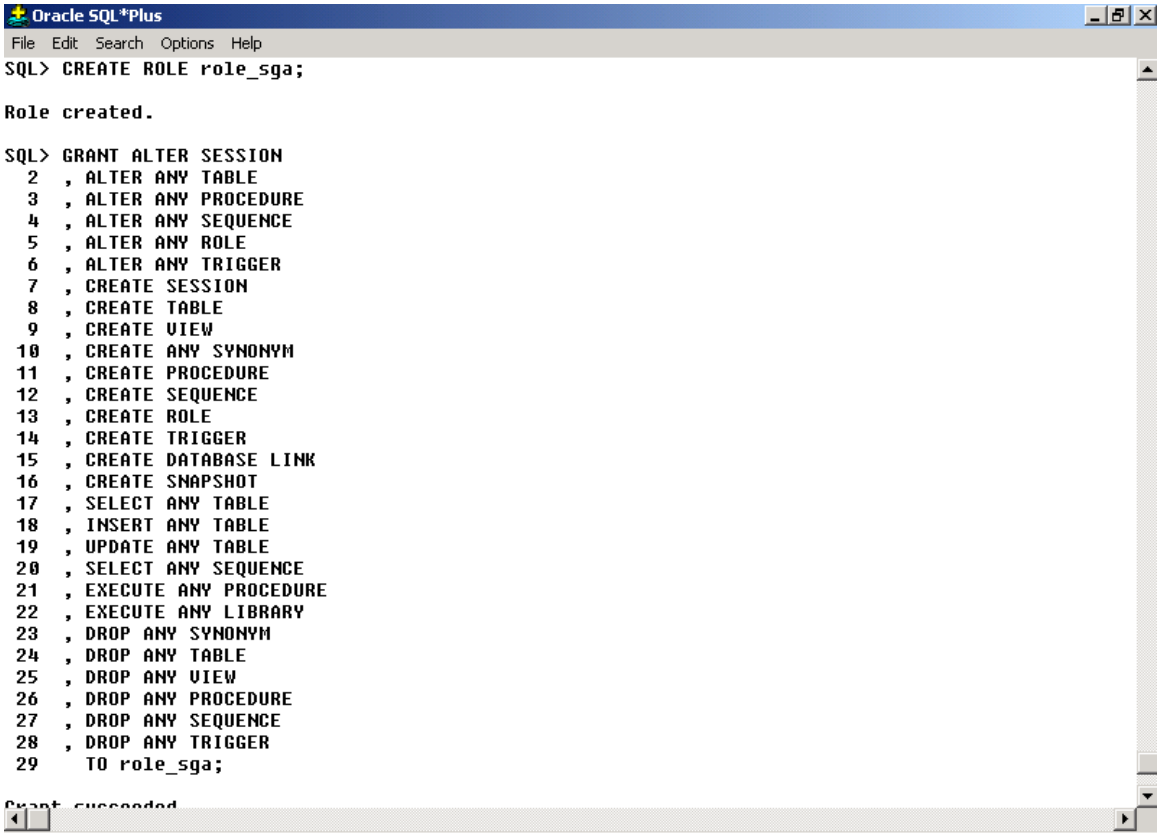
Grant resource to sga; **(ENTER)**

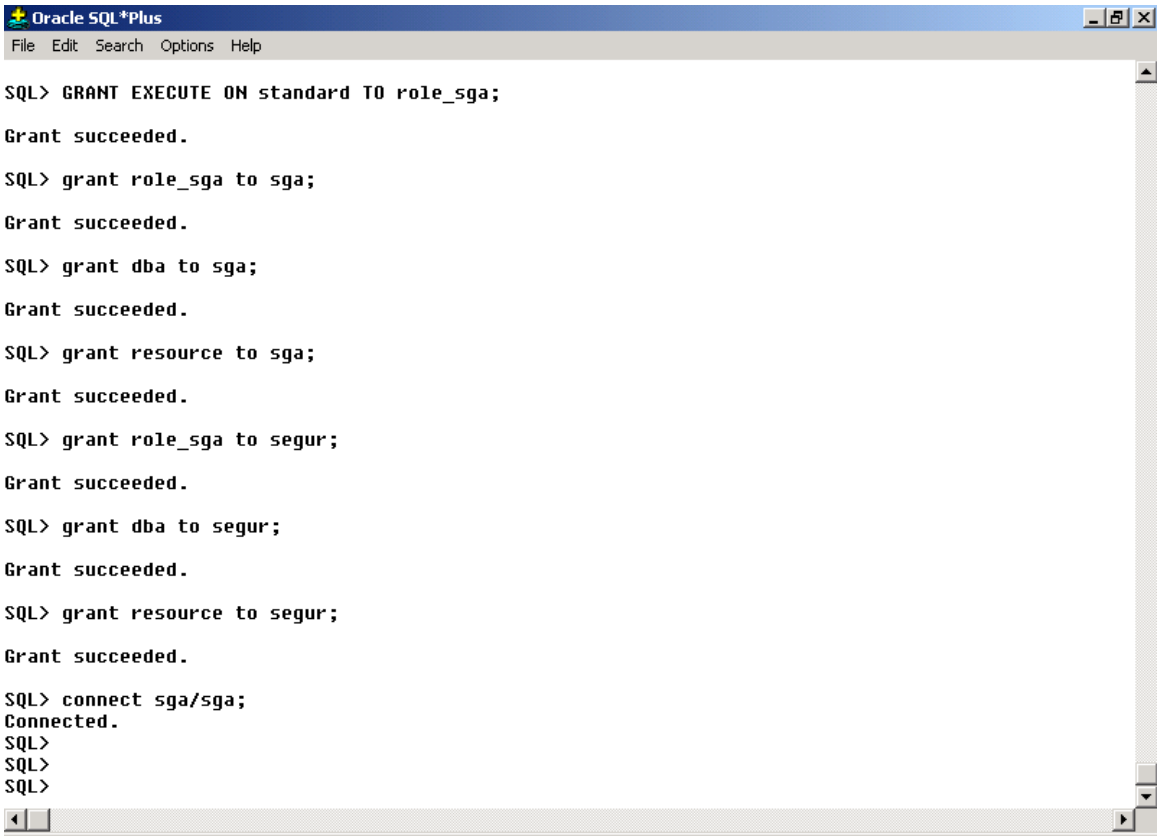
Grant role_sga to segur; **(ENTER)**

Grant dba to segur; **(ENTER)**

Grant resource to segur; **(ENTER)**

connect sga/sga; **(ENTER)**


Oracle SQL*Plus
File Edit Search Options Help
SQL> CREATE ROLE role_sga;
Role created.
SQL> GRANT ALTER SESSION
2 , ALTER ANY TABLE
3 , ALTER ANY PROCEDURE
4 , ALTER ANY SEQUENCE
5 , ALTER ANY ROLE
6 , ALTER ANY TRIGGER
7 , CREATE SESSION
8 , CREATE TABLE
9 , CREATE VIEW
10 , CREATE ANY SYNONYM
11 , CREATE PROCEDURE
12 , CREATE SEQUENCE
13 , CREATE ROLE
14 , CREATE TRIGGER
15 , CREATE DATABASE LINK
16 , CREATE SNAPSHOT
17 , SELECT ANY TABLE
18 , INSERT ANY TABLE
19 , UPDATE ANY TABLE
20 , SELECT ANY SEQUENCE
21 , EXECUTE ANY PROCEDURE
22 , EXECUTE ANY LIBRARY
23 , DROP ANY SYNONYM
24 , DROP ANY TABLE
25 , DROP ANY VIEW
26 , DROP ANY PROCEDURE
27 , DROP ANY SEQUENCE
28 , DROP ANY TRIGGER
29 TO role_sga;


Oracle SQL*Plus
File Edit Search Options Help
SQL> GRANT EXECUTE ON standard TO role_sga;
Grant succeeded.
SQL> grant role_sga to sga;
Grant succeeded.
SQL> grant dba to sga;
Grant succeeded.
SQL> grant resource to sga;
Grant succeeded.
SQL> grant role_sga to segur;
Grant succeeded.
SQL> grant dba to segur;
Grant succeeded.
SQL> grant resource to segur;
Grant succeeded.
SQL> connect sga/sga;
Connected.
SQL>
SQL>
SQL>

Anexo 5. INSTALACIÓN DE ORACLE V11g (CLIENTE)

MANUAL DE INSTALACION DE ORACLE 11g – CLIENTE

INSTALACION

Insertar CD de instalación y hacer clic en el archivo ejecutable setup.exe

1.- Enseguida aparecerá la siguiente pantalla y presione en la opción “Cliente de Oracle”, hacer clic en **SIGUIENTE**



2.- Luego aparecerá la siguiente pantalla, seleccionar la opción “RUNTIME” luego presionar el botón “SIGUIENTE”



3.- Luego aparecerá la siguiente pantalla, en la que puede cambiar el nombre y ruta donde se grabará el oracle (se recomienda la instalación en particiones con mayor de 300MB), y después presione el botón “Siguiente”.



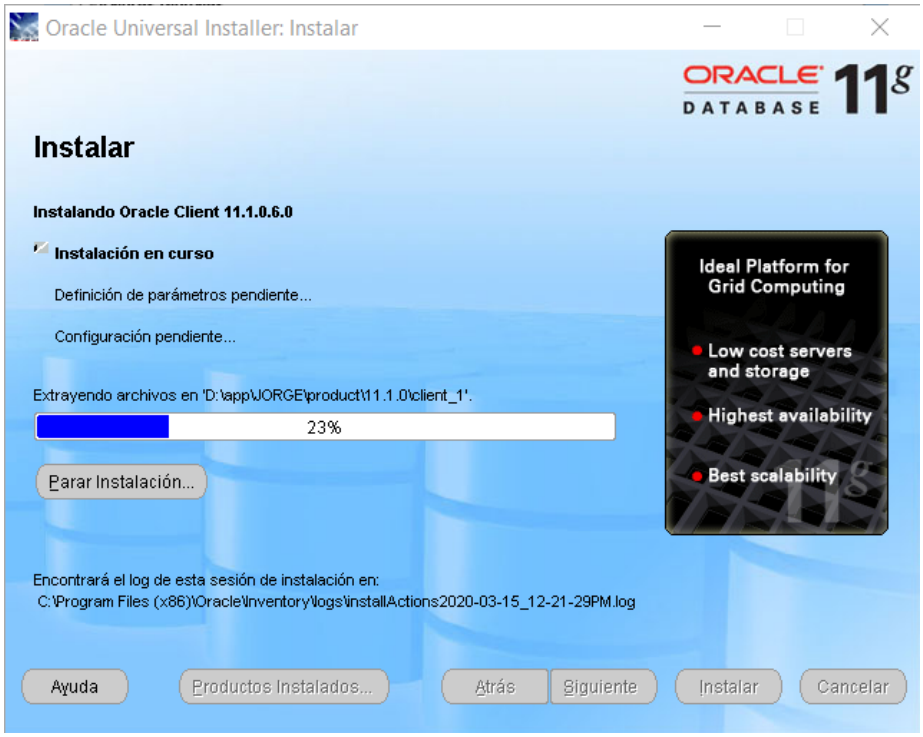
4.- La siguiente pantalla nos dirá los requisitos especificos del Producto a instalarse en la PC, dar clic en “SIGUIENTE”



5.- Después de la comprobación aparecerá la siguiente ventana resumen, hacer clic en “INSTALAR”



6.- Luego empieza la instalación del producto Oracle 11g



7.- Después aparecerá la ventana de Asistente de configuración:



8.- Enseguida aparecerá la siguiente ventana, y presionar el botón “Siguiente”:



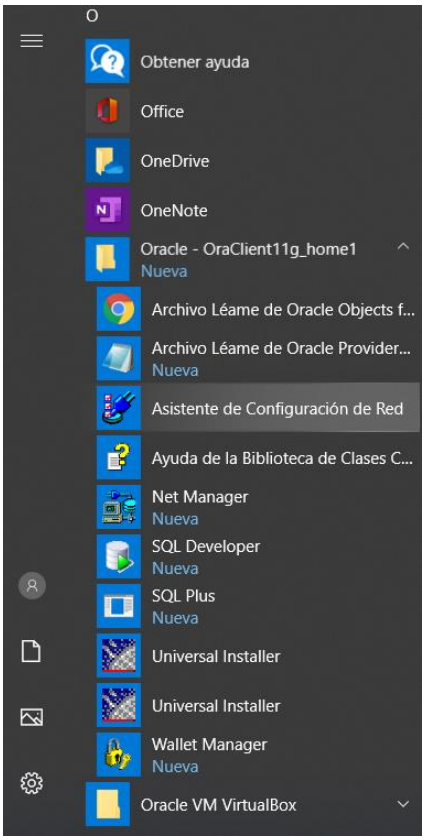
9.- En la siguiente ventana hacer clic en “TERMINAR”



10.- en la siguiente ventana hacer clic en “SALIR”



11.- Buscar la instalación y seleccionar “Asistente de Configuración de Red”



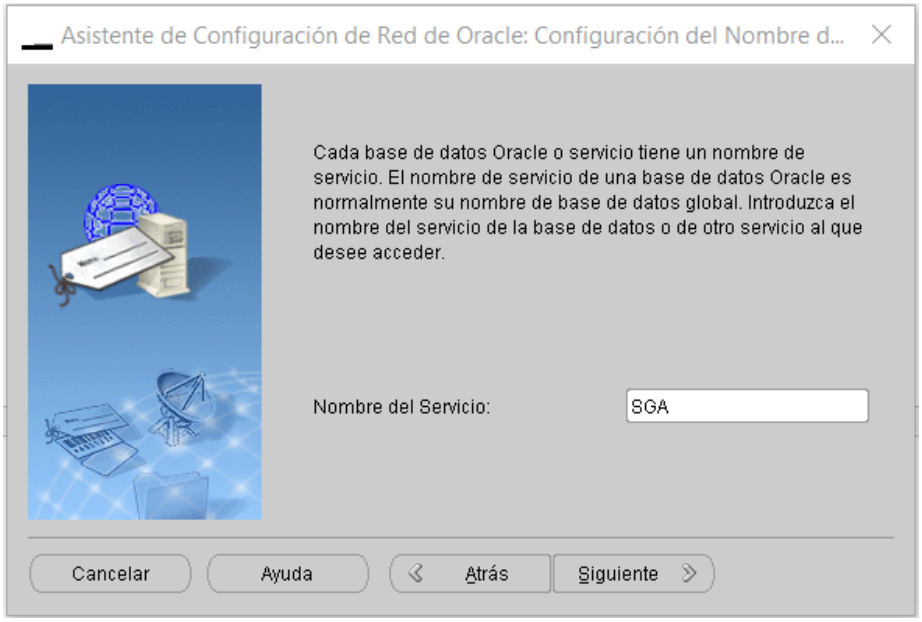
12.- En la siguiente imagen seleccionar “Configuración del Nombre del Servicio de Red Local” y hacer clic en “Siguiente”



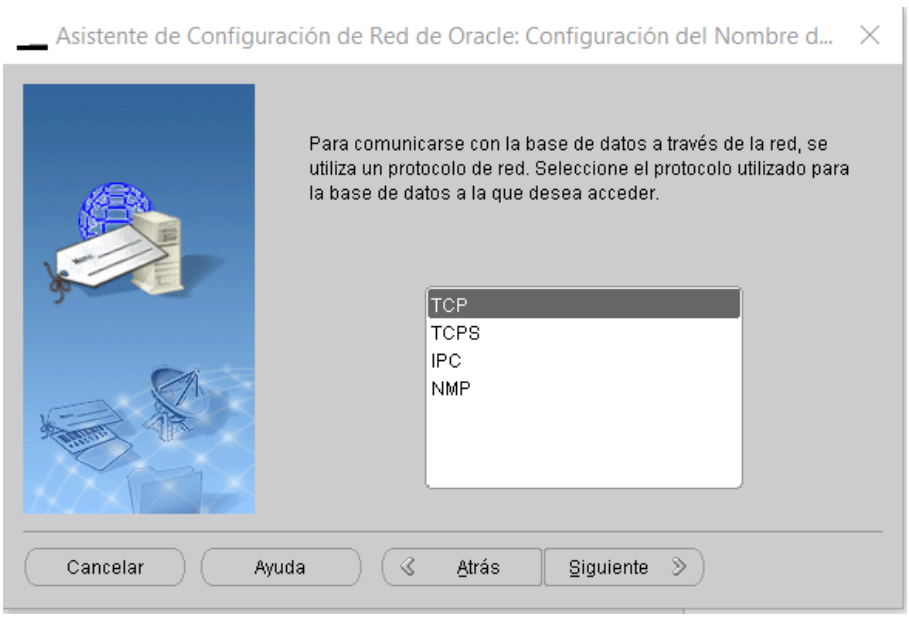
13.- En la siguiente imagen automáticamente esta seleccionado la opción **“Agregar”** debido a que es la primera instalación, hacer clic en **“Siguiente”**



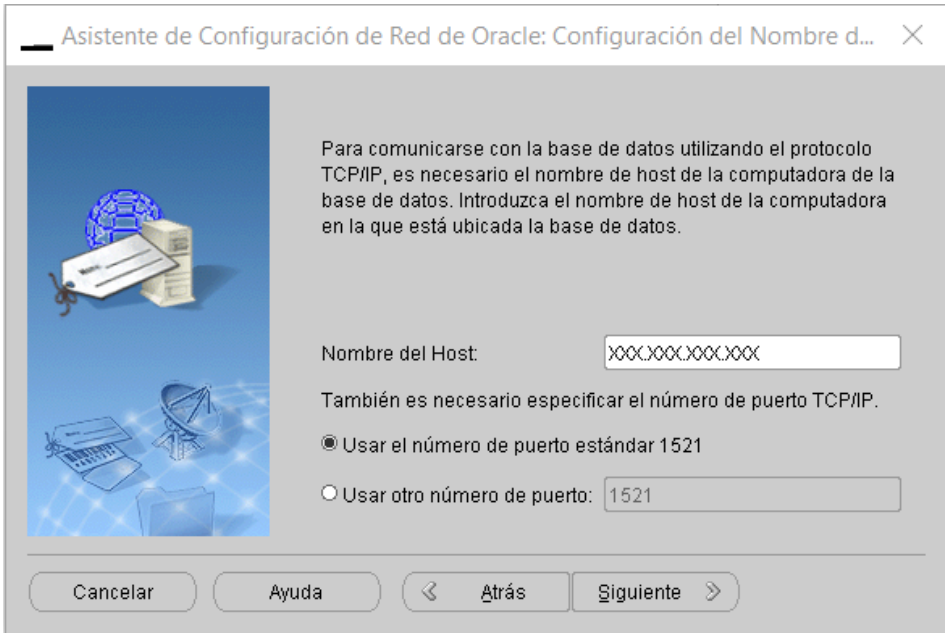
14.- Escribir como nombre de servicio “SGA” y presionar el botón “Siguiente”



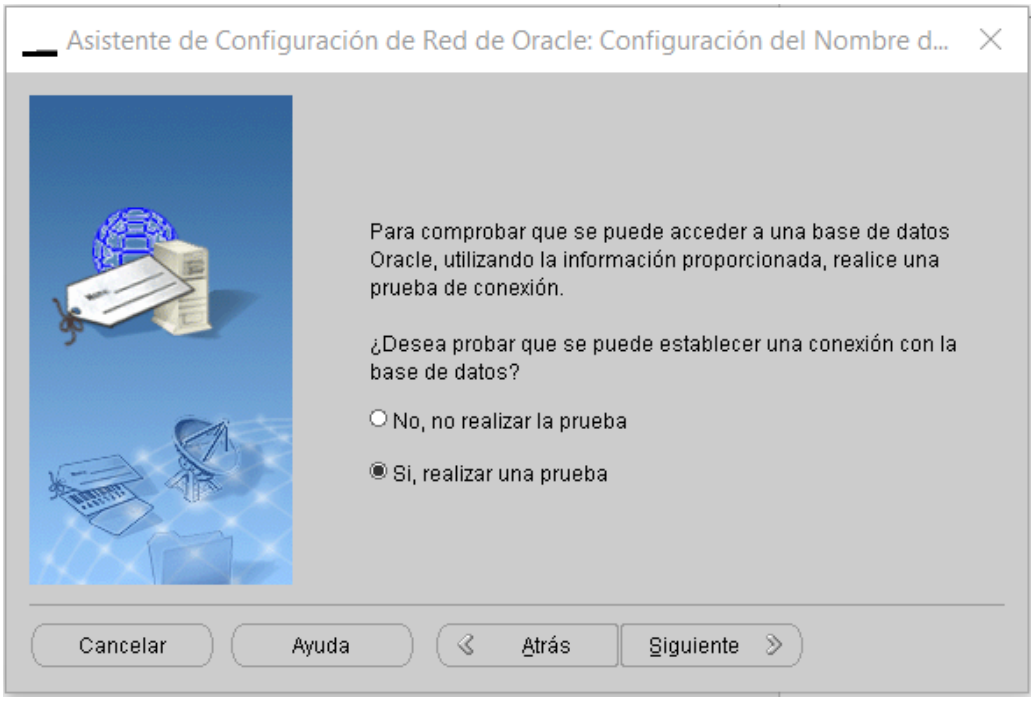
15.- Seleccionar como protocolo “TCP” y presionar el botón “Siguiente”



16.- Escribir como nombre de Host “XXX.XXX.XXX.XXX”, es recomendable colocar el IP donde se encuentra Instalado la base de datos de ORACLE, dejar en el puerto estándar en 1521 y presionar el botón “Siguiente”



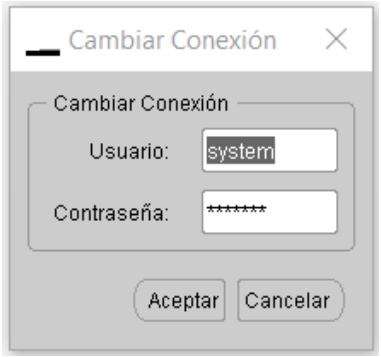
17.- Seleccionar la opción “Si, realizar una prueba”



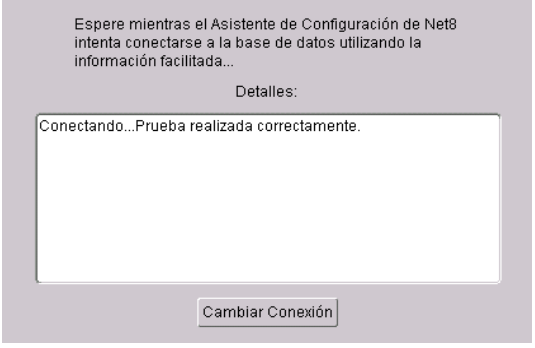
18.- Presionar el Botón “Cambiar conexión”



19.- Escribir como usuario “***” y password “***”, y luego presionar el botón “Aceptar”



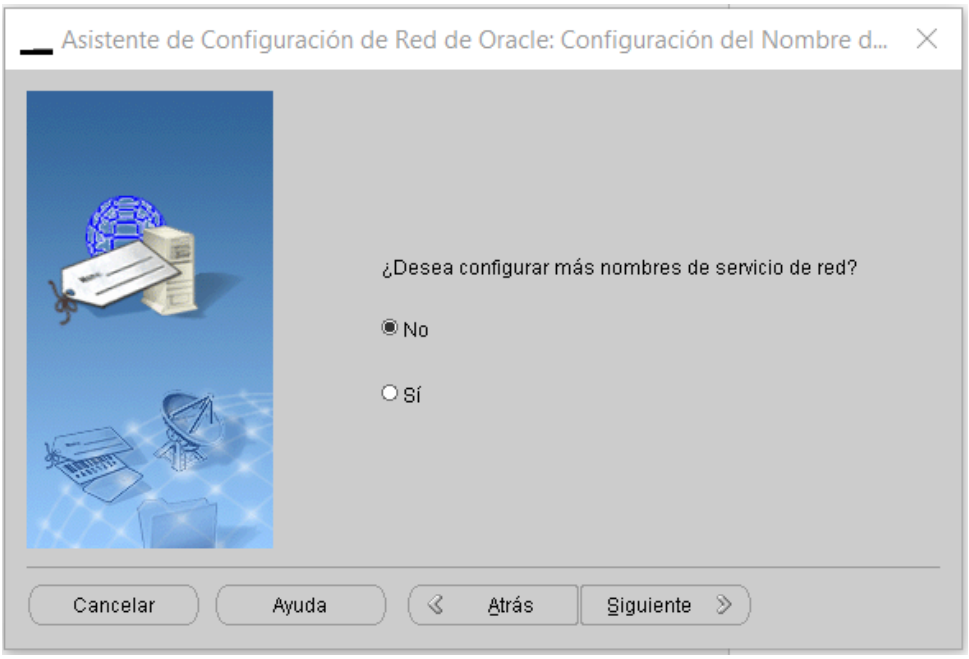
20.- Luego aparece la confirmación que se ha conectado correctamente y después presionar el botón “Siguiente”



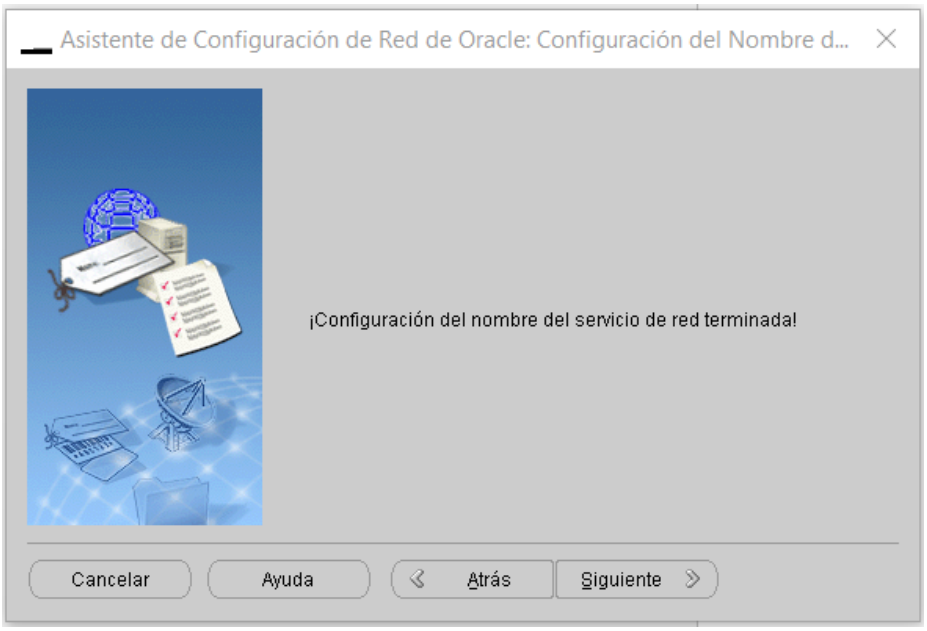
21.- Escribir como nombre del Servicio de Red que se registró en el Servidor con anterioridad y presionar el botón “Siguiente”:



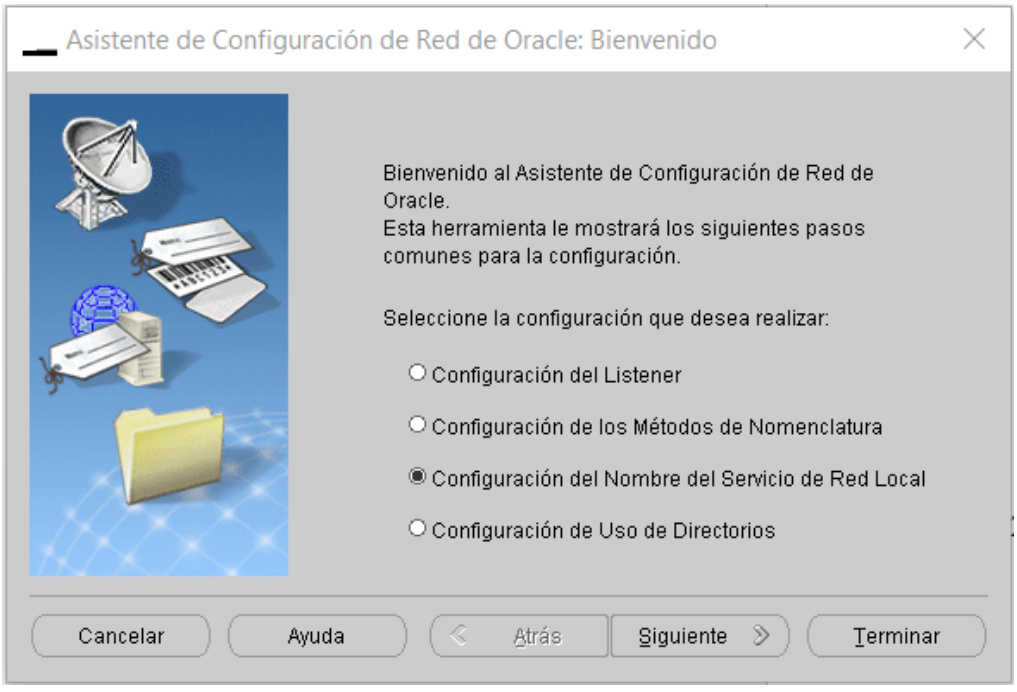
22.- Seleccionar la opción “No” y presionar el botón “Siguiente”



23.- Luego presionar el botón “Siguiente”:



24.- Finalmente presionar el botón “Terminar”:

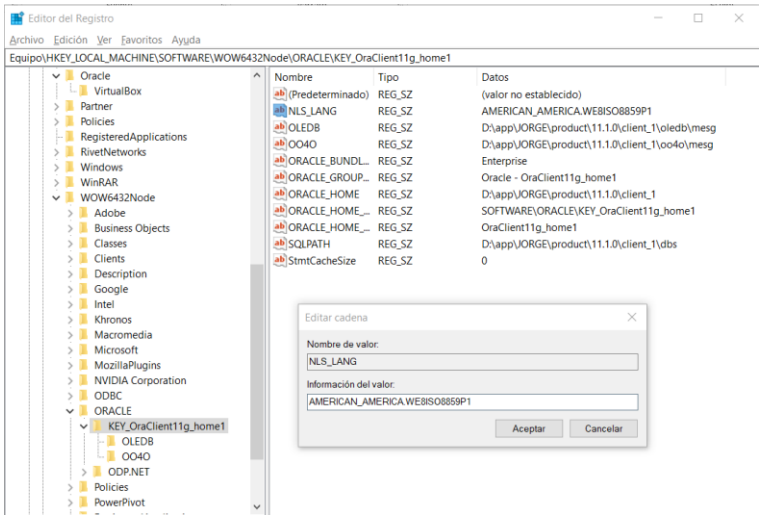


25.- CONFIGURACIÓN DE REGEDIT (OBLIGATORIO)

En el editor de Registro modificar los **NLS_LANG** su Información de valor por

AMERICAN_AMERICA.WE8ISO8859P1

Del Registro
Equipo\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ORACLE\KEY_OraClient11g_home1



Anexo 6. CÓDIGO FUENTE

CÓDIGO FUENTE SPRINT 1

Código fuente acceso al Sistema

```
datawindowchild		ldw_child
integer				li_error, li_yesno
s_arg_segur			lstr_arg
string		ls_referencia, ls_c_usuario,ls_c_fl_activo_log, ls_usuario, ls_facultad
//--------------------------------------------------------------------------

lstr_arg		= message.powerobjectparm
is_c_sistema	= lstr_arg.s[1]
//--------------------------------------------------------
//--picture
p_1.PictureName = gs_dir + '\Recursos\LogoURP.bmp'
pb_1.PictureName = gs_dir + '\Recursos\aceptar.jpg'
pb_3.PictureName = gs_dir + '\Recursos\cancelar.JPG'
choose case gs_c_modulo
	case '01'
		pb_2.PictureName = gs_dir + '\Recursos\Alumnos.bmp'
		//pb_3.PictureName = gs_dir + '\Recursos\EvalSoc.bmp'
		//pb_4.PictureName = gs_dir + '\Recursos\CentroMédico.bmp'
		//pb_5.PictureName = gs_dir + '\Recursos\Autoseguro.bmp'
	case '08'
		pb_2.PictureName = gs_dir + '\Recursos\AperturaAcad.bmp'
	case '05'
		pb_2.PictureName = gs_dir + '\Recursos\PlanEstudios.bmp'
	case '07'
		pb_2.PictureName = gs_dir + '\Recursos\Matricula.bmp'
	case '06'
		pb_2.PictureName = gs_dir + '\Recursos\horario.bmp'
	case '00'
		pb_2.PictureName = gs_dir + '\Recursos\Seguridad.bmp'
//	case '06'
//pb_3.PictureName = gs_dir + '\Recursos\lupa.bmp'
	case '09'
		pb_2.PictureName = gs_dir + '\Recursos\cierreacad.bmp'
	case '10'
		pb_2.PictureName = gs_dir + '\Recursos\registro.bmp'
	case '11'
		pb_2.PictureName = gs_dir + '\Recursos\traslado.bmp'
	case '12'
		pb_2.PictureName = gs_dir + '\Recursos\notas.bmp'
    case '13'
		pb_2.PictureName = gs_dir + '\Recursos\corriente.bmp'
	case '14'
		pb_2.PictureName = gs_dir + '\Recursos\gerencial.bmp'
	case '15'
		pb_2.PictureName = gs_dir + '\Recursos\auditoria.jpg'
end choose
//--------------------------------------------------------

dw_1.settransobject( sqlca )
dw_1.getchild("cod_facultad", iddw01)
dw_1.getchild("cod_carrera", iddw02)

iddw01.settransobject(sqlca)
iddw02.settransobject(sqlca)

if iddw01.retrieve(ls_usuario) < 1 then iddw01.insertrow(0)
```

```
if iddw02.retrieve(ls_usuario, ls_facultad) < 1 then iddw02.insertrow(0)

dw_1.insertrow( 0 )

RegistryGet( "HKEY_LOCAL_MACHINE\Network\logon", "username", gs_c_usuario)
RegistryGet( "HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\ComputerName\ComputerName",
"ComputerName", ls_referencia)
gs_c_referencia=trim(ls_referencia)
RegistryGet( "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\NetTrans\0002",
"IPAddress", ls_referencia)
gs_c_referencia=left(gs_c_referencia+" // "+ls_referencia,70)

gs_c_usuario= UPPER(gs_c_usuario)
if POS(gs_c_usuario,'¥') > 0 then
        gs_c_usuario= REPLACE(gs_c_usuario, POS(gs_c_usuario,'¥'),1,'Ñ')
end if

dw_1.setItem(1,"c_c_usuario",gs_c_usuario)
//iddw01.settransobject(sqlca)
//if iddw01.retrieve(gs_c_usuario) < 1 then iddw01.insertrow(0)

SELECT c_c_usuario, c_fl_activo_log
        INTO :ls_c_usuario, :ls_c_fl_activo_log
        FROM segur.USUARIO
        WHERE C_C_USUARIO = :gs_c_usuario using sqlca;

IF (sqlca.sqlcode <> 0) or ls_c_fl_activo_log = '0' THEN return

//li_yesno = messagebox( "Bienvenido....", 'Desea ingresar como "' + gs_c_usuario + '"', question!, yesno!, 2 )
//IF li_yesno = 2 THEN
return

// Recupera perfil segun el usuario elegido y el sistema
SELECT C_C_PERFIL
  INTO :gs_c_perfil
  FROM segur.USUARIO_PERFIL
  WHERE ( C_C_USUARIO = :gs_c_usuario ) AND
      ( C_C_SISTEMA = :gs_c_sistema )  and
                          ( c_c_modulo  = :gs_c_modulo  )using sqlca;

IF sqlca.sqlcode <> 0 THEN
        messagebox( "Aviso", "Usuario no tiene perfil en el Sistema" )
        lstr_arg.ch[1] = '0'
        closewithreturn( this, lstr_arg )
        return
END IF

lstr_arg.ch[1] = '1'
lstr_arg.s[2] = '99'
lstr_arg.s[3] = gs_c_usuario
closewithreturn( this, lstr_arg )
```

Código fuente Gestión de Usuarios

```
//this.triggerevent( "ue_update" )
long                                                    ll_getrow,ll_maxrow
s_arg                                                   lstr_arg
string                                          ls_usuario
//-----------------------------------------------------------

ll_getrow = dw_1.getrow()
IF ll_getrow < 1 THEN return

IF isnull( dw_1.getitemstring( ll_getrow, "c_c_usuario" ) ) THEN
        lstr_arg.b[1] = false
ELSE
        lstr_arg.b[1] = true
end if

ls_usuario          =dw_1.getitemstring( ll_getrow, "c_c_usuario" )
lstr_arg.s[1] = ls_usuario

openwithparm( w_m_ingrese_password_u, lstr_arg )
lstr_arg  =         message.powerobjectparm

if lstr_arg.s[2]="M" then
        dw_1.SetItem(dw_1.GetRow(),"c_t_nombre",lstr_arg.s[3])
        dw_1.SetItem(dw_1.GetRow(),"doc_id",lstr_arg.s[4])
        dw_1.SetItem(dw_1.GetRow(),"nro_doc_id",lstr_arg.s[5])
        dw_1.SetItem(dw_1.GetRow(),"c_c_cargo",lstr_arg.s[6])
        dw_1.SetItem(dw_1.GetRow(),"c_c_password",lstr_arg.s[7])
        return
else
        dw_1.TriggerEvent("ue_retrieve")
end if
```

Código fuente gestión de perfiles

```
CHOOSE CASE is_arg.s[2]
        CASE "d_m_perfil"
                parent.triggerevent( "ue_sistemas" )

        CASE "d_m_aplicativo"
                parent.triggerevent( "ue_modulo" )

        CASE "d_m_modulo"
                parent.triggerevent( "ue_menu" )

        CASE "d_m_menu"
                parent.triggerevent( "ue_items_menu" )

        CASE    "d_c_modulo_menu"
                parent.triggerevent( "ue_accesos" )

END CHOOSE

-------------------------------------------------------------

s_arg           lstr_arg
window  lw_sheet
long            ll_getrow
```

```
string     ls_sistema, ls_modulo

ll_getrow = dw_1.getrow()
IF ll_getrow < 1 THEN return

lstr_arg.s[1]        = "Mantenimiento de Perfiles del " + dw_1.GetItemString(ll_getrow,"c_t_dsc")
lstr_arg.s[2]        = "d_m_perfil"

ls_sistema = dw_1.GetItemString(dw_1.getrow(),"c_c_sistema")
if ls_sistema = '' or isnull(ls_sistema) then
          messagebox("Aviso", "No existe sistema")
          return
else
          lstr_arg.s[4]        = ls_sistema
end if

ls_modulo = dw_1.GetItemString(dw_1.getrow(),"c_c_modulo")
if ls_modulo = '' or isnull(ls_modulo) then
          messagebox("Aviso", "No existe modulo")
          return
else
          lstr_arg.s[5]        = ls_modulo
end if


opensheetwithparm( lw_sheet, lstr_arg, "w_m_tabla_main",this, gi_positem, layered! )
```

Código fuente gestión asignación de perfiles

```
long                                        ll_getrow
s_arg                                       lstr_arg
window                                lw_sheet
//-------------------------------------------------------------

ll_getrow = dw_1.getrow()
IF ll_getrow < 1 THEN return

lstr_arg.s[1] = dw_1.getitemstring( ll_getrow, "c_c_usuario" )
lstr_arg.s[2] = "U"

//          opensheetwithparm( lw_sheet, lstr_arg, "w_m_sistema_perfil_usuario", this, gi_positem, layered! )
openwithparm( w_m_usuario_sistema_perfil, lstr_arg)
```

```
long ll_maxrow, ll_row
string ls_sistema, ls_modulo, ls_perfil, ls_usuario

ls_usuario=is_c_usuario
dw_1.accepttext()
ll_maxrow=dw_1.rowcount()
For ll_row=1 to ll_maxrow
          if (dw_1.IsSelected(ll_Row)) then
                    ls_sistema          =dw_1.GetItemString(ll_row,"c_c_sistema")
                    ls_modulo           =dw_1.GetItemString(ll_row,"c_c_modulo")
                    ls_perfil  =dw_1.GetItemString(ll_row,"c_c_perfil")
          end if
End for

delete from segur.usuario_perfil
where c_c_usuario=:ls_usuario
and       c_c_sistema=:ls_sistema
and       c_c_modulo =:ls_modulo
using sqlca;
```

```
insert into segur.usuario_perfil (c_c_usuario,c_c_sistema,c_c_modulo, c_c_perfil)
values (:ls_usuario, :ls_sistema,:ls_modulo, :ls_perfil) using sqlca;

commit using sqlca;
```

CÓDIGO FUENTE SPRINT 2

Código fuente gestión de facultad

```
integer li_Cnt

this.x     =          0
this.y     =          0


IF IsValid (Message.powerobjectparm) THEN
          IF message.powerobjectparm.ClassName() = "str_par" THEN
                    istr_par = message.powerobjectparm
                    if istr_par.s_accion = 'consulta' then this.changemenu(m_grid_consulta)
          END IF
END IF

istr_par.po_caller =          this
istr_par.s_title              =          this.Title
istr_par.tv                                        =          tv_1
istr_par.dw                                        =          dw_1

TriggerEvent('ue_transaction')

IF          Not IsValid(itrans_obj)      THEN    f_mensaje(this.title,'deben configurar la transaction de su módulo,
evento ue_transaction')

IF dw_1.SetTransObject(itrans_obj) = -1 THEN
   f_mensaje(this.title,"Error al conectar dw_1")
ELSE
          dw_1.Reset()
END IF

SetPointer(HourGlass!)

// Captura los objetos TreeView y datawindow para poder modificar el tamaño
idrg_Vertical[1] = tv_1
idrg_Vertical[2] = dw_1
// Captura el color de fondo de ventana para hacer invisible el color de barra
il_HiddenColor = This.BackColor
st_Vertical.BackColor = il_HiddenColor
// Cambia de tamaño la barra
fw_ResizeBars()
// Cambia de tamaño los paneles (TreeView y ListView)
fw_ResizePanels()

//Parámetros a configurar:
ii_niveles = 3 //Número de niveles del TreeView
is_raiz    = "Facultades" //Cadena conteniendo el nombre de la raiz del arbol
ii_picture_select =  1//Numero del icono que sera usado cuando se selecciona un item del arbol
istr_param.i[1] = 0
// Crea DataStores para almacenar los datos del explorer
```

```
ids_Data[1] = Create DataStore
ids_Data[1].DataObject = "ds_facultades"

ids_Data[2] = Create DataStore
//ids_Data[2].DataObject = "ds_carrera"
ids_Data[2].DataObject = "ds_escuela"

ids_Data[3] = Create DataStore
ids_Data[3].DataObject = "ds_carrera_hijo"

// Realiza la conexion con el dbms
For li_Cnt = 1 To ii_niveles
        ids_Data[li_Cnt].SetTransObject(itrans_obj)
Next

// Invocar desde la ventana heredada al evento ue_populate
Post Event ue_populate()
```

```
string ls_facultad, ls_carrera

if dw_1.rowcount() < 1 and istr_par.s_accion <> 'new' then return

CHOOSE CASE dw_1.dataObject
        CASE 'd_pea_facultades'
                if istr_par.s_accion <> 'new' then
                        ls_facultad = dw_1.object.cod_facultad[dw_1.getrow()]
                        istr_par.s[1] = ls_facultad
                        istr_par.l[1] = 0  // update facultad
                else
                        istr_par.s[1] = istr_param.s[1] //''
                        istr_par.l[1] = 1  // nuevo facultad
                end if
                  istr_par.i[1] = 0

//   CASE 'd_pea_carreras'
   CASE 'd_pea_escuelas'
                if istr_par.s_accion <> 'new' then
                        ls_facultad = dw_1.object.cod_facultad[dw_1.getrow()]
                        ls_carrera = dw_1.object.cod_escuela[dw_1.getrow()]
                        istr_par.s[1] = ls_facultad
                        istr_par.s[3] = ls_carrera
                        istr_par.l[1] = 2  // update carrera
                else
                        istr_par.s[1] = istr_param.s[1]                  // codigo facultad
                        istr_par.l[1] = 2  // nuevo carrera
                end if
                  istr_par.i[1] = 1 // segundo nivel
   CASE 'd_pea_carreras_hijo'
                if istr_par.s_accion <> 'new' then
                        ls_facultad = dw_1.object.cod_facultad[dw_1.getrow()]
                        ls_carrera = dw_1.object.cod_carrera[dw_1.getrow()]
                        istr_par.s[1] = ls_facultad
                        istr_par.s[3] = ls_carrera
                        istr_par.l[1] = 3  // update carrera hijo

                else
                        istr_par.s[1] = istr_param.s[1]                  //facultad/carrera
                        istr_par.l[1] = 3  // nuevo carrera hijo
                end if
                  istr_par.i[1] = 2   //tercer nivel
END CHOOSE



if istr_par.s_accion = 'new' and istr_param.i[1] < 4 then
        istr_par.s_title      = istr_param.s[2]
        if dw_1.rowcount() > 0 then
                istr_par.s[2] = dw_1.object.cod_facultad[dw_1.getrow()]
```

```
		end if
		OpenSheetWithParm(w_pea_form_m_plan_estudio, istr_par, parentwindow(), gi_menu_posi,
original!)
//elseif ( istr_par.s_accion = 'fileopen' or istr_par.s_accion = 'consulta') and len(istr_par.s[1]) > 1 then
elseif ( istr_par.s_accion = 'fileopen' or istr_par.s_accion = 'consulta') then
		istr_par.s_title	= istr_param.s[2]
		if dw_1.rowcount() > 0 then
			istr_par.s[2] = dw_1.object.cod_facultad[dw_1.getrow()]
		end if
		OpenSheetWithParm(w_pea_form_m_plan_estudio, istr_par, parentwindow(), gi_menu_posi,
original!)
end if
```

Código fuente gestión de carrera profesional

```
string	ls_cod_item, ls_codigo, ls_facultad, ls_carrera, ls_null
long		ll_len , ll_escuela
Pointer oldpointer
Integer  li_op

IF dw_1.AcceptText() = -1 THEN	RETURN
TriggerEvent('ue_prenew')

IF	Message.ReturnValue	=	-1	THEN	RETURN
IF dw_1.DeletedCount()+dw_1.ModifiedCount() > 0 AND ib_modified THEN
  li_op = f_confirmax3('Confirmar',"¿Desea guardar los cambios realizados?",'&Si','&No','&Cancelar',1 )

  CHOOSE CASE li_op
    CASE 1
		TriggerEvent("ue_save")
    CASE 3
			RETURN
  END CHOOSE
END IF

Title = istr_par.s_title + ' (Nuevo)'
istr_par.s_accion = 'new'


CHOOSE CASE istr_par.l[1]
		CASE 2
		  dw_1.dataobject = 'd_pea_escuela_fform'
		CASE 3
		  dw_1.dataobject = 'd_pea_carreras_fform'
		CASE ELSE
			dw_1.dataobject = 'd_pea_facultades_fform'
END CHOOSE

dw_1.settransobject(itrans_obj)
dw_1.SetRedraw ( FALSE )
dw_1.Reset()
dw_1.scrolltorow(dw_1.InsertRow(0))
dw_1.Setfocus()
dw_1.SetRedraw ( TRUE )

CHOOSE CASE istr_par.l[1]
		CASE 0,1
			SELECT max(PLA_FACULTAD.COD_FACULTAD)
			INTO  :ls_facultad
			FROM  sga.PLA_FACULTAD
			WHERE PLA_FACULTAD.COD_FACULTAD < '99'
			USING gtrans_sig;
```

```
                    if isnull(ls_facultad) or ls_facultad = '' then
                            ls_facultad = '01'
                    else
                            ls_facultad = right('0' + string(integer(ls_facultad) + 1), 1)
                    end if
                    dw_1.object.cod_facultad[dw_1.getrow()] = ls_facultad
                    dw_1.object.fec_creacion[dw_1.getrow()] = gdt_fecha

            CASE 2
                    setnull(ls_null)
                    if istr_par.i[1] = 1 then
                            ls_facultad = istr_par.s[1]
                            dw_1.object.num_nivel[dw_1.getrow()] = 1
                            dw_1.object.cod_escuela_sup[dw_1.getrow()] = ls_null
                    else
                            ls_facultad = trim(mid(istr_par.s[1],1,gi_len_cod_facultad))
                            dw_1.object.num_nivel[dw_1.getrow()] = 1
                            dw_1.object.cod_escuela_sup[dw_1.getrow()] = ls_null  //trim(mid(istr_par.s[1],
gi_len_cod_facultad + 1, gi_len_cod_carrera))

                    end if
                    dw_1.object.cod_facultad[dw_1.getrow()] = ls_facultad
                    dw_1.object.fec_creacion[dw_1.getrow()] = gdt_fecha

            CASE 3
                    setnull(ls_null)
                    if istr_par.i[1] = 1 then
                            ls_facultad = istr_par.s[1]
                            dw_1.object.num_nivel[dw_1.getrow()] = 1
                            dw_1.object.cod_carrera_sup[dw_1.getrow()] = ls_null
                    else
                            ll_escuela = len(istr_par.s[1])
                            ls_facultad = trim(mid(istr_par.s[1],1,gi_len_cod_facultad))
                            dw_1.object.num_nivel[dw_1.getrow()] = 1
                            dw_1.object.cod_carrera_sup[dw_1.getrow()] = trim(mid(istr_par.s[1],
gi_len_cod_facultad + 1, ll_escuela - (gi_len_cod_facultad ) ))

                    end if
                    dw_1.object.cod_facultad[dw_1.getrow()] = ls_facultad
                    dw_1.object.fec_creacion[dw_1.getrow()] = gdt_fecha
            END CHOOSE

dw_1.object.cod_usuario[1] = gs_c_usuario
dw_1.object.fec_registro[1] = gdt_fecha
dw_1.object.tip_operacion[1] = 'I'
```

```
Pointer  oldpointer

IF dw_1.AcceptText() = -1 THEN RETURN

This.TriggerEvent ( "ue_presave" )

IF message.returnvalue = -1 THEN
        messagebox("Aviso", "El codigo ya existe")
        RETURN
END IF

IF message.returnvalue = -2 THEN
        messagebox("Aviso", "Ingrese Nombre de la Escuela ")
        RETURN
END IF

IF message.returnvalue = -3 THEN
        messagebox("Aviso", "Ingrese Nombre de la Facultad ")
        RETURN
```

```
END IF

IF dw_1.deletedCount() + dw_1.modifiedcount() > 0 AND ib_modified THEN
oldpointer = SetPointer (HourGlass!)

//          Eduardo

          IF        dw_1.Update() = 1           THEN
                    //          Invocar a un evento donde se actualizan otras tablas
                    This.TriggerEvent ( "ue_postupdate" )
                    IF message.returnvalue = -1 THEN
                              ROLLBACK USING itrans_obj;
                      SetPointer (oldpointer)
                      gw_inicio.SetMicroHelp ( "Los cambios realizados no se actualizaron..." )
                              RETURN
                    ELSE
                              COMMIT  USING itrans_obj;
                    END IF
          ELSE
                    ROLLBACK USING itrans_obj;
            SetPointer (oldpointer)
            gw_inicio.SetMicroHelp ( "Los cambios realizados no se actualizaron..." )
                    RETURN
          END IF
  ib_modified = FALSE
  gw_inicio.SetMicroHelp ( "Grabación Satisfactoria..." )
          istr_par.s_accion = 'fileopen'

  this.TriggerEvent ( "ue_postsave" )
  SetPointer (oldpointer)

END IF
```

Código fuente gestión tipo condición docente

```
string ls_carrera
//          Para implementar en el hijo
          this.x = 0
          this.y = 0
          ii_num_tabs = 4  //numero de hojas del tab
//          Inicializa las variables para que se recupere una sola vez
          fw_ini_array_boolean(ib_modi,FALSE)
          fw_ini_array_boolean(ib_recu,FALSE)

//          seteada en el evento ue_transaction
          fw_settransObject( tab_1.tabpage_1.dw_1 )
          fw_settransObject( tab_1.tabpage_2.dw_2 )
          fw_settransObject( tab_1.tabpage_3.dw_3 )
          fw_settransObject( tab_1.tabpage_2.dw_4 )
          fw_settransObject( tab_1.tabpage_2.dw_5 )
          fw_settransObject( tab_1.tabpage_4.dw_6 )

          tab_1.tabpage_2.dw_2.getchild("cod_curso", dwc01)
          dwc01.settransobject(itrans_obj)
          if dwc01.retrieve(ls_carrera) < 1 then dwc01.insertrow(0)
          is_sql = tab_1.tabpage_1.dw_1.GetSQLSelect()
          tab_1.tabpage_2.dw_5.getchild("cod_curso", dwc02)
          dwc02.settransobject(itrans_obj)
          if dwc02.retrieve(ls_carrera) < 1 then dwc02.insertrow(0)

          CHOOSE CASE istr_par.s_accion
                    CASE "new"
```

```
                        This.TriggerEvent( "ue_new" )
                CASE "fileopen", "consulta"
                        This.TriggerEvent("ue_fileopen")
                CASE ELSE
        f_mensaje (this.title,"Opción no válida en evento open.")
            END CHOOSE
```

```
CHOOSE CASE ii_tab
        CASE 2
                tab_1.tabpage_2.dw_4.accepttext()
                if tab_1.tabpage_2.dw_4.update() < 1 then
                        rollback using itrans_obj;
                        messagebox("Aviso", "No se puede actualizar
PRG_DISPONIBILIDAD_HORARIO_")
                        return
                end if
                tab_1.tabpage_2.dw_5.accepttext()
                if tab_1.tabpage_2.dw_5.update() < 1 then
                        rollback using itrans_obj;
                        messagebox("Aviso", "No se puede actualizar PRG_DESIDERATA")
                        return
                else
                        commit using itrans_obj;
                        messagebox("Aviso", "Datos Grabados Correctamente")
                end if
//      CASE 3
//              tab_1.tabpage_3.dw_3.accepttext()
//              if tab_1.tabpage_3.dw_3.update() < 1 then
//                      rollback using itrans_obj;
//                      messagebox("Aviso", "No se puede actualizar PRG_FACULTAD_DOCENTE")
//                      return
//              else
//                      commit using itrans_obj;
//                      messagebox("Aviso", "Datos Grabados Correctamente")
//              end if
        case 3
                tab_1.tabpage_4.dw_6.accepttext()
                if tab_1.tabpage_4.dw_6.update() < 1 then
                        rollback using itrans_obj;
                        messagebox("Aviso", "No se puede actualizar PRG_FACULTAD_DOCENTE")
                        return
                else
                        commit using itrans_obj;
                        messagebox("Aviso", "Datos Grabados Correctamente")
                end if
END CHOOSE
```

Código fuente gestión tipo categoría docente

```
string ls_docente, ls_secuencial, ls_carrera

if isnull(currentrow) or currentrow < 1 then return

selectrow(0, false)
selectrow(currentrow, true)

scrolltorow(currentrow)
ls_docente = object.cod_docente[currentrow]
ls_secuencial = object.cod_sec_desiderata[currentrow]
ls_carrera = object.cod_carrera[currentrow]

tab_1.tabpage_2.dw_5.settransobject(itrans_obj)
```

```
dwc02.settransobject(itrans_obj)
if dwc02.retrieve(ls_carrera) < 1 then dwc02.insertrow(0)
if tab_1.tabpage_2.dw_5.retrieve(ls_docente, ls_secuencial) < 1 then tab_1.tabpage_2.dw_5.insertrow(0)
integer li_op
choose case ii_tab
	case 2
		li_op = MessageBox("Aviso", "Esta seguro de borrar el registro(S/N)", Exclamation!,
OKCancel!, 2)
		IF li_op = 1 THEN	tab_1.tabpage_2.dw_2.deleterow(0) // Process OK.
	case 3
		li_op = MessageBox("Aviso", "Esta seguro de borrar el registro(S/N)", Exclamation!,
OKCancel!, 2)
		IF li_op = 1 THEN	tab_1.tabpage_3.dw_3.deleterow(0) // Process OK.
end choose
```

CÓDIGO FUENTE SPRINT 3

Código Fuente Plan de Estudio

```
string ls_secuencial, ls_curso, ls_carrera, ls_ciclo

choose case ii_tab
	case 1, 2

		if isnull(is_cod_carrera) or is_cod_carrera ='' then
			Messagebox("Aviso.","Seleccione la Escuela.")
			return
		end if
		tab_1.tabpage_1.dw_1.scrolltorow(tab_1.tabpage_1.dw_1.rowcount())
		tab_1.selectedtab = 2
		tab_1.tabpage_2.dw_2.reset()
		tab_1.tabpage_2.dw_2.scrolltorow(tab_1.tabpage_2.dw_2.insertrow(0))
		tab_1.tabpage_2.dw_2.object.cod_carrera[tab_1.tabpage_2.dw_2.getrow()] = is_cod_carrera
//		tab_1.tabpage_2.dw_2.object.cod_sec_plan[tab_1.tabpage_2.dw_2.getrow()] =
ls_secuencial
		tab_1.tabpage_2.dw_2.object.fec_registro[tab_1.tabpage_2.dw_2.getrow()] = gdt_fecha
		tab_1.tabpage_2.dw_2.object.cod_usuario[tab_1.tabpage_2.dw_2.getrow()] = gs_c_usuario
		tab_1.tabpage_2.dw_2.object.tip_operacion[tab_1.tabpage_2.dw_2.getrow()] = 'I'
		tab_1.tabpage_3.dw_3.reset()
		tab_1.tabpage_4.dw_4.reset()
		tab_1.tabpage_5.dw_5.reset()
	case 3, 4

		if tab_1.tabpage_2.dw_2.rowcount() < 1 then return
		dwc01.settransobject(itrans_obj)
		if dwc01.retrieve(is_cod_facultad) < 1 then dwc01.insertrow(0)

		tab_1.tabpage_4.dw_4.scrolltorow(tab_1.tabpage_4.dw_4.insertrow(0))
		tab_1.tabpage_3.dw_3.scrolltorow(tab_1.tabpage_3.dw_3.rowcount())

		tab_1.selectedtab = 4
		ls_secuencial =
tab_1.tabpage_2.dw_2.object.cod_sec_plan[tab_1.tabpage_2.dw_2.getrow()]
		tab_1.tabpage_4.dw_4.object.fec_registro[tab_1.tabpage_4.dw_4.getrow()] = gdt_fecha
		tab_1.tabpage_4.dw_4.object.cod_sec_plan[tab_1.tabpage_4.dw_4.getrow()] =
ls_secuencial
		tab_1.tabpage_4.dw_4.object.cod_carrera[tab_1.tabpage_4.dw_4.getrow()] = is_cod_carrera
		tab_1.tabpage_4.dw_4.object.cod_usuario[tab_1.tabpage_4.dw_4.getrow()] = gs_c_usuario
		tab_1.tabpage_4.dw_4.object.tip_operacion[tab_1.tabpage_4.dw_4.getrow()] = 'I'
```

```
                tab_1.tabpage_4.dw_4.object.cod_especialidad[tab_1.tabpage_4.dw_4.getrow()] =
is_cod_carrera

        tab_1.tabpage_4.dw_4.object.pla_d_plan_estudio_cod_facultad[tab_1.tabpage_4.dw_4.getrow()] =
mid(is_cod_carrera,1,1)
                tab_1.tabpage_4.dw_4.Setfocus()
                tab_1.tabpage_5.dw_5.reset()
        case 5
                if tab_1.tabpage_1.dw_1.rowcount() < 1 then return
                if tab_1.tabpage_3.dw_3.rowcount() < 1 then return

                ls_secuencial =
trim(tab_1.tabpage_1.dw_1.object.cod_sec_plan[tab_1.tabpage_1.dw_1.getrow()])
                ls_carrera =
trim(tab_1.tabpage_1.dw_1.object.cod_carrera[tab_1.tabpage_1.dw_1.getrow()])

                if dwc02.retrieve(ls_carrera, ls_secuencial) < 1 then dwc02.insertrow(0)
                tab_1.tabpage_5.dw_5.scrolltorow(tab_1.tabpage_5.dw_5.insertrow(0))
                tab_1.selectedtab = 5

                tab_1.tabpage_5.dw_5.object.cod_sec_plan[tab_1.tabpage_5.dw_5.getrow()] =
ls_secuencial
                ls_curso = tab_1.tabpage_3.dw_3.object.cod_curso[tab_1.tabpage_3.dw_3.getrow()]
                tab_1.tabpage_5.dw_5.object.cod_curso[tab_1.tabpage_5.dw_5.getrow()] = ls_curso
                tab_1.tabpage_5.dw_5.object.cod_carrera[tab_1.tabpage_5.dw_5.getrow()] = ls_carrera
        case 6
                if tab_1.tabpage_3.dw_3.rowcount() < 1 then return
                ls_secuencial =
trim(tab_1.tabpage_3.dw_3.object.cod_sec_plan[tab_1.tabpage_3.dw_3.getrow()])
                ls_carrera =
trim(tab_1.tabpage_3.dw_3.object.cod_carrera[tab_1.tabpage_3.dw_3.getrow()])
                ls_ciclo =
tab_1.tabpage_3.dw_3.object.pla_d_plan_estudio_cod_ciclo[tab_1.tabpage_3.dw_3.getrow()]

                dwc06.settransobject(itrans_obj)
                if dwc06.retrieve(ls_carrera, ls_secuencial) < 1 then dwc06.insertrow(0)

                dwc07.settransobject(itrans_obj)
                if dwc07.retrieve(ls_carrera, ls_secuencial, ls_ciclo) < 1 then dwc07.insertrow(0)

                tab_1.tabpage_6.dw_7.scrolltorow(tab_1.tabpage_6.dw_7.insertrow(0))
                tab_1.tabpage_6.dw_7.object.cod_sec_plan[tab_1.tabpage_6.dw_7.getrow()] =
ls_secuencial
                tab_1.tabpage_6.dw_7.object.cod_carrera[tab_1.tabpage_6.dw_7.getrow()] = ls_carrera
                tab_1.tabpage_6.dw_7.object.cod_ciclo[tab_1.tabpage_6.dw_7.getrow()] = ls_ciclo
                tab_1.tabpage_6.dw_7.object.cod_curso[tab_1.tabpage_6.dw_7.getrow()] =
tab_1.tabpage_3.dw_3.object.cod_curso[tab_1.tabpage_3.dw_3.getrow()]
end choose
```

Código Fuente Tipo Curso

```
string ls_data, ls_plan
long ll_i , ll_return
date ld_fec_inicio, ld_fec_termino
datetime ldt_null
decimal ldc_obligatorio, ldc_electivo, ldc_credito, ldc_hte, ldc_hp, ldc_hl, ldc_hta, ldc_ho, ldc_total
string  ls_cod_sec_plan , &
      ls_cod_curso , &
                  ls_cod_curadi , &
                  ls_cod_carrera

if istr_par.s_accion = 'delete' then
        tab_1.tabpage_5.dw_5.accepttext()
        if tab_1.tabpage_5.dw_5.update() < 0 then
```

```
                rollback using itrans_obj;
                messagebox("Aviso", "No se puede actualizar PLA_PRE_REQUISITO")
                return
        end if

        tab_1.tabpage_3.dw_3.accepttext()
        if tab_1.tabpage_3.dw_3.update() < 0 then
                rollback using itrans_obj;
                messagebox("Aviso", "No se puede actualizar PLA_D_PLAN_ESTUDIO")
                return
        end if

        tab_1.tabpage_2.dw_2.accepttext()
        if tab_1.tabpage_2.dw_2.update() < 0 then
                rollback using itrans_obj;
                messagebox("Aviso", "No se puede actualizar PLA_PLAN_ESTUDIO")
                return
        end if
        if tab_1.tabpage_6.dw_7.update() < 0 then
                rollback using itrans_obj;
                messagebox("Aviso", "No se puede actualizar PLA_PLAN_ESTUDIO")
                return
        end if

        commit using sqlca;

        if sqlca.sqlcode <> 0 then
                rollback using sqlca;
                messagebox("Aviso", "No se puede actualizar PLA_PLAN_ESTUDIO")
                return
        end if
else
        choose case ii_tab
                case 1, 2
                        tab_1.tabpage_2.dw_2.accepttext()
                        /*Validacion*/
                        ld_fec_inicio =
date(tab_1.tabpage_2.dw_2.object.fec_inicio_plan[tab_1.tabpage_2.dw_2.getrow()])
                        ld_fec_termino =
date(tab_1.tabpage_2.dw_2.object.fec_termino_plan[tab_1.tabpage_2.dw_2.getrow()])
                        if isnull(ld_fec_termino) or isnull(ld_fec_inicio) then
                        else
                                if ld_fec_termino < ld_fec_inicio then
                                        messagebox("Aviso", "La Fecha de Inicio debe ser menor que la
fecha de Termino")

        tab_1.tabpage_2.dw_2.object.fec_inicio_plan[tab_1.tabpage_2.dw_2.getrow()] = ldt_null
                                        return
                                end if
                        end if

                        ldc_obligatorio =
tab_1.tabpage_2.dw_2.object.num_tot_cred_obligatorio[tab_1.tabpage_2.dw_2.getrow()]
                        ldc_electivo =
tab_1.tabpage_2.dw_2.object.num_tot_cre_electivo[tab_1.tabpage_2.dw_2.getrow()]
                        if ldc_obligatorio < ldc_electivo then
                                messagebox("Aviso", "El número de creditos obligatorio debe ser mayor
que el número de creditos electivos ")
                                return
                        end if
                        /*Fin de la Validadcion*/

                        if isnull(tab_1.tabpage_2.dw_2.object.cod_periodo[tab_1.tabpage_2.dw_2.getrow()])
then
                                messagebox("Aviso", "Seleccione el Semestre de Inicio")
                                return
                        end if
```

```
                        if tab_1.tabpage_2.dw_2.update() < 0 then
                                rollback using itrans_obj;
                                messagebox("Aviso", "No se puede actualizar PLA_PLAN_ESTUDIO.")
                                return
                        end if

                        ls_plan = tab_1.tabpage_2.dw_2.object.cod_sec_plan[1]
                        if ls_plan = '' or isnull(ls_plan) then
                        else
                                tab_1.tabpage_2.dw_6.accepttext()
                                for ll_i = 1 to tab_1.tabpage_2.dw_6.rowcount()
                                        tab_1.tabpage_2.dw_6.object.cod_sec_plan[ll_i] = ls_plan
                                next

                                tab_1.tabpage_2.dw_6.accepttext()
                                if tab_1.tabpage_2.dw_6.update() < 0 then
                                        rollback using itrans_obj;
                                        messagebox("Aviso", "No se puede actualizar
PLA_PLAN_ESTUDIO")
                                        return
                                end if
                        end if

                case 3, 4
                        tab_1.tabpage_4.dw_4.accepttext()
                        /*Validación*/
                        ///INGRESO DE CURSO
                        ldc_hte =
tab_1.tabpage_4.dw_4.object.num_hora_teoria[tab_1.tabpage_4.dw_4.getrow()]
                        if ldc_hte < 0 then
                                messagebox("Aviso", "La Hora de Teoria debe ser mayor de cero ")
                                return
                        else
                                if isnull(ldc_hte) then ldc_hte = 0.00
                                ldc_total = ldc_total + ldc_hte
                        end if
                        ldc_hp =
tab_1.tabpage_4.dw_4.object.num_hora_practica[tab_1.tabpage_4.dw_4.getrow()]
                        if ldc_hp < 0 then
                                messagebox("Aviso", "La Hora de Practica debe ser mayor de cero ")
                                return
                        else
                                if isnull(ldc_hp) then ldc_hp = 0.00
                                ldc_total = ldc_total + ldc_hp
                        end if
                        ldc_hl =
tab_1.tabpage_4.dw_4.object.num_hora_laboratorio[tab_1.tabpage_4.dw_4.getrow()]
                        if ldc_hl < 0 then
                                messagebox("Aviso", "La Hora de Laboratorio debe ser mayor de cero ")
                                return
                        else
                                if isnull(ldc_hl) then ldc_hl = 0.00
                                ldc_total = ldc_total + ldc_hl
                        end if
                        ldc_hta =
tab_1.tabpage_4.dw_4.object.num_hora_taller[tab_1.tabpage_4.dw_4.getrow()]
                        if ldc_hta < 0 then
                                messagebox("Aviso", "La Hora de Taller debe ser mayor de cero ")
                                return
                        else
                                if isnull(ldc_hta) then ldc_hta = 0.00
                                ldc_total = ldc_total + ldc_hta
                        end if
                        ldc_ho =
tab_1.tabpage_4.dw_4.object.num_hora_otro[tab_1.tabpage_4.dw_4.getrow()]
                        if ldc_ho < 0 then
                                messagebox("Aviso", "La Hora Otros debe ser mayor de cero ")
```

```
                                        return
                                else
                                        if isnull(ldc_ho) then ldc_ho = 0.00
                                        ldc_total = ldc_total + ldc_ho
                                end if
                                ldc_credito =
tab_1.tabpage_4.dw_4.object.num_credito[tab_1.tabpage_4.dw_4.getrow()]
                                if ldc_credito < 0 then
                                        messagebox("Aviso", "La Número de Creditos debe ser mayor de cero ")
                                        return
                                end if

//                              ldc_total = ldc_hte + ldc_hp + ldc_hl + ldc_hta + ldc_ho
                                if ldc_total < 1 or isnull(ldc_total) then
                                        messagebox("Aviso", "Debe Ingresarse la Hora de Teoria o la Hora de
Practica o la Hora de Laboratorio o la Hora de Taller")
                                        return
                                end if
                                /*Fin de la Validación*/
                                //**********************//
                                //VERIFICACION DE CODIGO DE CURSO
                                //**********************//
                        ls_cod_sec_plan =
tab_1.tabpage_4.dw_4.Object.cod_sec_plan[tab_1.tabpage_4.dw_4.getrow()]
        ls_cod_curso = tab_1.tabpage_4.dw_4.Object.cod_curso[tab_1.tabpage_4.dw_4.getrow()]
                          ls_cod_curadi = tab_1.tabpage_4.dw_4.object.cod_curadi[tab_1.tabpage_4.dw_4.getrow()]
                          ls_cod_carrera =
tab_1.tabpage_4.dw_4.object.cod_carrera[tab_1.tabpage_4.dw_4.getrow()]

//                              ll_return =
SQLCA.CRM_VERIFICA_COD_CURSO_FN(ls_cod_sec_plan,ls_cod_curso,ls_cod_curadi,mid(ls_cod_carrer
a,1,1))
//                              if ll_return > 0 then
//                                      messagebox("Aviso", "El codigo del Curso Tiene diferente Nombre, en otro
Plan de Estudio de la UNIVERSIDAD.. No se ha grabado")
//                                      return
//                              end if


                                //**********************//
                                if tab_1.tabpage_4.dw_4.update() < 0 then
                                        rollback using itrans_obj;
                                        messagebox("Aviso", "No se puede actualizar PLA_D_PLAN_ESTUDIO")
                                        return
                                end if
                        case 5
                                tab_1.tabpage_5.dw_5.accepttext()
                                if tab_1.tabpage_5.dw_5.update() < 0 then
                                        rollback using itrans_obj;
                                        messagebox("Aviso", "No se puede actualizar PLA_D_PLAN_ESTUDIO")
                                        return
                                end if
                        case 6
                                tab_1.tabpage_6.dw_7.accepttext()
                                if tab_1.tabpage_6.dw_7.update() < 0 then
                                        rollback using itrans_obj;
                                        messagebox("Aviso", "No se puede actualizar PLA_D_PLAN_ESTUDIO")
                                        return
                                end if
                end choose
end if
commit using itrans_obj;
tab_1.tabpage_4.dw_4.modify("cod_curso.protect = 1")
messagebox("Aviso", "Datos Grabados Correctamente")
```

Código Fuente Tipo Condición de Curso

```
integer li_op, li_i, li_j

choose case ii_tab
	case 1, 2
		li_op = f_confirmax3('Confirmar',"¿Desea Borrar el Plan de Estudio?",'&Si','&No','&Cancel',1 )
		CHOOSE CASE li_op
			CASE 1                                  // Cerrar y grabar
				for li_i = 1 to tab_1.tabpage_5.dw_5.rowcount()
					tab_1.tabpage_5.dw_5.deleterow(0)
				next
				for li_j = 1 to tab_1.tabpage_3.dw_3.rowcount()
					tab_1.tabpage_3.dw_3.deleterow(0)
				next

				delete from sga.pla_plan_estudio
				where cod_carrera  =:is_cod_carrera
				and     cod_sec_plan = :is_cod_plan
				using sqlca;
			   tab_1.tabpage_1.dw_1.deleterow(0)

				ib_modi[2] = TRUE
				ib_modified = TRUE
				istr_par.s_accion = 'delete'
			CASE    ELSE
		// Cancel
				return 1
		END CHOOSE
	case 3, 4
		li_op = f_confirmax3('Confirmar',"¿Desea Borrar el Curso?",'&Si','&No','&Cancel',1 )
		CHOOSE CASE li_op
			CASE 1                                  // Cerrar y grabar
				//verifica si tiene equivalencia
				string ls_carrera, ls_plan, ls_curso
				long ll_count1, ll_count2

				ls_carrera =
tab_1.tabpage_3.dw_3.object.cod_carrera[tab_1.tabpage_3.dw_3.getrow()]
				ls_plan =
tab_1.tabpage_3.dw_3.object.cod_sec_plan[tab_1.tabpage_3.dw_3.getrow()]
				ls_curso =
tab_1.tabpage_3.dw_3.object.cod_curso[tab_1.tabpage_3.dw_3.getrow()]

				SELECT COUNT(*)
				INTO :ll_count1
				FROM sga.PLA_PLAN_EQUIVALENCIA
				WHERE COD_CARRERA = :ls_carrera and
					COD_SEC_PLAN = :ls_plan and
					COD_CURSO = :ls_curso
				USING itrans_obj;

				if ll_count1 > 0 then
					messagebox("Aviso", "El curso tiene EQUIVALENCIA con otro
curso de otro PLAN")
					return
				end if

				SELECT COUNT(*)
				INTO :ll_count2
				FROM sga.PLA_PLAN_EQUIVALENCIA
				WHERE COD_CARRERA = :ls_carrera and
					COD_SEC_PLAN_EQUIVALENTE = :ls_plan and
					COD_CURSO_EQUIVALENTE = :ls_curso
				USING itrans_obj;

				if ll_count2 > 0 then
```

```
                                        messagebox("Aviso", "El curso es EQUIVALENCIA de otro
curso")
                                        return
                                end if
                                //****
                                for li_i = 1 to tab_1.tabpage_5.dw_5.rowcount()
                                        tab_1.tabpage_5.dw_5.deleterow(0)
                                next
                                        tab_1.tabpage_3.dw_3.deleterow(0)
                          ib_modi[2] = TRUE
                          ib_modified = TRUE
                          istr_par.s_accion = 'delete'
                        CASE    ELSE
                // Cancel
                          return 1
                END CHOOSE
        case 5
                li_op = f_confirmax3('Confirmar',"¿Desea Borrar el Pre -   Requisito ?",'&Si','&No','&Cancel',1
)
                CHOOSE CASE li_op
                        CASE 1                                  // Cerrar y grabar
                                tab_1.tabpage_5.dw_5.deleterow(0)
                          ib_modi[2] = TRUE
                          ib_modified = TRUE
                          istr_par.s_accion = 'delete'
                        CASE    ELSE
                // Cancel
                          return 1
                END CHOOSE
        case 6
                li_op = f_confirmax3('Confirmar',"¿Desea Borrar el Paralelo.. ?",'&Si','&No','&Cancel',1 )
                CHOOSE CASE li_op
                        CASE 1                                  // Cerrar y grabar
                                tab_1.tabpage_6.dw_7.deleterow(0)
                          ib_modi[2] = TRUE
                          ib_modified = TRUE
                          istr_par.s_accion = 'delete'
                        CASE    ELSE
                // Cancel
                          return 1
                END CHOOSE

end choose
```

Código Fuente Curso

```
long ll_count, ll_count1

string ls_curso, ls_max, ls_carrera, ls_facultad, ls_txtcurso, ls_txt_abreviatura

tab_1.tabpage_2.dw_2.accepttext()

//          case 'COD_CURSO'

ls_facultad = tab_1.tabpage_2.dw_2.object.cod_facultad[tab_1.tabpage_2.dw_2.getrow()]

if ls_facultad = '' or isnull(ls_facultad) then
        messagebox("Aviso", "Falta seleccionar la facultad")
        return
end if

ls_txt_abreviatura = tab_1.tabpage_2.dw_2.object.txt_abreviatura[tab_1.tabpage_2.dw_2.getrow()]
if ls_txt_abreviatura = '' or isnull(ls_txt_abreviatura) then
        messagebox("Aviso", "Falta ingresar la abreviatura del nombre del curso")
```

```
		return
end if

ls_curso = tab_1.tabpage_2.dw_2.object.cod_curso[tab_1.tabpage_2.dw_2.getrow()]

if ls_curso = '' or isnull(ls_curso) then

		ls_txtcurso = trim(tab_1.tabpage_2.dw_2.object.txt_curso[tab_1.tabpage_2.dw_2.getrow()])
		if ls_txtcurso = '' or isnull(ls_txtcurso) then
			messagebox("Aviso", "Falta ingresar el nombre del curso")
			return
		else
			 SELECT count(*)
		    INTO :ll_count1
		    FROM sga.PLA_CURSO
		    WHERE ( COD_FACULTAD = :ls_facultad ) AND
		          ( TXT_CURSO = :ls_txtcurso )
			 USING itrans_obj;

			 if ll_count1 > 0 then
				messagebox("Aviso", "Nombre del curso Duplicado")
				return
			 end if
		end if

		SELECT max(COD_CURSO)
	INTO :ls_max
	FROM sga.PLA_CURSO
		USING itrans_obj;

		if ls_max = '' or isnull(ls_max) then
			ls_curso = '0000000001'
		else
			ls_curso = right('0000000000' + string(integer(ls_max) + 1), 10)
		end if
		tab_1.tabpage_2.dw_2.object.cod_curso[tab_1.tabpage_2.dw_2.getrow()]	= ls_curso
end if

tab_1.tabpage_2.dw_2.accepttext()

if tab_1.tabpage_1.dw_1.update() < 0 then
		rollback using itrans_obj;
		messagebox("Aviso", "No se puede actualizar PLA_CURSO")
		return
else
		commit using itrans_obj;

  long ll_step, ll_max
		tab_1.tabpage_2.hpb_1.visible = true
		SetPointer ( HourGlass! )

		ll_max = tab_1.tabpage_2.hpb_1.MaxPosition

		For ll_step = 1 to ll_max
			Yield ( )
			tab_1.tabpage_2.hpb_1.StepIt ( )
		Next
		tab_1.tabpage_2.hpb_1.visible = false
		messagebox("Aviso", "Datos Grabados Correctamente")

		istr_par.s_accion = 'fileopen'
end if
```

Código Fuente Sistema de Evaluación

```
integer li_Cnt

this.x     =          0
this.y     =          0


IF IsValid (Message.powerobjectparm) THEN
          IF message.powerobjectparm.ClassName() = "str_par" THEN
                    istr_par = message.powerobjectparm
                    if istr_par.s_accion = 'consulta' then this.changemenu(m_grid_consulta)
          END IF
END IF

istr_par.po_caller =          this
istr_par.s_title              =          this.Title
istr_par.tv                                        =          tv_1
istr_par.dw                                        =          dw_1

TriggerEvent('ue_transaction')

IF          Not IsValid(itrans_obj)      THEN    f_mensaje(this.title,'deben configurar la transaction de su módulo,
evento ue_transaction')

IF dw_1.SetTransObject(itrans_obj) = -1 THEN
  f_mensaje(this.title,"Error al conectar dw_1")
ELSE
          dw_1.Reset()
END IF

SetPointer(HourGlass!)

// Captura los objetos TreeView y datawindow para poder modificar el tamaño
idrg_Vertical[1] = tv_1
idrg_Vertical[2] = dw_1
// Captura el color de fondo de ventana para hacer invisible el color de barra
il_HiddenColor = This.BackColor
st_Vertical.BackColor = il_HiddenColor
// Cambia de tamaño la barra
fw_ResizeBars()
// Cambia de tamaño los paneles (TreeView y ListView)
fw_ResizePanels()

//Parámetros a configurar:
ii_niveles = 1 //Número de niveles del TreeView
is_raiz    = "Sistema de Evaluación" //Cadena conteniendo el nombre de la raiz del arbol
ii_picture_select =  1//Numero del icono que sera usado cuando se selecciona un item del arbol

// Crea DataStores para almacenar los datos del explorer
//ids_Data[1] = Create DataStore
//ids_Data[1].DataObject = "ds_facultad"

ids_Data[1] = Create DataStore
ids_Data[1].DataObject = "ds_sistema_evaluacion"

// Realiza la conexion con el dbms
For li_Cnt = 1 To ii_niveles
          ids_Data[li_Cnt].SetTransObject(itrans_obj)
Next


// Invocar desde la ventana heredada al evento ue_populate
Post Event ue_populate()
```

```
string     ls_cod_item, ls_codigo, ls_facultad, ls_area
long                ll_len
```

```
Pointer oldpointer
Integer  li_op

IF dw_1.AcceptText() = -1 THEN    RETURN
TriggerEvent('ue_prenew')

IF        Message.ReturnValue      =         -1         THEN    RETURN
IF dw_1.DeletedCount()+dw_1.ModifiedCount() > 0 AND ib_modified THEN
   li_op = f_confirmax3('Confirmar',"¿Desea guardar los cambios realizados?",'&Si','&No','&Cancelar',1 )

  CHOOSE CASE li_op
    CASE 1
                  TriggerEvent("ue_save")
    CASE 3
                        RETURN
  END CHOOSE
END IF

Title = istr_par.s_title + ' (Nuevo)'
istr_par.s_accion = 'new'

dw_1.settransobject(itrans_obj)
dw_1.SetRedraw ( FALSE )
dw_1.Reset()
dw_1.scrolltorow(dw_1.InsertRow(0))
dw_1.Setfocus()
dw_1.SetRedraw ( TRUE )
```

CÓDIGO FUENTE SPRINT 4

Código Fuente Matricula Web

```
PROCEDURE PRC_FM_FICHAMATRICULA_UK(PCID_USUARIO VARCHAR, PCID_ALUMNO
VARCHAR, PC_CARRERA VARCHAR,PC_PERIODO VARCHAR )
  AS
    w_c_codigo                  varchar2(10);
    w_c_tip_alumno_fm           ADA_ALUMNOS.TIP_ALUMNO%TYPE;
    w_c_tip_alumno             ADA_ALUMNOS.TIP_ALUMNO%TYPE;
    w_c_tip_condicion_al
CRM_FICHA_MATRICULA.TIP_CONDICION_ALUMNO%TYPE;
    w_n_cred_aprobado          number;
    w_c_promo                  varchar2(1);
    w_n_prom_alumno_regular    number;
    w_n_prom_alumno_destacado number;
    w_n_prom_alu_destacado_m   number;
    w_i_count8                 number;
    w_n_prom_alumno            number;
    w_n_credito_aprobado       number;
    w_c_plan_ultimo            varchar2(2);
    w_n_cred_ampliacion        number;
    w_n_cred_matricula         number;
    w_n_cred_aprob_ciclo_ant   number;
    w_c_periodo                CRM_FICHA_MATRICULA.COD_PERIODO%TYPE;
    w_c_periodo_ult            CRM_FICHA_MATRICULA.COD_PERIODO%TYPE;
    w_c_periodo_penult         CRM_FICHA_MATRICULA.COD_PERIODO%TYPE;
    w_i_count6                 number;
    w_c_periodo_ant            CRM_FICHA_MATRICULA.COD_PERIODO%TYPE;
    w_i_count5                 number;
    w_i_count7                 number;
```

```
    w_i_promocion             integer;
    w_i_sum_credito           number;
    w_n_cred_prom             number;
    w_i_count2                number;
    w_c_minciclo
CRM_D_AVANCE_CURRICULAR.COD_CICLO_CURSO%TYPE;
    w_n_suma                  number;
    w_n_i                     number;
    w_c_plan                  CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN%TYPE;
    w_c_curso                 CRM_D_AVANCE_CURRICULAR.COD_CURSO%TYPE;
    w_c_ciclo
CRM_D_AVANCE_CURRICULAR.COD_CICLO_CURSO%TYPE;
    w_c_tipo_curso            CRM_D_AVANCE_CURRICULAR.TIP_CURSO%TYPE;
    w_c_estado                CRM_D_AVANCE_CURRICULAR.EST_CURSO%TYPE;
    w_i_veces                 CRM_D_AVANCE_CURRICULAR.NUM_VECES%TYPE;
    w_i_creditos
CRM_D_AVANCE_CURRICULAR.NUM_CREDITOS_CURSO%TYPE;
    WNUM_PROM_POND_PERIODO_ANT
CRM_AVANCE_CURRICULAR.NUM_PROM_POND_PERIODO_ANT%TYPE;
    w_n_detalle               number;
    w_b_aprobo                boolean;
    w_i_aprobado              integer;
    w_i_desaprobado           integer;
    w_i_desaprob_ult          integer;
    w_i_desaprob_penult       integer;
    w_i_tdesaprob_ult         integer;
    w_i_tdesaprob_penult      integer;
    w_i_count                 integer;
    w_i_count3                number;
    w_i_count4                number;
    w_i_mincred               NUMBER;
    w_i_count1                number;
    w_c_curso_requisito       varchar2(10);
    w_i_count10                number;
    w_i_udesaprob             integer;
    w_c_estado_requisito      varchar2(2);
    w_c_tipo_credito          varchar2(3);
    w_n_total_credito         number;
    w_n_credito_obligatorio   number;
    w_n_credito_electivo      number;
    w_c_ciclo_minimo          varchar2(2);
    w_n_max_credito             number;
    w_n_max_curso             number;
    w_n_prom_alumno_irregular   number;
    w_n_promm_alumno_irregular  number;
    w_n_prom_ultimo             number;
    w_n_prom_dos_ultimo         number;
    w_n_prom_penultimo          number;
    w_flag_desaprob_dosultimo   varchar2(1);
    WNUM_CRED_APROB_OBLIGATORIO NUMBER;
    WNUM_CRED_APROB_ELECTIVO    NUMBER;
    WNUM_CREDITO                NUMBER;
    WNUM_CICLO                  NUMBER;
    WNUM_CICLO_ELECTIVO         NUMBER;
    WNUM_CICLO_RELATIVO
crm_avance_curricular.num_ciclo_relativo%type;
    WNUM_VARIABLE               NUMBER;
    w_n_max_ciclos_prom         number;
    WCOD_CICLO                  PLA_D_PLAN_ESTUDIO.COD_CICLO%TYPE;
    WCOD_CICLO_RELATIVO         PLA_D_PLAN_ESTUDIO.COD_CICLO%TYPE;
    WCICLO_INC                  NUMBER;
```

```
    w_flg_reprogramacion
PLA_D_PLAN_ESTUDIO.FLG_REPROGRAMACION%TYPE;
    v_i_prom_pre_aprobado       integer;
    WCOD_CURSO_REQUISITO_PROM
PLA_PRE_REQUISITO.COD_CURSO_REQUISITO%type;
    w_c_estado_requisito_pro    varchar2(2);
    v_n_ciclo_promocion         integer;
    v_c_tip_otro_req            varchar2(2);
    v_i_count20                 number;
    wcod_curso_des                CRM_D_FICHA_MATRICULA.COD_CURSO%TYPE;
    wcod_sec_plan_des
CRM_D_FICHA_MATRICULA.COD_SEC_PLAN%TYPE;
  BEGIN
   /*Para cierre de Matricula*/
   /***************************/
   delete from crm_d_ficha_matricula
   where cod_carrera = PC_CARRERA AND
         COD_ALUMNO = PCID_ALUMNO and
         COD_CURSO NOT IN( SELECT COD_CURSO FROM
SGA.CRM_D_CURSO_MATRICULADO WHERE COD_ALUMNO =PCID_ALUMNO);
        -- est_curso = '02';
   COMMIT;

   FOR r_genera_ficha IN ( SELECT DISTINCT A.COD_ALUMNO, B.TIP_ALUMNO,
A.NUM_CRED_APROB_PLAN_ACTUAL,A.NUM_CICLO_RELATIVO,NUM_PROM_POND_PERIODO_A
NT
                           FROM ADA_ALUMNOS B,
                                CRM_AVANCE_CURRICULAR A
                           WHERE ( A.COD_ALUMNO = B.COD_ALUMNO ) and
                                 ( B.COD_ESPECIALIDAD = PC_CARRERA ) AND
                                   ( A.COD_CARRERA = PC_CARRERA ) AND
                                 ( A.COD_ALUMNO = PCID_ALUMNO ) AND
                                 ( B.EST_SITUACION = '01' ) AND
                                 ( B.TIP_CONDICION IN ('N', 'I') ) AND
                                 ( A.EST_AVANCE_CURRICULAR = '01' ) AND
                                 ( B.COD_ALUMNO  IN ( SELECT DISTINCT
COD_ALUMNO
                                                      FROM
CRM_FICHA_MATRICULA
                                                      WHERE
COD_CARRERA = PC_CARRERA  ) )
                                                       ) LOOP
          w_c_codigo := ltrim(rtrim(r_genera_ficha.cod_alumno));
          w_c_tip_alumno := ltrim(rtrim(r_genera_ficha.tip_alumno));
          w_c_tip_alumno_fm := ltrim(rtrim(r_genera_ficha.tip_alumno));
          w_n_cred_aprobado :=
to_number(r_genera_ficha.num_cred_aprob_plan_actual);
          wnum_ciclo_relativo := r_genera_ficha.num_ciclo_relativo;
          WNUM_PROM_POND_PERIODO_ANT  :=
to_number(r_genera_ficha.NUM_PROM_POND_PERIODO_ANT);

          IF LENGTH(wnum_ciclo_relativo)= 1 then
             WCOD_CICLO_RELATIVO := '0'||to_char(wnum_ciclo_relativo);
          else
             WCOD_CICLO_RELATIVO := to_char(wnum_ciclo_relativo);
          end if;


          w_c_promo := '0';
```

```
          w_n_prom_alumno_regular :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '02', PC_CARRERA));
          w_n_prom_alumno_destacado :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '24', PC_CARRERA));
          w_n_prom_alumno_irregular :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '17', PC_CARRERA));
          w_n_promm_alumno_irregular :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '11', PC_CARRERA));
          w_n_max_ciclos_prom := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'03', PC_CARRERA));
          v_n_ciclo_promocion := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'30', PC_CARRERA));
            w_n_max_curso := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'09', PC_CARRERA));
          ---------------------------------------------
          --verificar existencia de avance curricular
          ---------------------------------------------
          SELECT DISTINCT COUNT(*)
          INTO w_i_count8
          FROM CRM_D_AVANCE_CURRICULAR
          WHERE ( COD_ALUMNO = w_c_codigo ) AND
                ( COD_CARRERA = PC_CARRERA )  ;

          IF w_i_count8 > 0 THEN
            SELECT NUM_TOT_PROM_POND_GENERAL,
                   NVL(NUM_CRED_APROB_PLAN_ACTUAL, 0)
            INTO   w_n_prom_alumno,
                   w_n_credito_aprobado
            FROM   CRM_AVANCE_CURRICULAR
            WHERE ( COD_ALUMNO = w_c_codigo ) AND
                  ( COD_CARRERA = PC_CARRERA ) AND
                  ( EST_AVANCE_CURRICULAR = '01' )  ;

            SELECT DISTINCT COD_SEC_PLAN
            INTO   w_c_plan_ultimo
            FROM CRM_D_AVANCE_CURRICULAR
            WHERE ( COD_ALUMNO = w_c_codigo ) AND
                  ( COD_CARRERA = PC_CARRERA )   ;

          END IF;
          ---------------------------------------------
          -- fin deverificar existencia de avance curricular
          ---------------------------------------------

          --unu

          w_n_prom_dos_ultimo := WNUM_PROM_POND_PERIODO_ANT;
          --falta
          if w_n_prom_dos_ultimo >= 10.5 AND w_n_prom_dos_ultimo <=11
then
             w_n_prom_dos_ultimo := round(w_n_prom_dos_ultimo);
          end if;
          ---------------------------------------------
          --DESAPROBADO
          --Calculo del numero de cursos desaprobados       w_i_udesaprob
          ---------------------------------------------
              SELECT COUNT(COD_CURSO)
              INTO   w_i_udesaprob
              FROM CRM_D_AVANCE_CURRICULAR
              WHERE ( CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )
AND
```

```
                    ( CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo )
AND
                    ( CRM_D_AVANCE_CURRICULAR.NUM_VECES >= 2  ) AND
                    ( CRM_D_AVANCE_CURRICULAR.EST_CURSO = '02' )  ;
              ------------------------------
              --MAS DE TRES VECES DESAPROBADO
              --Verificar si los dos ultimo periodos esta desaprobado en
todos los cursos
              ------------------------------
              SELECT COUNT(COD_CURSO)
              INTO w_i_desaprob_ult
              FROM HIS.HIS_ACTA_D_NOTAS
              WHERE COD_CARRERA = PC_CARRERA and
                     COD_ALUMNO = w_c_codigo and
                     cod_periodo = w_c_periodo_ult and
                     EST_CURSO IN ('03', '04', '06', '07', '10') ;

              SELECT COUNT(COD_CURSO)
              INTO w_i_tdesaprob_ult
              FROM HIS.HIS_ACTA_D_NOTAS
              WHERE COD_CARRERA = PC_CARRERA and
                     COD_ALUMNO = w_c_codigo and
                     cod_periodo = w_c_periodo_ult ;

              SELECT COUNT(COD_CURSO)
              INTO w_i_desaprob_penult
              FROM HIS.HIS_ACTA_D_NOTAS
              WHERE COD_CARRERA = PC_CARRERA and
                     COD_ALUMNO = w_c_codigo and
                     cod_periodo = w_c_periodo_penult and
                     EST_CURSO IN ('03', '04', '06', '07', '10') ;

              SELECT COUNT(COD_CURSO)
              INTO w_i_tdesaprob_penult
              FROM HIS.HIS_ACTA_D_NOTAS
              WHERE COD_CARRERA = PC_CARRERA and
                     COD_ALUMNO = w_c_codigo and
                     cod_periodo = w_c_periodo_penult ;

              if ( w_i_desaprob_penult = w_i_tdesaprob_penult ) and (
w_i_desaprob_ult = w_i_tdesaprob_ult ) then
                w_flag_desaprob_dosultimo := '1';  ----SI
              else
                w_flag_desaprob_dosultimo := '0';  ----NO
              end if;
              ------------------------------------------------
              --FIN DE DESAPROBADO TODOS LOS CURSOS EN LOS DOS ULTIMOS
PERIODOS
              ------------------------------------------------
          ------------------------------
          --- CREDITO OBLIGATORIO Y ELECTIVO
          ------------------------------
          SELECT sum(nvl(CRM_D_AVANCE_CURRICULAR.NUM_CREDITOS_CURSO, 0))
          INTO w_n_credito_obligatorio
          FROM CRM_D_AVANCE_CURRICULAR, PLA_D_PLAN_ESTUDIO
          WHERE ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN ) and
             ( PLA_D_PLAN_ESTUDIO.COD_CURSO =
CRM_D_AVANCE_CURRICULAR.COD_CURSO ) and
             ( CRM_D_AVANCE_CURRICULAR.COD_CARRERA =
PLA_D_PLAN_ESTUDIO.COD_CARRERA ) and
```

```
             ( PLA_D_PLAN_ESTUDIO.TIP_CURSO = '02' ) AND
             ( CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
             ( CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan_ultimo )
AND
             ( CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA ) AND
             ( CRM_D_AVANCE_CURRICULAR.EST_CURSO in ('03', '04', '06',
'07') )   ;

          SELECT sum(nvl(CRM_D_AVANCE_CURRICULAR.NUM_CREDITOS_CURSO, 0))
          INTO w_n_credito_electivo
          FROM CRM_D_AVANCE_CURRICULAR, PLA_D_PLAN_ESTUDIO
          WHERE ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN ) and
             ( PLA_D_PLAN_ESTUDIO.COD_CURSO =
CRM_D_AVANCE_CURRICULAR.COD_CURSO ) and
             ( CRM_D_AVANCE_CURRICULAR.COD_CARRERA =
PLA_D_PLAN_ESTUDIO.COD_CARRERA ) and
             ( PLA_D_PLAN_ESTUDIO.TIP_CURSO <> '02' ) AND
             ( CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
             ( CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan_ultimo )
AND
             ( CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA ) AND
             ( CRM_D_AVANCE_CURRICULAR.EST_CURSO in ('03', '04', '06',
'07') )   ;
          -------------------------------
          --- FIN CREDITO OBLIGATORIO Y ELECTIVO
          -------------------------------
           w_i_udesaprob := 0;
          ----------------------------
          --CONDICION DE ALUMNO
          ----------------------------
          if w_i_udesaprob > 0 then  ----DESAPROBADO MAYOR DE TRES VECES
VER AL FINAL
             if w_flag_desaprob_dosultimo = '1' then  -- DESAPROBADO EN
TODOS LOS CURSOS
                w_c_tip_condicion_al := '02';
                w_n_cred_ampliacion := 0;
                w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);

                if w_n_cred_matricula = 0 then
                   w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                end if;

             else
                  if w_i_udesaprob = 1 then
                     w_c_tip_condicion_al := '02';
                     w_n_cred_ampliacion := 0;
                     --w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '28', PC_CARRERA));
                     w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                  elsif w_i_udesaprob > 1 then
                        w_c_tip_condicion_al := '02';
                        w_n_cred_ampliacion := 0;
                       -- w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '29', PC_CARRERA));
```

```
                          w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                        --1
                        if  w_n_cred_matricula = 0 then
                          --  w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '28', PC_CARRERA));
                            w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                        end if;
                        --2
                 end if;
             end if;

          else
           IF w_n_prom_alumno_regular <= w_n_prom_dos_ultimo and
w_n_prom_dos_ultimo < w_n_prom_alumno_destacado THEN
                --ALUMNO REGULAR
                --Ultimo semestre del ciclo anterior este todos los
cursos aprobados
                w_c_tip_condicion_al := '01';
                w_n_cred_ampliacion := 0;
                w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '14', PC_CARRERA));
                if w_n_cred_matricula = 0 then
                   w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                end if;

           ELSIF w_n_prom_dos_ultimo >= w_n_prom_alumno_destacado THEN
                --ALUMNO DESTACADO
                --Ultimo semestre del ciclo anterior este todos los
cursos aprobados
                w_c_tip_condicion_al := '03';
                w_n_cred_ampliacion :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '25', PC_CARRERA));
                w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '14', PC_CARRERA));
                if w_n_cred_matricula = 0 then
                   w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                end if;

           ELSIF w_n_prom_dos_ultimo < w_n_prom_alumno_irregular THEN
                --ALUMNO IRREGULAR
                w_c_tip_condicion_al := '02';
                w_n_cred_ampliacion := 0;
                w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '21', PC_CARRERA));
           ELSIF w_n_promm_alumno_irregular <= w_n_prom_dos_ultimo and
w_n_prom_dos_ultimo < w_n_prom_alumno_regular THEN
                --ALUMNO MAX REGULAR
                w_c_tip_condicion_al := '02';
                w_n_cred_ampliacion := 0;
                w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '12', PC_CARRERA));
           END IF;
          end if;
```

```
          -------------------------------
          -- PROMOCION
          -------------------------------

          SELECT count(*)
          INTO w_i_count7
          FROM PLA_PLAN_ESTUDIO
          WHERE ( PLA_PLAN_ESTUDIO.COD_SEC_PLAN = w_c_plan_ultimo ) AND
                ( PLA_PLAN_ESTUDIO.COD_CARRERA = PC_CARRERA ) and
                ( PLA_PLAN_ESTUDIO.NUM_TOT_CRED_OBLIGATORIO > 0 ) ;

          if w_i_count7 > 0 then
            --verificar el plan de estudio solo se considera oblogatorio
y electivo
            SELECT nvl(PLA_PLAN_ESTUDIO.NUM_TOT_CRED_OBLIGATORIO, 0) +
nvl(PLA_PLAN_ESTUDIO.NUM_TOT_CRE_ELECTIVO, 0)
            INTO w_i_sum_credito
            FROM PLA_PLAN_ESTUDIO
            WHERE ( PLA_PLAN_ESTUDIO.COD_SEC_PLAN = w_c_plan_ultimo ) AND
               ( PLA_PLAN_ESTUDIO.COD_CARRERA = PC_CARRERA )   ;
          else
              w_i_sum_credito := 0;
          end if;

          w_i_promocion := 0;

          wnum_ciclo :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CICLOS_PLAN(PC_CARRERA,w_c_plan_ultimo)
;

          IF wnum_ciclo - (v_n_ciclo_promocion-1) <= wnum_ciclo_relativo
then
              w_n_cred_prom := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'15', PC_CARRERA));
              IF w_n_cred_prom = 0 THEN

                  wnum_ciclo :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CICLOS_PLAN(PC_CARRERA,w_c_plan_ultimo)
;
                  if length(wnum_ciclo) = 1 then
                     wcod_ciclo := '0'||to_char(wnum_ciclo);
                  else
                     wcod_ciclo := to_char(wnum_ciclo);
                  end if;

                  IF v_n_ciclo_promocion = 0 then
                    w_n_cred_prom :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo);
                  ELSE
                    w_n_cred_prom :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo);
                    FOR I IN 1..v_n_ciclo_promocion - 1 LOOP
                      if length(wnum_ciclo - I)=1 then
                        wcod_ciclo := '0'||to_char(wnum_ciclo - I);
                      else
                        wcod_ciclo := to_char(wnum_ciclo - I);
                      end if;
```

```
                      w_n_cred_prom := w_n_cred_prom +
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo);
                    END LOOP;
                END IF ;
            END IF;

            if TO_NUMBER(w_i_sum_credito) - w_n_credito_aprobado <
w_n_cred_prom then
                w_i_promocion := 1;
                w_c_tip_condicion_al := '04';
                w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '15', PC_CARRERA));
                if w_n_cred_matricula = 0 then
                   w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
                end if;
                IF w_n_prom_dos_ultimo < w_n_prom_alumno_irregular THEN
                   w_n_cred_ampliacion := 0;
                   w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '21', PC_CARRERA));
                END IF;
            ELSE
                w_i_promocion := 0;
            end if;
         END IF;

           update  SGA.CRM_FICHA_MATRICULA
           set   NUM_MAX_CRED_AMPLIACION = w_n_cred_ampliacion,
                 NUM_MAX_CRED_MATRICULA = w_n_cred_matricula,
                 TIP_CONDICION_ALUMNO = w_c_tip_condicion_al,
                 NUM_PROM_POND_PERIODO_ANT = WNUM_PROM_POND_PERIODO_ANT,
                 NUM_MAX_CUR_MATRICULA =   w_n_max_curso
           where COD_ALUMNO = w_c_codigo
           and   cod_carrera= PC_CARRERA;
       ------------------------------------------------------
       ---Ingresando datos en el detalle de la Ficha de Matricula
       ------------------------------------------------------
          w_n_i := 1;

          FOR r_genera_d_ficha IN ( SELECT
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN,
                                    CRM_D_AVANCE_CURRICULAR.COD_CURSO,

CRM_D_AVANCE_CURRICULAR.COD_CICLO_CURSO,
                                    CRM_D_AVANCE_CURRICULAR.TIP_CURSO,
                                    CRM_D_AVANCE_CURRICULAR.EST_CURSO,
                                    CRM_D_AVANCE_CURRICULAR.NUM_VECES,

CRM_D_AVANCE_CURRICULAR.NUM_CREDITOS_CURSO
                              FROM CRM_D_AVANCE_CURRICULAR
                              WHERE ( CRM_D_AVANCE_CURRICULAR.COD_ALUMNO
= w_c_codigo ) AND
                                    ( CRM_D_AVANCE_CURRICULAR.COD_CARRERA
= PC_CARRERA ) AND
                                    ( CRM_D_AVANCE_CURRICULAR.EST_CURSO
in ('01', '02', '08', '09') ) AND
                                    ( CRM_D_AVANCE_CURRICULAR.COD_CURSO
IN ( SELECT A.COD_CURSO
```

```
FROM PLA_D_PLAN_ESTUDIO A

WHERE  ( A.TIP_CONDICION = '01' ) and

A.COD_SEC_PLAN = w_c_plan_ultimo AND

A.COD_CARRERA = PC_CARRERA   ) )
                                              ORDER BY
CRM_D_AVANCE_CURRICULAR.COD_CICLO_CURSO,

CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN,

CRM_D_AVANCE_CURRICULAR.COD_CURSO    ) LOOP

               w_c_plan := ltrim(rtrim(r_genera_d_ficha.cod_sec_plan));
               w_c_curso := ltrim(rtrim(r_genera_d_ficha.cod_curso));
               w_c_ciclo :=
ltrim(rtrim(r_genera_d_ficha.cod_ciclo_curso));
               w_c_tipo_curso :=
ltrim(rtrim(r_genera_d_ficha.tip_curso));
               w_c_estado := ltrim(rtrim(r_genera_d_ficha.est_curso));
               w_i_veces := r_genera_d_ficha.num_veces;
               w_i_creditos := r_genera_d_ficha.num_creditos_curso;
  --DBMS_OUTPUT.put_line('procesando');
  --DBMS_OUTPUT.put_line(w_c_curso);
               SELECT flg_reprogramacion
               INTO  w_flg_reprogramacion
               FROM  SGA.PLA_D_PLAN_ESTUDIO
               WHERE COD_SEC_PLAN = w_c_plan
               AND   COD_CURSO = w_c_curso ;

               if w_c_estado = '02' then
               ---------------
               --Desaprobados
               ---------------
                  SELECT COUNT(*)
                  INTO w_n_detalle
                  FROM CRM_D_FICHA_MATRICULA
                  WHERE COD_ALUMNO = w_c_codigo AND
                        COD_SEC_PLAN = w_c_plan AND
                        COD_CURSO = w_c_curso ;

                  IF w_n_detalle = 0 THEN
                     INSERT INTO CRM_D_FICHA_MATRICULA
                            ( COD_ALUMNO,
                           COD_SEC_PLAN,
                           COD_CURSO,
                           COD_CARRERA,
                           COD_CICLO_CURSO,
                           TIP_CURSO,
                           EST_CURSO,
                           NUM_VECES,
                           NUM_CREDITO_CURSO,
                           FLG_PROMOCION,
                           FLG_REPROGRAMACION )
                     VALUES ( w_c_codigo,
                           w_c_plan,
                           w_c_curso,
                           PC_CARRERA,
                            w_c_ciclo,
```

```
                            '01',
                            '02',
                            w_i_veces,
                            w_i_creditos,
                            '0',
                            w_flg_reprogramacion )  ;
                  END IF;
               elsif w_c_estado = '01' OR w_c_estado = '08' OR w_c_estado
= '09' then
                     -------------
                     --Pendientes
                     -------------
                     w_b_aprobo := false;
                     w_i_aprobado := 0;
                     w_i_desaprobado := 0;

                     SELECT COUNT(*)
                     INTO w_i_count
                     FROM ADA_ALUMNOS
                     WHERE COD_ALUMNO = w_c_codigo AND
                           TIP_ALUMNO = 'ING' AND
                           COD_MODALIDAD_INGRESO NOT IN ('04', '05',
'06') AND
                           EST_SITUACION = '01' ;

                     IF w_i_count = 1 THEN
                      -------------------
                      --Para Ingresantes
                      -------------------
                        IF w_c_ciclo = '01' THEN
                           SELECT COUNT(*)
                           INTO w_n_detalle
                           FROM CRM_D_FICHA_MATRICULA
                           WHERE COD_ALUMNO = w_c_codigo AND
                                   COD_SEC_PLAN = w_c_plan AND
                                   COD_CURSO = w_c_curso ;

                           IF w_n_detalle = 0 THEN
                             INSERT INTO CRM_D_FICHA_MATRICULA
                             ( COD_ALUMNO,
                               COD_SEC_PLAN,
                               COD_CURSO,
                               COD_CARRERA,
                               COD_CICLO_CURSO,
                               TIP_CURSO,
                               EST_CURSO,
                               NUM_VECES,
                               NUM_CREDITO_CURSO,
                               FLG_PROMOCION,
                               FLG_REPROGRAMACION )
                             VALUES ( w_c_codigo,
                                  w_c_plan,
                                  w_c_curso,
                                  PC_CARRERA,
                                  w_c_ciclo,
                                  '02',
                                  '02',
                                   w_i_veces,
                                   w_i_creditos,
                                   '0',
                                   w_flg_reprogramacion )  ;
```

```
                          END IF;
                        END IF;
                      ELSIF w_i_count < 1 THEN
                        SELECT TIP_OTRO_REQ
                        INTO   v_c_tip_otro_req
                        FROM  SGA.PLA_D_PLAN_ESTUDIO
                        WHERE COD_CURSO = w_c_curso
                        AND  COD_SEC_PLAN = w_c_plan ;

                        IF v_c_tip_otro_req = '02' THEN
                          SELECT COUNT(*)
                          INTO  v_i_count20
                          FROM  SGA.CRM_D_AVANCE_CURRICULAR
                          WHERE COD_ALUMNO = w_c_codigo
                          AND   COD_CARRERA = PC_CARRERA
                          AND   EST_CURSO IN('01','02','08','09')
                          AND    TO_NUMBER(COD_CICLO_CURSO) <
TO_NUMBER(w_c_ciclo)
                          AND   COD_CURSO NOT IN (SELECT COD_CURSO
                                       FROM  SGA.PLA_D_PLAN_ESTUDIO
                                       WHERE  COD_SEC_PLAN =
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN
                                       AND    COD_CARRERA = PC_CARRERA
                                       AND    TIP_CURSO <> '02') ;

                          IF  v_i_count20 = 0 THEN

                             ------------------------------------
                             --Verificamos el numero creditos minimo
permitidos
                             ------------------------------------
                             SELECT COUNT(*)
                             INTO w_i_count4
                             FROM PLA_D_PLAN_ESTUDIO
                             WHERE ( PLA_D_PLAN_ESTUDIO.COD_CARRERA =
PC_CARRERA ) AND
                                   ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
w_c_plan ) AND
                                   ( PLA_D_PLAN_ESTUDIO.COD_CURSO =
w_c_curso ) AND
                                   ( NUM_MIN_CREDITOS > 0 );
                                 IF w_i_count4 > 0 THEN
                                  SELECT NVL(NUM_MIN_CREDITOS, 0),
TIP_CREDITO_REQ
                                  INTO w_i_mincred, w_c_tipo_credito
                                  FROM   PLA_D_PLAN_ESTUDIO
                                  WHERE ( PLA_D_PLAN_ESTUDIO.COD_CARRERA =
PC_CARRERA ) AND
                                        ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN
= w_c_plan ) AND
                                        ( PLA_D_PLAN_ESTUDIO.COD_CURSO =
w_c_curso )   ;
                                 ELSE
                                  w_i_mincred := 0;
                                 END IF;
                                 IF w_i_mincred IS NULL OR w_i_mincred < 1
THEN
                                   ------------------------------------
                                   --Si el curso no tiene credito minimos
                                   ------------------------------------
```

```
                                    SELECT COUNT(*)
                                    INTO w_i_count1
                                    FROM PLA_PRE_REQUISITO
                                    WHERE ( PLA_PRE_REQUISITO.COD_SEC_PLAN
= w_c_plan ) AND
                                          ( PLA_PRE_REQUISITO.COD_CURSO =
w_c_curso );

                                    if w_i_count1 > 0 then
                                      FOR r_pre_requisito IN ( SELECT
PLA_PRE_REQUISITO.COD_CURSO_REQUISITO
                                                               FROM
PLA_PRE_REQUISITO
                                                               WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                                                     (
PLA_PRE_REQUISITO.COD_CURSO = w_c_curso )
) LOOP
                                                w_c_curso_requisito :=
ltrim(rtrim(r_pre_requisito.cod_curso_requisito));

                                                SELECT count(*)
                                                INTO w_i_count3
                                                FROM CRM_D_AVANCE_CURRICULAR
                                                WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                  (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                  (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                                  (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                                if w_i_count3 > 0 then
                                                   SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                                   INTO w_c_estado_requisito
                                                   FROM
CRM_D_AVANCE_CURRICULAR
                                                   WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                     (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                     (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                                     (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                                   IF w_c_estado_requisito =
'03' OR w_c_estado_requisito = '04' OR w_c_estado_requisito = '06' OR
w_c_estado_requisito = '07' OR w_c_estado_requisito = '10' THEN
/*APROBADO*/
                                                      w_i_aprobado := 1;
                                                   ELSE
                                                      w_i_desaprobado := 1;
                                                   END IF;
                                                end if;
                                    END LOOP;
                                    IF w_i_desaprobado = 0  THEN
                                        SELECT COUNT(*)
                                        INTO w_n_detalle
```

```
                                        FROM CRM_D_FICHA_MATRICULA
                                        WHERE COD_ALUMNO = w_c_codigo
AND
                                                  COD_SEC_PLAN =
w_c_plan AND
                                                  COD_CURSO =
w_c_curso ;

                                        IF w_n_detalle = 0 THEN
                                           INSERT INTO
CRM_D_FICHA_MATRICULA
                                         ( COD_ALUMNO,
                                           COD_SEC_PLAN,
                                             COD_CURSO,
                                             COD_CARRERA,
                                             COD_CICLO_CURSO,
                                             TIP_CURSO,
                                             EST_CURSO,
                                             NUM_VECES,
                                             NUM_CREDITO_CURSO,
                                             FLG_PROMOCION,
                                             FLG_REPROGRAMACION )
                                          VALUES ( w_c_codigo,
                                                w_c_plan,
                                                w_c_curso,
                                                PC_CARRERA,
                                                w_c_ciclo,
                                                '02',
                                                '02',
                                                w_i_veces,
                                                w_i_creditos,
                                                '0',
                                                w_flg_reprogramacion )
;
                                        END IF;
                                        COMMIT;
                                    ELSE
                                        IF w_i_promocion = 1 THEN
                                        -- verifica los niveles de
acptacion de cursos paralelos
                                            v_i_prom_pre_aprobado := 0;
                                           FOR r_pro_pre_requisito IN
( SELECT PLA_PRE_REQUISITO.COD_CURSO_REQUISITO

FROM PLA_PRE_REQUISITO

WHERE ( PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND

( PLA_PRE_REQUISITO.COD_CURSO = w_c_curso_requisito ) /*AND

( PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP

WCOD_CURSO_REQUISITO_PROM:= r_pro_pre_requisito.COD_CURSO_REQUISITO;
                                                  SELECT count(*)
                                                  INTO w_i_count10
                                                  FROM
CRM_D_AVANCE_CURRICULAR
                                                  WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                    (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
```

```
                                                    (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                    (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                                  if w_i_count10 > 0
then
                                                     SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                                     INTO
w_c_estado_requisito_pro
                                                     FROM
CRM_D_AVANCE_CURRICULAR
                                                     WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                           (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                           (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                           (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                                     IF
w_c_estado_requisito = '03' OR w_c_estado_requisito = '04' OR
w_c_estado_requisito = '06' OR w_c_estado_requisito = '07' OR
w_c_estado_requisito = '10' THEN /*APROBADO*/
                                                        w_i_aprobado :=
1;
                                                     ELSE

v_i_prom_pre_aprobado := v_i_prom_pre_aprobado + 1;
                                                     END IF;
                                                  end if;


                                               END LOOP;
                                                if v_i_prom_pre_aprobado =
0 then
                                                   INSERT INTO
CRM_D_FICHA_MATRICULA
                                                            (
COD_ALUMNO,

COD_SEC_PLAN,
                                                              COD_CURSO,

COD_CARRERA,

COD_CICLO_CURSO,
                                                              TIP_CURSO,
                                                              EST_CURSO,
                                                              NUM_VECES,

NUM_CREDITO_CURSO,

FLG_PROMOCION,

FLG_REPROGRAMACION )
VALUES ( w_c_codigo,
w_c_plan,
w_c_curso,
```

```
PC_CARRERA,
w_c_ciclo,
'02', '02',
w_i_veces,
w_i_creditos,
'1',
w_flg_reprogramacion )  ;
                                              end if;

                                          END IF;
                                       END IF;
                                    else
                                       SELECT COUNT(*)
                                       INTO w_n_detalle
                                       FROM CRM_D_FICHA_MATRICULA
                                       WHERE COD_ALUMNO = w_c_codigo AND
                                                   COD_SEC_PLAN =
w_c_plan AND
                                                    COD_CURSO = w_c_curso
;

                                       IF w_n_detalle = 0 THEN
                                         INSERT INTO CRM_D_FICHA_MATRICULA
                                          ( COD_ALUMNO,
                                            COD_SEC_PLAN,
                                            COD_CURSO,
                                            COD_CARRERA,
                                            COD_CICLO_CURSO,
                                            TIP_CURSO,
                                            EST_CURSO,
                                            NUM_VECES,
                                            NUM_CREDITO_CURSO,
                                            FLG_PROMOCION,
                                            FLG_REPROGRAMACION )
                                         VALUES ( w_c_codigo,
                                                w_c_plan,
                                                w_c_curso,
                                                PC_CARRERA,
                                                w_c_ciclo,
                                                '02',
                                                '02',
                                                w_i_veces,
                                                w_i_creditos,
                                                '0',
                                                w_flg_reprogramacion )  ;
                                          END IF;
                                     END IF;
                                     --FIN PRE - REQUISITO
                                  ELSE
                                    ----------------------------------
                                    --TIENE NUMERO DE CREDITOS MINIMOS
                                    ----------------------------------
                                    if w_c_tipo_credito = '01' then
/*Obligatorio*/
w_n_total_credito := w_n_credito_obligatorio;
elsif w_c_tipo_credito = '02' then /*Electivo*/
      w_n_total_credito := w_n_credito_electivo;
elsif w_c_tipo_credito = '03' then /*Obligatorio + Electivo*/
      w_n_total_credito := w_n_cred_aprobado;
elsif w_c_tipo_credito = '04' then /*Todos los cursos anteriores
Obligatorio*/
```

```
                                    --modificacion
       SELECT sum(nvl(CRM_D_AVANCE_CURRICULAR.NUM_CREDITOS_CURSO, 0))
       INTO w_n_cred_aprob_ciclo_ant
       FROM CRM_D_AVANCE_CURRICULAR, PLA_D_PLAN_ESTUDIO
       WHERE ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN ) and
                                                 (
PLA_D_PLAN_ESTUDIO.COD_CURSO = CRM_D_AVANCE_CURRICULAR.COD_CURSO ) and
                                                 (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PLA_D_PLAN_ESTUDIO.COD_CARRERA )
and
                                                 (
PLA_D_PLAN_ESTUDIO.TIP_CURSO = '02' ) AND
                                                 (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                 (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                 (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA ) AND
                                                 (
CRM_D_AVANCE_CURRICULAR.COD_CICLO_CURSO < w_c_ciclo ) AND
                                                 (
CRM_D_AVANCE_CURRICULAR.EST_CURSO in ('03', '04', '06', '07') )   ;
                                          w_n_total_credito :=
w_n_cred_aprob_ciclo_ant;
                                  end if;

                                  if  w_n_total_credito >= w_i_mincred
then
                                    ---------------------------------
                                    --Verificando el Pre-requisito
                                    ---------------------------------
                                    SELECT COUNT(*)
                                    INTO w_i_count1
                                    FROM PLA_PRE_REQUISITO
                                    WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                           ( PLA_PRE_REQUISITO.COD_CURSO =
w_c_curso ) /*AND
                                           (
PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ;

                                    if w_i_count1 > 0 then
                                       FOR r_pre_requisito IN ( SELECT
PLA_PRE_REQUISITO.COD_CURSO_REQUISITO
                                                                FROM
PLA_PRE_REQUISITO
                                                                WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                                                      (
PLA_PRE_REQUISITO.COD_CURSO = w_c_curso ) /*AND
                                                                      (
PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP

                                           w_c_curso_requisito :=
ltrim(rtrim(r_pre_requisito.cod_curso_requisito));

                                           SELECT count(*)
                                           INTO w_i_count3
                                           FROM CRM_D_AVANCE_CURRICULAR
```

```
                                          WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                            (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                            (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                            (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                        if w_i_count3 > 0 then
                                           SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                           INTO w_c_estado_requisito
                                           FROM
CRM_D_AVANCE_CURRICULAR
                                           WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                             (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                             (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                             (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                           IF w_c_estado_requisito =
'03' OR w_c_estado_requisito = '04' OR w_c_estado_requisito = '06' OR
w_c_estado_requisito = '07' OR w_c_estado_requisito = '10' THEN
                                              --APROBADO
                                              w_i_aprobado := 1;
                                           ELSE
                                              w_i_desaprobado := 1;
                                           END IF;
                                        end if;
                                     END LOOP;

                                     IF w_i_desaprobado = 0 THEN
                                             SELECT COUNT(*)
                                             INTO w_n_detalle
                                             FROM
CRM_D_FICHA_MATRICULA
                                             WHERE COD_ALUMNO =
w_c_codigo AND
                                                   COD_SEC_PLAN =
w_c_plan AND
                                                   COD_CURSO =
w_c_curso ;

                                             IF w_n_detalle = 0 THEN
                                               INSERT INTO
CRM_D_FICHA_MATRICULA
                                              ( COD_ALUMNO,
                                                COD_SEC_PLAN,
                                                COD_CURSO,
                                                COD_CARRERA,
                                                COD_CICLO_CURSO,
                                                TIP_CURSO,
                                                EST_CURSO,
                                                NUM_VECES,
                                                NUM_CREDITO_CURSO,
                                                FLG_PROMOCION,
                                                FLG_REPROGRAMACION )
```

```
                                                VALUES ( w_c_codigo,
                                                     w_c_plan,
                                                     w_c_curso,
                                                     PC_CARRERA,
                                                     w_c_ciclo,
                                                     '02',
                                                     '02',
                                                     w_i_veces,
                                                     w_i_creditos,
                                                     '0',

w_flg_reprogramacion )  ;
                                                END IF;
                                        ELSE


                                            IF w_i_promocion = 1 THEN
                                              -- verifica los niveles de
acptacion de cursos paralelos
                                              v_i_prom_pre_aprobado := 0;
                                              FOR r_pro_pre_requisito2 IN
( SELECT PLA_PRE_REQUISITO.COD_CURSO_REQUISITO

FROM PLA_PRE_REQUISITO

WHERE ( PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND

( PLA_PRE_REQUISITO.COD_CURSO = w_c_curso_requisito ) /*AND

( PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP

WCOD_CURSO_REQUISITO_PROM:= r_pro_pre_requisito2.COD_CURSO_REQUISITO;
                                                SELECT count(*)
                                                INTO w_i_count10
                                                FROM
CRM_D_AVANCE_CURRICULAR
                                                WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                                if w_i_count10 > 0
then
                                                   SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                                   INTO
w_c_estado_requisito_pro
                                                   FROM
CRM_D_AVANCE_CURRICULAR
                                                   WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                         (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                         (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                         (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;
```

```
                                                    IF
w_c_estado_requisito = '03' OR w_c_estado_requisito = '04' OR
w_c_estado_requisito = '06' OR w_c_estado_requisito = '07' OR
w_c_estado_requisito = '10' THEN /*APROBADO*/
                                                       w_i_aprobado :=
1;
                                                    ELSE

v_i_prom_pre_aprobado := v_i_prom_pre_aprobado + 1;
                                                    END IF;
                                                end if;


                                            END LOOP;
                                            if v_i_prom_pre_aprobado =
0 then

                                             INSERT INTO
CRM_D_FICHA_MATRICULA
                                                   ( COD_ALUMNO,
                                                     COD_SEC_PLAN,
                                                     COD_CURSO,
                                                     COD_CARRERA,
                                                     COD_CICLO_CURSO,
                                                     TIP_CURSO,
                                                     EST_CURSO,
                                                     NUM_VECES,
                                                     NUM_CREDITO_CURSO,
                                                     FLG_PROMOCION,
                                                     FLG_REPROGRAMACION
)
                                             VALUES ( w_c_codigo,
                                                      w_c_plan,
                                                      w_c_curso,
                                                      PC_CARRERA,
                                                      w_c_ciclo,
                                                      '02',
                                                      '02',
                                                      w_i_veces,
                                                      w_i_creditos,
                                                      '1',

w_flg_reprogramacion )  ;
                                                END IF;
                                               END IF;
                                          END IF;
                                     else

                                         SELECT COUNT(*)
                                         INTO w_n_detalle
                                         FROM CRM_D_FICHA_MATRICULA
                                         WHERE COD_ALUMNO = w_c_codigo AND
                                               COD_SEC_PLAN = w_c_plan AND
                                               COD_CURSO = w_c_curso ;

                                         IF w_n_detalle = 0 THEN
                                             INSERT INTO
CRM_D_FICHA_MATRICULA
                                              ( COD_ALUMNO,
                                                COD_SEC_PLAN,
```

```
                                                COD_CURSO,
                                                COD_CARRERA,
                                                COD_CICLO_CURSO,
                                                TIP_CURSO,
                                                EST_CURSO,
                                                NUM_VECES,
                                                NUM_CREDITO_CURSO,
                                                FLG_PROMOCION,
                                                FLG_REPROGRAMACION )
                                             VALUES ( w_c_codigo,
                                                 w_c_plan,
                                                 w_c_curso,
                                                 PC_CARRERA,
                                                 w_c_ciclo,
                                                 '02',
                                                 '02',
                                                 w_i_veces,
                                                 w_i_creditos,
                                                 '0',
                                                 w_flg_reprogramacion )  ;
                                         END IF;

                                        end if;
                                        ---------------------------------
                                        --fin Verificando el Pre-requisito
                                        ---------------------------------
                                      end if;
                                    END IF;--Fin de 1
                                    -------------------------------------
                                    --fin Verificamos el numero creditos
minimo permitidos
                                    -------------------------------------

                                 END IF;
                               ELSE
                                ---------------------------------------
                                --Verificamos el numero creditos minimo
permitidos
                                ---------------------------------------
                                SELECT COUNT(*)
                                INTO w_i_count4
                                FROM PLA_D_PLAN_ESTUDIO
                                WHERE ( PLA_D_PLAN_ESTUDIO.COD_CARRERA =
PC_CARRERA ) AND
                                      ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
w_c_plan ) AND
                                      ( PLA_D_PLAN_ESTUDIO.COD_CURSO = w_c_curso
) AND
                                      ( NUM_MIN_CREDITOS > 0 );

                                IF w_i_count4 > 0 THEN
                                 SELECT NVL(NUM_MIN_CREDITOS, 0), TIP_CREDITO_REQ
                                 INTO w_i_mincred, w_c_tipo_credito
                                 FROM   PLA_D_PLAN_ESTUDIO
                                 WHERE ( PLA_D_PLAN_ESTUDIO.COD_CARRERA =
PC_CARRERA ) AND
                                       ( PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
w_c_plan ) AND
                                       ( PLA_D_PLAN_ESTUDIO.COD_CURSO = w_c_curso
)   ;
                                ELSE
```

```
                         w_i_mincred := 0;
                       END IF;
                       IF w_i_mincred IS NULL OR w_i_mincred < 1 THEN
                         -------------------------------------
                         --Si el curso no tiene credito minimos
                         -------------------------------------

                          SELECT COUNT(*)
                          INTO w_i_count1
                          FROM PLA_PRE_REQUISITO
                          WHERE ( PLA_PRE_REQUISITO.COD_SEC_PLAN =
w_c_plan ) AND
                                ( PLA_PRE_REQUISITO.COD_CURSO =
w_c_curso ) /*AND
                                ( PLA_PRE_REQUISITO.EST_REQUISITO = '01'
)*/ ;

                          if w_i_count1 > 0 then
                            --w_i_aprobado := 0;
                            w_i_desaprobado := 0;

                            FOR r_pre_requisito IN ( SELECT
PLA_PRE_REQUISITO.COD_CURSO_REQUISITO
                                                     FROM
PLA_PRE_REQUISITO
                                                     WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                                           (
PLA_PRE_REQUISITO.COD_CURSO = w_c_curso ) /*AND
                                                           (
PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP

                                  w_c_curso_requisito :=
ltrim(rtrim(r_pre_requisito.cod_curso_requisito));

                                  SELECT count(*)
                                  INTO w_i_count3
                                  FROM CRM_D_AVANCE_CURRICULAR
                                  WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                    (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                    ( CRM_D_AVANCE_CURRICULAR.COD_CURSO
= w_c_curso_requisito ) AND
                                  (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                  if w_i_count3 > 0 then
                                     SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                     INTO w_c_estado_requisito
                                     FROM CRM_D_AVANCE_CURRICULAR
                                     WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                       (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                       (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                       (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;
```

```
                                        IF w_c_estado_requisito = '03' OR
w_c_estado_requisito = '04' OR w_c_estado_requisito = '06' OR
w_c_estado_requisito = '07' OR w_c_estado_requisito = '10' THEN
/*APROBADO*/
                                           w_i_aprobado := 1;
                                        ELSE
                                           w_i_desaprobado := 1;
                                        END IF;
                                     end if;
                                END LOOP;
                                IF w_i_desaprobado = 0  THEN
                                    SELECT COUNT(*)
                                    INTO w_n_detalle
                                    FROM CRM_D_FICHA_MATRICULA
                                    WHERE COD_ALUMNO = w_c_codigo AND
                                              COD_SEC_PLAN = w_c_plan AND
                                              COD_CURSO = w_c_curso ;

                                    IF w_n_detalle = 0 THEN
                                       INSERT INTO CRM_D_FICHA_MATRICULA
                                     ( COD_ALUMNO,
                                       COD_SEC_PLAN,
                                         COD_CURSO,
                                         COD_CARRERA,
                                         COD_CICLO_CURSO,
                                         TIP_CURSO,
                                         EST_CURSO,
                                         NUM_VECES,
                                         NUM_CREDITO_CURSO,
                                         FLG_PROMOCION,
                                         FLG_REPROGRAMACION )
                                      VALUES ( w_c_codigo,
                                            w_c_plan,
                                            w_c_curso,
                                            PC_CARRERA,
                                            w_c_ciclo,
                                            '02',
                                            '02',
                                            w_i_veces,
                                            w_i_creditos,
                                            '0',
                                            w_flg_reprogramacion )  ;
                                    END IF;
                                    COMMIT;
                                ELSE
                                    IF w_i_promocion = 1 THEN


                                     --
DBMS_OUTPUT.put_line(w_c_curso_requisito);
                                    -- verifica los niveles de acptacion de
cursos paralelos
                                       v_i_prom_pre_aprobado := 0;
                                       FOR r_pro_pre_requisito IN ( SELECT
PLA_PRE_REQUISITO.COD_CURSO_REQUISITO
                                                                    FROM
PLA_PRE_REQUISITO
                                                                   WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                                                         (
PLA_PRE_REQUISITO.COD_CURSO = w_c_curso_requisito ) /*AND
```

```
                                                                    (
PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP
                                                WCOD_CURSO_REQUISITO_PROM:=
r_pro_pre_requisito.COD_CURSO_REQUISITO;
                                                --
DBMS_OUTPUT.put_line(WCOD_CURSO_REQUISITO_PROM);
                                                SELECT count(*)
                                                INTO w_i_count10
                                                FROM CRM_D_AVANCE_CURRICULAR
                                                WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;
--DBMS_OUTPUT.put_line('cantidad');
--DBMS_OUTPUT.put_line(to_char(w_i_count10));
                                                if w_i_count10 > 0 then
                                                   SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                                   INTO
w_c_estado_requisito_pro
                                                   FROM
CRM_D_AVANCE_CURRICULAR
                                                   WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                      (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                      (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                      (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;
--DBMS_OUTPUT.put_line('estado');
--DBMS_OUTPUT.put_line(w_c_estado_requisito_pro);
                                                   IF w_c_estado_requisito_pro
= '03' OR w_c_estado_requisito_pro = '04' OR w_c_estado_requisito_pro =
'06' OR w_c_estado_requisito_pro = '07' OR w_c_estado_requisito_pro =
'10' THEN /*APROBADO*/
                                                      w_i_aprobado := 1;
                                                      --v_i_prom_pre_aprobado
:= 0;
                                                   ELSE
                                                      v_i_prom_pre_aprobado :=
v_i_prom_pre_aprobado + 1;
                                                   END IF;
 --DBMS_OUTPUT.put_line('xx0');
--DBMS_OUTPUT.put_line(to_char(v_i_prom_pre_aprobado));
                                                end if;

 -- DBMS_OUTPUT.put_line('xx1');
--DBMS_OUTPUT.put_line(to_char(v_i_prom_pre_aprobado));
                                             END LOOP;

                                              if v_i_prom_pre_aprobado = 0 then
                                              --DBMS_OUTPUT.put_line('insert');
                                              --DBMS_OUTPUT.put_line(W_C_CURSO);
                                                 SELECT COUNT(*)
                                                 INTO w_n_detalle
                                                 FROM CRM_D_FICHA_MATRICULA
```

```
                                        WHERE COD_ALUMNO = w_c_codigo
AND
                                              COD_SEC_PLAN =
w_c_plan AND
                                              COD_CURSO =
w_c_curso ;

                                        IF w_n_detalle = 0 THEN
                                        INSERT INTO
CRM_D_FICHA_MATRICULA
                                              ( COD_ALUMNO,
                                                COD_SEC_PLAN,
                                                COD_CURSO,
                                                COD_CARRERA,
                                                COD_CICLO_CURSO,
                                                TIP_CURSO,
                                                EST_CURSO,
                                                NUM_VECES,
                                                NUM_CREDITO_CURSO,
                                                FLG_PROMOCION,
                                                FLG_REPROGRAMACION
)
                                         VALUES ( w_c_codigo,
                                                  w_c_plan,
                                                  w_c_curso,
                                                  PC_CARRERA,
                                                  w_c_ciclo,
                                                  '02',
                                                  '02',
                                                   w_i_veces,
                                                   w_i_creditos,
                                                   '1',
                                                   w_flg_reprogramacion
)  ;
                                        COMMIT;
                                        end if;
                                     end if;

                                  END IF;
                               END IF;
                            else
                               SELECT COUNT(*)
                               INTO w_n_detalle
                               FROM CRM_D_FICHA_MATRICULA
                               WHERE COD_ALUMNO = w_c_codigo AND
                                           COD_SEC_PLAN = w_c_plan AND
                                            COD_CURSO = w_c_curso ;

                               IF w_n_detalle = 0 THEN
                                 INSERT INTO CRM_D_FICHA_MATRICULA
                                  ( COD_ALUMNO,
                                    COD_SEC_PLAN,
                                    COD_CURSO,
                                    COD_CARRERA,
                                    COD_CICLO_CURSO,
                                    TIP_CURSO,
                                    EST_CURSO,
                                    NUM_VECES,
                                    NUM_CREDITO_CURSO,
                                    FLG_PROMOCION,
                                    FLG_REPROGRAMACION )
```

```
                                VALUES ( w_c_codigo,
                                      w_c_plan,
                                      w_c_curso,
                                      PC_CARRERA,
                                      w_c_ciclo,
                                      '02',
                                      '02',
                                      w_i_veces,
                                      w_i_creditos,
                                      '0',
                                      w_flg_reprogramacion )  ;
                                END IF;
                             END IF;
                             --FIN PRE - REQUISITO
                          ELSE
                            ---------------------------------
                            --TIENE NUMERO DE CREDITOS MINIMOS
                            ---------------------------------
                            if w_c_tipo_credito = '01' then /*Obligatorio*/
                               w_n_total_credito :=
w_n_credito_obligatorio;
                            elsif w_c_tipo_credito = '02' then /*Electivo*/
                                  w_n_total_credito :=
w_n_credito_electivo;
                            elsif w_c_tipo_credito = '03' then
/*Obligatorio + Electivo*/
                                  w_n_total_credito := w_n_cred_aprobado;
                            elsif w_c_tipo_credito = '04' then /*Todos los
cursos anteriores Obligatorio*/
                            --modificacion
                                     SELECT
sum(nvl(CRM_D_AVANCE_CURRICULAR.NUM_CREDITOS_CURSO, 0))
                                     INTO w_n_cred_aprob_ciclo_ant
                                     FROM CRM_D_AVANCE_CURRICULAR,
PLA_D_PLAN_ESTUDIO
                                     WHERE (
PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN = CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN )
and
                                           ( PLA_D_PLAN_ESTUDIO.COD_CURSO =
CRM_D_AVANCE_CURRICULAR.COD_CURSO ) and
                                           (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PLA_D_PLAN_ESTUDIO.COD_CARRERA )
and
                                           ( PLA_D_PLAN_ESTUDIO.TIP_CURSO =
'02' ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_CICLO_CURSO < w_c_ciclo ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.EST_CURSO in ('03', '04', '06', '07') )   ;
                                        w_n_total_credito :=
w_n_cred_aprob_ciclo_ant;
                            end if;

                            if  w_n_total_credito >= w_i_mincred then
                              ---------------------------------
```

```
                              --Verificando el Pre-requisito
                              --------------------------------
                              SELECT COUNT(*)
                              INTO w_i_count1
                              FROM PLA_PRE_REQUISITO
                              WHERE ( PLA_PRE_REQUISITO.COD_SEC_PLAN =
w_c_plan ) AND
                                    ( PLA_PRE_REQUISITO.COD_CURSO =
w_c_curso ) /*AND
                                    ( PLA_PRE_REQUISITO.EST_REQUISITO =
'01' )*/ ;

                              if w_i_count1 > 0 then
                                 FOR r_pre_requisito IN ( SELECT
PLA_PRE_REQUISITO.COD_CURSO_REQUISITO
                                                          FROM
PLA_PRE_REQUISITO
                                                          WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                                                (
PLA_PRE_REQUISITO.COD_CURSO = w_c_curso ) /*AND
                                                                (
PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP

                                  w_c_curso_requisito :=
ltrim(rtrim(r_pre_requisito.cod_curso_requisito));

                                  SELECT count(*)
                                  INTO w_i_count3
                                  FROM CRM_D_AVANCE_CURRICULAR
                                  WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                        (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                        (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                        (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                  if w_i_count3 > 0 then
                                     SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                     INTO w_c_estado_requisito
                                     FROM CRM_D_AVANCE_CURRICULAR
                                     WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = w_c_curso_requisito ) AND
                                           (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                     IF w_c_estado_requisito = '03' OR
w_c_estado_requisito = '04' OR w_c_estado_requisito = '06' OR
w_c_estado_requisito = '07' OR w_c_estado_requisito = '10' THEN
                                        --APROBADO
                                        w_i_aprobado := 1;
                                     ELSE
                                        w_i_desaprobado := 1;
                                     END IF;
```

```
                                    end if;
                                END LOOP;

                                IF w_i_desaprobado = 0 THEN
                                        SELECT COUNT(*)
                                        INTO w_n_detalle
                                        FROM CRM_D_FICHA_MATRICULA
                                        WHERE COD_ALUMNO = w_c_codigo AND
                                              COD_SEC_PLAN = w_c_plan AND
                                              COD_CURSO = w_c_curso ;

                                        IF w_n_detalle = 0 THEN
                                          INSERT INTO
CRM_D_FICHA_MATRICULA
                                         ( COD_ALUMNO,
                                           COD_SEC_PLAN,
                                           COD_CURSO,
                                           COD_CARRERA,
                                           COD_CICLO_CURSO,
                                           TIP_CURSO,
                                           EST_CURSO,
                                           NUM_VECES,
                                           NUM_CREDITO_CURSO,
                                           FLG_PROMOCION,
                                           FLG_REPROGRAMACION )
                                         VALUES ( w_c_codigo,
                                              w_c_plan,
                                              w_c_curso,
                                              PC_CARRERA,
                                              w_c_ciclo,
                                              '02',
                                              '02',
                                              w_i_veces,
                                              w_i_creditos,
                                              '0',
                                              w_flg_reprogramacion )  ;
                                         END IF;
                                ELSE


                                    IF w_i_promocion = 1 THEN
                                      -- verifica los niveles de
acptacion de cursos paralelos
                                      v_i_prom_pre_aprobado := 0;
                                      FOR r_pro_pre_requisito2 IN (
SELECT PLA_PRE_REQUISITO.COD_CURSO_REQUISITO
                                                                    FROM
PLA_PRE_REQUISITO
                                                                   WHERE (
PLA_PRE_REQUISITO.COD_SEC_PLAN = w_c_plan ) AND
                                                                         (
PLA_PRE_REQUISITO.COD_CURSO = w_c_curso_requisito ) /*AND
                                                                          (
PLA_PRE_REQUISITO.EST_REQUISITO = '01' )*/ ) LOOP
                                        WCOD_CURSO_REQUISITO_PROM:=
r_pro_pre_requisito2.COD_CURSO_REQUISITO;
                                        SELECT count(*)
                                        INTO w_i_count10
                                        FROM CRM_D_AVANCE_CURRICULAR
                                        WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
```

```
                                                (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                             if w_i_count10 > 0 then
                                                SELECT
CRM_D_AVANCE_CURRICULAR.EST_CURSO
                                                INTO
w_c_estado_requisito_pro
                                                FROM
CRM_D_AVANCE_CURRICULAR
                                                WHERE (
CRM_D_AVANCE_CURRICULAR.COD_ALUMNO = w_c_codigo ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_SEC_PLAN = w_c_plan ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_CURSO = WCOD_CURSO_REQUISITO_PROM ) AND
                                                   (
CRM_D_AVANCE_CURRICULAR.COD_CARRERA = PC_CARRERA )   ;

                                                IF w_c_estado_requisito =
'03' OR w_c_estado_requisito = '04' OR w_c_estado_requisito = '06' OR
w_c_estado_requisito = '07' OR w_c_estado_requisito = '10' THEN
/*APROBADO*/
                                                   w_i_aprobado := 1;
                                                ELSE
                                                   v_i_prom_pre_aprobado :=
v_i_prom_pre_aprobado + 1;
                                                END IF;
                                             end if;


                                          END LOOP;
                                          if v_i_prom_pre_aprobado = 0 then
                                             SELECT COUNT(*)
                                              INTO w_n_detalle
                                              FROM CRM_D_FICHA_MATRICULA
                                              WHERE COD_ALUMNO = w_c_codigo
AND
                                                        COD_SEC_PLAN =
w_c_plan AND
                                                        COD_CURSO =
w_c_curso ;

                                             IF w_n_detalle = 0 THEN
                                           INSERT INTO CRM_D_FICHA_MATRICULA
                                                 ( COD_ALUMNO,
                                                   COD_SEC_PLAN,
                                                   COD_CURSO,
                                                   COD_CARRERA,
                                                   COD_CICLO_CURSO,
                                                   TIP_CURSO,
                                                   EST_CURSO,
                                                   NUM_VECES,
                                                   NUM_CREDITO_CURSO,
                                                   FLG_PROMOCION,
                                                   FLG_REPROGRAMACION )
                                           VALUES ( w_c_codigo,
```

```
                                                w_c_plan,
                                                w_c_curso,
                                                PC_CARRERA,
                                                w_c_ciclo,
                                                '02',
                                                '02',
                                                w_i_veces,
                                                w_i_creditos,
                                                '1',
                                                w_flg_reprogramacion )
;
                                          end if;
                                       END IF;
                                      END IF;
                                   END IF;
                              else

                                  SELECT COUNT(*)
                                  INTO w_n_detalle
                                  FROM CRM_D_FICHA_MATRICULA
                                  WHERE COD_ALUMNO = w_c_codigo AND
                                        COD_SEC_PLAN = w_c_plan AND
                                        COD_CURSO = w_c_curso ;

                                  IF w_n_detalle = 0 THEN
                                      INSERT INTO CRM_D_FICHA_MATRICULA
                                       ( COD_ALUMNO,
                                         COD_SEC_PLAN,
                                         COD_CURSO,
                                         COD_CARRERA,
                                         COD_CICLO_CURSO,
                                         TIP_CURSO,
                                         EST_CURSO,
                                         NUM_VECES,
                                         NUM_CREDITO_CURSO,
                                         FLG_PROMOCION,
                                         FLG_REPROGRAMACION )
                                      VALUES ( w_c_codigo,
                                          w_c_plan,
                                          w_c_curso,
                                          PC_CARRERA,
                                          w_c_ciclo,
                                          '02',
                                          '02',
                                          w_i_veces,
                                          w_i_creditos,
                                          '0',
                                          w_flg_reprogramacion )  ;
                                  END IF;

                                  end if;
                                  ----------------------------------
                                  --fin Verificando el Pre-requisito
                                  ----------------------------------
                               end if;
                            END IF;--Fin de 1
                            -------------------------------------
                            --fin Verificamos el numero creditos minimo
permitidos
                            -------------------------------------
                            END IF;
```

```
                    END IF;
               end if;

               w_n_max_credito := to_number(GEN_PARAMETROS_FN(pc_periodo,
'19', pc_carrera));

               WCICLO_INC := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'13', PC_CARRERA));

               IF  TO_NUMBER(WNUM_CICLO_RELATIVO) >= WCICLO_INC THEN
                    WCICLO_INC := 0;
               ELSE
                    WCICLO_INC := 0;
               END IF;

               UPDATE crm_avance_curricular
               SET NUM_MAX_CICLO_MATRICULARSE = w_n_max_credito +
TO_NUMBER(WNUM_CICLO_RELATIVO) + TO_NUMBER(WCICLO_INC) ,
                   tip_condicion_alumno = w_c_tip_condicion_al,
                   NUM_CICLO_RELATIVO = WNUM_CICLO_RELATIVO,
                   NUM_INCRE_CICLO_MATRICULARSE = 0
               WHERE COD_ALUMNO = w_c_codigo AND
                     COD_CARRERA = PC_CARRERA ;



               SELECT nvl(NUM_CICLO_RELATIVO,0)
               INTO   WNUM_CICLO_RELATIVO
               FROM   CRM_AVANCE_CURRICULAR
               WHERE ( COD_ALUMNO = w_c_codigo ) AND
                     ( COD_CARRERA = PC_CARRERA )  ;

               IF WNUM_CICLO_RELATIVO = 0 THEN
                  WNUM_CICLO_RELATIVO := 1;
               END IF;
               IF WNUM_CICLO_RELATIVO < 10 THEN
                  w_c_ciclo_minimo := '0' ||
TO_CHAR(WNUM_CICLO_RELATIVO);
               ELSE
                  w_c_ciclo_minimo := TO_CHAR(WNUM_CICLO_RELATIVO);
               END IF;

               UPDATE crm_d_ficha_matricula
               SET   COD_CICLO_ACTUAL = w_c_ciclo_minimo
               WHERE COD_ALUMNO = w_c_codigo AND
                     COD_CARRERA = PC_CARRERA ;
               commit;
          END LOOP;

     SELECT COUNT(*)
     INTO w_i_count
     FROM crm_d_ficha_matricula
     WHERE COD_ALUMNO = w_c_codigo
     AND   COD_CARRERA = PC_CARRERA
     AND   NUM_VECES >= 2
     AND   nvl(FLG_REPROGRAMACION,'0') = '0';
     IF w_i_count = 1 and w_c_tip_condicion_al <> '04' then

        ---CASO DE AMNISTIA----------------------------------
        SELECT COD_CURSO , COD_SEC_PLAN
        INTO wcod_curso_des, wcod_sec_plan_des
```

```
        FROM crm_d_ficha_matricula
        WHERE COD_ALUMNO = w_c_codigo
        AND   COD_CARRERA = PC_CARRERA
        AND   NUM_VECES >= 2
        AND   nvl(FLG_REPROGRAMACION,'0') = '0';

        SELECT COUNT(*)
        INTO    w_i_count
        FROM  PLA_D_PLAN_ESTUDIO , HIS.HIS_ACTA_D_NOTAS
        WHERE PLA_D_PLAN_ESTUDIO.COD_CURADI =
HIS.HIS_ACTA_D_NOTAS.COD_CURSO
        AND   PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
HIS.HIS_ACTA_D_NOTAS.COD_SEC_PLAN
        AND   PLA_D_PLAN_ESTUDIO.COD_CURSO = wcod_curso_des
        AND   PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN = wcod_sec_plan_des
        AND   HIS.HIS_ACTA_D_NOTAS.COD_ALUMNO = w_c_codigo
        AND   HIS.HIS_ACTA_D_NOTAS.COD_CARRERA = PC_CARRERA
        AND   HIS.HIS_ACTA_D_NOTAS.COD_PERIODO >= '2016-II';
        ------------------------------------------------------------
        IF  w_i_count>= 2 THEN
        w_n_cred_matricula := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'28', PC_CARRERA));
        UPDATE CRM_FICHA_MATRICULA
        SET      NUM_MAX_CRED_AMPLIACION = 0,
                 NUM_MAX_CRED_MATRICULA =  w_n_cred_matricula
        WHERE COD_ALUMNO = w_c_codigo
        AND   COD_CARRERA = PC_CARRERA;
        COMMIT;
        ELSE
        UPDATE CRM_FICHA_MATRICULA
        SET      NUM_MAX_CRED_AMPLIACION = 0
        WHERE COD_ALUMNO = w_c_codigo
        AND   COD_CARRERA = PC_CARRERA;
        COMMIT;
        END IF;
     ELSE
        IF w_i_count > 1 and w_c_tip_condicion_al <> '04' then
            ------------------------------------------------------------
--------------------
            ---POR AMNISTIA
            ------------------------------------------------------------
--------------------
             FOR rdesaprobadox IN (  SELECT COD_CURSO,COD_SEC_PLAN
                                     FROM crm_d_ficha_matricula
                                     WHERE COD_ALUMNO = w_c_codigo
                                     AND   COD_CARRERA = PC_CARRERA
                                     AND   NUM_VECES >= 2
                                     AND
nvl(FLG_REPROGRAMACION,'0') = '0'
                                  )  LOOP
                   wcod_curso_des:= rdesaprobadox.COD_CURSO;
                   wcod_sec_plan_des:= rdesaprobadox.COD_SEC_PLAN;
                    ------- CASO DE AMNISTIA-----------------------------
-----

                    SELECT COUNT(*)
                    INTO    w_i_count
                    FROM  PLA_D_PLAN_ESTUDIO , HIS.HIS_ACTA_D_NOTAS
                    WHERE PLA_D_PLAN_ESTUDIO.COD_CURADI =
HIS.HIS_ACTA_D_NOTAS.COD_CURSO
```

```
                    AND   PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
HIS.HIS_ACTA_D_NOTAS.COD_SEC_PLAN
                    AND   PLA_D_PLAN_ESTUDIO.COD_CURSO = wcod_curso_des
                    AND   PLA_D_PLAN_ESTUDIO.COD_SEC_PLAN =
wcod_sec_plan_des
                    AND   HIS.HIS_ACTA_D_NOTAS.COD_ALUMNO = w_c_codigo
                    AND   HIS.HIS_ACTA_D_NOTAS.COD_CARRERA = PC_CARRERA
                    AND   HIS.HIS_ACTA_D_NOTAS.COD_PERIODO >= '2016-II';
                    ------------------------------------------------------
---------
                    if w_i_count > 1 then
                     exit;
                    end if;
            END LOOP;

             IF  w_i_count>= 2 THEN
               w_n_cred_matricula :=
to_number(GEN_PARAMETROS_FN(PC_PERIODO, '29', PC_CARRERA));
               UPDATE CRM_FICHA_MATRICULA
               SET      NUM_MAX_CRED_AMPLIACION = 0,
                        NUM_MAX_CRED_MATRICULA =  w_n_cred_matricula
               WHERE COD_ALUMNO = w_c_codigo
               AND   COD_CARRERA = PC_CARRERA;
               COMMIT;
             ELSE
               UPDATE CRM_FICHA_MATRICULA
               SET      NUM_MAX_CRED_AMPLIACION = 0
               WHERE COD_ALUMNO = w_c_codigo
               AND   COD_CARRERA = PC_CARRERA;
               COMMIT;
             END IF;
        END IF;
     END IF;
     --
**********************************************************************
     --CALCULO NRO CREDITOS MAXIMOS A MATRICULAR DESAPROBADO MAS DE DOS
VECES
     --
**********************************************************************

     if w_c_tip_alumno_fm = 'ING' then

          w_n_cred_ampliacion := 0;
          w_n_cred_matricula := to_number(GEN_PARAMETROS_FN(PC_PERIODO,
'14', PC_CARRERA));
          if w_n_cred_matricula = 0 then
             w_n_cred_matricula :=
PKC_PL_MANTENIMIENTO_TABLA.FNC_PL_CREDITOS_CICLO(PC_CARRERA,
w_c_plan_ultimo,wcod_ciclo_relativo);
          end if;
          UPDATE CRM_FICHA_MATRICULA
          SET      NUM_MAX_CRED_AMPLIACION = 0,
                   NUM_MAX_CRED_MATRICULA =  w_n_cred_matricula
          WHERE COD_ALUMNO = w_c_codigo
          AND   COD_CARRERA = PC_CARRERA;
          COMMIT;

     end if;

   END LOOP;
```

```
  END PRC_FM_FICHAMATRICULA_UK;
```

Código Fuente Tasa Educacional

```
FUNCTION   FNC_MAW_TASA_MATRICULA_IK(PCOD_USUARIO VARCHAR, PCOD_ALUMNO
VARCHAR, PCOD_CARRERA VARCHAR, PCOD_PERIODO VARCHAR) RETURN NUMBER


 AS
    vnum_return                       number;
    vnum_desaprobado                  integer;
    vnum_reprogramado                 integer;
    vnum_reingreso                    integer;
    vnum_primer_puesto                integer;
    wclv_escuela                      TA_MATW1ALUMNO.CLV_ESCUELA%TYPE;
    wcod_modalidad                    TA_MATW1ALUMNO.CLV_MODALIDAD%TYPE;
    wtip_colegio                      TA_MATW1ALUMNO.TIP_COLEGIO%TYPE;
    wcod_periodo_ingreso              TA_MATW1ALUMNO.CLV_PERIODO%TYPE;
    wcod_permat                       TA_MATW1ALUMNO.CLV_PERIODOMAT%TYPE;
    vnum_count                        integer;
    west_ficha_matricula
TA_MAW1FICHA_MATRICULA.EST_FICHA%TYPE;
    vnum_curso                        integer;
    wcod_periodo_puesto               TA_MATW1ALUMNO.CLV_PERIODO%TYPE;
    wcod_anho_ingreso                 TA_MATW1ALUMNO.CLV_PERIODO%TYPE;
    wcod_anho_inicio                  TA_MATW1ALUMNO.CLV_PERIODO%TYPE;
    wnum_periodo_puesto               integer;
    wnum_pariente                     integer;
    wnum_danza                        integer;
    wnum_prom_pond_per_ant
SIGAUW.TA_MAW1FICHA_MATRICULA.NUM_PROM_POND_PER_ANT%TYPE;
    vnum_anho_gratuidad               integer;
    vnum_licencia                    integer;
    wcod_periodo_licencia            TA_MATW1ALUMNO.CLV_PERIODO%TYPE;
    vnum_periodo_licencia            integer;
 BEGIN
    vnum_return := 1;

    SELECT count(*)
    INTO   vnum_count
    FROM   sigauw.ta_maw2registro_matricula
    WHERE  CLV_ALUMNO = PCOD_ALUMNO;

    IF SQL%FOUND THEN
       ------------
       --- FOUND
       ------------
       IF (vnum_count = 0) THEN
             vnum_return := 0; /* no exite curso matriculado*/
       ELSE
        --------------------
        --- REGISTRO EXISTE
        --------------------
        IF  substr(pcod_periodo,6,1) = '0' THEN
        ----------------------------
        -- MATRICULA CICLO VERANO
        ----------------------------
```

```
               SELECT COUNT(*)
               INTO vnum_curso
               FROM SIGAUW.TA_MAW2REGISTRO_MATRICULA B
               WHERE b.CLV_ALUMNO=  PCOD_ALUMNO;

               INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                    NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
                   SELECT   PCOD_ALUMNO,
                            CLV_ESCUELA,
                            CLV_SEC_ITEM,
                            NUM_MONTO*vnum_curso,
                            NUM_MONTO*vnum_curso,
                            TIP_PAGO,
                            EST_PAGO,
                            PCOD_USUARIO,
                            SYSDATE,
                            'I'
                   FROM     SIGAUW.TA_APW2TASAACADEMICA
                   WHERE     CLV_PERIODO = PCOD_PERIODO
                   AND       CLV_ESCUELA  = PCOD_CARRERA
                   AND       CLV_SEC_ITEM ='21';          -- VALOR POR
CURSO DE VERANO
           ----------------------------
           -- FIN MATRICULA CICLO VERANO
           ----------------------------

           ELSE
           ----------------------------
           -- MATRICULA  PERIODO REGULAR
           ----------------------------
           --------------------------------
           --------------------------------
           -- proceso de tasa educacional
           --------------------------------
           --------------------------------
           ----------------------
           --Modalidad de Ingreso
           ----------------------
           select tip_postula, tip_colegio ,clv_periodo
,clv_periodomat,clv_escuela,substr(clv_periodo,1,4),substr(clv_periodo_in
icio,1,4)
           into   wcod_modalidad       ,
wtip_colegio,wcod_periodo_ingreso,wcod_permat,wclv_escuela,wcod_anho_ingr
eso,wcod_anho_inicio
           from SIGAUW.TA_MATW1ALUMNO
           where  clv_alumno = pcod_alumno;

           ----------------------
           --verificar reingreso
           ----------------------
           select count(*)
           into  vnum_reingreso
           from  sigauw.ta_ogw1solicitud
```

```
        where clv_alumno = pcod_alumno
        and   clv_periodo = pcod_periodo
        and   tip_solicitud = '01'  ---REINGRESO
        and    est_solicitud = '03';---APROBADO

        IF vnum_reingreso > 0 THEN
             --- reingreso
            select count(*)
            into vnum_reingreso
            from ta_apw2periodo
            where clv_periodo > wcod_permat--periodo ultimo matriculado
            and   clv_periodo < PCOD_PERIODO--periodo actual
            and   substr(clv_periodo,6,1) <> '0';
            ---revisar ciclo regular 2017-0 --aaa

            INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                     NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_MONTO*vnum_reingreso,
                                NUM_MONTO*vnum_reingreso,
                                TIP_PAGO,
                                EST_PAGO,
                                PCOD_USUARIO,
                                SYSDATE,
                                'I'
                         FROM     SIGAUW.TA_APW2TASAACADEMICA
                         WHERE     CLV_PERIODO = PCOD_PERIODO
                         AND       CLV_ESCUELA  = PCOD_CARRERA
                         AND       CLV_SEC_ITEM ='13'; -- PAGO POR SEMESTRE
DE REINGRESO


        END IF;
        ----------------------
        --fin de verificar reingreso
        ----------------------

        IF   wcod_periodo_ingreso = PCOD_PERIODO THEN
        ---------------------------------
        --verificar si es ingresante
        ---------------------------------
         ---------------------------------
          --verifica MODALIDAD
          ---------------------------------
          IF wcod_modalidad = '00' or wcod_modalidad = '01' or
wcod_modalidad = '02' or wcod_modalidad = '04' or wcod_modalidad = '05'
or wcod_modalidad = '06' or wcod_modalidad = '07' or wcod_modalidad =
'13' or wcod_modalidad = '14' or wcod_modalidad = '15' or wcod_modalidad
= '17' THEN
          ---X
          IF wtip_colegio = '0' THEN
```

```
            ------------
            --ESTATAL
            ------------
               INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                     NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='02';          --COLEGIO ESTATAL
MATRICULA

               INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                     NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='15';          --CARNET
UNIVERSITARIO

               INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                     NUM_PAGO_MIN,
                                    TIP_PAGO,
```

```
                            EST_PAGO,
                            CLV_USUARIO,
                            DT_REGISTRO,
                            TIP_OPERACION)
     SELECT   PCOD_ALUMNO,
              CLV_ESCUELA,
              CLV_SEC_ITEM,
              NUM_MONTO,
              NUM_MONTO,
              TIP_PAGO,
              EST_PAGO,
              PCOD_USUARIO,
              SYSDATE,
              'I'
     FROM     SIGAUW.TA_APW2TASAACADEMICA
     WHERE     CLV_PERIODO = PCOD_PERIODO
     AND       CLV_ESCUELA  = PCOD_CARRERA
     AND       CLV_SEC_ITEM ='18';         --CARNET BIBLIOTECA
    -------------aaa
    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                            CLV_ESCUELA,
                            CLV_SEC_ITEM,
                            NUM_PAGO,
                             NUM_PAGO_MIN,
                            TIP_PAGO,
                            EST_PAGO,
                            CLV_USUARIO,
                            DT_REGISTRO,
                            TIP_OPERACION)
     SELECT   PCOD_ALUMNO,
              CLV_ESCUELA,
              CLV_SEC_ITEM,
              NUM_MONTO,
              NUM_MONTO,
              TIP_PAGO,
              EST_PAGO,
              PCOD_USUARIO,
              SYSDATE,
              'I'
     FROM     SIGAUW.TA_APW2TASAACADEMICA
     WHERE     CLV_PERIODO = PCOD_PERIODO
     AND       CLV_ESCUELA  = PCOD_CARRERA
     AND       CLV_SEC_ITEM ='20';         --Cautoseguro
    ---------xaaa


------------
--FIN DE ESTATAL
------------

ELSE
------------
--PRIVADO
------------
 IF wtip_colegio = '1' THEN --PRIVADO
      INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
```

```
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='03';         --COLEGIO PRIVADO

              INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                     NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     NUM_MONTO,
                     NUM_MONTO,
                     TIP_PAGO,
                     EST_PAGO,
                     PCOD_USUARIO,
                     SYSDATE,
                     'I'
            FROM     SIGAUW.TA_APW2TASAACADEMICA
            WHERE     CLV_PERIODO = PCOD_PERIODO
            AND       CLV_ESCUELA  = PCOD_CARRERA
            AND       CLV_SEC_ITEM ='15';         --CARNET
UNIVERSITARIO

              INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                     NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     NUM_MONTO,
                     NUM_MONTO,
                     TIP_PAGO,
```

```
                EST_PAGO,
                PCOD_USUARIO,
                SYSDATE,
                'I'
        FROM    SIGAUW.TA_APW2TASAACADEMICA
        WHERE    CLV_PERIODO = PCOD_PERIODO
        AND      CLV_ESCUELA  = PCOD_CARRERA
        AND      CLV_SEC_ITEM ='18';         --CARNET BIBLIOTECA

        ------------aaa
         INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                 NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
        SELECT   PCOD_ALUMNO,
                CLV_ESCUELA,
                CLV_SEC_ITEM,
                NUM_MONTO,
                NUM_MONTO,
                TIP_PAGO,
                EST_PAGO,
                PCOD_USUARIO,
                SYSDATE,
                'I'
        FROM    SIGAUW.TA_APW2TASAACADEMICA
        WHERE    CLV_PERIODO = PCOD_PERIODO
        AND      CLV_ESCUELA  = PCOD_CARRERA
        AND      CLV_SEC_ITEM ='20';         --Cautoseguro
        ---------xaaa

      ELSE  --OTROS--- falta preguntar
           INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     NUM_MONTO,
                     NUM_MONTO,
                     TIP_PAGO,
                     EST_PAGO,
                     PCOD_USUARIO,
                     SYSDATE,
                     'I'
            FROM     SIGAUW.TA_APW2TASAACADEMICA
            WHERE     CLV_PERIODO = PCOD_PERIODO
            AND       CLV_ESCUELA  = PCOD_CARRERA
            AND       CLV_SEC_ITEM ='03';         --COLEGIO OTROS
```

```
                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                         NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='15';          --CARNET
UNIVERSITARIO

                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                         NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='18';          --CARNET BIBLIOTECA

                -------------aaa
                 INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                         NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
```

```
                                DT_REGISTRO,
                                TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     NUM_MONTO,
                     NUM_MONTO,
                     TIP_PAGO,
                     EST_PAGO,
                     PCOD_USUARIO,
                     SYSDATE,
                     'I'
            FROM     SIGAUW.TA_APW2TASAACADEMICA
            WHERE     CLV_PERIODO = PCOD_PERIODO
            AND       CLV_ESCUELA  = PCOD_CARRERA
            AND       CLV_SEC_ITEM ='20';         --Cautoseguro
            ---------xaaa

          END IF;
          ------------
          --FIN DE PRIVADO
          ------------
         END IF;
         --X
        ELSE
         --Y
         ---
           IF wcod_modalidad ='03' THEN -- graduado y titulados

                 INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                 SELECT   PCOD_ALUMNO,
                          CLV_ESCUELA,
                          CLV_SEC_ITEM,
                          NUM_MONTO,
                          NUM_MONTO,
                          TIP_PAGO,
                          EST_PAGO,
                          PCOD_USUARIO,
                          SYSDATE,
                          'I'
                 FROM     SIGAUW.TA_APW2TASAACADEMICA
                 WHERE     CLV_PERIODO = PCOD_PERIODO
                 AND       CLV_ESCUELA  = PCOD_CARRERA
                 AND       CLV_SEC_ITEM ='20';  ---autoseguro

                 INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
```

```
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             NUM_MONTO,
                             NUM_MONTO,
                             TIP_PAGO,
                             EST_PAGO,
                             PCOD_USUARIO,
                             SYSDATE,
                             'I'
                    FROM     SIGAUW.TA_APW2TASAACADEMICA
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='12';  ---MATRICULA ORDINARIA
PARA GRADUADOS

                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             NUM_MONTO,
                             0,
                             TIP_PAGO,
                             EST_PAGO,
                             PCOD_USUARIO,
                             SYSDATE,
                             'I'
                    FROM     SIGAUW.TA_APW2TASAACADEMICA
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='10';  --PAGO TOTAL

                           --
                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             0,
                             NUM_MONTO,
                             TIP_PAGO,
```

```
                        EST_PAGO,
                        PCOD_USUARIO,
                        SYSDATE,
                        'I'
                FROM    SIGAUW.TA_APW2TASAACADEMICA
                WHERE    CLV_PERIODO = PCOD_PERIODO
                AND      CLV_ESCUELA  = PCOD_CARRERA
                AND      CLV_SEC_ITEM ='11';         --  pago
fraccionario

            ELSE  --LAS DEMAS MODALIDADES
              ----
                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                        CLV_ESCUELA,
                        CLV_SEC_ITEM,
                        NUM_MONTO,
                        NUM_MONTO,
                        TIP_PAGO,
                        EST_PAGO,
                        PCOD_USUARIO,
                        SYSDATE,
                        'I'
                FROM    SIGAUW.TA_APW2TASAACADEMICA
                WHERE    CLV_PERIODO = PCOD_PERIODO
                AND      CLV_ESCUELA  = PCOD_CARRERA
                AND      CLV_SEC_ITEM ='20';         --  avance
curricular


                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                        CLV_ESCUELA,
                        CLV_SEC_ITEM,
                        NUM_MONTO,
                        NUM_MONTO,
                        TIP_PAGO,
                        EST_PAGO,
                        PCOD_USUARIO,
                        SYSDATE,
                        'I'
                FROM    SIGAUW.TA_APW2TASAACADEMICA
                WHERE    CLV_PERIODO = PCOD_PERIODO
```

```
                  AND       CLV_ESCUELA  = PCOD_CARRERA
                  AND       CLV_SEC_ITEM ='15';         -- CARNET
universitario

                ---PREGUNTAR POR PAGO DE CARNET UNIVERSITARIO -> 15??
             END IF;
          ---
          --Y
         END IF;
        -----------------------------------
        --fin verificar si es ingresante
        -----------------------------------
        ELSE
        -----------------------------------
        --verificar no es ingresante
        -----------------------------------
          ---------------------------------
          ---PAGO EN COMUN
          ---------------------------------
          if wcod_modalidad <> '08' then
              ---------------------------------------
              -- PAGO POR PERDIDA DE GRATUIDAD OBSERVADO
              ---------------------------------------
              SELECT NUM_PROM_POND_PER_ANT
              INTO   wnum_prom_pond_per_ant
              FROM SIGAUW.TA_MAW1FICHA_MATRICULA
              WHERE CLV_ESCUELA = wclv_escuela
              AND CLV_ALUMNO = pcod_alumno  ;

              IF  wnum_prom_pond_per_ant < 10.5 THEN  --CONSULTAR SI SOLO
ES 2016-II Y 2017-0
                 INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA
                      (CLV_ALUMNO,
                                      CLV_ESCUELA,
                                      CLV_SEC_ITEM,
                                      NUM_PAGO,
                                      NUM_PAGO_MIN,
                                      TIP_PAGO,
                                      EST_PAGO,
                                      CLV_USUARIO,
                                      DT_REGISTRO,
                                      TIP_OPERACION)
                  SELECT   PCOD_ALUMNO,
                           CLV_ESCUELA,
                           CLV_SEC_ITEM,
                           NUM_MONTO,
                           NUM_MONTO,
                           TIP_PAGO,
                           EST_PAGO,
                           PCOD_USUARIO,
                           SYSDATE,
                           'I'
                  FROM     SIGAUW.TA_APW2TASAACADEMICA
                  WHERE     CLV_PERIODO = PCOD_PERIODO
                  AND       CLV_ESCUELA  = PCOD_CARRERA
                  AND       CLV_SEC_ITEM ='06';         --G
              END IF;
              -------------------------------
              ---PAGO POR PERDIDA DE GRATUIDAD REZAGADOS  Y  (9) POR
PERDIDA DE GRATUIDAD
              -------------------------------
```

```
              IF wclv_escuela ='1410' OR wclv_escuela = '1212' THEN
                 vnum_anho_gratuidad :=
to_number(substr(PCOD_PERIODO,1,4)) - 6;
                 IF TO_NUMBER(wcod_anho_inicio) <= vnum_anho_gratuidad
THEN
                  INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA
                      (CLV_ALUMNO,
                                      CLV_ESCUELA,
                                      CLV_SEC_ITEM,
                                      NUM_PAGO,
                                      NUM_PAGO_MIN,
                                      TIP_PAGO,
                                      EST_PAGO,
                                      CLV_USUARIO,
                                      DT_REGISTRO,
                                      TIP_OPERACION)
                  SELECT   PCOD_ALUMNO,
                           CLV_ESCUELA,
                           CLV_SEC_ITEM,
                           NUM_MONTO,
                           NUM_MONTO,
                           TIP_PAGO,
                           EST_PAGO,
                           PCOD_USUARIO,
                           SYSDATE,
                           'I'
                  FROM     SIGAUW.TA_APW2TASAACADEMICA
                  WHERE     CLV_PERIODO = PCOD_PERIODO
                  AND       CLV_ESCUELA  = PCOD_CARRERA
                  AND       CLV_SEC_ITEM ='07';         --GRATUIDAD
REZAGADO
                 ELSE
                  IF wnum_prom_pond_per_ant >= 10.5 THEN
                   ---REVISAR SI TIENE DESP Y NSP
                   SELECT COUNT(*)
                   INTO vnum_count
                   FROM SIGAUW.TA_HIW1HISTORICO_NOTAS
                   WHERE CLV_ESCUELA = wclv_escuela
                   AND  CLV_ALUMNO = pcod_alumno
                   AND  CLV_PERIODO = wcod_permat
                   AND  EST_CURSO IN ('09','02');

                   IF vnum_count > 0 THEN
                     INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                      CLV_ESCUELA,
                                      CLV_SEC_ITEM,
                                      NUM_PAGO,
                                      NUM_PAGO_MIN,
                                      TIP_PAGO,
                                      EST_PAGO,
                                      CLV_USUARIO,
                                      DT_REGISTRO,
                                      TIP_OPERACION)
                   SELECT   PCOD_ALUMNO,
                            CLV_ESCUELA,
                            CLV_SEC_ITEM,
                            NUM_MONTO,
                            NUM_MONTO,
                            TIP_PAGO,
                            EST_PAGO,
                            PCOD_USUARIO,
```

```
                            SYSDATE,
                            'I'
                     FROM     SIGAUW.TA_APW2TASAACADEMICA
                     WHERE     CLV_PERIODO = PCOD_PERIODO
                     AND       CLV_ESCUELA  = PCOD_CARRERA
                     AND       CLV_SEC_ITEM ='09';         --GRATUIDAD
                      END IF;

                     END IF;
                    END IF;

                  ELSE
                    IF wclv_escuela ='1211' THEN
                     vnum_anho_gratuidad :=
to_number(substr(PCOD_PERIODO,1,4)) - 7;
                     IF TO_NUMBER(wcod_anho_inicio) <= vnum_anho_gratuidad
THEN
                      INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                        NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                      SELECT   PCOD_ALUMNO,
                               CLV_ESCUELA,
                               CLV_SEC_ITEM,
                               NUM_MONTO,
                               NUM_MONTO,
                               TIP_PAGO,
                               EST_PAGO,
                               PCOD_USUARIO,
                               SYSDATE,
                               'I'
                      FROM     SIGAUW.TA_APW2TASAACADEMICA
                      WHERE     CLV_PERIODO = PCOD_PERIODO
                      AND       CLV_ESCUELA  = PCOD_CARRERA
                      AND       CLV_SEC_ITEM ='07';         --GRATUIDAD
REZAGADO
                     ELSE
                      IF wnum_prom_pond_per_ant >= 10.5 THEN
                       ---REVISAR SI TIENE DESP Y NSP
                       SELECT COUNT(*)
                       INTO vnum_count
                       FROM SIGAUW.TA_HIW1HISTORICO_NOTAS
                       WHERE CLV_ESCUELA = wclv_escuela
                       AND  CLV_ALUMNO = pcod_alumno
                       AND  CLV_PERIODO = wcod_permat
                       AND  EST_CURSO IN ('09','02');

                       IF vnum_count > 0 THEN
                         INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                        NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
```

```
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             NUM_MONTO,
                             NUM_MONTO,
                             TIP_PAGO,
                             EST_PAGO,
                             PCOD_USUARIO,
                             SYSDATE,
                             'I'
                    FROM     SIGAUW.TA_APW2TASAACADEMICA
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='09';         --GRATUIDAD
                     END IF;

                    END IF;
                   END IF;

                  ELSE
                   vnum_anho_gratuidad :=
to_number(substr(PCOD_PERIODO,1,4)) - 5;
                   IF TO_NUMBER(wcod_anho_inicio) <= vnum_anho_gratuidad
THEN
                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                    CLV_ESCUELA,
                                    CLV_SEC_ITEM,
                                    NUM_PAGO,
                                    NUM_PAGO_MIN,
                                    TIP_PAGO,
                                    EST_PAGO,
                                    CLV_USUARIO,
                                    DT_REGISTRO,
                                    TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             NUM_MONTO,
                             NUM_MONTO,
                             TIP_PAGO,
                             EST_PAGO,
                             PCOD_USUARIO,
                             SYSDATE,
                             'I'
                    FROM     SIGAUW.TA_APW2TASAACADEMICA
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='07';         --GRATUIDAD
REZAGADO
                   ELSE
                      IF wnum_prom_pond_per_ant >= 10.5 THEN
                         ---REVISAR SI TIENE DESP Y NSP
                         SELECT COUNT(*)
                         INTO vnum_count
                         FROM SIGAUW.TA_HIW1HISTORICO_NOTAS
                         WHERE CLV_ESCUELA = wclv_escuela
                         AND  CLV_ALUMNO = pcod_alumno
                         AND  CLV_PERIODO = wcod_permat
```

```
                        AND  EST_CURSO IN ('09','02');

                        IF vnum_count > 0 THEN
                          INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA
(CLV_ALUMNO,
                                                CLV_ESCUELA,
                                                CLV_SEC_ITEM,
                                                NUM_PAGO,
                                                NUM_PAGO_MIN,
                                                TIP_PAGO,
                                                EST_PAGO,
                                                CLV_USUARIO,
                                                DT_REGISTRO,
                                                TIP_OPERACION)
                        SELECT   PCOD_ALUMNO,
                                 CLV_ESCUELA,
                                 CLV_SEC_ITEM,
                                 NUM_MONTO,
                                 NUM_MONTO,
                                 TIP_PAGO,
                                 EST_PAGO,
                                 PCOD_USUARIO,
                                 SYSDATE,
                                 'I'
                        FROM     SIGAUW.TA_APW2TASAACADEMICA
                        WHERE     CLV_PERIODO = PCOD_PERIODO
                        AND       CLV_ESCUELA  = PCOD_CARRERA
                        AND       CLV_SEC_ITEM ='09';         --GRATUIDAD
                         END IF;

                     END IF;
                   END IF;

                  END IF;
                END IF;
                ----------------------------
                --cursos desaprobados (5)
                ----------------------------
                SELECT COUNT(*)
                INTO vnum_desaprobado
                FROM SIGAUW.TA_MAW2FICHA_MATRICULA A,
SIGAUW.TA_MAW2REGISTRO_MATRICULA B
                WHERE A.CLV_ALUMNO= B.CLV_ALUMNO AND
                      A.CLV_CURSO = B.CLV_CURSO  AND
                      A.CLV_PLAN =B.CLV_PLAN AND
                      A.TIP_CURSO ='01'  AND   -- DESAPROBADO
                      B.CLV_ALUMNO = PCOD_ALUMNO;
                IF vnum_desaprobado > 0 THEN
                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                        NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                        SELECT   PCOD_ALUMNO,
                                 CLV_ESCUELA,
                                 CLV_SEC_ITEM,
```

```
                                NUM_MONTO*vnum_desaprobado,
                                NUM_MONTO*vnum_desaprobado,
                                TIP_PAGO,
                                EST_PAGO,
                                PCOD_USUARIO,
                                SYSDATE,
                                'I'
                        FROM    SIGAUW.TA_APW2TASAACADEMICA
                        WHERE    CLV_PERIODO = PCOD_PERIODO
                        AND      CLV_ESCUELA  = PCOD_CARRERA
                        AND      CLV_SEC_ITEM ='05';
                END IF;

                ----------------------------
                --cursos reprogramados (14)
                ----------------------------
                SELECT COUNT(*)
                INTO vnum_reprogramado
                FROM SIGAUW.TA_MAW2FICHA_MATRICULA A,
SIGAUW.TA_MAW2REGISTRO_MATRICULA B
                WHERE A.CLV_ALUMNO= B.CLV_ALUMNO AND
                      A.CLV_CURSO = B.CLV_CURSO  AND
                      A.CLV_PLAN =B.CLV_PLAN AND
                      A.FLG_REPROGRAMACION ='1'  AND   -- DESAPROBADO
                      B.CLV_ALUMNO = PCOD_ALUMNO;
                IF vnum_reprogramado > 0 THEN
                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
                        SELECT   PCOD_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_MONTO*vnum_reprogramado,
                                NUM_MONTO*vnum_reprogramado,
                                TIP_PAGO,
                                EST_PAGO,
                                PCOD_USUARIO,
                                SYSDATE,
                                'I'
                        FROM    SIGAUW.TA_APW2TASAACADEMICA
                        WHERE    CLV_PERIODO = PCOD_PERIODO
                        AND      CLV_ESCUELA  = PCOD_CARRERA
                        AND      CLV_SEC_ITEM ='14';
                END IF;
               ------------------------------------
              --verificar mat extemporanea
              ------------------------------------
                SELECT EST_FICHA
                INTO  west_ficha_matricula
                FROM SIGAUW.TA_MAW1FICHA_MATRICULA
                WHERE CLV_ALUMNO = PCOD_ALUMNO
                AND   CLV_ESCUELA = PCOD_CARRERA;

                IF west_ficha_matricula = '07' THEN
```

```
                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                        NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                 SELECT   PCOD_ALUMNO,
                          CLV_ESCUELA,
                          CLV_SEC_ITEM,
                          NUM_MONTO,
                          NUM_MONTO,
                          TIP_PAGO,
                          EST_PAGO,
                          PCOD_USUARIO,
                          SYSDATE,
                          'I'
                 FROM     SIGAUW.TA_APW2TASAACADEMICA
                 WHERE     CLV_PERIODO = PCOD_PERIODO
                 AND       CLV_ESCUELA  = PCOD_CARRERA
                 AND       CLV_SEC_ITEM ='04';         --MATRICULA
EXTEMPORANEA


              END IF;

           end if;
           ---------------------------------
           ---FIN DE PAGO EN COMUN
           ---------------------------------
           ---------------------------------
           --verifica MODALIDAD
           ---------------------------------
             IF wcod_modalidad = '00' or wcod_modalidad = '01' or
wcod_modalidad = '02' or wcod_modalidad = '04' or wcod_modalidad = '05'
or wcod_modalidad = '06' or wcod_modalidad = '07' or wcod_modalidad ='12'
or wcod_modalidad ='13'  or wcod_modalidad ='14' or wcod_modalidad = '15'
or wcod_modalidad = '17' THEN
            -------------
            --- XXXX
            -------------
              ------------------------
              -- primeros puestos
              ------------------------
              IF LENGTH(pcod_periodo) = 7 THEN
                wcod_periodo_puesto:= SUBSTR(PCOD_PERIODO,1,6);
              ELSE
                wcod_periodo_puesto:= SUBSTR(PCOD_PERIODO,1,4);
                wnum_periodo_puesto:= to_number(wcod_periodo_puesto) - 1;
                wcod_periodo_puesto:=
concat(to_char(wnum_periodo_puesto),'-II');
              END IF;

               --PRIMER PUESTO
               select count(*)
               into  vnum_primer_puesto
               from  sigauw.ta_ogw1solicitud
               where clv_alumno = pcod_alumno
```

```
and   clv_periodo = wcod_periodo_puesto
and   tip_solicitud = '22'
and    est_solicitud = '03';
--PARIENTE ADMINISTRATIVO
select count(*)
into  wnum_pariente
from  sigauw.ta_ogw1solicitud
where clv_alumno = pcod_alumno
and   clv_periodo = PCOD_PERIODO
and   tip_solicitud = '23'
and    est_solicitud = '03';

--DANZA DEPORTISTA
select count(*)
into  wnum_danza
from  sigauw.ta_ogw1solicitud
where clv_alumno = pcod_alumno
and   clv_periodo = PCOD_PERIODO
and   tip_solicitud = '24'
and    est_solicitud = '03';

---------------
---PAGO DE MATRICULA
---------------

IF (vnum_primer_puesto > 0) or (wnum_danza > 0) THEN
   INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                           CLV_ESCUELA,
                           CLV_SEC_ITEM,
                           NUM_PAGO,
                           NUM_PAGO_MIN,
                           TIP_PAGO,
                           EST_PAGO,
                           CLV_USUARIO,
                           DT_REGISTRO,
                           TIP_OPERACION)
    SELECT   PCOD_ALUMNO,
             CLV_ESCUELA,
             CLV_SEC_ITEM,
             NUM_MONTO,
             NUM_MONTO,
             TIP_PAGO,
             EST_PAGO,
             PCOD_USUARIO,
             SYSDATE,
             'I'
    FROM     SIGAUW.TA_APW2TASAACADEMICA
    WHERE     CLV_PERIODO = PCOD_PERIODO
    AND       CLV_ESCUELA  = PCOD_CARRERA
    AND       CLV_SEC_ITEM ='15'; -- CARNET univeristario

   INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                           CLV_ESCUELA,
                           CLV_SEC_ITEM,
                           NUM_PAGO,
                           NUM_PAGO_MIN,
                           TIP_PAGO,
                           EST_PAGO,
                           CLV_USUARIO,
                           DT_REGISTRO,
                           TIP_OPERACION)
```

```
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='20';          --  autoseguro

                ---PREGUNTAR POR PAGO DE CARNET UNIVERSITARIO -> 15??
            ELSE
              IF  wnum_pariente > 0 THEN

                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                        NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO/2,
                         NUM_MONTO/2,
                         TIP_PAGO,
                         EST_PAGO,
                         PCOD_USUARIO,
                         SYSDATE,
                         'I'
                FROM     SIGAUW.TA_APW2TASAACADEMICA
                WHERE     CLV_PERIODO = PCOD_PERIODO
                AND       CLV_ESCUELA  = PCOD_CARRERA
                AND       CLV_SEC_ITEM ='01'; -- MATRICULA ORDINARIA

                INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                        CLV_ESCUELA,
                                        CLV_SEC_ITEM,
                                        NUM_PAGO,
                                        NUM_PAGO_MIN,
                                        TIP_PAGO,
                                        EST_PAGO,
                                        CLV_USUARIO,
                                        DT_REGISTRO,
                                        TIP_OPERACION)
                SELECT   PCOD_ALUMNO,
                         CLV_ESCUELA,
                         CLV_SEC_ITEM,
                         NUM_MONTO,
                         NUM_MONTO,
                         TIP_PAGO,
                         EST_PAGO,
```

```
                    PCOD_USUARIO,
                    SYSDATE,
                    'I'
             FROM   SIGAUW.TA_APW2TASAACADEMICA
             WHERE   CLV_PERIODO = PCOD_PERIODO
             AND     CLV_ESCUELA  = PCOD_CARRERA
             AND     CLV_SEC_ITEM ='20';  ---Autoseguro

              INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                  CLV_ESCUELA,
                                  CLV_SEC_ITEM,
                                  NUM_PAGO,
                                  NUM_PAGO_MIN,
                                  TIP_PAGO,
                                  EST_PAGO,
                                  CLV_USUARIO,
                                  DT_REGISTRO,
                                  TIP_OPERACION)
             SELECT   PCOD_ALUMNO,
                      CLV_ESCUELA,
                      CLV_SEC_ITEM,
                      NUM_MONTO,
                      NUM_MONTO,
                      TIP_PAGO,
                      EST_PAGO,
                      PCOD_USUARIO,
                      SYSDATE,
                      'I'
             FROM     SIGAUW.TA_APW2TASAACADEMICA
             WHERE     CLV_PERIODO = PCOD_PERIODO
             AND       CLV_ESCUELA  = PCOD_CARRERA
             AND       CLV_SEC_ITEM ='15';  ---carnet

            ---PREGUNTAR POR PAGO DE CARNET UNIVERSITARIO -> 15??
        ELSE
            -----------------------
            --matricula regular (1)
            -----------------------
             INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                  CLV_ESCUELA,
                                  CLV_SEC_ITEM,
                                  NUM_PAGO,
                                  NUM_PAGO_MIN,
                                  TIP_PAGO,
                                  EST_PAGO,
                                  CLV_USUARIO,
                                  DT_REGISTRO,
                                  TIP_OPERACION)
             SELECT   PCOD_ALUMNO,
                      CLV_ESCUELA,
                      CLV_SEC_ITEM,
                      NUM_MONTO,
                      NUM_MONTO,
                      TIP_PAGO,
                      EST_PAGO,
                      PCOD_USUARIO,
                      SYSDATE,
                      'I'
             FROM     SIGAUW.TA_APW2TASAACADEMICA
             WHERE     CLV_PERIODO = PCOD_PERIODO
             AND       CLV_ESCUELA  = PCOD_CARRERA
```

```
                AND       CLV_SEC_ITEM ='01'; -- MATRICULA ORDINARIA

              INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
               SELECT   PCOD_ALUMNO,
                        CLV_ESCUELA,
                        CLV_SEC_ITEM,
                        NUM_MONTO,
                        NUM_MONTO,
                        TIP_PAGO,
                        EST_PAGO,
                        PCOD_USUARIO,
                        SYSDATE,
                        'I'
               FROM     SIGAUW.TA_APW2TASAACADEMICA
               WHERE     CLV_PERIODO = PCOD_PERIODO
               AND       CLV_ESCUELA  = PCOD_CARRERA
               AND       CLV_SEC_ITEM ='15';  ---CARNET universitario
            ---PREGUNTAR POR PAGO DE CARNET UNIVERSITARIO -> 15??
                END IF;
               END IF;
             ---------------
              ---FIN PAGO DE MATRICULA
              ---------------
            -------------------------------------
            --fin verificar mat extemporanea
            -------------------------------------

             vnum_return := 1;
          ELSE
             IF wcod_modalidad ='03' THEN -- graduado y titulados

               INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
               SELECT   PCOD_ALUMNO,
                        CLV_ESCUELA,
                        CLV_SEC_ITEM,
                        NUM_MONTO,
                        NUM_MONTO,
                        TIP_PAGO,
                        EST_PAGO,
                        PCOD_USUARIO,
                        SYSDATE,
                        'I'
               FROM     SIGAUW.TA_APW2TASAACADEMICA
```

```
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='20';  ---Auto seguro
                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                          CLV_ESCUELA,
                                          CLV_SEC_ITEM,
                                          NUM_PAGO,
                                          NUM_PAGO_MIN,
                                          TIP_PAGO,
                                          EST_PAGO,
                                          CLV_USUARIO,
                                          DT_REGISTRO,
                                          TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             NUM_MONTO,
                             NUM_MONTO,
                             TIP_PAGO,
                             EST_PAGO,
                             PCOD_USUARIO,
                             SYSDATE,
                             'I'
                    FROM     SIGAUW.TA_APW2TASAACADEMICA
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='12';  ---MATRICULA ORDINARIA
PARA GRADUADOS
                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                          CLV_ESCUELA,
                                          CLV_SEC_ITEM,
                                          NUM_PAGO,
                                          NUM_PAGO_MIN,
                                          TIP_PAGO,
                                          EST_PAGO,
                                          CLV_USUARIO,
                                          DT_REGISTRO,
                                          TIP_OPERACION)
                    SELECT   PCOD_ALUMNO,
                             CLV_ESCUELA,
                             CLV_SEC_ITEM,
                             NUM_MONTO,
                             0,
                             TIP_PAGO,
                             EST_PAGO,
                             PCOD_USUARIO,
                             SYSDATE,
                             'I'
                    FROM     SIGAUW.TA_APW2TASAACADEMICA
                    WHERE     CLV_PERIODO = PCOD_PERIODO
                    AND       CLV_ESCUELA  = PCOD_CARRERA
                    AND       CLV_SEC_ITEM ='10';    --PAGO TOTAL

                    INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                          CLV_ESCUELA,
                                          CLV_SEC_ITEM,
                                          NUM_PAGO,
                                          NUM_PAGO_MIN,
                                          TIP_PAGO,
                                          EST_PAGO,
                                          CLV_USUARIO,
```

```
                                DT_REGISTRO,
                                TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     0,
                     NUM_MONTO,
                     TIP_PAGO,
                     EST_PAGO,
                     PCOD_USUARIO,
                     SYSDATE,
                     'I'
            FROM     SIGAUW.TA_APW2TASAACADEMICA
            WHERE     CLV_PERIODO = PCOD_PERIODO
            AND       CLV_ESCUELA  = PCOD_CARRERA
            AND       CLV_SEC_ITEM ='11';  --  pago fraccionario

        ELSE  --LAS DEMAS MODALIDADES
            INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     NUM_MONTO,
                     NUM_MONTO,
                     TIP_PAGO,
                     EST_PAGO,
                     PCOD_USUARIO,
                     SYSDATE,
                     'I'
            FROM     SIGAUW.TA_APW2TASAACADEMICA
            WHERE     CLV_PERIODO = PCOD_PERIODO
            AND       CLV_ESCUELA  = PCOD_CARRERA
            AND       CLV_SEC_ITEM ='20';  --  avance curricular

            INSERT INTO SIGAUW.TA_MAW1TASAACADEMICA (CLV_ALUMNO,
                                CLV_ESCUELA,
                                CLV_SEC_ITEM,
                                NUM_PAGO,
                                NUM_PAGO_MIN,
                                TIP_PAGO,
                                EST_PAGO,
                                CLV_USUARIO,
                                DT_REGISTRO,
                                TIP_OPERACION)
            SELECT   PCOD_ALUMNO,
                     CLV_ESCUELA,
                     CLV_SEC_ITEM,
                     NUM_MONTO,
                     NUM_MONTO,
                     TIP_PAGO,
                     EST_PAGO,
                     PCOD_USUARIO,
```

```
                            SYSDATE,
                            'I'
                    FROM    SIGAUW.TA_APW2TASAACADEMICA
                    WHERE    CLV_PERIODO = PCOD_PERIODO
                    AND      CLV_ESCUELA  = PCOD_CARRERA
                    AND      CLV_SEC_ITEM ='15';  ---carnet universitario
                END IF;

                vnum_return := 1;
             END IF;
             vnum_return := 1; /*ok*/
           END IF;
           END IF;
            COMMIT;
          END IF;

    ELSE
         vnum_return := 0; /* El alumno no está registrado*/
    END IF;

      return(vnum_return);

   EXCEPTION
   WHEN NO_DATA_FOUND THEN
        vnum_return := 0; --El Usuario no está registrado
     return(vnum_return);

 END   FNC_MAW_TASA_MATRICULA_IK;
 END PKC_MAW_MANTENIMIENTO_TABLA ;
```

Código Fuente Boleta de Matricula Web

```
PROCEDURE   PRC_MAW_BOLETA1_MAT_VW(PCURSOR OUT CUR_BOLETAMAT1
,PCID_USUARIO VARCHAR, PC_ALUMNO VARCHAR, PC_ESCUELA VARCHAR )
   AS
    BEGIN
      OPEN PCURSOR  for
        SELECT TA_MAW1FICHA_MATRICULA.CLV_ALUMNO,
         TA_MAW1FICHA_MATRICULA.CLV_ESCUELA,
         TA_MAW1FICHA_MATRICULA.CLV_PERIODO,
         TA_MAW1FICHA_MATRICULA.NUM_CRED_MATRICULADO,
         TA_MAW1FICHA_MATRICULA.NUM_CUR_MATRICULADO,
         TA_MAW1FICHA_MATRICULA.DT_MATRICULA,
         TA_MAW1FICHA_MATRICULA.DT_REGISTRO,
         TA_PLW1FACULTAD.TXT_FACULTAD,
         TA_PLW1ESCUELA.TXT_ESCUELA,
         TA_MATW1ALUMNO.TXT_NOMBRE_COMPLETO,
         TA_PLW1PLAN_ESTUDIO.CLV_PERIODO AS PLAN_ESTUDIO,
         TA_PLW2PLAN_ESTUDIO.CLV_CURSO_PLAN,
         TA_PLW1CURSO.TXT_CURSO,
         TA_PLW2PLAN_ESTUDIO.CLV_CICLO_CURRICULAR,
         TA_PLW2PLAN_ESTUDIO.NUM_CREDITO,

decode(TA_MAW2FICHA_MATRICULA.TIP_CURSO,'01','D','02','P','03','F') as
TXT_EST_CURSO,
         TA_CAW1GRUPO.TXT_GRUPO
    FROM sigauw.TA_MAW1FICHA_MATRICULA,
         sigauw.TA_MATW1ALUMNO,
```

```
         sigauw.TA_PLW1FACULTAD,
         sigauw.TA_PLW1ESCUELA,
         sigauw.TA_PLW1PLAN_ESTUDIO,
         sigauw.TA_MAW2REGISTRO_MATRICULA,
         sigauw.TA_PLW2PLAN_ESTUDIO,
         sigauw.TA_PLW1CURSO,
         sigauw.TA_MAW2FICHA_MATRICULA,
         sigauw.TA_CAW1GRUPO
   WHERE ( TA_MAW1FICHA_MATRICULA.CLV_ALUMNO = TA_MATW1ALUMNO.CLV_ALUMNO
) and
         ( TA_MAW1FICHA_MATRICULA.CLV_ESCUELA =
TA_MATW1ALUMNO.CLV_ESCUELA ) and
         ( TA_MATW1ALUMNO.CLV_FACULTAD = TA_PLW1FACULTAD.CLV_FACULTAD )
and
         ( TA_PLW1ESCUELA.CLV_ESCUELA =
TA_MAW1FICHA_MATRICULA.CLV_ESCUELA ) and
            ( TA_PLW1PLAN_ESTUDIO.CLV_PLAN =
TA_MAW2FICHA_MATRICULA.CLV_PLAN ) and
         ( TA_PLW1PLAN_ESTUDIO.CLV_ESCUELA =
TA_MAW1FICHA_MATRICULA.CLV_ESCUELA ) and
         ( TA_MAW2REGISTRO_MATRICULA.CLV_ESCUELA =
TA_MAW1FICHA_MATRICULA.CLV_ESCUELA )   and
         ( TA_MAW2REGISTRO_MATRICULA.CLV_ALUMNO =
TA_MAW1FICHA_MATRICULA.CLV_ALUMNO )   and
         ( TA_PLW2PLAN_ESTUDIO.CLV_PLAN =
TA_MAW2REGISTRO_MATRICULA.CLV_PLAN  )   and
         ( TA_PLW2PLAN_ESTUDIO.CLV_CURSO =
TA_MAW2REGISTRO_MATRICULA.CLV_CURSO  )   and
         ( TA_PLW1CURSO.CLV_CURSO = TA_MAW2REGISTRO_MATRICULA.CLV_CURSO )
and
         ( TA_MAW2FICHA_MATRICULA.CLV_ESCUELA =
TA_MAW2REGISTRO_MATRICULA.CLV_ESCUELA ) and
         ( TA_MAW2FICHA_MATRICULA.CLV_ALUMNO =
TA_MAW2REGISTRO_MATRICULA.CLV_ALUMNO ) and
         ( TA_MAW2FICHA_MATRICULA.CLV_PLAN =
TA_MAW2REGISTRO_MATRICULA.CLV_PLAN ) and
         ( TA_MAW2FICHA_MATRICULA.CLV_CURSO =
TA_MAW2REGISTRO_MATRICULA.CLV_CURSO ) and
         ( TA_CAW1GRUPO.CLV_ESCUELA =
TA_MAW2REGISTRO_MATRICULA.CLV_ESCUELA ) and
         ( TA_CAW1GRUPO.CLV_GRUPO = TA_MAW2REGISTRO_MATRICULA.CLV_GRUPO )
and
          TA_MAW1FICHA_MATRICULA.CLV_ALUMNO = PC_ALUMNO;

   END PRC_MAW_BOLETA1_MAT_VW;
 PROCEDURE   PRC_MAW_BOLETA2_MAT_VW(PCURSOR OUT CUR_BOLETAMAT2
,PCID_USUARIO VARCHAR, PC_ALUMNO VARCHAR, PC_ESCUELA VARCHAR )
   AS
    BEGIN
      OPEN PCURSOR  for
      SELECT  VI_MAWCURSO_GRUPO_MATRICULA.ALUMNO ,
              VI_MAWCURSO_GRUPO_MATRICULA.SEC_PLAN ,
              VI_MAWCURSO_GRUPO_MATRICULA.CURSO,
              VI_MAWCURSO_GRUPO_MATRICULA.GRUPO ,
              VI_MAWCURSO_GRUPO_MATRICULA.TIPO ,
              VI_MAWCRUCE_HORARIO.TXT_CURSO ,
              VI_MAWCRUCE_HORARIO.DIA ,
              VI_MAWCRUCE_HORARIO.DOCENTE ,
              VI_MAWCRUCE_HORARIO.TXT_DOCENTE ,
              VI_MAWCRUCE_HORARIO.CARRERA ,
              VI_MAWCRUCE_HORARIO.CRUCE ,
```

```
              VI_MAWCURSO_GRUPO_MATRICULA.CREDITO ,
              VI_MAWCURSO_GRUPO_MATRICULA.CICLO
    FROM sigauw.VI_MAWCRUCE_HORARIO ,
         sigauw.VI_MAWCURSO_GRUPO_MATRICULA
    WHERE ( VI_MAWCRUCE_HORARIO.SEC_PLAN =
VI_MAWCURSO_GRUPO_MATRICULA.SEC_PLAN )
     and          ( ( VI_MAWCRUCE_HORARIO.CURSO =
VI_MAWCURSO_GRUPO_MATRICULA.CURSO )
     And          ( VI_MAWCRUCE_HORARIO.SECCION =
VI_MAWCURSO_GRUPO_MATRICULA.GRUPO )
     and          ( VI_MAWCURSO_GRUPO_MATRICULA.ALUMNO = PC_ALUMNO) )
;
   END PRC_MAW_BOLETA2_MAT_VW;
END PKC_MAW_REPORTE_DOC;
```

Código Fuente Reporte Avance Curricular Web

```
PROCEDURE   PRC_MAW_AVANCE1_CURRICULAR_VW(PCURSOR OUT CUR_MAWAVANCE1
,PCID_USUARIO VARCHAR, PC_ALUMNO VARCHAR, PC_ESCUELA VARCHAR )

   AS
    BEGIN
      OPEN PCURSOR  for
        SELECT TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ALUMNO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA,
         TA_OGW1AVANCE_PLAN_ESTUDIO.TIP_CONDICION_ALUMNO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ULT_PERIODO_MAT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_PLAN_ULTIMO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.FLG_AMPLIACION_CREDITO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.FLG_BIBLIOTECA,
         TA_OGW1AVANCE_PLAN_ESTUDIO.FLG_PAGO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.EST_AVANCE_PLAN,
         TA_MATW1ALUMNO.TXT_NOMBRE_COMPLETO,
         TA_PLW1ESCUELA.TXT_ESCUELA,
         TA_PLW1FACULTAD.TXT_FACULTAD,
         TA_PLW1PLAN_ESTUDIO.CLV_PERIODO AS CLV_PLAN_EST
    FROM SIGAUW.TA_OGW1AVANCE_PLAN_ESTUDIO,
         SIGAUW.TA_MATW1ALUMNO ,
         SIGAUW.TA_PLW1ESCUELA,
         SIGAUW.TA_PLW1FACULTAD,
         SIGAUW.TA_PLW1PLAN_ESTUDIO
    WHERE ( TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ALUMNO = PC_ALUMNO ) and
          ( TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA = PC_ESCUELA ) AND
          (TA_MATW1ALUMNO.CLV_ALUMNO =
TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ALUMNO) AND
          (TA_PLW1ESCUELA.CLV_ESCUELA =
TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA) AND
          ( TA_PLW1FACULTAD.CLV_FACULTAD =TA_PLW1ESCUELA.CLV_FACULTAD)
AND
          (TA_PLW1PLAN_ESTUDIO.CLV_ESCUELA =
TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA ) AND
```

```
          (TA_PLW1PLAN_ESTUDIO.CLV_PLAN =
TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_PLAN_ULTIMO );

   END PRC_MAW_AVANCE1_CURRICULAR_VW;


   PROCEDURE   PRC_MAW_AVANCE2_CURRICULAR_VW(PCURSOR OUT CUR_MAWAVANCE2
,PCID_USUARIO VARCHAR, PC_ALUMNO VARCHAR, PC_ESCUELA VARCHAR )

   AS


    BEGIN
      OPEN PCURSOR  for    SELECT TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_ALUMNO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_PLAN,
         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_CURSO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_ESCUELA,

         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_SIST_EVALUACION,

         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_CICLO_CURSO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.EST_CURSO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.TIP_CURSO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.FLG_CONVALIDACION,
         TA_OGW2AVANCE_PLAN_ESTUDIO.NUM_NOTA_PARCIAL,
         TA_OGW2AVANCE_PLAN_ESTUDIO.NUM_NOTA_PRACTICA,
         TA_OGW2AVANCE_PLAN_ESTUDIO.NUM_NOTA_FINAL,

            '',
            '',

         TA_PLW2PLAN_ESTUDIO.CLV_CURSO_PLAN,
         TA_PLW1CURSO.TXT_CURSO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.NUM_CRED_CURSO,

fnc_pre_requisito_rep(TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_PLAN,TA_OGW2AVANCE_P
LAN_ESTUDIO.CLV_CURSO,TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_ESCUELA) as pre_req,
          GEN_MAESTRA_A.TXT_DESCRIP_CORTA,
         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_GRUPO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.NUM_VECES,
         TA_OGW2AVANCE_PLAN_ESTUDIO.NUM_NOTA_PROMEDIO,
         TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_PER_ULT_MATRICULA,
           GEN_MAESTRA_B.TXT_DESCRIP_CORTA  AS ESTADO ,
        TA_OGW2AVANCE_PLAN_ESTUDIO.TXT_OBSERVACION,
           1  AS QUIEBRE
    FROM TA_OGW2AVANCE_PLAN_ESTUDIO,
         TA_GEW1MAESTRA  GEN_MAESTRA_A,
         TA_GEW1MAESTRA  GEN_MAESTRA_B ,
         TA_PLW2PLAN_ESTUDIO ,
         TA_PLW1CURSO
   WHERE ( TA_OGW2AVANCE_PLAN_ESTUDIO.TIP_CURSO = GEN_MAESTRA_A.CLV_ITEM
) and
         ( TA_OGW2AVANCE_PLAN_ESTUDIO.EST_CURSO = GEN_MAESTRA_B.CLV_ITEM
) and
         ( TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_ALUMNO = PC_ALUMNO ) AND
         ( TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_ESCUELA = PC_ESCUELA ) AND
         ( TA_OGW2AVANCE_PLAN_ESTUDIO.EST_CURSO <> '05' ) AND
          GEN_MAESTRA_A.CLV_TABLA = '48' AND
          GEN_MAESTRA_B.CLV_TABLA = '78' AND
          (TA_PLW2PLAN_ESTUDIO.CLV_CURSO =
TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_CURSO ) AND
```

```
          (TA_PLW2PLAN_ESTUDIO.CLV_PLAN =
TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_PLAN ) AND
          (TA_PLW1CURSO.CLV_CURSO = TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_CURSO
)

    ORDER BY  TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_ALUMNO,   QUIEBRE,
TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_CICLO_CURSO,
         TA_PLW2PLAN_ESTUDIO.CLV_CURSO_PLAN,
TA_OGW2AVANCE_PLAN_ESTUDIO.CLV_CURSO;

    END PRC_MAW_AVANCE2_CURRICULAR_VW;
   PROCEDURE   PRC_MAW_AVANCE3_CURRICULAR_VW(PCURSOR OUT CUR_MAWAVANCE2
,PCID_USUARIO VARCHAR, PC_ALUMNO VARCHAR, PC_ESCUELA VARCHAR )

   AS


    BEGIN
      OPEN PCURSOR  for   SELECT TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ALUMNO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA,
         TA_OGW1AVANCE_PLAN_ESTUDIO.TIP_CONDICION_ALUMNO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ULT_PERIODO_MAT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_PLAN_ULTIMO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.FLG_AMPLIACION_CREDITO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.FLG_BIBLIOTECA,
         TA_OGW1AVANCE_PLAN_ESTUDIO.FLG_PAGO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.EST_AVANCE_PLAN,

       --  TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_ORDEN_MERITO_GEN_MATRICULA,

         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_ORD_FICHA_MATRICULA,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_CRED_APROB_PLAN_ACT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_PROM_POND_PER_ANT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_CRED_APROB_PER_ANT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_TOTAL_CURSO_APROBADO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_TOTAL_CRED_APROBADO,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_TOTAL_PROM_POND_GRAL,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_CRED_TOTAL_DESAPROB,

     --    TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_ORDEN_MERITO_PER_ANT_CARR,

         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_CRED_APROB_PER_ANT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_CRED_INSC_PER_ANT,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_TOTAL_CRED_ACUM_APROB,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_PROM_APROB_PLAN_ACT  ,
         TA_OGW1AVANCE_PLAN_ESTUDIO.NUM_PROMED_ULT_PER
    FROM SIGAUW.TA_OGW1AVANCE_PLAN_ESTUDIO
   WHERE ( TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ALUMNO = PC_ALUMNO) AND
         ( TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA = PC_ESCUELA)    ;


   END PRC_MAW_AVANCE3_CURRICULAR_VW;

   PROCEDURE   PRC_MAW_AVANCE4_CURRICULAR_VW(PCURSOR OUT CUR_MAWAVANCE2
,PCID_USUARIO VARCHAR, PC_ALUMNO VARCHAR, PC_ESCUELA VARCHAR )

   AS


    BEGIN
      OPEN PCURSOR  for    SELECT TA_PLW1PLAN_ESTUDIO.NUM_TOT_CRED_COMP,
```

```
         TA_PLW1PLAN_ESTUDIO.NUM_TOT_CRED_PRACT,
         TA_PLW1PLAN_ESTUDIO.NUM_TOT_CRED_ELECT,
         TA_PLW1PLAN_ESTUDIO.NUM_TOT_CRED_OBLIGA,
         TA_PLW1PLAN_ESTUDIO.NUM_TOT_CRED_EXTRA
    FROM TA_PLW1PLAN_ESTUDIO,
         TA_OGW1AVANCE_PLAN_ESTUDIO
   WHERE ( TA_PLW1PLAN_ESTUDIO.CLV_ESCUELA =
TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ESCUELA ) and
         ( ( TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_ALUMNO = PC_ALUMNO ) AND
         ( TA_PLW1PLAN_ESTUDIO.CLV_ESCUELA = PC_ESCUELA ) AND
         ( TA_PLW1PLAN_ESTUDIO.CLV_PLAN =
TA_OGW1AVANCE_PLAN_ESTUDIO.CLV_PLAN_ULTIMO ) ) ;
   END  PRC_MAW_AVANCE4_CURRICULAR_VW;
```

Anexo 7. REUNIÓN DE TRABAJOS – TASKBOARD



