

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO E IMPLEMENTACIÓN DEL PORTAL WEB PARA LA GESTIÓN DE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA EN SISTEMAS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

ESCOBAR CÁRDENAS JOSÉ LUIS

jlec1991@hotmail.com

MOSQUERA ESPINOSA EVELYN MARCELA

marcela.mosquera.e@gmail.com

DIRECTOR: ING. BOLÍVAR OSWALDO PALÁN TAMAYO MSC

bolivar.palan@epn.edu.ec

QUITO, AGOSTO 2015

DECLARACIÓN

Nosotros, José Luis Escobar Cárdenas y Evelyn Marcela Mosquera Espinosa, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

José Luis Escobar Cárdenas

Evelyn Marcela Mosquera Espinosa

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por José Luis Escobar Cárdenas y Evelyn Marcela Mosquera Espinosa bajo mi supervisión.

Ing. Bolívar Palán

DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradezco principalmente a mi familia, a mis padres y a mi hermana, que estuvieron a mi lado durante mi carrera y durante la realización de este proyecto apoyándome incondicionalmente. Agradezco a mis amigos que son más como mis hermanos que de alguna forma me ayudaron con un granito de arena y en especial a Marcela con quien hemos compartido alegrías y también tristezas durante el difícil desarrollo de este proyecto de titulación. A nuestro director de tesis el Ing. Bolívar Palán, quien nos ha guiado con sus consejos y experiencia durante los últimos años de formación profesional y ahora durante esta última etapa de nuestra carrera de pregrado.

José Luis

AGRADECIMIENTO

A mis padres y a mi hermana gracias por todo el cariño y apoyo brindado, ustedes son la principal razón para superarme cada día.

A mi prima Pati con quien he compartido muchas cosas a lo largo de mi vida, gracias amistad por evitar que me dé por vencida, te quiero mucho.

A José Luis, mi mejor amigo, hemos pasado por tantos momentos buenos y malos, estoy segura que nuestra amistad es para siempre.

A todos mis amig@s y mi familia que de una u otra manera siempre han estado junto a mí.

A nuestro director, el Ing. Bolívar Palán por su paciencia y por su dedicación en la dirección de este proyecto de titulación.

Marcela

DEDICATORIA

Quiero dedicar este proyecto de titulación principalmente a mis padres, quienes me enseñaron a ser fuerte y no rendirme nunca incluso cuando las cosas se ponen difíciles. A mis amigos del alma, que son prácticamente mi segunda familia y han estado conmigo apoyándome aunque sea con una sonrisa las 24 horas del día los 365 días del año.

José Luis

DEDICATORIA

Este proyecto de titulación va dedicado con mucho cariño para mi hermana, espero ansiosamente ver que cumplas todas tus metas.

Marcela

CONTENIDO

1	CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	1
1.1	DESCRIPCIÓN DEL PROBLEMA	1
1.1.1	ENTORNO DE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA DE SISTEMAS	1
1.1.1.1	Misión.....	1
1.1.1.2	Visión	2
1.1.1.3	Valores.....	2
1.1.2	ESTRUCTURA DE LA DIRECTIVA DE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA DE SISTEMAS	2
1.1.2.1	Descripción de las Funciones de la Directiva de la AEIS.....	3
1.1.3	SERVICIOS Y ACTIVIDADES QUE OFRECE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA DE SISTEMAS	3
1.1.4	DESCRIPCIÓN DE PROCESOS QUE REALIZA LA AEIS	3
1.1.4.1	Afiliación de Estudiantes	4
1.1.4.3	Matriculación en Clubes.....	6
1.1.4.4	Venta de Snacks y Bebidas	7
1.1.4.5	Préstamo de Dispositivos.....	9
1.1.4.6	Devolución de Dispositivos	10
1.1.4.7	Alquiler de Casilleros	12
1.1.4.8	Organización de Eventos Sociales.....	13
1.1.4.9	Organización del Campeonato Interno de Deportes	15
1.1.5	PROBLEMAS QUE SE PRESENTAN EN LOS PROCESOS QUE REALIZA LA AEIS.....	16
1.1.5.1	Proceso de Afiliación.....	16
1.1.5.2	Oferta de Clubes	16

1.1.5.3	Venta de Snacks y Bebidas	17
1.1.5.4	Préstamo de Dispositivos.....	17
1.1.5.5	Alquiler de Casilleros	17
1.1.5.6	Gestión de Inventario.....	17
1.1.5.7	Eventos	17
1.1.5.8	Campeonato Interno de Deportes	17
1.1.5.9	Encuesta a Estudiantes	17
1.1.6	SITUACIÓN PROPUESTA.....	21
1.2	DESCRIPCIÓN DEL ENTORNO DE DESARROLLO CON LA METODOLOGÍA SCRUM.....	22
1.2.1	ORÍGENES.....	22
1.2.2	ROLES	23
1.2.2.1	Product Owner	23
1.2.2.2	Scrum Master.....	23
1.2.2.3	Equipo.....	24
1.2.3	ARTEFACTOS.....	24
1.2.3.1	Product Backlog	24
1.2.3.2	Sprint Backlog	25
1.2.3.3	Burndown Chart	25
1.2.4	EVENTOS	25
1.2.4.1	Sprint Planning.....	25
1.2.4.2	Sprint Review	26
1.2.4.3	Daily Scrum Meeting	26
1.2.4.4	Sprint Retrospective	26
1.3	SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO.....	27
1.3.1	GESTOR DE BASE DE DATOS.....	27
1.3.2	LENGUAJE DE PROGRAMACIÓN	28

1.3.3	ENTORNO DE DESARROLLO INTEGRADO	28
1.3.4	SERVIDOR DE APLICACIONES.....	29
1.3.5	FRAMEWORK PARA FRONTEND (ANGULARJS).....	30
1.3.6	ENTORNO DE EJECUCIÓN EN EL FRONTEND (NODEJS)	30
2	CAPÍTULO 2: DESARROLLO DEL PORTAL WEB	32
2.1	RECOLECCIÓN DE LAS HISTORIAS DE USUARIOS	32
2.1.1	HU01	34
2.1.2	HU02	34
2.1.3	HU03	35
2.1.4	HU04	35
2.1.5	HU05	35
2.1.6	HU06	36
2.1.7	HU07	36
2.1.8	HU08	36
2.1.9	HU09	37
2.1.10	HU10	37
2.2	DISEÑO DEL PRODUCT BACKLOG	37
2.2.1	LISTA INICIAL DEL PRODUCTO.....	37
2.2.2	REFINAMIENTO DEL PRODUCT BACKLOG.....	38
2.2.3	DEFINICIÓN DE LOS SPRINTS	41
2.2.3.1	Primer Sprint	42
2.2.3.2	Segundo Sprint	44
2.2.3.3	Tercer Sprint	46
2.2.3.4	Cuarto Sprint	48
2.2.4	LISTA FINAL DEL PRODUCTO	50
2.3	DISEÑO E IMPLEMENTACIÓN DE LOS SPRINTS	53
2.3.1	PRIMER SPRINT.....	53

2.3.1.1	Diseño	53
2.3.1.2	Implementación	68
2.3.2	SEGUNDO SPRINT	74
2.3.2.1	Diseño	74
2.3.2.2	Implementación	77
2.3.3	TERCER SPRINT	81
2.3.3.1	Diseño	81
2.3.3.2	Implementación	85
2.3.4	CUARTO SPRINT	91
2.3.4.1	Diseño	91
2.3.4.2	Implementación	95
2.4	SPRINT REVIEW	98
2.4.1	PRIMER SPRINT	99
2.4.2	SEGUNDO SPRINT	101
2.4.3	TERCER SPRINT	103
2.4.4	CUARTO SPRINT	107
2.4.5	ANÁLISIS DEL PROYECTO	108
2.4.5.1	Gráfico de Esfuerzo	108
3	CAPÍTULO III: EVALUACIÓN DEL PORTAL	113
3.1	RECOLECCIÓN DE LOS DATOS DE EVALUACIÓN DE LA AEIS	113
3.1.1	INFORMACIÓN DE ESTUDIANTES	113
3.1.2	INFORMACIÓN DE PRODUCTOS Y PROVEEDORES	113
3.1.3	INFORMACIÓN DE INVENTARIO	114
3.2	IMPLANTACIÓN DEL SISTEMA	114
3.2.1	REQUISITOS	115
3.2.1.1	Hardware	115
3.2.1.2	Software	115

3.2.2	PROCESO DE INSTALACIÓN	116
3.2.2.1	Instalación Backend	116
3.2.2.2	Instalación Frontend.....	119
3.3	PUESTA EN PRODUCCIÓN DEL PORTAL	122
3.3.1	PRUEBAS DE COMPATIBILIDAD	122
3.3.2	ACCESO A LAS PÁGINAS	123
3.4	ANÁLISIS DE RESULTADOS.....	124
3.4.1	RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN	124
3.4.2	EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE ..	131
3.4.2.1	Funcionalidad.....	132
3.4.2.2	Usabilidad	133
3.4.2.3	Apariencia del Portal	134
3.4.2.4	Satisfacción del Usuario.....	137
4	CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES	138
4.1	CONCLUSIONES	138
4.2	RECOMENDACIONES	139
	BIBLIOGRAFÍA	141
	ANEXOS	143

ÍNDICE DE FIGURAS

Figura 1.1 Estructura de la Directiva de la AEIS	2
Figura 1.2 Procesos de la AEIS	4
Figura 1.3 Afiliación de Estudiantes	5
Figura 1.4 Matriculación en Clubes	7
Figura 1.5 Hoja de Control de Ventas	8
Figura 1.6 Venta de Snacks y Bebidas	9
Figura 1.7 Préstamo de Dispositivos	10
Figura 1.8 Devolución de Dispositivos	11
Figura 1.9 Alquiler de Casilleros.....	13
Figura 1.10 Organización de Eventos Sociales.....	14
Figura 1.11 Inscripción en el Campeonato Interno de Deportes	16
Figura 1.12 Respuesta Primera Pregunta	18
Figura 1.13 Respuesta Segunda Pregunta	18
Figura 1.14 Respuesta Tercera Pregunta	19
Figura 1.15 Respuesta Cuarta Pregunta	19
Figura 1.16 Respuesta Quinta Pregunta	19
Figura 1.17 Respuesta Sexta Pregunta	20
Figura 1.18 Respuesta Séptima Pregunta	20
Figura 1.19 Respuesta Octava Pregunta	21
Figura 1.20 Scrum.....	23
Figura 2.1 Esfuerzo Realizado en el Primer Sprint.....	43
Figura 2.2 Esfuerzo Realizado en el Segundo Sprint.....	45
Figura 2.3 Esfuerzo Realizado en el Tercer Sprint.....	48
Figura 2.4 Esfuerzo Realizado en el Cuarto Sprint	50
Figura 2.5 Arquitectura AEIS WEB APP	54
Figura 2.6 Ejemplo de Objeto en formato JSON	55
Figura 2.7 Esquema de Seguridad Frontend.....	56
Figura 2.8 Esquema de Seguridad Backend	57
Figura 2.9 Modelo Navegacional.....	58
Figura 2.10 Modelo Conceptual de Datos	59
Figura 2.11 Modelo Físico de la Base de Datos.....	65
Figura 2.12 Interfaz de Registro de Estudiantes	67

Figura 2.13 Interfaz de Gestión de Aportes.....	67
Figura 2.14 Interfaz de Registro de Pagos.....	68
Figura 2.15 Código SQL.....	69
Figura 2.16 Ejemplo de RESTful Web Services.....	70
Figura 2.17 Código Fuente para buscar un estudiante.....	70
Figura 2.18 Código Fuente para actualizar los datos de un estudiante.....	71
Figura 2.19 Código Fuente para validar cédula.....	71
Figura 2.20 Código Fuente para validar el número telefónico.....	72
Figura 2.21 Código Fuente para registrar estudiante.....	73
Figura 2.22 Código Fuente para registrar el pago del aporte.....	73
Figura 2.23 Código Fuente para registrar la desafiliación de un estudiante.....	74
Figura 2.24 Código Fuente para emitir mensajes.....	74
Figura 2.25 Interfaz de Gestión de Proveedores.....	75
Figura 2.26 Interfaz de Gestión de Productos.....	75
Figura 2.27 Interfaz de Registro de Ventas.....	76
Figura 2.28 Interfaz de Gestión de Clubes.....	76
Figura 2.29 Interfaz de Matriculación en Clubes.....	77
Figura 2.30 Interfaz de Registro de Aprobación.....	77
Figura 2.31 Código Fuente para buscar un producto.....	79
Figura 2.32 Código Fuente para registrar la venta.....	79
Figura 2.33 Código Fuente para mostrar clubes.....	80
Figura 2.34 Código Fuente para registrar a un estudiante en un club.....	80
Figura 2.35 Código Fuente para registrar la aprobación del estudiante.....	80
Figura 2.36 Interfaz para Publicar Eventos.....	81
Figura 2.37 Interfaz para Gestión de Multas.....	82
Figura 2.38 Interfaz para Registrar Asistencia.....	82
Figura 2.39 Interfaz para Gestión de Disciplinas Deportivas.....	83
Figura 2.40 Interfaz para Registrar Equipo.....	83
Figura 2.41 Interfaz para Registrar Jugador.....	84
Figura 2.42 Interfaz para Registrar Jugador en las Disciplinas Deportivas.....	84
Figura 2.43 Interfaz para Gestión de Dispositivos.....	85
Figura 2.44 Interfaz para registrar el préstamo de un dispositivo.....	85
Figura 2.45 Código Fuente para registrar un evento.....	87

Figura 2.46 Código Fuente para mostrar eventos	87
Figura 2.47 Código Fuente para asignar multas	88
Figura 2.48 Código Fuente para registrar un equipo.....	88
Figura 2.49 Código Fuente para registrar un jugador.....	89
Figura 2.50 Código Fuente para mostrar categorías disponibles	89
Figura 2.51 Código Fuente para buscar un dispositivo	90
Figura 2.52 Código Fuente para registrar el préstamo de un dispositivo	90
Figura 2.53 Código Fuente para mostrar los dispositivos prestados.....	91
Figura 2.54 Interfaz para Gestión de Casilleros	92
Figura 2.55 Interfaz para Alquiler de Casilleros.....	92
Figura 2.56 Interfaz para Gestión de Usuarios.....	93
Figura 2.57 Interfaz para Inicio de Sesión	93
Figura 2.58 Interfaz para mostrar la misión de la AEIS	94
Figura 2.59 Interfaz para mostrar la visión de la AEIS	94
Figura 2.60 Código Fuente para mostrar casilleros disponibles.....	95
Figura 2.61 Código Fuente para registrar el alquiler de un casillero	96
Figura 2.62 Código Fuente para la autenticación de usuario	96
Figura 2.63 Código Fuente para mostrar el menú de acuerdo al tipo de usuario .	97
Figura 2.64 Gráfico de Esfuerzo del Primer Sprint.....	109
Figura 2.65 Gráfico de avance de tareas en el Primer Sprint.....	109
Figura 2.66 Gráfico de Esfuerzo del Segundo Sprint	110
Figura 2.67 Gráfico de avance de tareas en el Segundo Sprint.....	110
Figura 2.68 Gráfico de Esfuerzo del Tercer Sprint	111
Figura 2.69 Gráfico de avance de tareas en el Tercer Sprint.....	111
Figura 2.70 Gráfico de Esfuerzo del Cuarto Sprint.....	112
Figura 2.71 Gráfico de avance de tareas en el Segundo Sprint.....	112
Figura 3.1 Creación de la Base de Datos – Parte I	116
Figura 3.2 Creación de la Base de Datos – Parte II	117
Figura 3.3 Restauración de la Base de Datos – Parte I	117
Figura 3.4 Restauración de la Base de Datos – Parte II	118
Figura 3.5 Ejecución del Script Setup Glassfish.....	118
Figura 3.6 Ejecución del Script Setup Backend.....	119
Figura 3.7 Despliegue del Servidor de Backend	119

Figura 3.8 Instalación de Node.js	120
Figura 3.9 Instalación de GIT – Parte I.....	120
Figura 3.10 Instalación de GIT – Parte II.....	121
Figura 3.11 Instalación de GIT – Parte III.....	121
Figura 3.12 Página de Inicio del Portal en el navegador Chrome	122
Figura 3.13 Página de Inicio del Portal en el navegador Internet Explorer.....	123
Figura 3.14 Página de Inicio del Portal en el navegador Firefox	123
Figura 3.15 Resultados Primera Pregunta	132
Figura 3.16 Resultados Segunda Pregunta	133
Figura 3.17 Resultados Tercera Pregunta	133
Figura 3.18 Resultados Cuarta Pregunta	134
Figura 3.19 Resultados Quinta Pregunta	134
Figura 3.20 Resultados Sexta Pregunta.....	134
Figura 3.21 Resultados Séptima Pregunta.....	135
Figura 3.22 Resultados Octava Pregunta	135
Figura 3.23 Resultados Novena Pregunta	136
Figura 3.24 Resultados Décima Pregunta.....	136
Figura 3.25 Resultados Undécima Pregunta.....	136
Figura 3.26 Resultados Duodécima Pregunta.....	137

ÍNDICE DE TABLAS

Tabla 1.1 Descripción de la Directiva de la AEIS	3
Tabla 1.2 Criterios para seleccionar el Gestor de Base de Datos	27
Tabla 1.3 Resultados de la Selección del Gestor de Base de Datos	27
Tabla 1.4 Criterios para seleccionar el IDE	29
Tabla 1.5 Resultados de la Selección del IDE.....	29
Tabla 2.1 Roles Scrum para el Portal Web de la AEIS	32
Tabla 2.2 Procesos que realiza la AEIS	32
Tabla 2.3 Formato para Describir Historias de Usuario.....	33
Tabla 2.4 Product Backlog Inicial	38
Tabla 2.5 Refinamiento del Product Backlog.....	38
Tabla 2.6 Primer Sprint Backlog.....	42
Tabla 2.7 Segundo Sprint Backlog	44
Tabla 2.8 Tercer Sprint Backlog	46
Tabla 2.9 Cuarto Sprint Backlog.....	48
Tabla 2.10 Tareas añadidas al Product Backlog	50
Tabla 2.11 Product Backlog Final.....	50
Tabla 2.12 Formato para analizar la creación de las entidades	60
Tabla 2.13 Formato Pruebas de Aceptación	98
Tabla 3.1 Recolección de información de estudiantes	113
Tabla 3.2 Recolección de información de proveedores.....	113
Tabla 3.3 Recolección de información de productos.....	114
Tabla 3.4 Recolección de información de inventario	114
Tabla 3.5 Requisitos de Hardware para los servidores de Backend	115
Tabla 3.6 Requisitos de Hardware para los servidores de Frontend.....	115
Tabla 3.7 Requisitos de Software para el servidor de Backend	115
Tabla 3.8 Requisitos de Software para el servidor de Frontend.....	115
Tabla 3.9 Medición de carga de las páginas del portal	124
Tabla 3.10 Resultados Pruebas de aceptación	124
Tabla 3.11 Prueba Registrar Estudiante	125
Tabla 3.12 Prueba Registrar Pago Aporte - Afiliación	126
Tabla 3.13 Prueba Registrar Pago Aporte - Desafiliación	127
Tabla 3.14 Prueba Registrar Ventas	127

Tabla 3.15 Prueba Matricular Estudiante en Club	128
Tabla 3.16 Prueba Registrar Aprobación del estudiante	128
Tabla 3.17 Prueba Publicar Información Evento	129
Tabla 3.18 Prueba Inscripción Equipo Deportivo	129
Tabla 3.19 Prueba Inscripción Equipo Deportivo	130
Tabla 3.20 Prueba Inscripción Equipo Deportivo	131
Tabla 3.21 Encuesta de Satisfacción para Usuarios Finales	132

RESUMEN

En el presente proyecto de titulación se desarrolló el Portal Web para la gestión de la Asociación de Estudiantes de la Facultad de Ingeniería de Sistemas (AEIS). Se realizó un análisis de la situación actual de la AEIS, que abarcó: estructura organizacional, servicios y actividades que brinda en beneficio de los estudiantes. Posteriormente, se describe de manera formal los procesos de la AEIS usando el Lenguaje de Modelado Unificado (UML).

En base a la descripción de los procesos se elaboraron las historias de usuario, estas historias priorizadas pasaron a formar el “Product Backlog”. Para la consecución del producto fue necesario la implementación de cuatro sprints de 9 días cada uno, al finalizar cada sprint, realizaron pruebas en conjunto con el Product Owner para validar los requisitos implementados.

El desarrollo se hizo usando el framework de JavaScript AngularJS con un backend funcionando sobre Java Enterprise Edition 7, la combinación de estas dos tecnologías permitió cumplir con los tiempos previstos para el desarrollo.

Luego de poner en producción el portal AEIS WEB APP, se realizó una encuesta a los usuarios finales y se concluye que se encuentran satisfechos cien por ciento con la funcionalidad del portal; además de opinar que el mismo tiene un diseño atractivo con un menú que facilita el acceso a los contenidos y que muestra de manera clara todas las actividades que se pueden realizar en él.

PRESENTACIÓN

Con el avance de la tecnología podemos generar soluciones innovadoras y eficaces para satisfacer las necesidades de las personas. En este proyecto de titulación con el fin de solventar las necesidades de la Asociación de Estudiantes de Ingeniería en Sistemas (AEIS) de la Escuela Politécnica Nacional se propuso la elaboración de un portal web llamado "AEIS WEB APP" que permite gestionar los procesos que realiza la AEIS, durante el desarrollo se aplicó la metodología ágil Scrum.

1 CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

El presente trabajo de titulación se encuentra orientado al entorno estudiantil, específicamente a la Asociación de Estudiantes de Ingeniería de Sistemas (AEIS), la cual es el ente de representación de los estudiantes ante las autoridades de la Escuela Politécnica Nacional (EPN) y la sociedad. Mediante este proyecto planteamos proveer una solución de software a varias de las problemáticas con respecto a la administración de la AEIS las cuales actualmente no cuentan con un proceso definido ni automatizado de tal forma que se mejore la atención y los servicios que se prestan a los estudiantes y a la EPN.

1.1 DESCRIPCIÓN DEL PROBLEMA

1.1.1 ENTORNO DE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA DE SISTEMAS

La AEIS es una organización gremial cuya principal misión es la de proporcionar una representación de los estudiantes en los aspectos académicos, culturales y deportivos dentro y fuera de la Escuela Politécnica Nacional. La AEIS está constituida por todos los estudiantes matriculados en la Facultad de Ingeniería de Sistemas de la Escuela Politécnica Nacional excepto aquellos que manifestaren por escrito su voluntad de no pertenecer a ella. [1]

A la vez la AEIS como ente gremial se encuentra relacionado directamente con las autoridades de la Facultad de Ingeniería de Sistemas y de igual manera forma parte de la agrupación conformada por las asociaciones de todas las carreras de la Escuela Politécnica Nacional la cual es la Federación de Estudiantes de la Politécnica Nacional (FEPON).

1.1.1.1 Misión

Representar y promover los derechos estudiantiles en los órganos de gobierno de la Universidad y fuera de ella, conforme a la legislación pertinente y al estatuto orgánico de la EPN, estimular la participación de los estudiantes en la vida interna de la Universidad.

1.1.1.2 Visión

Ser una asociación de calidad mediante la colaboración y participación de todos los estudiantes de la Facultad, para consolidarnos como una de las asociaciones más reconocidas a nivel nacional.

1.1.1.3 Valores

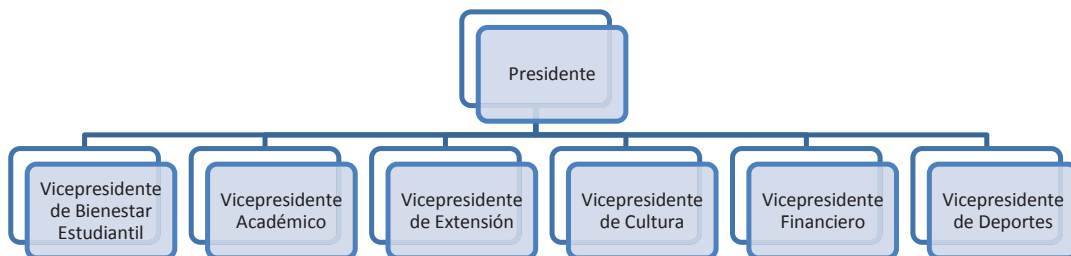
Los valores de la AEIS se enmarcan en el código de ética de la Escuela Politécnica Nacional, estos son:

- Respeto hacia sí mismo y hacia los demás
- Honestidad
- Verdad
- Compromiso con la Institución
- Liderazgo
- Calidad
- Transparencia

1.1.2 ESTRUCTURA DE LA DIRECTIVA DE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA DE SISTEMAS

La directiva estará conformada por el Presidente, Vicepresidente de Bienestar Estudiantil, Vicepresidente Académico, Vicepresidente de Extensión, Vicepresidente de Cultura, Vicepresidente Financiero, Vicepresidente de Deportes; los suplentes y vocales [1], como se observa en la Figura 1.1.

Figura 1.1 Estructura de la Directiva de la AEIS



Elaborado por: Los autores

1.1.2.1 Descripción de las Funciones de la Directiva de la AEIS

En la Tabla 1.1 se presentan las funciones de cada miembro de la directiva de la AEIS.

Tabla 1.1 Descripción de la Directiva de la AEIS

Cargo	Función
Vicepresidente Académico	Se encarga de los eventos académicos de la asociación.
Vicepresidente Financiero	Cumple las funciones de tesorero de la asociación.
Vicepresidente de Bienestar Estudiantil	Se encarga de velar por el Bienestar de los Estudiantes de la asociación.
Vicepresidente de Extensión	Cumple las funciones de secretario de la asociación.
Vicepresidente de Deportes	Se encarga de los eventos deportivos de la asociación.
Vicepresidente de Cultura	Se encarga de los eventos culturales de la asociación.

Elaborado por: Los autores

1.1.3 SERVICIOS Y ACTIVIDADES QUE OFRECE LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA DE SISTEMAS

La AEIS además de ser un ente de representación estudiantil brinda diferentes tipos de servicios a sus miembros, como son:

- Servicio de Venta de Snacks y Bebidas
- Servicio de Préstamo de Dispositivos
- Alquiler de Casilleros

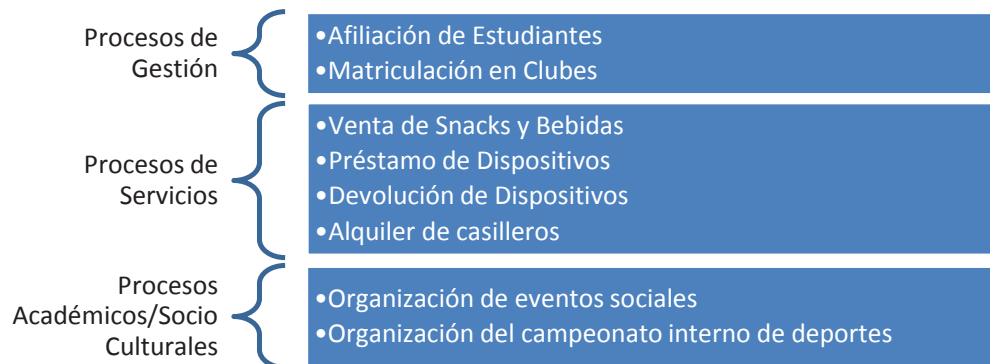
De igual forma la asociación organiza diferentes actividades con el objetivo de motivar la integración estudiantil dentro de la Facultad. Las actividades que se realizan cada semestre son las siguientes:

- Organización de eventos: Bautizo a los estudiantes de primer semestre, Baile de Gala, asambleas informativas.
- Organización del Campeonato Interno de deportes.
- Clubes para los estudiantes de primer semestre.

1.1.4 DESCRIPCIÓN DE PROCESOS QUE REALIZA LA AEIS

En la Figura 1.2 se pueden observar los procesos que realiza la AEIS:

Figura 1.2 Procesos de la AEIS



Elaborado por: Los autores

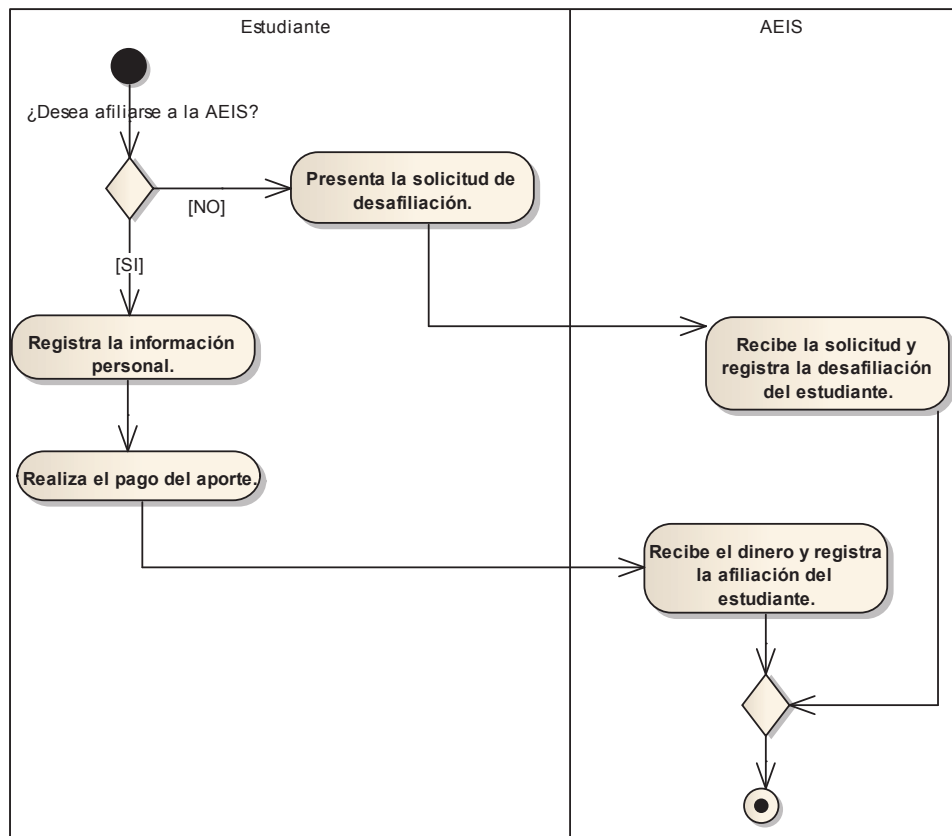
1.1.4.1 Afiliación de Estudiantes

Nombre: Proceso de Afiliación
Descripción: <p>La directiva de la AEIS inicia sus actividades semestrales cobrando un aporte voluntario a todos los estudiantes de la Facultad de Sistemas que deseen formar parte de la misma, en caso de que un estudiante no desee afiliarse durante un periodo académico, tiene que realizar una solicitud de desafiliación por duplicado dirigida al presidente de la AEIS.</p>
Actores: <ol style="list-style-type: none"> 1. Estudiante 2. Vicepresidente Financiero
Pre Condiciones: <ul style="list-style-type: none"> - Para afiliarse a la AEIS el estudiante debe estar legalmente matriculado en la Facultad de Ingeniería de Sistemas. - El estudiante no debe poseer impedimento para pertenecer a asociaciones estudiantiles ya sea por sanción o disposición de alguna autoridad de la EPN.
Post condiciones <ul style="list-style-type: none"> - Si el estudiante realiza el pago del aporte queda registrado como

afiliado. - Si el estudiante no realiza el pago del aporte queda registrado como desafiado.		
Entrada	Actividades	Salida
El estudiante decide afiliarse a la AEIS.	- El estudiante registra los siguientes datos: cédula, nombres completos, número único - El estudiante realiza el pago del aporte.	Comprobante de afiliación.

Flujo

Figura 1.3 Afiliación de Estudiantes



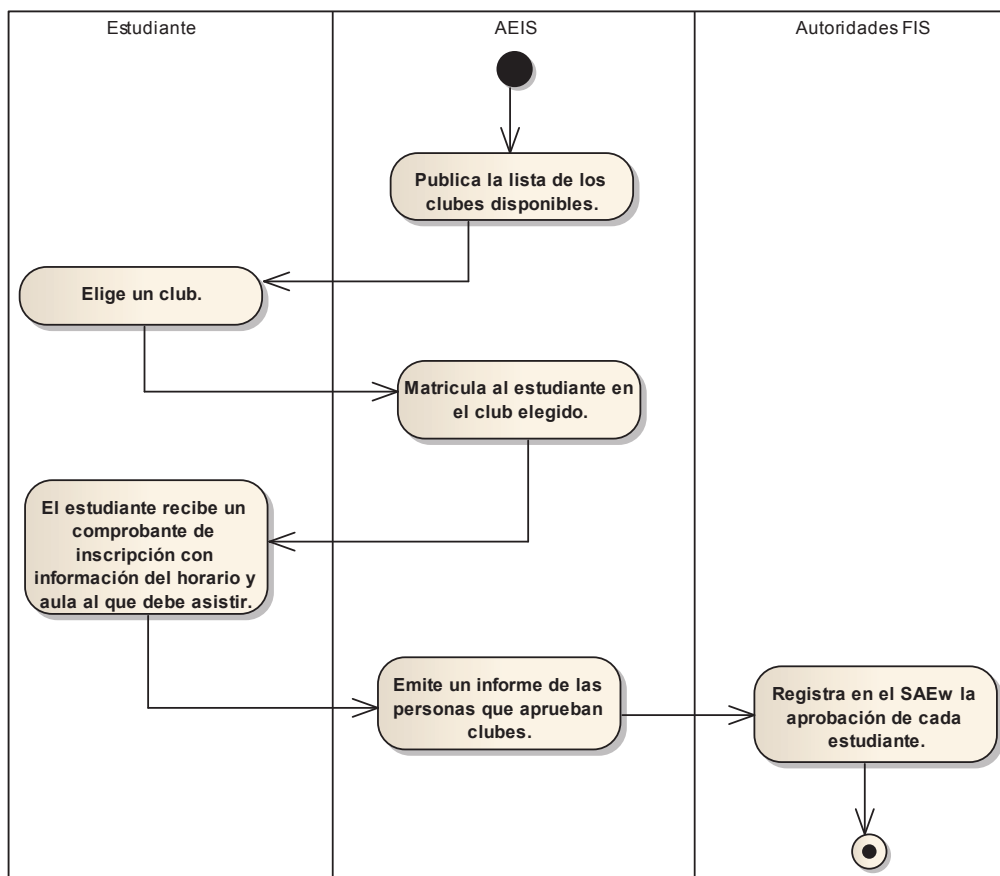
Elaborado por: Los autores

1.1.4.3 Matriculación en Clubes

Nombre: Oferta de Clubes		
Descripción: La AEIS oferta diferentes clubes con actividades culturales que forman parte de una materia obligatoria para los estudiantes de primer semestre.		
Actores: <ul style="list-style-type: none"> - Estudiante - Vicepresidente Académico - Autoridades FIS 		
Pre Condiciones: <ul style="list-style-type: none"> - El estudiante debe estar legalmente matriculado en la Facultad de Ingeniería de Sistemas. 		
Post condiciones <ul style="list-style-type: none"> - El estudiante queda registrado como alumno de un club ofertado por la AEIS. 		
Entrada	Actividades	Salida
Lista de clubes publicada por la AEIS.	<ul style="list-style-type: none"> - El estudiante elige un club al cual inscribirse registrando su nombre completo, cédula y número único. - La AEIS matricula al estudiante en el club escogido. - El estudiante recibe un comprobante de inscripción con su horario e información sobre su club y profesor. - Al final del semestre la AEIS emite un informe con los alumnos aprobados y reprobados. 	Lista de estudiantes que aprobaron clubes para el registro en el SAEW.

Flujo

Figura 1.4 Matriculación en Clubes



Elaborado por: Los autores

1.1.4.4 Venta de Snacks y Bebidas

Nombre: Venta de Snacks y Bebidas**Descripción:**

El bar de la AEIS ofrece a los estudiantes diferentes snacks y bebidas. El bar está a cargo de dos personas que trabajan a medio tiempo que se encargan de registrar manualmente las ventas diarias en una hoja de control de acuerdo a la Figura 1.5.

Otra de las tareas de los encargados de las ventas es realizar los pedidos de productos para el bar cuando existen pocos productos en stock.

Figura 1.5 Hoja de Control de Ventas

CONTROL DE VENTAS AEIS			
Fecha:			
Producto	Precio	Cantidad de Articulos Vendidos	Total
LÁCTEOS			
Alpina con Fruta y Regeneris	0.65		
Avena	0.55		
Avena Leche y Frutas	0.65		
Bonyurt Galleta	0.75		
Caffe lato	1.10		
Gelatoni	0.65		
Leche Entera	0.50		
Leche Toni	0.70		
Manjar de Leche	0.35		
Tampico	0.40		
Tampico 350cc	0.50		
Tampico ½l	0.60		
Toni Frush	0.75		
Toni mix	0.75		
Toni Bebible	0.65		
Yogurt Bebible	0.65		
Yogurt Met	0.75		
Yogurt Mix	0.75		

Elaborado por: Los Autores

Actores:

- Empleado
- Cliente

Pre Condiciones:

- El cliente debe poseer dinero para realizar la compra.

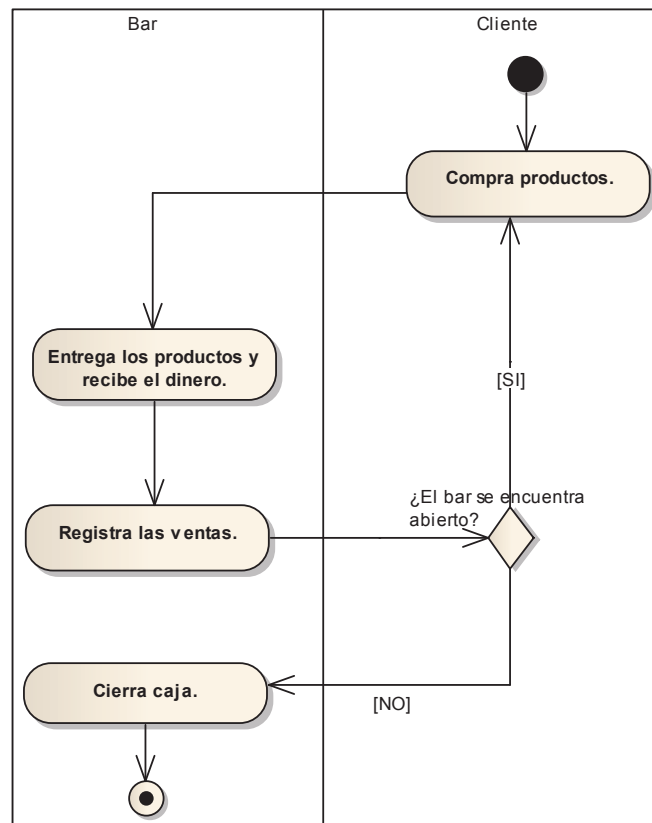
Post condiciones

- Se registra la venta en el sistema.

Entrada	Actividades	Salida
Cliente compra productos en el bar de la AEIS.	<ul style="list-style-type: none"> - El bar entrega los productos y recibe dinero a cambio. - La venta se registra con los siguientes datos: nombre del producto y cantidad. 	Reporte diario de ventas.

Flujo

Figura 1.6 Venta de Snacks y Bebidas



Elaborado por: Los autores

1.1.4.5 Préstamo de Dispositivos

Nombre: Préstamo de Dispositivos**Descripción:**

El préstamo de dispositivos que realiza la AEIS incluye el alquiler de proyectores, computadores y equipos de red.

Actores:

- Estudiante
- Empleado

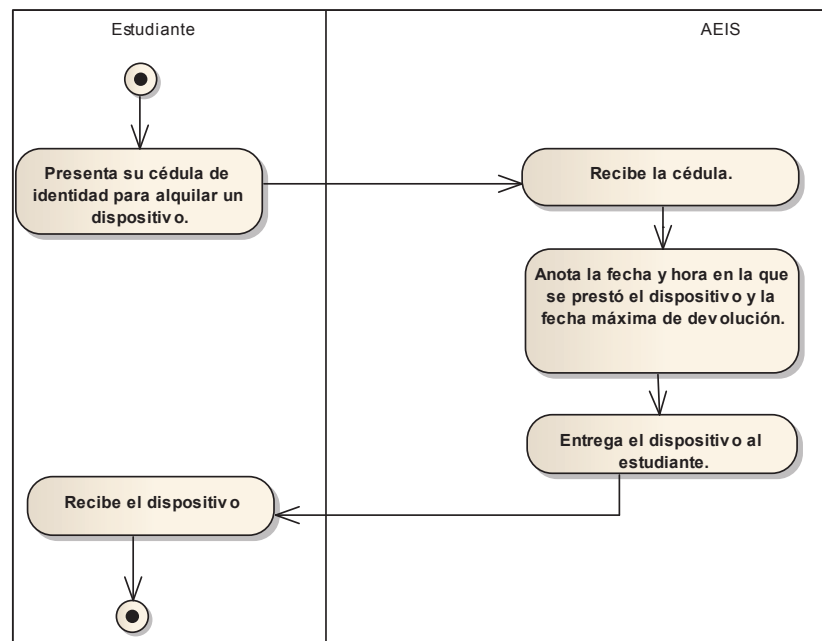
Pre Condiciones:

- El estudiante debe ser miembro de la AEIS para que se le pueda prestar el dispositivo.
- El estudiante debe poseer su cédula para autorizar el préstamo.

Post condiciones		
- Queda registrado el préstamo de un dispositivo a nombre de un estudiante.		
Entrada	Actividades	Salida
El estudiante solicita el préstamo de un equipo dejando como garantía su cédula.	- La AEIS recibe la cédula y registra el préstamo con los siguientes datos: Nombre del estudiante al que se prestó, su cédula y la fecha en la que se realiza el mismo y la fecha máxima de devolución.	Se registra el préstamo del dispositivo.

Flujo

Figura 1.7 Préstamo de Dispositivos



Elaborado por: Los autores

1.1.4.6 Devolución de Dispositivos

Nombre: Devolución de Dispositivos

Descripción:

La devolución de dispositivos que realizan los estudiantes a la AEIS.

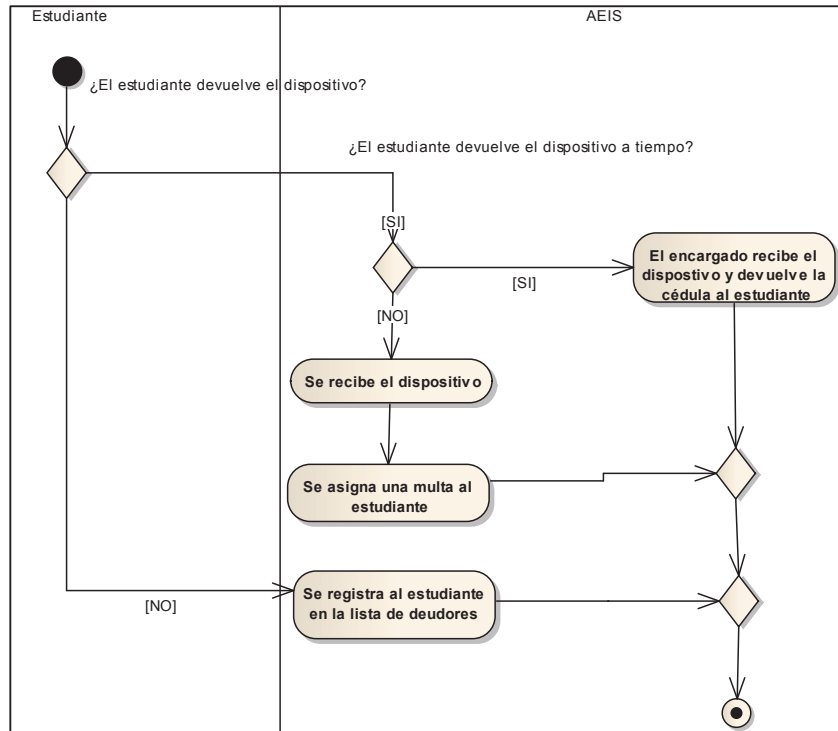
Actores:

- Estudiante

<ul style="list-style-type: none"> - Empleado 		
Pre Condiciones: <ul style="list-style-type: none"> - El dispositivo debe estar en el mismo estado en el que fue prestado. 		
Post condiciones <ul style="list-style-type: none"> - Queda registrada la devolución del dispositivo. 		
Entrada El estudiante entrega el dispositivo.	Actividades <ul style="list-style-type: none"> - La AEIS recibe el dispositivo y verifica que se encuentre en las mismas condiciones en las que fue prestado en caso de que no se haya devuelto el dispositivo en el tiempo establecido se le asigna una multa al estudiante. - La AEIS devuelve la cédula al estudiante. 	Salida Se registra la devolución del dispositivo.

Flujo

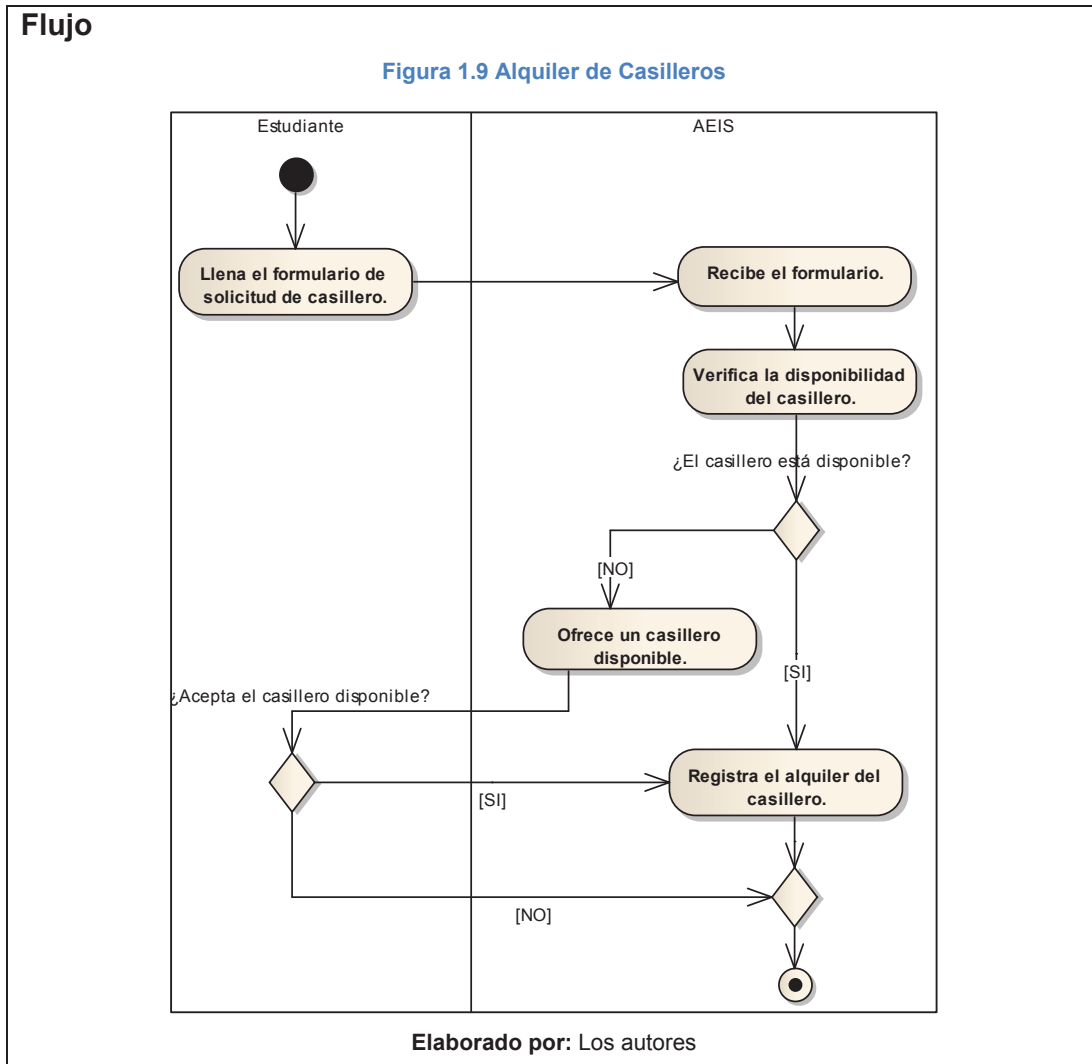
Figura 1.8 Devolución de Dispositivos



Elaborado por: Los autores

1.1.4.7 Alquiler de Casilleros

Nombre: Alquiler de Casilleros		
Descripción: La AEIS pone a disposición de los estudiantes casilleros que pueden arrendar semestralmente.		
Actores: <ul style="list-style-type: none"> - Empleado - Estudiante 		
Pre Condiciones: <ul style="list-style-type: none"> - El estudiante debe ser miembro de la AEIS para que se pueda alquilar un casillero. 		
Post condiciones <ul style="list-style-type: none"> - El casillero queda registrado a nombre de un estudiante durante un semestre. 		
Entrada	Actividades	Salida
El estudiante solicita el alquiler del casillero llenando un formulario con los siguientes datos: Nombre completo, número único, cédula y casillero que desea alquilar.	<ul style="list-style-type: none"> - La AEIS recibe este formulario y verifica la disponibilidad de este casillero. - En caso de que el casillero solicitado esté disponible, se ofrece otro casillero. 	Registro del casillero alquilado por un estudiante.



1.1.4.8 Organización de Eventos Sociales

Nombre: Organización de Eventos Sociales

Descripción:

A lo largo del semestre la AEIS organiza diferentes eventos para los estudiantes, esta información se publica de manera impresa en las carteleras de la Facultad.

Si un estudiante no asiste a un evento de carácter obligatorio se asigna una multa.

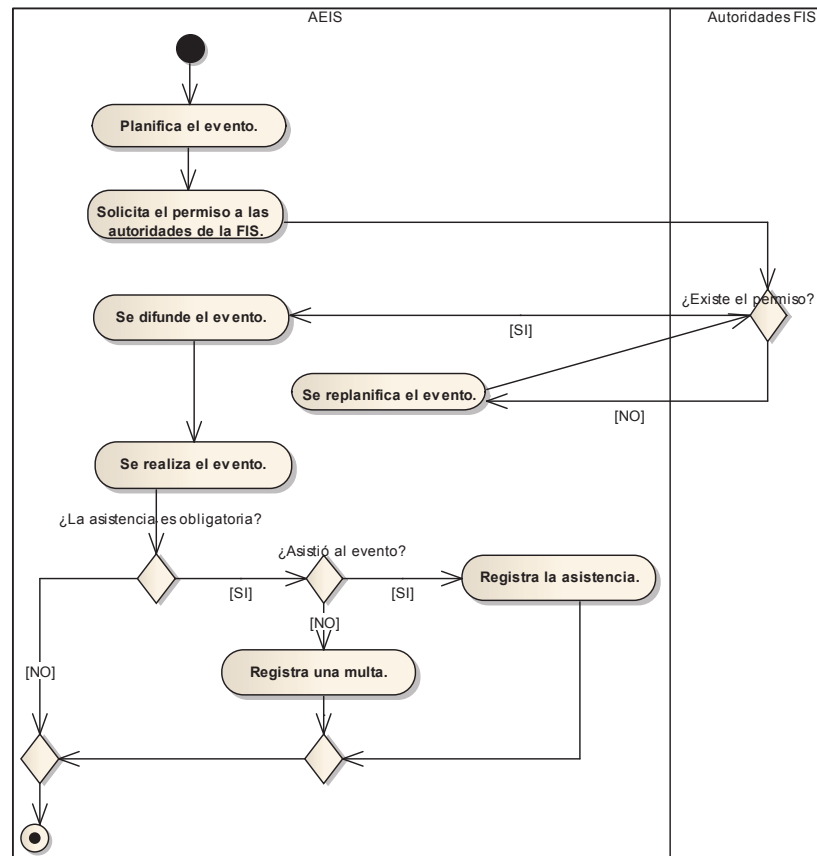
Actores:

- Autoridades FIS
- Vicepresidente de Cultura

- Estudiante		
Pre Condiciones:		
- Para la asistencia a eventos oficiales de la AEIS es necesario que el estudiante sea afiliado a la misma.		
Post condiciones		
- La asistencia del estudiante al evento queda registrada.		
- Si el estudiante no asiste a un evento obligatorio recibe una multa.		
Entrada	Actividades	Salida
La AEIS solicita el permiso para realizar un evento a las autoridades de la FIS.	<ul style="list-style-type: none"> - Una vez otorgado el permiso se difunde la información sobre el evento. - Se realiza el evento en donde la AEIS registra la asistencia. 	Registro de multas por inasistencia.

Flujo

Figura 1.10 Organización de Eventos Sociales



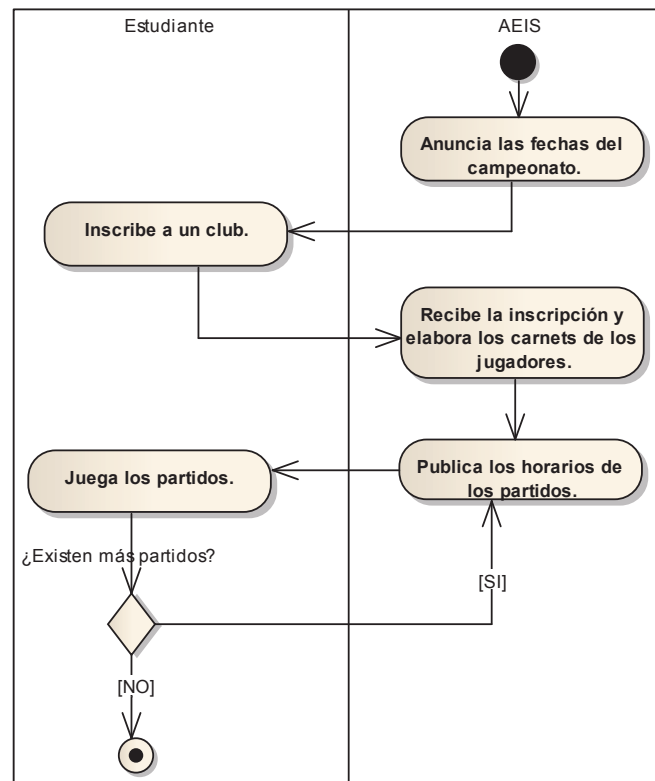
Elaborado por: Los autores

1.1.4.9 Organización del Campeonato Interno de Deportes

Nombre: Organización del Campeonato Interno de Deportes		
Descripción: Para participar en el campeonato interno de deportes cada equipo registra a los jugadores en las respectivas disciplinas que van a participar. Posteriormente el Vicepresidente de Deportes organiza las fechas de los partidos, y lleva el control de la tabla de posiciones para premiar a los equipos ganadores.		
Actores: <ul style="list-style-type: none"> - Estudiante - Vicepresidente de Deportes 		
Pre Condiciones: <ul style="list-style-type: none"> - Los estudiantes que deseen participar deben estar legalmente matriculados en la EPN o a su vez trabajar en la Institución. 		
Post condiciones <ul style="list-style-type: none"> - El equipo queda inscrito para jugar el campeonato interno. 		
Entrada	Actividades	Salida
Información personal de cada jugador.	<ul style="list-style-type: none"> - La AEIS anuncia las fechas del campeonato. - El estudiante inscribe a su club deportivo con los siguientes datos: nombre del club, representante, miembros y disciplinas en las que jugará. - La AEIS recibe la inscripción y elabora los carnets de los jugadores. 	Lista de equipos inscritos.

Flujo

Figura 1.11 Inscripción en el Campeonato Interno de Deportes



Elaborado por: Los autores

1.1.5 PROBLEMAS QUE SE PRESENTAN EN LOS PROCESOS QUE REALIZA LA AEIS

1.1.5.1 Proceso de Afiliación

No existen datos completos de los afiliados, o se encuentran desactualizados.

1.1.5.2 Oferta de Clubes

Existe dificultad debido a que los estudiantes no conocen los clubes que se ofertan durante un semestre, además de que el proceso de inscripción en los mismos se realiza de forma manual, por lo que los registros podrían extraviarse y no se puede pasar la nota del estudiante para el registro en el SAEw.

1.1.5.3 Venta de Snacks y Bebidas

El control del stock no se encuentra estandarizado y se lo realiza de manera intuitiva por lo que hay ocasiones en las que se tiene más producto del que se vende y se invierte más dinero del necesitado.

Al final de cada turno, el empleado realiza el cierre de caja de manera manual siendo esta una tarea difícil, tediosa e imprecisa además de que conlleva mucho tiempo ya que se necesita hacer un conteo manual de las ventas realizadas y cuadrarlas con la caja del bar de la AEIS.

1.1.5.4 Préstamo de Dispositivos

No se mantiene un control de las personas que han pedido un dispositivo y no lo han devuelto por lo que es difícil llevar un control del inventario.

1.1.5.5 Alquiler de Casilleros

No se mantiene un registro del nivel de ocupación de los casilleros. Por lo que no se analiza la rentabilidad de implementar más casilleros.

1.1.5.6 Gestión de Inventario.

No se tiene un control riguroso, ya que el usar una hoja para anotar información importante ésta se puede perder o deteriorar poniendo en riesgo la información contenida.

1.1.5.7 Eventos

Los eventos que realiza la AEIS no se promocionan de manera adecuada, lo que conlleva a la poca afluencia de estudiantes a los mismos.

En el caso de los eventos que requieren entrada, el pago se registra de manera manual lo que dificulta llevar un control de ingresos.

1.1.5.8 Campeonato Interno de Deportes

Existe dificultad al tabular la información de los equipos y estudiantes participantes dentro del campeonato, y llevar un control adecuado sobre el número de jugadores inscritos en cada disciplina.

1.1.5.9 Encuesta a Estudiantes

Se realizó una encuesta a los estudiantes de la Facultad de Ingeniería de Sistemas que se encuentra entre primer y octavo semestre de la carrera. La

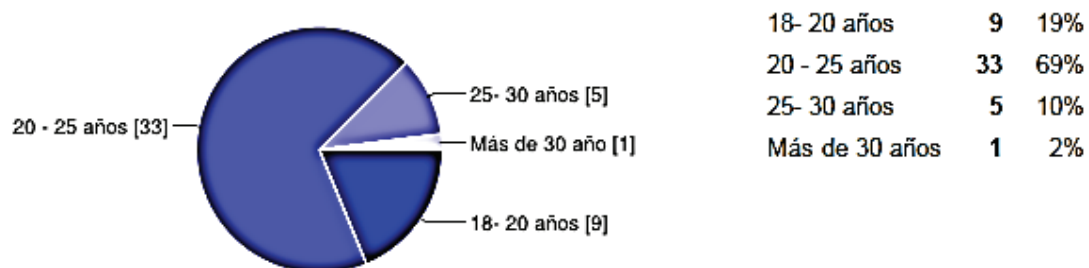
encuesta consistió en 8 preguntas de respuesta múltiple que se enfocaron en obtener información acerca de la opinión general sobre la forma en que se lleva la gestión administrativa de la AEIS y además de preguntar cuan necesario consideran la implementación de una plataforma que además de ayudar con la gestión a la asociación pueda servir como medio para difundir información entre los estudiantes.

La encuesta arrojó los siguientes resultados:

1.1.5.9.1 Primera Pregunta

¿Su edad se encuentra entre?

Figura 1.12 Respuesta Primera Pregunta

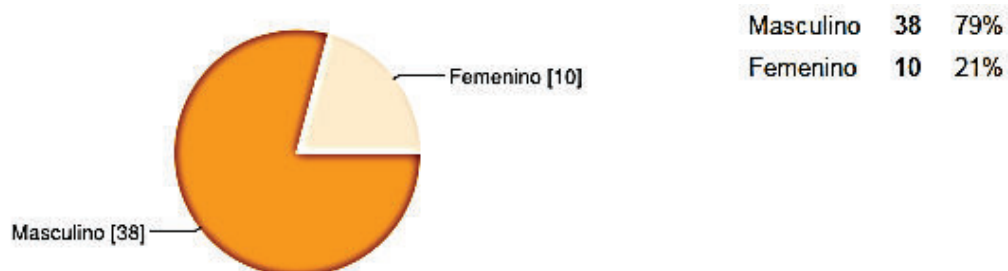


Elaborado por: Los autores

1.1.5.9.2 Segunda Pregunta

Género

Figura 1.13 Respuesta Segunda Pregunta

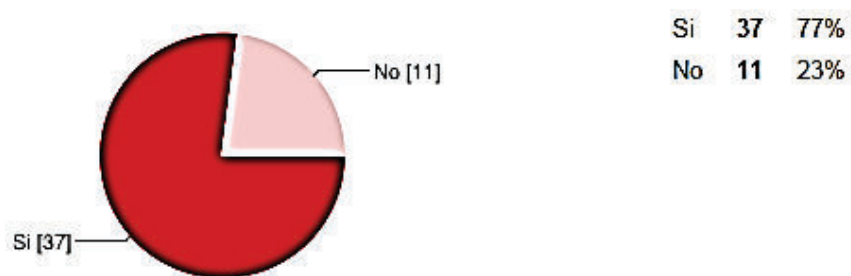


Elaborado por: Los autores

1.1.5.9.3 Tercera Pregunta

¿Se encuentra usted afiliado a la AEIS?

Figura 1.14 Respuesta Tercera Pregunta

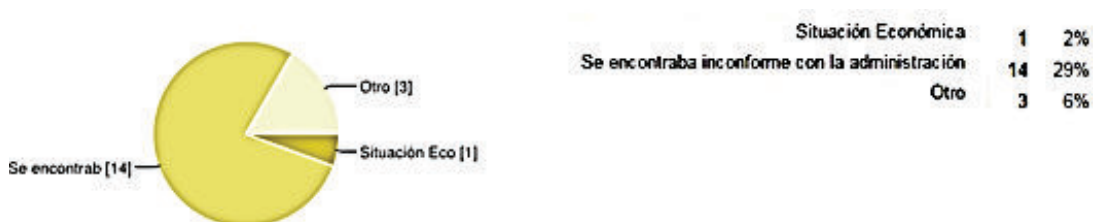


Elaborado por: Los autores

1.1.5.9.4 Cuarta Pregunta

En caso de que se haya desafiado de la AEIS. ¿Cuál fue la razón de que haya tomado esa decisión?

Figura 1.15 Respuesta Cuarta Pregunta

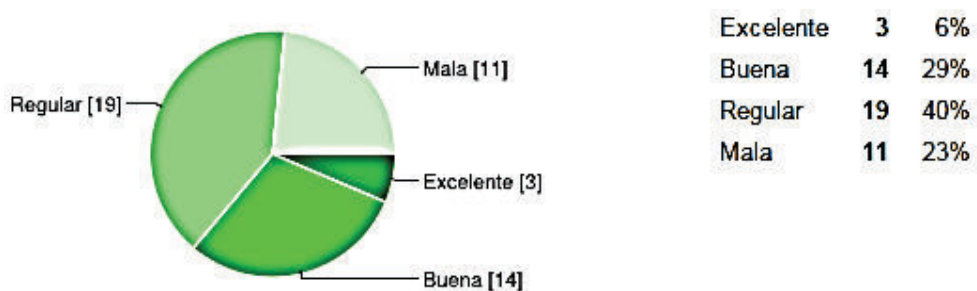


Elaborado por: Los autores

1.1.5.9.5 Quinta Pregunta

¿Cómo calificaría usted la eficiencia con la que se realizan las gestiones administrativas de la AEIS?

Figura 1.16 Respuesta Quinta Pregunta

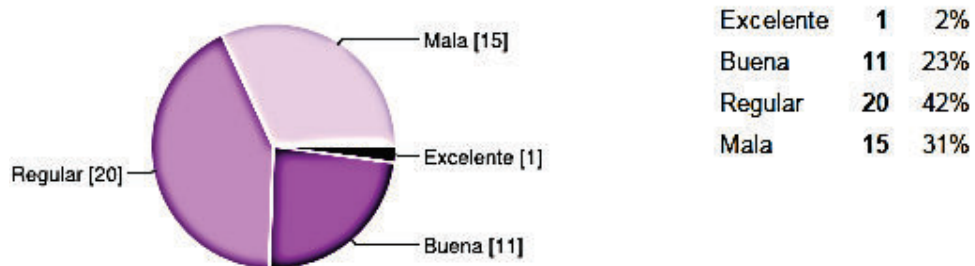


Elaborado por: Los autores

1.1.5.9.6 Sexta Pregunta

¿Cómo calificaría actualmente la forma en la que la AEIS proporciona información importante a los estudiantes?

Figura 1.17 Respuesta Sexta Pregunta

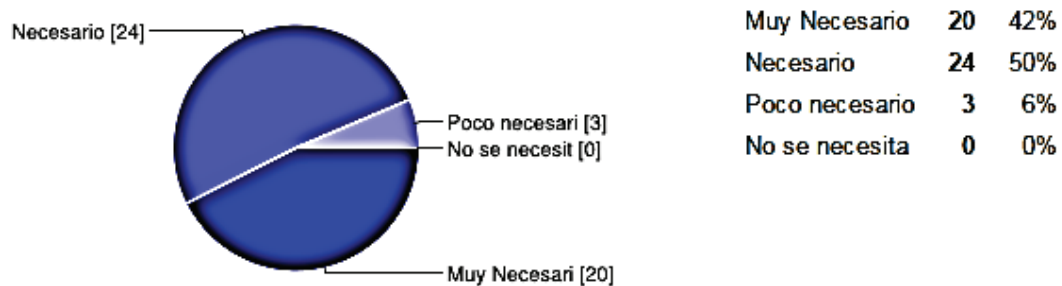


Elaborado por: Los autores

1.1.5.9.7 Séptima Pregunta

¿Cómo calificaría usted la necesidad de un portal web en dónde se muestra información importante a los estudiantes?

Figura 1.18 Respuesta Séptima Pregunta

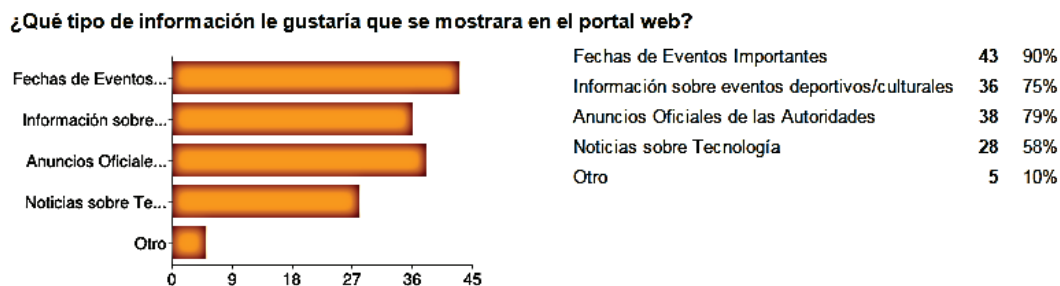


Elaborado por: Los autores

1.1.5.9.8 Octava Pregunta

¿Qué tipo de información le gustaría que se mostrara en el portal web?

Figura 1.19 Respuesta Octava Pregunta



Elaborado por: Los autores

A partir de estos resultados se pudieron sacar las siguientes conclusiones:

- La mayor parte de los estudiantes de la Facultad son hombres que se encuentran entre 20 y 30 años.
- Existe un porcentaje alto de desafiliación el cual es del 22%
- De las personas desafiadas el 29 % se desafilió por estar en desacuerdo con la administración.
- El 40% califica a la administración de la AEIS como regular y el 42% califica de la misma manera a la forma en la que la asociación difunde información a sus afiliados.
- El 50 % considera que es necesario y el 42% considera que es muy necesario un portal web donde se muestre información a los estudiantes.
- En orden de prioridad los estudiantes demandan información sobre eventos importantes, fechas de eventos culturales y deportivos, anuncios oficiales y noticias sobre tecnología.

1.1.6 SITUACIÓN PROPUESTA

Luego de analizar los problemas que se presentan en cada uno de los procesos que realiza la AEIS, se plantea la implantación de un portal web en dónde los estudiantes tendrán los siguientes servicios:

- Informarse sobre los eventos que se realizan en la EPN, resoluciones académicas, reuniones y eventos propios de la AEIS como: el campeonato interno de deportes, clubes, bautizo, convivencias y el alquiler de casilleros.
- Ser un medio de comunicación directa con la directiva de la AEIS, para que puedan transmitir sus comentarios y sugerencias en cuanto a los ámbitos: de cultura, bienestar estudiantil y deportes.

- Consultar si tienen multa por inasistencia a cualquier evento realizado por la asociación.

Para los afiliados a la AEIS el portal web permitirá:

- Gestionar la oferta de clubes (materia obligatoria) para los estudiantes de primer semestre.
- Realizar el registro de afiliación y los cobros cada semestre.
- Llevar un control de las ventas del bar de la AEIS.
- Gestionar el campeonato interno: inscripciones y publicar los horarios de los partidos.
- Llevar el inventario de los bienes de la AEIS.
- Realizar el alquiler de casilleros para los estudiantes.
- Registrar el préstamo de dispositivos.

1.2 DESCRIPCIÓN DEL ENTORNO DE DESARROLLO CON LA METODOLOGÍA SCRUM

Para desarrollar el proyecto se optó por seguir la metodología de ágil: Scrum.

1.2.1 ORÍGENES

Scrum es una metodología ágil para gestionar proyectos de software que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80. Aunque surgió como práctica en el desarrollo de productos tecnológicos, resulta válido en los entornos que trabajan con requisitos inestables, y necesitan rapidez y flexibilidad; situaciones habituales en el desarrollo de algunos sistemas de software. [2]

Scrum es una metodología de desarrollo muy simple que requiere de mucho trabajo, dado que su gestión no se basa en el seguimiento de un plan, sino en la adaptación continua sobre la marcha. Scrum presenta las siguientes características: [2]

- Es un modo de desarrollo de carácter adaptable.
- Orientado a las personas antes que los procesos.
- Emplea desarrollo ágil iterativo e incremental.

En la Figura 1.20 se muestran los roles, artefactos y eventos de Scrum.

Figura 1.20 Scrum



Fuente: Isla Visual [3]

1.2.2 ROLES

En Scrum existen tres roles principales: Product Owner, Scrum Master y Equipo.

1.2.2.1 Product Owner

El dueño del producto es una persona (no una comisión) que tiene conocimiento del entorno de negocio del cliente y de la visión del producto. Es el responsable de la financiación necesaria para el proyecto y también de decidir cómo debe ser el resultado final, del lanzamiento y del retorno de la inversión. [2]

Se encarga de gestionar el Product Backlog: expresa de manera clara los elementos del Product Backlog y se asegura que el equipo entienda estos elementos, es decir que el Product Backlog sea visible, transparente y claro para el equipo. [4]

1.2.2.2 Scrum Master

El Scrum Master es un líder que está al servicio del Equipo Scrum, se asegura que el equipo trabaje de acuerdo a la teoría, prácticas y reglas de Scrum. [4]

El Scrum master trabaja junto al Dueño del Producto para que éste logre crear y mantener el Product Backlog y trabaja junto al equipo de Desarrollo para

encontrar e implementar las prácticas técnicas que les permitirán tener un incremento de producto al final de cada Sprint. [5]

1.2.2.3 Equipo

Es el grupo de personas multidisciplinar que cubre todas las habilidades necesarias para construir el producto que el cliente requiere [2]. Este equipo se auto gestiona, auto organiza y dispone de atribuciones suficientes para tomar decisiones sobre cómo realizar su trabajo y crear un incremento de producto. [2] [4]

El Dueño del Producto crea una lista ordenada de lo que hay que hacer y los miembros del equipo de desarrollo hacen un pronóstico de cuánto pueden realizar en un Sprint y deciden cómo lo van a llevar a cabo. [5]

1.2.3 ARTEFACTOS

Los artefactos de Scrum representan trabajo o valor en diversas formas, están diseñados para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto. [4]

1.2.3.1 Product Backlog

El Product Backlog es una lista que contiene las funcionalidades, requisitos, mejoras y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones de desarrollo. [2]

El Product Backlog es una herramienta de referencia para el equipo y por lo tanto es recomendable que al menos contenga la siguiente información:

- Identificador único de la funcionalidad o trabajo
- Descripción de la funcionalidad
- Campo o sistema de priorización
- Estimación

Es importante notar que se pueden añadir algunos otros campos dependiendo del tipo de proyecto, funcionamiento del equipo y la organización ya que el formato del Product Backlog no es cerrado. [2]

El dueño del producto es el responsable del Product Backlog: del contenido, disponibilidad y ordenación [4].

1.2.3.2 Sprint Backlog

El Sprint Backlog es una lista parcial de los elementos que componen el Product Backlog que han sido seleccionados para implementar en el Sprint (ciclo de corta duración en los cuales se lleva a cabo el trabajo) con el fin de entregar el incremento de producto.

El Sprint Backlog es una imagen visible en tiempo real del trabajo que el equipo se encuentra ejecutando o ha completado durante el Sprint. [4]

En esta lista se asigna la persona responsable que va a llevar a cabo cada tarea, se estima el tiempo de trabajo que le puede tomar realizarla.

En otras palabras, el sprint backlog descompone el proyecto en tareas de un tamaño adecuado para determinar un avance diario y además permite identificar riesgos y problemas que se puedan presentar y sin necesidad de grandes procesos de gestión.

Además tiene las siguientes características:

- Se lo realiza en forma conjunta con todos los miembros del equipo.
- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Solo puede ser modificado por el equipo.
- El rango permitido de duración de cada tarea es de 4 a 16 horas de trabajo.
- Tiene que ser visible para todo el equipo, de preferencia se usa una pizarra o una pared ubicada en el espacio donde trabaja el equipo.

1.2.3.3 Burndown Chart

El burndown chart es una manera de visualizar la correlación entre la cantidad de trabajo restante en cualquier punto y el progreso del equipo. [6]

1.2.4 EVENTOS

1.2.4.1 Sprint Planning

En esta reunión se planifica el trabajo que se va a realizar durante el Sprint y tiene como objetivo responder a las siguientes preguntas: [4]

- P1: ¿Qué puede entregarse en el incremento resultante del Sprint que comienza?
- P2: ¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento? [4]

Estas dos preguntas se resuelven en dos partes:

- Primera Parte (P1): se decide qué elementos del Product Backlog se van a desarrollar durante el Sprint.
- Segunda Parte (P2): se determinan tareas, se estima el esfuerzo necesario y se asigna un responsable para realizar cada tarea. [2]

En el Sprint Planning participan el Product Owner, el Scrum Master y el equipo, la duración puede variar entre 1 y 4 horas dependiendo del tamaño del Sprint.

1.2.4.2 Sprint Review

Es una reunión que se realiza al final de cada Sprint en dónde el equipo presenta al propietario del producto el incremento construido en el Sprint. [2]

En el Sprint Planning participan el Cliente, el Product Owner, el Scrum Master y el equipo, la duración puede variar de 1 a 3 horas aquí se recibe la retroalimentación del cliente.

1.2.4.3 Daily Scrum Meeting

Es una reunión diaria que tiene el equipo de desarrollo y dirigida por el Scrum Master, durante 15 minutos cada miembro indica las tareas que está realizando, los impedimentos que han surgido, las tareas que ha finalizado y el trabajo que presentará al día siguiente. [2]

1.2.4.4 Sprint Retrospective

Es una reunión que se realiza luego del Sprint Review y antes del Sprint Planning, el equipo tiene la oportunidad de inspeccionarse a sí mismo y crear un plan de mejoras para el siguiente Sprint.

El Scrum Master participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso Scrum recae sobre él. El propósito de la Retrospectiva de Sprint es: [4]

- Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas;
- Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras; y,
- Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

1.3 SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO

Para el desarrollo del portal web se usará software libre y Open Source, debido a que la AEIS no posee los recursos económicos suficientes para cubrir el costo de licenciamiento de un lenguaje de programación y base de datos propietarios.

1.3.1 GESTOR DE BASE DE DATOS

Un gestor de base de datos es un componente software encargado de garantizar el correcto, eficiente, íntegro y seguro acceso y almacenamiento de los datos. [7]

En la Tabla 1.2 se muestran los criterios que se tomaron en cuenta para la selección del gestor de base de datos.

Tabla 1.2 Criterios para seleccionar el Gestor de Base de Datos

Criterio	Descripción	Valoración	
Multiplataforma	El gestor funciona sobre diferentes sistemas operativos.	Cumple	1
		No Cumple	0
Cumplimiento ACID	Cumplimiento de: Atomicidad, Consistencia, Aislamiento y Durabilidad	Cumple	1
		No Cumple	0
Concurrencia	Permite que varios usuarios puedan acceder a la base de datos.	Cumple	1
		No Cumple	0
Soporte	Existe soporte del Software y recursos de capacitación	Cumple	1
		No Cumple	0
Nivel de conocimiento	Experiencia por parte de los desarrolladores en el uso del gestor de base de datos.	Ninguno	0
		Poco	1
		Mucho	2

Elaborado por: Los autores

Para seleccionar el gestor de base de datos se realizó el análisis entre MySQL y PostgreSQL. En la Tabla 1.3 se muestran los resultados obtenidos.

Tabla 1.3 Resultados de la Selección del Gestor de Base de Datos

Criterio	MySQL	PostgreSQL
Multiplataforma	1	1
Cumplimiento ACID	1	1
Concurrencia	1	1
Soporte	1	1
Nivel de conocimiento	1	2
TOTAL	5	6

Elaborado por: Los autores

El sistema gestor de base de datos que se utilizará es PostgreSQL, un sistema de base de datos relacional de código abierto, que posee las siguientes características: [8]

- Es una base de datos 100% ACID
- Integridad referencial
- Replicación asincrónica/sincrónica
- Commit de 2 fases
- PITR - point in time recovery
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL
- Completa documentación

1.3.2 LENGUAJE DE PROGRAMACIÓN

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. [9]

Para el desarrollo de la aplicación se utilizará el lenguaje Java.

1.3.3 ENTORNO DE DESARROLLO INTEGRADO

Un entorno de desarrollo integrado es una aplicación que proporciona utilidades para crear aplicaciones. Un IDE consta de un editor de código fuente, un depurador, un compilador, un sistema de control de versiones.

En la Tabla 1.4 se muestran los criterios que se tomaron en cuenta para la selección del entorno de desarrollo.

Tabla 1.4 Criterios para seleccionar el IDE

Criterio	Descripción	Valoración	
Multiplataforma	El IDE sobre diferentes sistemas operativos.	Cumple	1
		No Cumple	0
Control de Versiones	El IDE facilita el control de cambios en los archivos y permite recuperar versiones específicas.	Si	2
		Mediante Plugins	1
		No	0
Soporte	Existe soporte del IDE y recursos de capacitación	Cumple	1
		No Cumple	0
Nivel de conocimiento	Experiencia por parte de los desarrolladores en el uso del gestor de base de datos.	Ninguno	0
		Poco	1
		Mucho	2

Elaborado por: Los autores

Para seleccionar el gestor de base de datos se realizó el análisis entre NetBeans y Eclipse. En la Tabla 1.5 se muestran los resultados obtenidos.

Tabla 1.5 Resultados de la Selección del IDE

Criterio	NetBeans	Eclipse
Multiplataforma	1	1
Control de Versiones	2	1
Soporte	1	1
Nivel de conocimiento	2	1
TOTAL	6	4

Elaborado por: Los autores

Para el desarrollo de la aplicación web se utilizará el IDE NetBeans que provee un entorno de desarrollo Integrado tanto para programación de Frontend como programación de Backend al igual que soporta el estándar JEE 7 y ofrece compatibilidad con Frameworks de trabajo en JavaScript para Frontend como Backbone, NodeJS y AngularJS.

1.3.4 SERVIDOR DE APLICACIONES

Un servidor de aplicaciones proporciona servicios que soportan la ejecución y disponibilidad de aplicaciones desplegadas. [10] Un servidor de aplicaciones

recibe solicitudes, ejecuta la lógica del negocio y las transacciones solicitadas contactándose a la base de datos.

NetBeans viene incluido con el servidor de aplicaciones GlassFish por defecto, el cual será usado para soportar la aplicación que brindará servicios a la parte del Frontend.

1.3.5 FRAMEWORK PARA FRONTEND (ANGULARJS)

La programación de Frontend se la realizará usando HTML, CSS, JavaScript y todo esto bajo AngularJS el cual es un framework para JavaScript que permite implementar el patrón de arquitectura Modelo Vista Controlador del lado del cliente, además de facilitar la mantenibilidad del código mediante un desarrollo modular y de bajo acoplamiento. [11]

AngularJS permite programar interfaces personalizadas de manera simple y ordenada gracias al manejo de directivas las cuales permiten abstraer las declaraciones HTML y programar comportamientos especiales para las mismas así como ayuda a mejorar la interacción con los Web Services ya que posee una API para interactuar con Restful Services que abstrae la parte “complicada” de las llamadas a tramas JSON y las reduce a simples llamadas a métodos GET, POST y DELETE con muy pocos parámetros.

La Interfaz abstrae todo su comportamiento en el controlador de la misma el cual es el que se encarga de gestionar los modelos y la lógica del negocio, De igual forma todo, cambio realizado en la vista se gestiona y actualiza de manera asíncrona con la base de datos gracias al enlazamiento de datos bidireccional que provee AngularJS el cual se encarga de estar pendiente de los cambios realizados en la vista para actualizar el modelo y viceversa.

1.3.6 ENTORNO DE EJECUCIÓN EN EL FRONTEND (NODEJS)

NodeJS es un entorno de ejecución escrito totalmente en JavaScript que permite la construcción rápida y escalable de aplicaciones web. NodeJs usa una arquitectura orientada a eventos e implementa un modelo de entrada y salida asíncrono que lo hace ligero, eficiente e ideal para aplicaciones que manejan grandes cantidades de datos en tiempo real y que corren a la vez en diferentes dispositivos. [12]

NodeJS permite la instalación de módulos adicionales gracias a su propio gestor de paquetes llamado NPM. En el caso del desarrollo de la aplicación AEIS Web APP se usará un módulo llamado Grunt, el cual permite automatizar tareas tales como despliegue, compilación y pruebas lo que proporciona una mejora en la eficiencia y rapidez del desarrollo.

2 CAPÍTULO 2: DESARROLLO DEL PORTAL WEB

En el Capítulo I en la Sección 1.1.4 se describe los procesos que realiza la AEIS, en la Sección 1.1.5 se analizan los problemas que se presentan en estos procesos, en la Sección 1.1.6 se propone una solución: desarrollar un Portal Web utilizando la metodología ágil Scrum.

De acuerdo a la metodología seleccionada es necesario definir roles y responsabilidades como se indicó en el Capítulo I en la Sección 1.2.2, en la Tabla 2.1 se definen los roles que tomaran las personas que trabajan en el desarrollo de este proyecto.

Tabla 2.1 Roles Scrum para el Portal Web de la AEIS

Rol	Persona
Propietario del Producto	Sr. Christian Yaselga Presidente AEIS 2014
Scrum Master	Ing. Bolívar Palán
Equipo	José Luis Escobar, Marcela Mosquera

Elaborado por: Los autores

2.1 RECOLECCIÓN DE LAS HISTORIAS DE USUARIOS

Las historias de usuario fueron obtenidas en base a la descripción de los procesos que realiza la AEIS que se detallan en la Sección 1.1.4, en la Tabla 2.2 se presenta información general que contiene: el nombre del proceso, el usuario involucrado, la funcionalidad que se quiere implementar, el número de historia de usuario que va a describir esta funcionalidad y la prioridad asignada. Para el proceso de organización del campeonato interno de deportes, solamente se desarrollará la parte de inscripción de equipos.

Tabla 2.2 Procesos que realiza la AEIS

Proceso	Usuario	Requerimiento	Historia de Usuario	Prioridad
Afiliación	Presidente	Registrar datos personales.	HU01	Alta
Afiliación	Vicepresidente Financiero	Registrar el pago del aporte.	HU02	Alta
Matriculación en Clubes	Vicepresidente Académico	Matricular estudiante en un Club.	HU03	Media
Matriculación en Clubes	Profesor	Registrar la aprobación del estudiante.	HU04	Media

Venta de Snacks y Bebidas	Empleado	Registrar ventas.	HU05	Alta
Préstamo de Dispositivos	Empleado	Registrar el préstamo o devolución de un dispositivo.	HU06	Baja
Devolución de Dispositivos				
Alquiler de casilleros	Empleado	Registrar el alquiler del casillero.	HU07	Baja
Organización de Eventos Sociales	Vicepresidente de Cultura	Publicar la información de un evento.	HU08	Media
Organización de Eventos Sociales	Vicepresidente de Cultura	Registrar la inasistencia	HU09	Media
Inscripción en el Campeonato Interno de Deportes	Vicepresidente de Deportes	Inscripción de Equipo Deportivo	HU10	Media

Elaborado por: Los autores

Para describir las historias de usuario se usará el formato indicado en la Tabla 2.3 este formato contiene:

- **Número:** El número secuencial de la historia de usuario.
- **Nombre Historia:** Texto descriptivo de la historia de usuario.
- **Usuario:** Se refiere al rol de la persona que realiza la funcionalidad
- **Descripción:** Texto que detalla la historia de usuario.
- **Condiciones de Satisfacción:** Describe el resultado esperado.
- **Prerrequisitos:** Son los requisitos previos para realizar lo que describe la historia de usuario.

Tabla 2.3 Formato para Describir Historias de Usuario

HISTORIA DE USUARIO	Número
Nombre Historia:	
Usuario:	
Descripción:	
Condiciones de Satisfacción:	
Prerrequisitos	

Elaborado por: Los autores

2.1.1 HU01

HISTORIA DE USUARIO	HU01
Nombre Historia: Registrar datos personales del estudiante	
Usuario: Estudiante	
Descripción:	
<ol style="list-style-type: none"> 1. Consultar si el estudiante se encuentra registrado realizando una búsqueda con el número de cédula. 2. Si el estudiante se encuentra registrado, desplegar los datos y permitir actualizar los mismos. 3. Si el estudiante no se encuentra registrado tiene que ingresar los siguientes datos: cédula, nombre, apellido, número único, semestre, teléfono, dirección, correo electrónico, y género. 4. Validar los datos ingresados 5. Guardar los datos en la tabla estudiante. 	
Condiciones de Satisfacción	
<ol style="list-style-type: none"> 1. La información del estudiante queda registrada. 	

2.1.2 HU02

HISTORIA DE USUARIO	HU02
Nombre Historia: Registrar el pago del aporte	
Usuario: Vicepresidente Financiero	
Descripción:	
<ol style="list-style-type: none"> 1. Consultar los datos del estudiante mediante su número de cédula. 2. Registrar la desafiliación en caso de que el estudiante decida no afiliarse a la AEIS. 3. En caso de que el estudiante decida afiliarse, se despliegan los rubros a pagar que componen el aporte total y las multas registradas. 4. Cuando el estudiante cancele el aporte se guarda el valor total cancelado, la fecha en la tabla aporte. Y se registra la afiliación del estudiante. 	
Condiciones de Satisfacción:	
<ol style="list-style-type: none"> 1. Si el estudiante realiza el pago del aporte se registra la afiliación. 2. Si el estudiante no realiza el pago se registra la desafiliación. 	
Prerrequisitos:	
Tabla Aporte: Registrar, actualizar y eliminar aportes.	

2.1.3 HU03

HISTORIA DE USUARIO	HU03
Nombre Historia: Matricular estudiante en un Club	
Usuario: Estudiante	
Descripción:	
<ol style="list-style-type: none"> 1. El estudiante selecciona un club. 2. El estudiante se matricula en el club seleccionado. 	
Condiciones de Satisfacción	
<ol style="list-style-type: none"> 1. El estudiante queda matriculado en el club. 	
Prerrequisitos:	
Tabla Club: Registrar, actualizar y eliminar clubes.	

2.1.4 HU04

HISTORIA DE USUARIO	HU04
Nombre Historia: Registrar la aprobación del estudiante	
Usuario: Vicepresidente Académico	
Descripción:	
<ol style="list-style-type: none"> 1. Ingresar la aprobación de un estudiante en el club. 2. Imprimir un registro de estudiantes aprobados y reprobados. 	
Condiciones de Satisfacción:	
<ol style="list-style-type: none"> 1. Se registra la aprobación del estudiante. 	

2.1.5 HU05

HISTORIA DE USUARIO	HU05
Nombre Historia: Registrar Ventas	
Usuario: Empleado	
Descripción:	
<ol style="list-style-type: none"> 1. Registrar la cantidad de ventas realizadas en el bar de la AEIS al fin del día con los siguientes datos: fecha, producto y cantidad. 2. Actualizar el stock de productos, descontando la cantidad de productos vendidos durante el día. 3. Registrar el total de ventas diario. 	
Condiciones de Satisfacción:	
<ol style="list-style-type: none"> 1. Se registra cada venta y se obtiene un reporte de ventas diarias. 	
Prerrequisitos:	
Tabla Proveedor: Registrar, actualizar y eliminar proveedores.	
Tabla Producto: Registrar, actualizar y eliminar productos.	

2.1.6 HU06

HISTORIA DE USUARIO	HU06
Nombre Historia: Registrar el préstamo o devolución de un dispositivo	
Usuario: Empleado	
Descripción:	
<ol style="list-style-type: none"> 1. Registrar el préstamo de un dispositivo a un estudiante, se registran los siguientes datos: fecha. 2. Registrar la devolución del dispositivo. 	
Condiciones de Satisfacción:	
<ol style="list-style-type: none"> 1. Se registra el préstamo/devolución del dispositivo 	
Prerrequisitos:	
Tabla Activo: Registrar, actualizar y eliminar activos.	

2.1.7 HU07

HISTORIA DE USUARIO	HU07
Nombre Historia: Registrar el alquiler del casillero.	
Usuario: Empleado	
Descripción:	
<ol style="list-style-type: none"> 1. Registrar el alquiler de casillero con los siguientes datos: número casillero, persona a la que se alquila. 	
Condiciones de Satisfacción:	
<ol style="list-style-type: none"> 1. El estado de una casillero pasa de libre a ocupado y un estudiante pasa a estar relacionado con ese casillero 	
Prerrequisitos:	
Tabla Casillero: Registrar, actualizar y eliminar casillero.	

2.1.8 HU08

HISTORIA DE USUARIO	HU08
Nombre Historia: Publicar información de un evento	
Usuario: Vicepresidente de cultura	
Descripción:	
<ol style="list-style-type: none"> 1. Publicar información sobre los eventos que realiza la AEIS, fecha del evento, descripción. 	
Condiciones de Satisfacción:	
<ol style="list-style-type: none"> 1. La información de cada evento se muestra en el Portal. 	

2.1.9 HU09

HISTORIA DE USUARIO	HU09
Nombre Historia: Registrar la inasistencia a un evento	
Usuario: Empleado	
Descripción:	
1. Asignar multas a los estudiantes que no asistieron a eventos de carácter obligatorio.	
Condiciones de Satisfacción:	
1. La asistencia o multa queda registrada.	
Prerrequisitos:	
Tabla Multa: Registrar, actualizar y eliminar multas.	

2.1.10 HU10

HISTORIA DE USUARIO	HU10
Nombre Historia: Inscripción de Equipo Deportivo	
Usuario: Vicepresidente Deportivo	
Descripción:	
1. Registrar el equipo deportivo para su participación en el campeonato con los siguientes datos: Nombre del equipo, año de inscripción.	
2. Registrar a los jugadores que pertenecen a un equipo deportivo con los siguientes datos: Nombre, apellido, tipo de jugador (Si es de la facultad o externo), disciplinas en la que participa.	
Condiciones de Satisfacción:	
1. El equipo deportivo y queda registrado.	
Prerrequisitos:	
Tabla Disciplina: Registrar, actualizar y eliminar disciplinas.	

2.2 DISEÑO DEL PRODUCT BACKLOG

2.2.1 LISTA INICIAL DEL PRODUCTO

Para realizar el Product Backlog Inicial se tomaron las historias de usuario organizadas de acuerdo a la prioridad: alta, media, baja. Las historias de usuario de prioridad alta son: HU00, HU01, HU02, HU5, las historias que tienen prioridad media: HU03, HU04, HU08, HU09, HU10, HU11 y finalmente las historias de usuario con prioridad baja: HU06, HU07. El Product Backlog Inicial se muestra en la Tabla 2.4 que posee las siguientes columnas:

- **Orden:** prioridad del requerimiento
- **Requerimiento:** texto descriptivo de la historia de usuario
- **Historia de Usuario:** es el número de la historia.

Tabla 2.4 Product Backlog Inicial

Orden	Requerimiento	Historia de Usuario
1	Definir la arquitectura de la aplicación	Definido por los autores
2	Registrar datos personales del estudiante	HU01
3	Registrar el pago del aporte	HU02
4	Registrar ventas	HU05
5	Matricular estudiante en un Club	HU03
6	Registrar la aprobación del estudiante	HU04
7	Publicar la información de un evento	HU08
8	Registrar la inasistencia a un evento	HU09
9	Inscripción de Equipo Deportivo	HU10
10	Registrar el préstamo o devolución de un dispositivo	HU06
11	Registrar el alquiler del casillero.	HU07
12	Iniciar Sesión	Definido por los autores

Elaborado por: Los autores

2.2.2 REFINAMIENTO DEL PRODUCT BACKLOG

En el refinamiento del Product Backlog se añaden tareas que deben realizarse para cumplir con cada requerimiento. En la Tabla 2.5 se muestra el Refinamiento del Product Backlog.

Tabla 2.5 Refinamiento del Product Backlog

Orden	Requerimiento	Tarea
1	Diseño de la arquitectura	1.1 Definir la arquitectura de la aplicación.
		1.2 Crear el modelo conceptual de la base de datos.
		1.3 Crear el modelo físico de la base de datos.
		1.4 Generar el código SQL.
		1.5 Generar RESTful web services a partir de la base de datos.
2	Registrar datos personales del estudiante (HU01)	2.1 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		2.2 Crear un método que permita actualizar los datos del estudiante.
		2.3 Validar los datos ingresados.
		2.4 Crear un método que permita registrar al estudiante, en caso de que la búsqueda no devuelva ningún resultado.
		2.5 Crear la interfaz para el registro de estudiantes.
		2.6 Emitir un mensaje de confirmación al guardar/actualizar la información personal del estudiante.

3	Registrar el pago del aporte (HU02)	<p>3.1 Crear la interfaz para la gestión de aportes.</p> <p>3.2 Implementar la lógica de la interfaz para gestión de aportes.</p> <p>3.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.</p> <p>3.4 Crear un método que permita registrar el pago que realiza el estudiante.</p> <p>3.5 Crear la interfaz para el registro de pagos.</p> <p>3.6 Crear un método que permita registrar la desafiliación del estudiante.</p> <p>3.7 Emitir un mensaje de confirmación al registrar el pago del aporte.</p>
4	Registrar ventas (HU05)	<p>4.1 Crear la interfaz para la gestión de proveedores y productos.</p> <p>4.2 Implementar la lógica de la interfaz para gestión de proveedores y productos.</p> <p>4.3 Crear un método que permita buscar un producto.</p> <p>4.4 Crear un método que permita registrar la venta.</p> <p>4.5 Crear la interfaz para registrar las ventas.</p> <p>4.6 Emitir un mensaje de confirmación al registrar la venta.</p>
5	Matricular estudiante en un Club (HU03)	<p>5.1 Crear la interfaz para la gestión de clubes.</p> <p>5.2 Implementar la lógica de la interfaz para gestión de clubes.</p> <p>5.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.</p> <p>5.4 Crear un método que muestre la lista de clubes a dictarse durante el semestre.</p> <p>5.5 Crear un método que permita registrar al estudiante en el club seleccionado.</p> <p>5.6 Crear la interfaz para matricularse en un club.</p> <p>5.7 Emitir un mensaje de confirmación al matricular al estudiante en el club.</p>
6	Registrar la aprobación del estudiante (HU04)	<p>6.1 Crear un método que muestre la lista de clubes que se dictaron durante el semestre.</p> <p>6.2 Crear un método que despliegue la lista de los estudiantes matriculados en el club seleccionado.</p> <p>6.3 Crear un método que permita registrar la aprobación del estudiante.</p> <p>6.4 Crear la interfaz para registrar la aprobación del estudiante.</p> <p>6.5 Emitir un mensaje de confirmación al registrar la aprobación del estudiante.</p>
7	Publicar la información de un evento (HU08)	<p>7.1 Crear un método que permita registrar un evento.</p> <p>7.2 Crear la interfaz para la publicación de eventos.</p> <p>7.3 Emitir un mensaje de confirmación al publicar un</p>

		evento.
8	Registrar la inasistencia a un evento (HU09)	8.1 Crear la interfaz para la gestión de multas.
		8.2 Implementar la lógica de la interfaz para la gestión de multas.
		8.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		8.4 Crear un método que muestre una lista de eventos realizados en un determinado periodo de tiempo.
		8.5 Crear un método que permita registrar la inasistencia de un estudiante a un evento obligatorio.
		8.6 Emitir mensajes de confirmación.
		8.7 Crear la interfaz para el registro de inasistencia.
9	Inscripción de Equipo Deportivo (HU10)	9.1 Crear la interfaz para la gestión de disciplinas deportivas.
		9.2 Implementar la lógica de la interfaz para la gestión de disciplinas.
		9.3 Crear un método que permita registrar un equipo.
		9.4 Crear la interfaz para registrar un equipo.
		9.5 Crear un método que permita registrar al jugador.
		9.6 Crear la interfaz para registrar un jugador.
		9.7 Crear un método que muestre las categorías disponibles en las que un jugador pueda participar.
		9.8 Crear la interfaz para el registrar a un jugador en las disciplinas.
10	Registrar el préstamo o devolución de un dispositivo (HU06)	10.1 Crear la interfaz para la gestión de dispositivos.
		10.2 Implementar la lógica de la interfaz para gestión de dispositivos.
		10.3 Crear un método que permita buscar un dispositivo.
		10.4 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		10.5 Crear un método que permita registrar el préstamo del dispositivo seleccionado a un estudiante.
		10.6 Crear un método que muestre los dispositivos que se encuentran prestados a un estudiante.
		10.7 Crear la interfaz para el registro de préstamo de un dispositivo.
		10.8 Emitir un mensaje de confirmación al registrar el préstamo de un dispositivo.
11	Registrar el alquiler de un casillero (HU07)	11.1 Crear la interfaz para la gestión de casilleros.
		11.2 Implementar la lógica de la interfaz para gestión de casilleros.
		11.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		11.4 Crear un método que muestre la lista de

		casilleros disponibles.
		11.5 Crear un método que permita registrar el alquiler de un casillero a un estudiante.
		11.6 Crear un método que muestre los casilleros disponibles.
		11.7 Crear la interfaz para el alquiler de casilleros.
		11.8 Emitir un mensaje de confirmación al registrar el alquiler de un casillero
12	Iniciar Sesión	12.1 Crear la interfaz para la gestión de usuarios.
		12.2 Implementar la lógica para la gestión de usuarios.
		12.3 Crear la interfaz para el inicio de sesión
		12.4 Crear un método de autenticación de usuarios.
		12.5 Crear un método que muestre el menú de acuerdo al tipo de usuario.

Elaborado por: Los autores

2.2.3 DEFINICIÓN DE LOS SPRINTS

Luego de definir el Product Backlog y siguiendo lo que indica la metodología Scrum, se realizó el Sprint Planning Meeting para determinar las tareas que conformarán el Sprint Backlog y se implementarán en los Sprints.

Se estimó la duración de cada tarea en función de Story Points obtenidos al usar el método de Planning Poker:

- Cada miembro del equipo de desarrollo recibe un conjunto de cartas numeradas con la serie de Fibonacci (0, 1, 2, 3, 5, 8, 13). Estas cartas se utilizan para reflejar la estimación de cada tarea que compone el Sprint Backlog.
- El Scrum Master presenta al equipo de desarrollo cada tarea que se va a realizar y pide a cada miembro del equipo que elija una carta con la estimación que considere más conveniente.
- El Scrum Master asigna la media de las estimaciones.
- En caso de existir estimaciones que salgan del promedio o valores extremos el Scrum Master pide que se expongan las razones por las que cada miembro escogió esa estimación y se realiza la votación nuevamente hasta llegar a un consenso.

El Sprint Planning Meeting y la estimación de las tareas del Sprint Backlog usando el método de Poker Planning se realizaron en cada uno de los cuatro Sprints necesarios para completar el proyecto.

2.2.3.1 Primer Sprint

El primer Sprint inició el 02/02/2015 y terminó el 12/02/2015 tuvo una duración total de 9 días. Para el primer Sprint se diseñaron e implementaron las siguientes historias de usuario: HU00, HU01, HU02 consideradas de prioridad alta. Al completar estas historias se tiene el primer entregable funcional correspondiente al proceso de Afiliación. En la Tabla 2.6 se muestran las tareas y estimaciones correspondientes al primer Sprint Backlog.

2.2.3.1.1 Objetivo

El objetivo del primer Sprint es: diseñar la arquitectura de la aplicación, crear la base de datos e implementar las primeras funcionalidades del Portal Web: registrar los datos personales del estudiante y registrar el pago del aporte con estos requisitos cubren el Proceso de Afiliación.

Tabla 2.6 Primer Sprint Backlog

Orden	Requerimiento	Tarea	Estimación
1	Diseño de la arquitectura	1.1 Definir la arquitectura de la aplicación.	13
		1.2 Crear el modelo conceptual de la base de datos.	8
		1.3 Crear el modelo físico de la base de datos.	5
		1.4 Generar el código SQL.	5
		1.5 Generar RESTful web services a partir de la base de datos.	5
2	Registrar datos personales del estudiante (HU01)	2.1 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.	8
		2.2 Crear un método que permita actualizar los datos del estudiante.	5
		2.3 Validar los datos ingresados.	3
		2.4 Crear un método que permita registrar al estudiante, en caso de que la búsqueda no devuelva ningún resultado.	8
		2.5 Crear la interfaz para el registro de estudiantes.	8
		2.6 Emitir un mensaje de confirmación al guardar/actualizar la información personal del estudiante.	2
3	Registrar el pago del aporte (HU02)	3.1 Crear la interfaz para la gestión de aportes.	5
		3.2 Implementar la lógica de la interfaz para gestión de aportes.	8

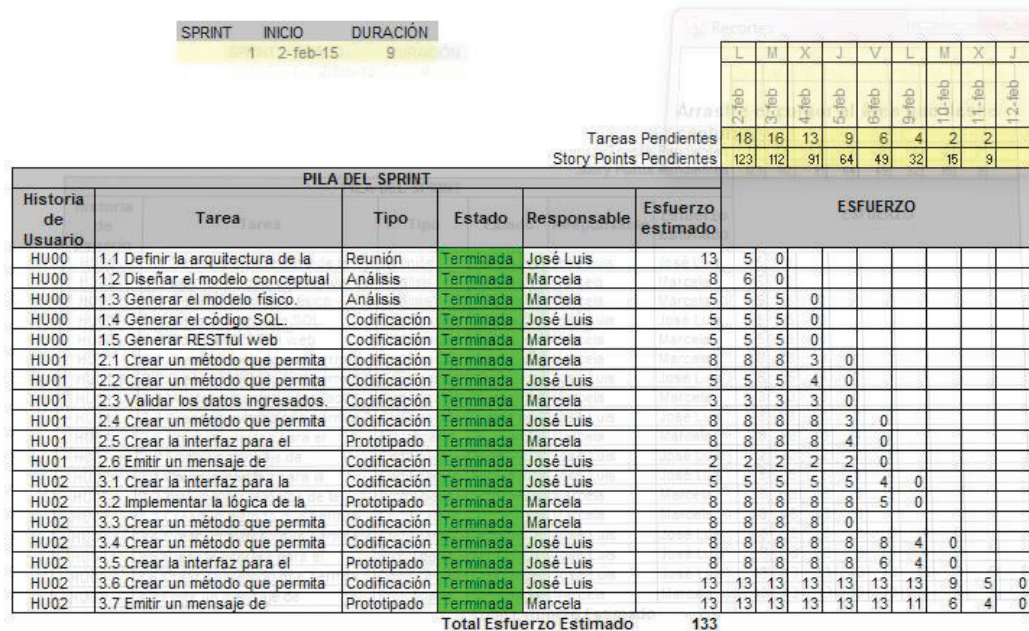
	3.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.	8
	3.4 Crear un método que permita registrar el pago que realiza el estudiante.	8
	3.5 Crear la interfaz para el registro de pagos.	8
	3.6 Crear un método que permita registrar la desafiliación del estudiante.	13
	3.7 Emitir un mensaje de confirmación al registrar el pago del aporte.	13
TOTAL		133

Elaborado por: Los autores

2.2.3.1.2 Scrum Diario

Las reuniones de Scrum Diario no se documentan, el objetivo de estas reuniones es que cada miembro del equipo indique las tareas realizadas, y las tareas pendientes. La Figura 2.1 muestra el esfuerzo diario que se ha realizado durante el primer Sprint.

Figura 2.1 Esfuerzo Realizado en el Primer Sprint



Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.2.3.2 Segundo Sprint

El segundo Sprint inició el 13/02/2015 y terminó el 25/02/2015 tuvo una duración total de 9 días, pero solamente solo 7 días laborables debido al feriado por carnaval (16 y 17 de febrero). Para el segundo Sprint se diseñaron e implementaron las siguientes historias de usuario: HU05 de prioridad alta, HU03 y HU04 de prioridad media. Al terminar estas historias se entrega la funcionalidad correspondiente al proceso de Venta de Snacks y Bebidas y también al proceso de Matriculación en Clubes.

En la 2.7 se muestran las tareas y estimaciones correspondientes al segundo Sprint Backlog.

2.2.3.2.1 Objetivo

El objetivo del segundo Sprint es: implementar las funcionalidades del Portal Web correspondientes al proceso de venta de snacks y bebidas y al proceso de matriculación en clubes.

Tabla 2.7 Segundo Sprint Backlog

Orden	Requerimiento	Tarea	Estimación
4	Registrar ventas (HU05)	4.1 Crear la interfaz para la gestión de proveedores y productos.	8
		4.2 Implementar la lógica de la interfaz para gestión de proveedores y productos.	8
		4.3 Crear un método que permita buscar un producto.	8
		4.4 Crear un método que permita registrar la venta.	8
		4.5 Crear la interfaz para registrar las ventas.	5
		4.6 Emitir un mensaje de confirmación al registrar la venta.	2
5	Matricular estudiante en un Club (HU03)	5.1 Crear la interfaz para la gestión de clubes.	5
		5.2 Implementar la lógica de la interfaz para gestión de clubes.	8
		5.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.	8
		5.4 Crear un método que muestre la lista de clubes a dictarse durante el semestre.	8
		5.5 Crear un método que permita registrar al estudiante en el club	8

		seleccionado.	
		5.6 Crear la interfaz para matricularse en un club.	8
		5.7 Emitir un mensaje de confirmación al matricular al estudiante en el club.	2
6	Registrar la aprobación del estudiante (HU04)	6.1 Crear un método que muestre la lista de clubes que se dictaron durante el semestre.	8
		6.2 Crear un método que despliegue la lista de los estudiantes matriculados en el club seleccionado.	8
		6.3 Crear un método que permita registrar la aprobación del estudiante.	8
		6.4 Crear la interfaz para registrar la aprobación del estudiante.	8
		6.5 Emitir un mensaje de confirmación al registrar la aprobación del estudiante.	2
TOTAL			120

Elaborado por: Los autores

2.2.3.2.2 Scrum Diario

La Figura 2.2 muestra el esfuerzo diario que se ha realizado durante el segundo Sprint.

Figura 2.2 Esfuerzo Realizado en el Segundo Sprint

SPRINT	INICIO	DURACIÓN
2	13-feb-15	9

	V	X	J	V	L	M	X
Tareas	16	12	10	8	5	2	
Story Points Pendientes	92	74	58	44	29	9	

PILA DEL SPRINT						ESFUERZO							
Historia de Usuario	Tarea	Tipo	Estado	Responsable	Esfuerzo estimado								
HU05	4.1 Crear la interfaz para la gestión	Codificación	Terminada	José Luis	8	0							
HU05	4.2 Implementar la lógica de la interfaz	Codificación	Terminada	Marcela	8	0							
HU05	4.3 Crear un método que permita registrar	Prototipado	Terminada	Marcela	8	3	0						
HU05	4.4 Crear un método que permita registrar	Codificación	Terminada	José Luis	8	2	0						
HU05	4.5 Crear la interfaz para registrar	Prototipado	Terminada	Marcela	5	4	5						
HU05	4.6 Emitir un mensaje de confirmación	Codificación	Terminada	José Luis	2	2	0						
HU03	5.1 Crear la interfaz para la gestión	Prototipado	Terminada	Marcela	5	5	3	0					
HU03	5.2 Implementar la lógica de la interfaz	Codificación	Terminada	José Luis	8	8	4	0					
HU03	5.3 Crear un método que permita registrar	Codificación	Terminada	Marcela	8	8	7	5	4	0			
HU03	5.4 Crear un método que muestre la lista	Codificación	Terminada	José Luis	8	8	8	3	0				
HU03	5.5 Crear un método que permita registrar	Codificación	Terminada	José Luis	8	8	8	6	2	0			
HU03	5.6 Crear la interfaz para matricularse	Codificación	Terminada	Marcela	8	8	8	8	4	0			
HU03	5.7 Emitir un mensaje de confirmación	Prototipado	Terminada	Marcela	2	2	2	2	0				
HU04	6.1 Crear un método que muestre la lista	Codificación	Terminada	José Luis	8	8	8	8	8	5	0		
HU04	6.2 Crear un método que despliegue la lista	Codificación	Terminada	Marcela	8	8	8	8	8	7	0		
HU04	6.3 Crear un método que permita registrar	Codificación	Terminada	José Luis	8	8	8	8	8	8	4	0	
HU04	6.4 Crear la interfaz para registrar	Codificación	Terminada	Marcela	8	8	8	8	8	7	5	0	
HU04	6.5 Emitir un mensaje de confirmación	Codificación	Terminada	José Luis	2	2	2	2	2	2	0	0	
Total Esfuerzo Estimado					120								

Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.2.3.3 Tercer Sprint

El tercer Sprint inició el 26/02/2015 y terminó el 10/03/2015 tuvo una duración total de 9 días. Para el tercer Sprint se diseñaron e implementaron las siguientes historias de usuario: HU08, HU09, HU10 de prioridad media, HU06 de prioridad baja. Al terminar estas historias se entrega las diferentes funcionalidades correspondientes a los procesos de: Organización de Eventos Sociales, Inscripción en el Campeonato Interno de Deportes y Préstamo de Dispositivos.

En la Tabla 2.8 se muestran las tareas y estimaciones correspondientes al tercer Sprint Backlog.

2.2.3.3.1 Objetivo

El objetivo del tercer Sprint es: implementar las funcionalidades del Portal Web correspondientes a los procesos de Organización de Eventos Sociales, Inscripción en el Campeonato Interno de Deportes y Préstamo de Dispositivos.

Tabla 2.8 Tercer Sprint Backlog

Orden	Requerimiento	Tarea	Estimación
7	Publicar la información de un evento (HU08)	7.1 Crear un método que permita registrar un evento.	5
		7.2 Crear la interfaz para la publicación de eventos.	5
		7.3 Emitir un mensaje de confirmación al publicar un evento.	2
8	Registrar la inasistencia a un evento (HU09)	8.1 Crear la interfaz para la gestión de multas.	5
		8.2 Implementar la lógica de la interfaz para la gestión de multas.	5
		8.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.	8
		8.4 Crear un método que muestre una lista de eventos realizados en un determinado periodo de tiempo.	5
		8.5 Crear un método que permita registrar inasistencia de un estudiante a un evento obligatorio.	8
		8.6 Emitir mensajes de confirmación.	2
		8.7 Crear la interfaz para el registro de inasistencia.	5
9	Inscripción de Equipo Deportivo (HU10)	9.1 Crear la interfaz para la gestión de disciplinas deportivas.	3
		9.2 Implementar la lógica de la interfaz para la gestión de disciplinas.	5
		9.3 Crear un método que permita	5

		registrar un equipo.	
		9.4 Crear la interfaz para registrar un equipo.	3
		9.5 Crear un método que permita registrar al jugador.	5
		9.6 Crear la interfaz para registrar un jugador.	5
		9.7 Crear un método que muestre las categorías disponibles en las que un jugador pueda participar.	5
		9.8 Crear la interfaz para el registrar a un jugador en las disciplinas.	8
10	Registrar el préstamo o devolución de un dispositivo (HU06)	10.1 Crear la interfaz para la gestión de dispositivos.	5
		10.2 Implementar la lógica de la interfaz para gestión de dispositivos.	8
		10.3 Crear un método que permita buscar un dispositivo.	5
		10.4 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.	8
		10.5 Crear un método que permita registrar el préstamo del dispositivo seleccionado a un estudiante.	8
		10.6 Crear un método que muestre los dispositivos que se encuentran prestados a un estudiante.	5
		10.7 Crear la interfaz para el registro de préstamo de un dispositivo.	2
		10.8 Emitir un mensaje de confirmación al registrar el préstamo de un dispositivo.	8
TOTAL			138

Elaborado por: Los autores

2.2.3.3.2 Scrum Diario

La Figura 2.3 muestra el esfuerzo diario que se ha realizado durante el tercer Sprint.

Figura 2.3 Esfuerzo Realizado en el Tercer Sprint

SPRINT		INICIO	DURACIÓN													
		3	26-feb-15	9												
					Tareas Pendientes	21	18	15	15	11	8	5	2			
					Story Points Pendientes	144	121	104	85	63	50	26	7			
						J	V	L	M	X	J	V	L	M		
						26-feb	27-feb	2-mar	3-mar	4-mar	5-mar	6-mar	3-mar	10-mar		
PILA DEL SPRINT						ESFUERZO										
Historia de Usual	Tarea	Tipc	Esta	Respons	Esfuerzo Estimado											
HU08	7.1 Crear un método que permita registrar un alquiler de un casillero	Codificación	Terminada	Marcela	5	0										
HU08	7.2 Crear la interfaz para la publicación de un anuncio	Prototipado	Terminada	José Luis	5	0										
HU08	7.3 Emitir un mensaje de confirmación de un alquiler	Codificación	Terminada	José Luis	2	0										
HU09	8.1 Crear la interfaz para la gestión de alquileres	Prototipado	Terminada	Marcela	5	5	0									
HU09	8.2 Implementar la lógica de la interfaz para la gestión de alquileres	Codificación	Terminada	José Luis	5	4	0									
HU09	8.3 Crear un método que permita buscar un alquiler	Codificación	Terminada	Marcela	8	8	4	0								
HU09	8.4 Crear un método que permita actualizar un alquiler	Codificación	Terminada	Marcela	5	2	2	0								
HU09	8.5 Crear un método que permita asignar un alquiler	Codificación	Terminada	José Luis	8	8	8	0								
HU09	8.6 Emitir mensajes de confirmación de un alquiler	Codificación	Terminada	José Luis	2	2	2	2	0							
HU09	8.7 Crear la interfaz para el registro de un alquiler	Prototipado	Terminada	José Luis	5	5	5	3	0							
HU10	9.1 Crear la interfaz para la gestión de alquileres	Prototipado	Terminada	José Luis	3	3	3	2	0							
HU10	9.2 Implementar la lógica de la interfaz para la gestión de alquileres	Prototipado	Terminada	Marcela	5	5	5	5	1	0						
HU10	9.3 Crear un método que permita registrar un alquiler	Codificación	Terminada	Marcela	5	5	5	3	0							
HU10	9.4 Crear la interfaz para registrar un alquiler	Prototipado	Terminada	José Luis	3	3	3	3	2	0						
HU10	9.5 Crear un método que permita registrar un alquiler	Codificación	Terminada	Marcela	5	5	5	5	5	2	0					
HU10	9.6 Crear la interfaz para registrar un alquiler	Prototipado	Terminada	José Luis	5	5	5	4	2	0						
HU10	9.7 Crear un método que muestre la disponibilidad de un alquiler	Codificación	Terminada	Marcela	5	5	5	5	4	2	0					
HU10	9.8 Crear la interfaz para el registro de un alquiler	Prototipado	Terminada	José Luis	8	8	8	6	4	2	0					
HU06	10.1 Crear la interfaz para la gestión de alquileres	Prototipado	Terminada	Marcela	5	5	5	5	5	5	0					
HU06	10.2 Implementar la lógica de la interfaz para la gestión de alquileres	Prototipado	Terminada	José Luis	8	8	8	8	8	6	0					
HU06	10.3 Crear un método que permita buscar un alquiler	Codificación	Terminada	Marcela	5	5	5	5	5	4	0					
HU06	10.4 Crear un método que permita actualizar un alquiler	Prototipado	Terminada	José Luis	8	8	8	8	8	8	5	0				
HU06	10.5 Crear un método que permita asignar un alquiler	Codificación	Terminada	Marcela	8	8	8	8	8	8	3	0				
HU06	10.6 Crear reportes que muestren los alquileres	Codificación	Terminada	José Luis	5	5	5	5	5	5	5	0				
HU06	10.7 Crear la interfaz para el registro de un alquiler	Prototipado	Terminada	José Luis	2	2	2	2	2	2	2	2	2	2	2	0
HU06	10.8 Emitir un mensaje de confirmación de un alquiler	Codificación	Terminada	Marcela	8	8	8	8	8	8	8	8	8	8	5	0
Total Esfuerzo Estimado					138											

Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.2.3.4 Cuarto Sprint

El cuarto Sprint inició el 11/03/2015 y terminó el 23/03/2015 tuvo una duración total de 9 días. Para el cuarto Sprint se diseñó e implementó la historia de usuario: HU06 de prioridad baja, correspondiente al proceso de alquiler de casilleros. Adicionalmente

En la Tabla 2.9 se muestran las tareas y estimaciones correspondientes al cuarto Sprint Backlog.

2.2.3.4.1 Objetivo

El objetivo del tercer Sprint es: implementar las funcionalidades del Portal Web correspondientes al Proceso de Alquiler de Casilleros. También se va implementar un menú y un login.

Tabla 2.9 Cuarto Sprint Backlog

Orden	Requerimiento	Tarea	Estimación
11	Registrar el alquiler de un casillero (HU07)	11.1 Crear la interfaz para la gestión de casilleros.	5
		11.2 Implementar la lógica de la interfaz para gestión de casilleros.	8
		11.3 Crear un método que permita	8

		buscar a un estudiante de acuerdo a su cédula.	
		11.4 Crear un método que muestre la lista de casilleros disponibles.	8
		11.5 Crear un método que permita registrar el alquiler de un casillero a un estudiante.	8
		11.6 Crear reportes que muestren los casilleros alquilados a los estudiantes.	8
		11.7 Crear la interfaz para el alquiler de casilleros.	13
		11.8 Emitir un mensaje de confirmación al registrar el alquiler de un casillero	2
12	Iniciar Sesión	12.1 Crear la interfaz para la gestión de usuarios.	5
		12.2 Implementar la lógica para la gestión de usuarios.	8
		12.3 Crear la interfaz para el inicio de sesión	3
		12.4 Crear un método de autenticación de usuarios.	5
		12.5 Crear un método que muestre el menú de acuerdo al tipo de usuario.	5
13	Diseñar la página de inicio	13.1 Crear la página de inicio e incluir el texto de visión y misión de la AEIS	8
		13.2 Colocar enlaces a otras páginas relacionadas con la AEIS, la página oficial de la EPN y de la FIS.	5
TOTAL			99

Elaborado por: Los autores

2.2.3.4.2 Scrum Diario

La Figura 2.4 muestra el esfuerzo diario que se ha realizado durante el cuarto Sprint.

Figura 2.4 Esfuerzo Realizado en el Cuarto Sprint

SPRINT	INICIO	DURACIÓN																
4	11-mar.-15	9																
			X	J	V	L	M	X	J	V	L							
			11-mar.	12-mar.	13-mar.	16-mar.	17-mar.	18-mar.	19-mar.	20-mar.	23-mar.							
			Tareas Pendientes	13	11	10	9	8	6	4	2							
			Story Points Pendientes	84	70	54	41	36	29	19	6							
PILA DEL SPRINT												ESFUERZO						
Historia de Usuari	Tarea	Tipo	Estado	Responsable	Esfuerzo Estimado													
HU07	11.1 Crear la interfaz para la gestión	Codificación	Terminada	Marcela	5	0												
HU07	11.2 Implementar la lógica de la interfaz	Codificación	Terminada	José Luis	8	0												
HU07	11.3 Crear un método que permita la gestión	Codificación	Terminada	Marcela	8	6	0											
HU07	11.4 Crear un método que muestre la información	Codificación	Terminada	José Luis	8	8	0											
HU07	11.5 Crear un método que permita la gestión	Prototipado	Terminada	Marcela	8	8	8	0										
HU07	11.6 Crear reportes que muestren la información	Codificación	Terminada	José Luis	8	8	8	2	0									
HU07	11.7 Crear la interfaz para el alquiler	Codificación	Terminada	Marcela	13	13	13	11	6	2	0							
HU07	11.8 Emitir un mensaje de confirmación	Codificación	Terminada	Marcela	2	2	2	2	2	2	0							
Definido por el equipo	12.1 Crear la interfaz para la gestión	Codificación	Terminada	José Luis	5	5	5	5	1	0								
	12.2 Implementar la lógica para la gestión	Prototipado	Terminada	José Luis	8	8	8	8	6	6	3	0						
	12.3 Crear la interfaz para el inicio de sesión	Prototipado	Terminada	Marcela	3	3	3	3	3	3	0							
	12.4 Crear un método de autenticación	Codificación	Terminada	Marcela	5	5	5	5	5	5	4	0						
Definido por el cliente	12.5 Crear un método que muestre la información	Codificación	Terminada	José Luis	5	5	5	5	5	5	3	0						
	13.1 Crear la página de inicio e incluir el texto de visión y misión de la AEIS	Prototipado	Terminada	Marcela	8	8	8	8	8	8	8	4	0					
	13.2 Colocar enlaces a otras páginas relacionadas con la AEIS, la página oficial de la EPN y de la FIS.	Prototipado	Terminada	José Luis	5	5	5	5	5	5	4	2	0					
Total Esfuerzo Estimado					99													

Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.2.4 LISTA FINAL DEL PRODUCTO

Durante el desarrollo del Portal Web se ha refinado el Product Backlog, en la Tabla 2.10 se presenta las tareas que se añadieron como parte de los requisitos del cliente.

Tabla 2.10 Tareas añadidas al Product Backlog

Orden	Requerimiento	Tarea
13	Diseñar la página de inicio	13.1 Crear la página de inicio e incluir el texto de visión y misión de la AEIS
		13.2 Colocar enlaces a otras páginas relacionadas con la AEIS, la página oficial de la EPN y de la FIS.

Elaborado por: Los Autores

En la Tabla 2.11 se presenta el Product Backlog final:

Tabla 2.11 Product Backlog Final

Orden	Requerimiento	Tarea
1	Diseño de la arquitectura	1.1 Definir la arquitectura de la aplicación.
		1.2 Crear el modelo conceptual de la base de datos.
		1.3 Crear el modelo físico de la base de datos.
		1.4 Generar el código SQL.

		1.5 Generar RESTful web services a partir de la base de datos.
2	Registrar datos personales del estudiante (HU01)	2.1 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		2.2 Crear un método que permita actualizar los datos del estudiante.
		2.3 Validar los datos ingresados.
		2.4 Crear un método que permita registrar al estudiante, en caso de que la búsqueda no devuelva ningún resultado.
		2.5 Crear la interfaz para el registro de estudiantes.
		2.6 Emitir un mensaje de confirmación al guardar/actualizar la información personal del estudiante.
3	Registrar el pago del aporte (HU02)	3.1 Crear la interfaz para la gestión de aportes.
		3.2 Implementar la lógica de la interfaz para gestión de aportes.
		3.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		3.4 Crear un método que permita registrar el pago que realiza el estudiante.
		3.5 Crear la interfaz para el registro de pagos.
		3.6 Crear un método que permita registrar la desafiliación del estudiante.
		3.7 Emitir un mensaje de confirmación al registrar el pago del aporte.
4	Registrar ventas (HU05)	4.1 Crear la interfaz para la gestión de proveedores y productos
		4.2 Implementar la lógica de la interfaz para gestión de proveedores y productos.
		4.3 Crear un método que permita buscar un producto.
		4.4 Crear un método que permita registrar una venta.
		4.5 Crear la interfaz para registrar las ventas.
		4.6 Emitir un mensaje de confirmación al registrar la venta.
5	Matricular estudiante en un Club (HU03)	5.1 Crear la interfaz para la gestión de clubes.
		5.2 Implementar la lógica de la interfaz para gestión de clubes.
		5.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		5.4 Crear un método que muestre la lista de clubes a dictarse durante el semestre.
		5.5 Crear un método que permita registrar al estudiante en el club seleccionado.
		5.6 Crear la interfaz para matricularse en un club.
		5.7 Emitir un mensaje de confirmación al matricular al

		estudiante en el club.
6	Registrar la aprobación del estudiante (HU04)	<p>6.1 Crear un método que muestre la lista de clubes que se dictaron durante el semestre.</p> <p>6.2 Crear un método que despliegue la lista de los estudiantes matriculados en el club seleccionado.</p> <p>6.3 Crear un método que permita registrar la aprobación del estudiante.</p> <p>6.4 Crear la interfaz para registrar la aprobación del estudiante.</p> <p>6.5 Emitir un mensaje de confirmación al registrar la aprobación del estudiante.</p>
7	Publicar la información de un evento (HU08)	<p>7.1 Crear un método que permita registrar un evento.</p> <p>7.2 Crear la interfaz para la publicación de eventos.</p> <p>7.3 Emitir un mensaje de confirmación al publicar un evento.</p>
8	Registrar la inasistencia a un evento (HU09)	<p>8.1 Crear la interfaz para la gestión de multas.</p> <p>8.2 Implementar la lógica de la interfaz para la gestión de multas.</p> <p>8.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.</p> <p>8.4 Crear un método que muestre una lista de eventos realizados en un determinado periodo de tiempo.</p> <p>8.5 Crear un método que permita asignar una multa al estudiante que no asistió a un evento obligatorio.</p> <p>8.6 Emitir mensajes de confirmación.</p> <p>8.7 Crear la interfaz para el registro de inasistencia.</p>
9	Inscripción de Equipo Deportivo (HU10)	<p>9.1 Crear la interfaz para la gestión de disciplinas deportivas.</p> <p>9.2 Implementar la lógica de la interfaz para la gestión de disciplinas.</p> <p>9.3 Crear un método que permita registrar un equipo.</p> <p>9.4 Crear la interfaz para registrar un equipo.</p> <p>9.5 Crear un método que permita registrar al jugador.</p> <p>9.6 Crear la interfaz para registrar un jugador.</p> <p>9.7 Crear un método que muestre las categorías disponibles en las que un jugador pueda participar.</p> <p>9.8 Crear la interfaz para el registrar a un jugador en las disciplinas.</p>
10	Registrar el préstamo o devolución de un dispositivo (HU06)	<p>10.1 Crear la interfaz para la gestión de dispositivos.</p> <p>10.2 Implementar la lógica de la interfaz para gestión de dispositivos.</p> <p>10.3 Crear un método que permita buscar un dispositivo.</p> <p>10.4 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.</p>

		10.5 Crear un método que permita registrar el préstamo del dispositivo seleccionado a un estudiante.
		10.6 Crear un método que muestre los dispositivos que se encuentran prestados a un estudiante.
		10.7 Crear la interfaz para el registro de préstamo de un dispositivo.
		10.8 Emitir un mensaje de confirmación al registrar el préstamo de un dispositivo.
11	Registrar el alquiler de un casillero (HU07)	11.1 Crear la interfaz para la gestión de casilleros.
		11.2 Implementar la lógica de la interfaz para gestión de casilleros.
		11.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
		11.4 Crear un método que muestre la lista de casilleros disponibles.
		11.5 Crear un método que permita registrar el alquiler de un casillero a un estudiante.
		11.6 Crear un método que muestre los casilleros disponibles.
		11.7 Crear la interfaz para el alquiler de casilleros.
		11.8 Emitir un mensaje de confirmación al registrar el alquiler de un casillero
12	Iniciar Sesión	12.1 Crear la interfaz para la gestión de usuarios.
		12.2 Implementar la lógica para la gestión de usuarios.
		12.3 Crear la interfaz para el inicio de sesión
		12.4 Crear un método de autenticación de usuarios.
		12.5 Crear un método que muestre el menú de acuerdo al tipo de usuario.
13	Diseñar la página de inicio	13.1 Crear la página de inicio e incluir el texto de visión y misión de la AEIS
		13.2 Colocar enlaces a otras páginas relacionadas con la AEIS, la página oficial de la EPN y de la FIS.

Elaborado por: Los autores

2.3 DISEÑO E IMPLEMENTACIÓN DE LOS SPRINTS

Después de planificar cada Sprint se procede a desarrollar las tareas planificadas en cada Sprint Backlog.

2.3.1 PRIMER SPRINT

2.3.1.1 Diseño

Las tareas que requieren diseño en el primer Sprint son:

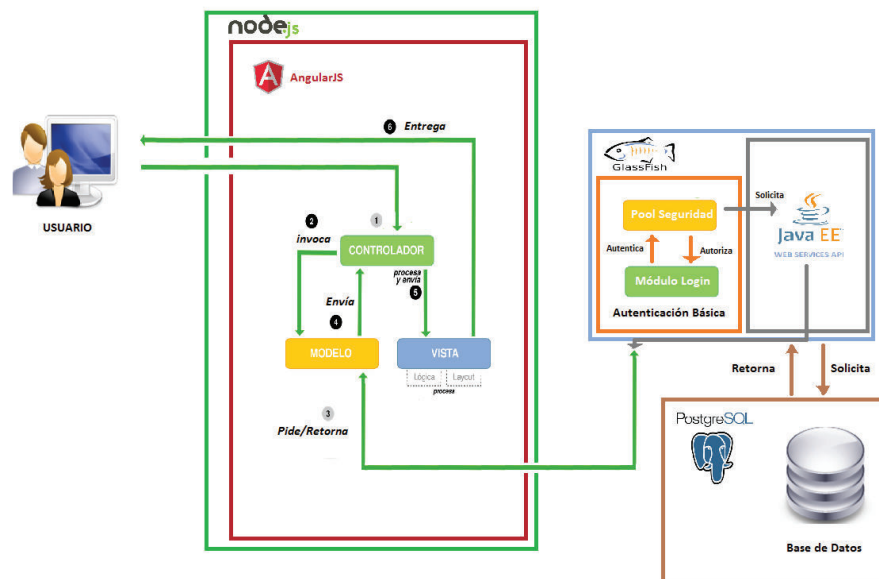
- 1.1 Definir la arquitectura de la aplicación
- 1.2 Diseñar el modelo conceptual de la base de datos.

- 1.3 Generar el modelo físico de la base de datos.
- 2.5 Crear la interfaz para el registro de estudiantes.
- 3.1 Crear la interfaz para la gestión de aportes.
- 3.5 Crear la interfaz para el registro de pagos.

2.3.1.1.1 Arquitectura de la Aplicación

La aplicación se comprende de dos grandes componentes el Backend y el Frontend. En cada uno de estos se utiliza dos tecnologías independientes pero que interactúan a través de una interfaz de abstracción de datos tal y como se describe en la Figura 2.5

Figura 2.5 Arquitectura AEIS WEB APP



Elaborado por: Los autores

- Frontend

La aplicación en el Frontend está escrita totalmente en JavaScript usando el framework AngularJS. La aplicación se ejecuta sobre un entorno de ejecución que se encuentra escrito también en JavaScript llamado Node.Js, Este entorno permite optimizar la carga de la aplicación y su escalabilidad, además que permite acelerar el proceso de desarrollo gracias a que permite hacer “live-reloads” la cual es una tecnología que permite visualizar los cambios realizados en el código en tiempo real sin necesidad de volver a compilar y desplegar la aplicación. Node.JS también cuenta con su gestor de paquetes integrado el cual permite

instalar o desinstalar componentes de la aplicación usando solo unos pocos comandos acortando el tiempo de desarrollo.

El Frontend sigue el patrón modelo-vista-controlador el modelo se conecta con el backend a través de servicios los cuales retornan datos que son gestionados en el controlador y presentados en la vista al usuario.

Una de las principales ventajas de una aplicación escrita totalmente sobre JavaScript es la velocidad con la que responde a las peticiones lo cual mejora notablemente la experiencia del usuario y su navegación.

Al realizarse la mayoría del trabajo a nivel de cliente se reduce la cantidad de peticiones remotas al servidor a simples llamadas a Web Services reduciendo la carga de trabajo en el servidor y aumentando la velocidad de navegación.

- Backend

En la parte de Backend se utiliza una API de WebServices implementada en Java EE7 usando un servidor de aplicaciones Glassfish v4.0.

Las peticiones a los Web Services se gestionan a través del módulo de autenticación básica que viene implementado en el servidor Glassfish. Este se encarga de autorizar o denegar la petición dependiendo del usuario y su rol. Una vez autenticado el servidor se encarga de gestionar la petición y devolver el objeto solicitado en formato JSON (De acuerdo a la Figura 2.8), consultándolo en la base de datos a través de la API de persistencia de Java (JPA).

Figura 2.6 Ejemplo de Objeto en formato JSON

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

▼<estudiantes>
  ▼<estudiante>
    <apellidoestudiante>Escobar Cárdenas</apellidoestudiante>
    <cedula>1721783213</cedula>
    <correoelectronicoestudiante>asd@hotmail.com</correoelectronicoestudiante>
    <direccionestudiante>asdas</direccionestudiante>
    <estadoafiliacionestudiante>Afiliado</estadoafiliacionestudiante>
    <estadoaprobacionclubes>Aprobado</estadoaprobacionclubes>
    <fechanacimientoestudiante>2015-06-24T00:00:00-05:00</fechanacimientoestudiante>
    <generoestudiante>77</generoestudiante>
    <nombreestudiante>José Luis</nombreestudiante>
    <numerounicoestudiante>201010238</numerounicoestudiante>
    <semestrereferencial>Egresado</semestrereferencial>
    <telefonoestudiante>022635498</telefonoestudiante>
  </estudiante>
</estudiantes>

```

Elaborado por: Los autores

Arquitectura de Seguridad

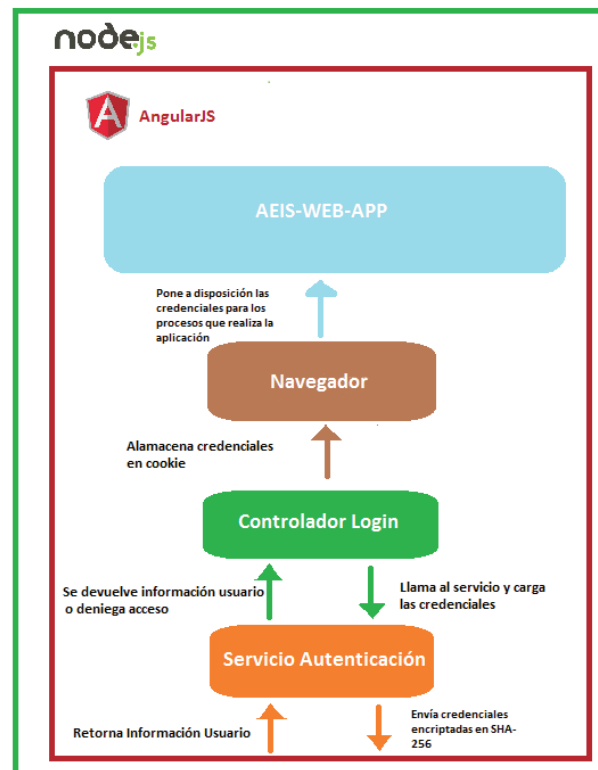
Para la aplicación se han diseñado dos módulos principales de seguridad los cuales permiten autenticar a los usuarios para que dependiendo de sus privilegios puedan acceder a las distintas opciones del programa.

- Seguridad en el Frontend

En el Frontend la seguridad está determinada por el servicio de autenticación el cual durante el proceso de login se encarga de encriptar las credenciales bajo el algoritmo simple de hash SHA-256 para su posterior procesamiento en el servidor. Una vez que llega la respuesta el servicio de autenticación deniega o aprueba el acceso pasando esta información al controlador del login.

El controlador del login se encarga de almacenar estas credenciales en una cookie que perdurará hasta que el usuario cierre la sesión de igual manera se habilitan y deshabilitan ciertas páginas de acuerdo al rol del usuario. En esta cookie se almacenan el nombre de usuario y contraseña el cual se enviará en cada petición hacia el servidor, el cual se encargará de autorizar o no los Web Services basándose en el rol del usuario, indicado en a Figura 2.7.

Figura 2.7 Esquema de Seguridad Frontend



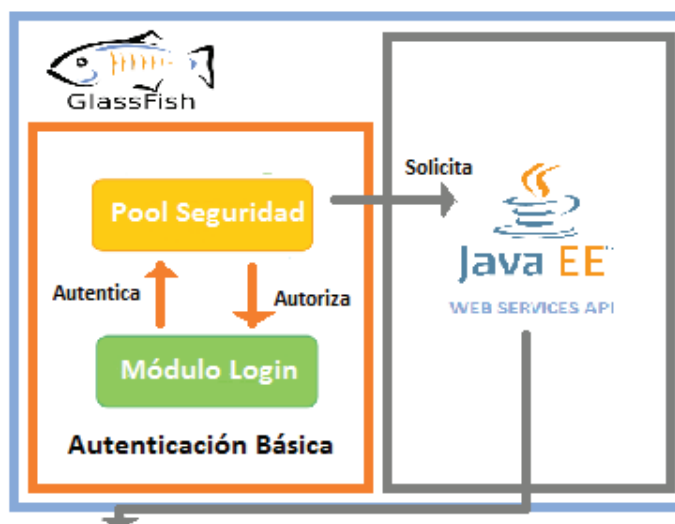
Elaborado por: Los autores

- Seguridad en el Backend

En el Backend la seguridad consta de dos módulos principales, el primero es un Web Service que se encarga de validar las credenciales enviadas por el cliente, este devuelve una respuesta al cliente en donde se especifica si las credenciales fueron aceptadas o rechazadas.

En caso de que las credenciales sean aceptadas éstas se devuelven al cliente el cual en cada llamada a los Web Services envía de regreso éstas credenciales para ser autenticadas (Autenticación Básica). El servidor de aplicaciones Glassfish se encarga de comprobar que este usuario se encuentre dentro del pool de seguridad (Figura 2.7) y le asigna los permisos establecidos dentro del descriptor de despliegue de la aplicación Web. XML (Figura 2.8)

Figura 2.8 Esquema de Seguridad Backend



Elaborado por: Los autores

Modelo Navegacional

El diagrama de la Figura 2.9, muestra el modelo navegacional del sistema AEIS WEB APP escrito en notación UWL. UWL son las siglas de “UML based Web Language” el cuál es una extensión de UML para Ingeniería Web. UWL usa puramente notación y diagramas UML para el análisis y diseño de páginas web. Para características específicas de la web UWL usa estereotipos definidos para los elementos de modelado.

Un modelo navegacional se encuentra definido por nodos y enlaces a estos. Un nodo se define como una unidad de navegación. UWL provee algunos

estereotipos para la construcción de modelos navegacionales. En nuestro caso se usaron los siguientes.

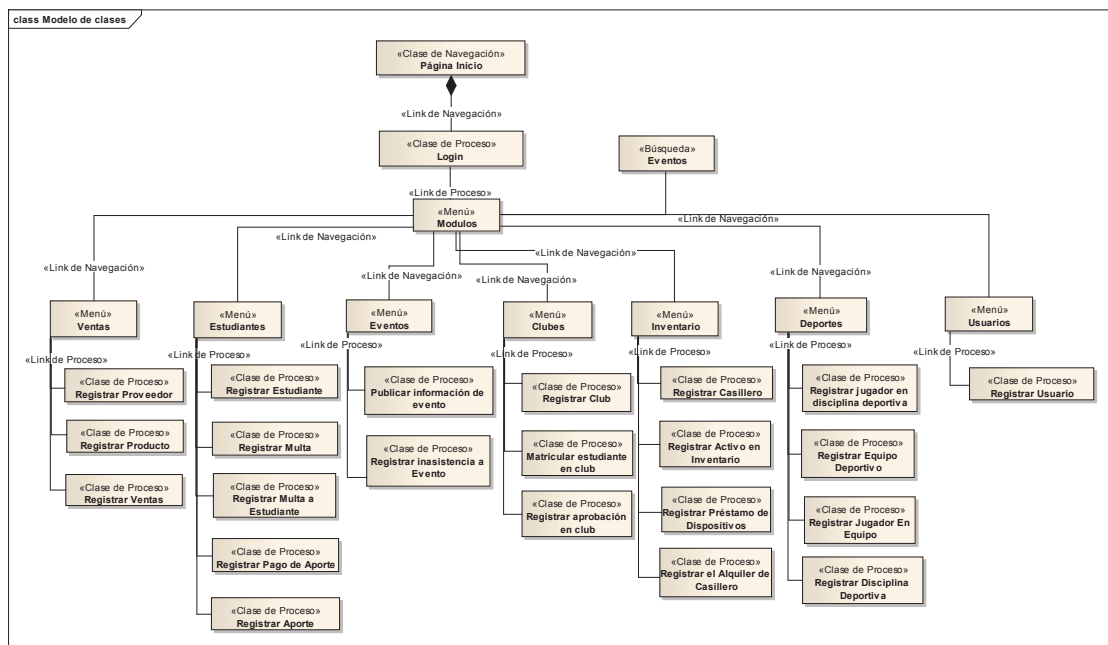
Nodos

- **Clase de Navegación:** Se refiere a una unidad de navegación que presenta estrictamente información estática o enlaces a otras páginas.
- **Clase de Proceso:** Se refiere a una unidad de navegación que ejecuta código y presenta resultados al cliente.
- **Menú:** Es una unidad de navegación que presenta un conjunto de enlaces a diferentes tipos de nodos.
- **Búsqueda:** Es una unidad de navegación que ejecuta una búsqueda en la base de datos y presenta esta información al cliente.

Enlaces (Links)

- **Link de Navegación:** Se refiere a un enlace hacia una clase de navegación.
- **Link de Proceso:** Se refiere a un enlace hacia una clase de proceso.

Figura 2.9 Modelo Navegacional

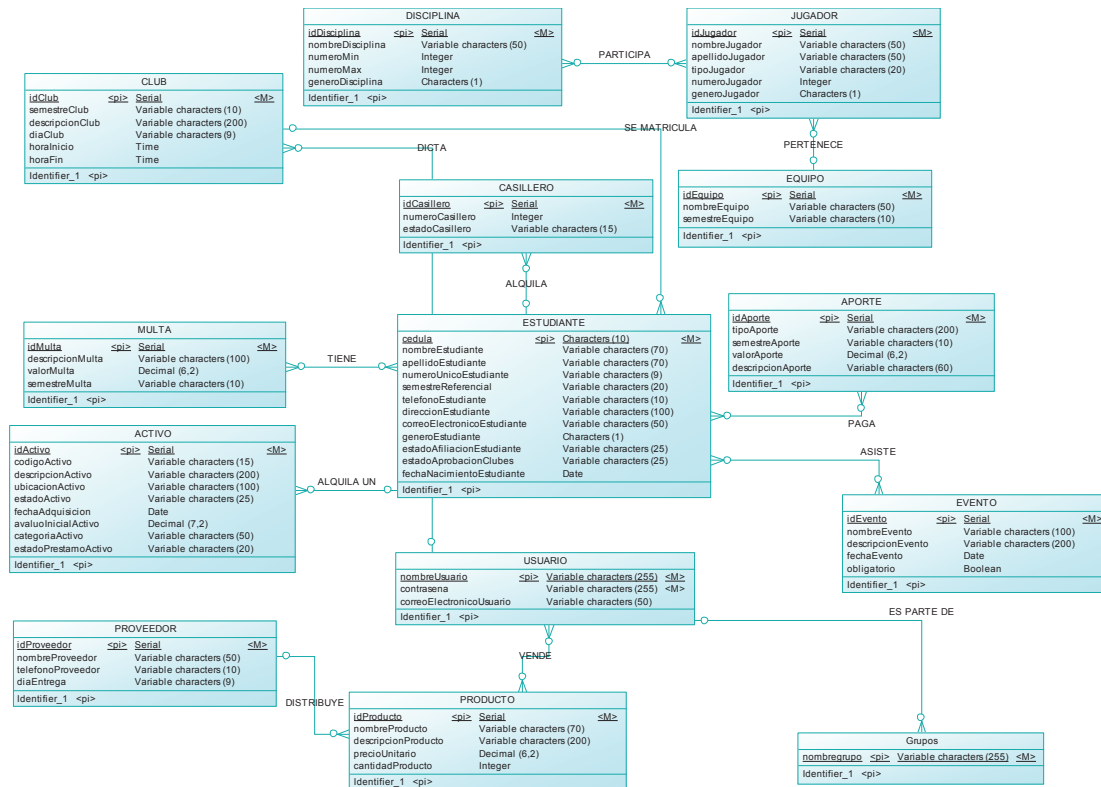


Elaborado por: Los autores

2.3.1.1.2 Modelo Conceptual de Datos

El modelo conceptual de datos se creó luego de analizar los requerimientos del sistema. La Figura 2.10 presenta el modelo conceptual de datos diseñado usando PowerDesigner.

Figura 2.10 Modelo Conceptual de Datos



Elaborado por: Los autores

Para realizar el análisis se usó el formato indicado en la Tabla 2.12, este formato contiene:

- **Entidad:** Nombre de la entidad
- **Requerimiento:** Indica el/los requerimiento(s) de acuerdo a las historias de usuario que justifican la creación de la entidad.
- **Atributo:** Nombre del atributo de la entidad
- **Tipo de Dato:** Indica el tipo de dato del atributo de la entidad
- **Longitud:** Indica la longitud del tipo de dato del atributo de la entidad.
- **Descripción:** Detalla el atributo

Tabla 2.12 Formato para analizar la creación de las entidades

ENTIDAD			
Requerimiento:			
Atributo	Tipo de Dato	Longitud	Descripción

Elaborado por: Los autores

Diseño de la entidad Activo

ACTIVO			
Requerimiento:			
Registrar el préstamo o devolución de un dispositivo (HU06)			
Atributo	Tipo de Dato	Longitud	Descripción
idActivo	Autoincremental		Identificador del registro.
codigoActivo	Varchar	15	Código Identificador otorgado por el departamento de bienes.
descripcionActivo	Varchar	200	Descripción del activo comprado.
ubicacionActivo	Varchar	100	Lugar dentro de la AEIS en donde se encuentra el activo.
estadoActivo	Varchar	25	Estado en el que se encuentra un activo: Bueno, Regular, Malo, Dado de baja.
fechaAdquisicion	Fecha		Fecha en la que se adquirió el activo.
avaluoInicialActivo	Decimal		Precio inicial por el cual se ha comprado el activo.
categoriaActivo	Varchar	20	Categoría del activo: Mueble, Dispositivo Electrónico, Electrodoméstico.
estadoPrestamoActivo	Varchar	20	Permite especificar si el activo se encuentra: Disponible o Prestado

Diseño de la entidad Aporte

APORTE			
Requerimiento:			
Registrar el pago del aporte (HU02)			
Atributo	Tipo de Dato	Longitud	Descripción
idAporte	Autoincremental		Identificador del registro
tipoAporte	Varchar	200	El tipo de aporte se refiere al valor por Afiliación, o por

			Deportes establecido por la AEIS
semestreAporte	Varchar	10	Semestre en el que se va a cobrar el aporte
valorAporte	Decimal		Valor en dólares del aporte.
descripcionAporte	Varchar	60	La descripción del aporte: puede ser: Baile de Gala, Clubes, Bautizo, etc.

Diseño de la entidad Casillero

CASILLERO			
Requerimiento: Registrar el alquiler de un casillero (HU07)			
Atributo	Tipo de Dato	Longitud	Descripción
idCasillero	Autoincremental		Identificador del registro.
numeroCasillero	Entero		Número del casillero.
estadoCasillero	Varchar	15	Estado en el que se encuentra un casillero: Disponible, No Disponible.

Diseño de la entidad Club

CLUB			
Requerimiento: Matricular estudiante en un Club (HU03) Registrar la aprobación del estudiante (HU04)			
Atributo	Tipo de Dato	Longitud	Descripción
idClub	Autoincremental		Identificador del registro.
semestreClub	Varchar	20	Semestre en el que se dicta el club.
descripcionClub	Varchar	200	Descripción del Club a dictarse.
diaClub	Varchar	9	Día en el que se dicta el club.
horalnicio	Hora		Hora de inicio del club.
horaFin	Hora		Hora de fin del club.

Diseño de la entidad Disciplina

DISCIPLINA			
Requerimiento: Registrar un equipo (HU11)			
Atributo	Tipo de Dato	Longitud	Descripción
idDisciplina	Autoincremental		Identificador del registro.

nombreDisciplina	Varchar	50	Nombre de la Disciplina Deportiva.
numeroMin	Entero		Número mínimo de jugadores admitidos en una disciplina deportiva.
numeroMax	Entero		Número máximo de jugadores admitidos en una disciplina deportiva.
generoDisciplina	Varchar	1	Género de los jugadores que pueden participar en la disciplina deportiva: Femenino (F), Masculino (M), Mixto (X).

Diseño de la entidad Equipo

EQUIPO			
Requerimiento: Registrar un equipo (HU11)			
Atributo	Tipo de Dato	Longitud	Descripción
idEquipo	Autoincremental		Identificador del registro.
nombreEquipo	Varchar	50	Nombre del Equipo Deportivo.
semestreEquipo	Varchar	10	Semestre en el que se registra el equipo.

Diseño de la Entidad Estudiante

ESTUDIANTE			
Requerimiento: Registrar datos personales del estudiante (HU01) Matricular estudiante en un Club (HU03)			
Atributo	Tipo de Dato	Longitud	Descripción
Cedula	Varchar	10	Identificador del registro.
nombreEstudiante	Varchar	70	Nombre del Estudiante.
apellidoEstudiante	Varchar	70	Apellido del Estudiante.
numeroUnicoEstudiante	Varchar	9	Número único.
semestreReferencial	Varchar	20	Semestre referencial en el que se encuentra un estudiante.
telefonoEstudiante	Varchar	10	Teléfono del estudiante.
direccionEstudiante	Varchar	100	Dirección domiciliaria del estudiante.
correoElectronicoEstudiante	Varchar	50	Dirección de correo electrónico del estudiante que se registra en la AEIS.

generoEstudiante	Varchar	1	Género del estudiante.
estadoAfiliacionEstudiante	Varchar	25	Estado de un estudiante con respecto a su afiliación a la AEIS: Afiliado, Desafiliado, Pendiente.
estadoAprobacionClubes	Varchar	25	Estado de un estudiante con respecto a su aprobación de Clubes: Aprobado, No Toma, Fallido.
fechaNacimientoEstudiante	Fecha		Fecha de Nacimiento del Estudiante.

Diseño de la entidad Evento

EVENTO			
Requerimiento: Publicar la información de un evento (HU08)			
Atributo	Tipo de Dato	Longitud	Descripción
idEvento	Autoincremental		Identificador del registro.
nombreEvento	Varchar	100	Nombre del evento.
descripcionEvento	Varchar	200	Descripción del evento a realizarse.
fechaEvento	Fecha		Fecha en la que la AEIS realizará un evento.
obligatorio	Booleano		Indica si el evento es obligatorio o no.

Diseño de la entidad Grupos

GRUPOS			
Requerimiento: Inicio de sesión			
Atributo	Tipo de Dato	Longitud	Descripción
nombreGrupo	Varchar	50	Nombre del grupo al cuál pertenecen los usuarios.

Diseño de la entidad Jugador

Debido a que existen varios Jugadores que pertenecen a un mismo equipo se creó una tabla específica para ello

JUGADOR			
Requerimiento: Registrar un equipo (HU11)			
Atributo	Tipo de Dato	Longitud	Descripción
idJugador	Autoincremental		Identificador del registro.

nombreJugador	Varchar	50	Nombre del Jugador.
apellidoJugador	Varchar	50	Apellido del Jugador.
tipoJugador	Varchar	1	Tipo de Jugador: Extranjero, Trabajador, Estudiante Afiliado, Estudiante no afiliado.
numeroJugador	Entero		Número del jugador.
generoJugador	Varchar	1	Género del Jugador.

Diseño de la entidad Multa

MULTA			
Requerimiento: Registrar la inasistencia a un evento (HU10)			
Atributo	Tipo de Dato	Longitud	Descripción
idMulta	Autoincremental		Identificador del registro.
descripcionMulta	Varchar	100	Descripción de la multa.
valorMulta	Decimal		Valor de una multa en dólares.
semestreMulta	Varchar	10	Semestre en el que se creó la multa.

Diseño de la entidad Producto

PRODUCTO			
Requerimiento: Registrar ventas (HU05)			
Atributo	Tipo de Dato	Longitud	Descripción
idProducto	Autoincremental		Identificador del registro.
nombreProducto	Varchar	70	Nombre del Producto.
descripcionProducto	Varchar	200	Descripción del Producto.
precioUnitario	Decimal		Precio unitario del Producto.
cantidadProducto	Entero		Cantidad de producto disponible para vender.

Diseño de la entidad Proveedor

Debido a que existen varios proveedores que distribuyen productos se creó una tabla específica para ello.

PROVEEDOR			
Requerimiento: Registrar ventas (HU05)			
Atributo	Tipo de Dato	Longitud	Descripción
idProveedor	Autoincremental		Identificador del registro.

nombreProveedor	Varchar	50	Nombre del Proveedor.
telefonoProveedor	Varchar	10	Teléfono del Proveedor.
diaentrega	Varchar	9	Día de entrega de los productos por parte del proveedor.

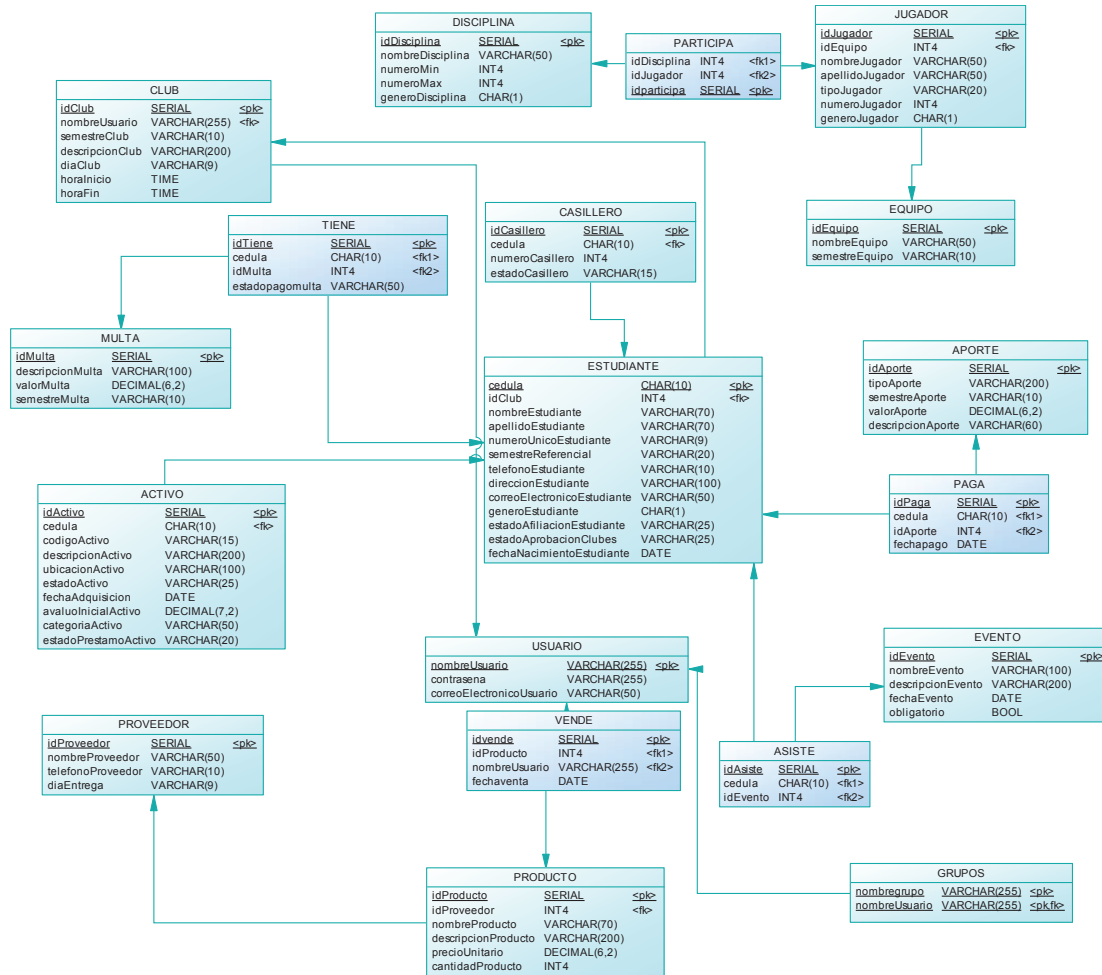
Diseño de la entidad Usuario

USUARIO			
Requerimiento: Inicio de Sesión			
Atributo	Tipo de Dato	Longitud	Descripción
nombreUsuario	Varchar	15	Nombre del Usuario, para ingresar al sistema.
nombreCompleto	Varchar	50	Nombre completo del usuario.
contrasena	Varchar	100	Contraseña del usuario para ingresar al sistema.
correoElectronicoUsuario	Varchar	50	Dirección de correo electrónico del estudiante que se registra en la AEIS.

2.3.1.1.3 Modelo Físico de la Base de Datos

La Figura 2.11 muestra el modelo físico que se generó en base al modelo conceptual utilizando la herramienta PowerDesigner.

Figura 2.11 Modelo Físico de la Base de Datos



Elaborado por: Los autores

2.3.1.1.4 Diseño de las Interfaces

En la Figura 2.12 se presenta el prototipo de la interfaz que permite el registro de estudiantes.

Figura 2.12 Interfaz de Registro de Estudiantes

Elaborado por: Los autores

En la Figura 2.13 se presenta el prototipo de la interfaz que permite la gestión de aportes.

Figura 2.13 Interfaz de Gestión de Aportes

Elaborado por: Los autores

En la Figura 2.14 se presenta el diseño de la interfaz que permite el registro de pagos.

Figura 2.14 Interfaz de Registro de Pagos

Elaborado por: Los autores

2.3.1.2 Implementación

Las tareas que requieren implementación en el primer Sprint son:

- 1.4 Generar el código SQL.
- 1.5 Generar RESTful web services a partir de la base de datos.
- 2.1 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
- 2.2 Crear un método que permita actualizar los datos del estudiante.
- 2.3 Validar los datos ingresados.
- 2.4 Crear un método que permita registrar al estudiante, en caso de que la búsqueda no devuelva ningún resultado.
- 2.6 Emitir un mensaje de confirmación al guardar/actualizar la información personal del estudiante.
- 3.2 Implementar la lógica de la interfaz para gestión de aportes.
- 3.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
- 3.4 Crear un método que permita registrar el pago que realiza el estudiante.
- 3.6 Crear un método que permita registrar la desafiliación del estudiante.
- 3.7 Emitir un mensaje de confirmación al registrar el pago del aporte.

2.3.1.2.1 Código SQL

El código de la base de datos fue generado usando la herramienta PowerDesigner, la Figura 2.15 muestra parte del script de creación de la base de datos, el script completo se encuentra en el Anexo D.

Figura 2.15 Código SQL

```

/*=====*/
/* Table: ACTIVO */
/*=====*/
create table ACTIVO (
  IDACTIVO          SERIAL          not null,
  CEDULA            CHAR(10)        null,
  IDUSUARIO         INT4            null,
  CODIGOACTIVO      VARCHAR(15)     null,
  DESCRIPCIONACTIVO VARCHAR(200)     null,
  UBICACIONACTIVO  VARCHAR(100)    null,
  ESTADOACTIVO     VARCHAR(25)     null,
  FECHAADQUISICION DATE            null,
  AVALUOINICIALACTIVO DECIMAL(7,2)  null,
  CATEGORIAACTIVO  VARCHAR(20)     null,
  constraint PK_ACTIVO primary key (IDACTIVO)
);

/*=====*/
/* Table: APORTE */
/*=====*/
create table APORTE (
  IDAPORTE          SERIAL          not null,
  TIPOAPORTE        VARCHAR(200)    null,
  SEMESTREAPORTE    VARCHAR(10)     null,
  VALORAPORTE       DECIMAL(6,2)    null,
  DESCRIPCIONAPORTE VARCHAR(60)     null,
  constraint PK_APORTE primary key (IDAPORTE)
);

```

Elaborado por: Los autores

2.3.1.2.2 RESTful web services

La Figura 2.16 muestra un ejemplo de los RESTful web services que se generaron para cada una de las tablas a partir de la base de datos.

Figura 2.16 Ejemplo de RESTful Web Services

```

29 @Stateless
30 @Path("estudiante")
31 public class EstudianteFacadeREST extends AbstractFacade<Estudiante> {
32     @PersistenceContext(unitName = "com.mycompany_TesisBackend_war_1.0-SNAPSHOTPU")
33     private EntityManager em;
34
35     public EstudianteFacadeREST() {
36         super(Estudiante.class);
37     }
38
39     @POST
40     @Override
41     @Consumes({"application/xml", "application/json"})
42     public void create(Estudiante entity) {
43         super.create(entity);
44     }
45
46     @PUT
47     @Path("/{id}")
48     @Consumes({"application/xml", "application/json"})
49     public void edit(@PathParam("id") String id, Estudiante entity) {
50         super.edit(entity);
51     }
52

```

Elaborado por: Los autores

La Figura 2.17 muestra el método que realiza la búsqueda de un estudiante ingresando la cédula.

Figura 2.17 Código Fuente para buscar un estudiante

```

190 $scope.buscarEstudiante = function (idEstudiante) {
191
192     estudianteServicio.get({id: idEstudiante}, function (data) {
193         if (typeof (data.nombreestudiante) === 'undefined') {
194             dialogs.error("Error", "No existe estudiante");
195             $scope.esEstudianteNuevo = true;
196
197         }
198         else {
199             $scope.estudiante = data;
200             $scope.valorBotonRegistro = "Actualizar Datos Estudiante";
201             $scope.isUpdate = true;
202         }
203
204
205
206     }, function () {
207         dialogs.error("Error", "Ocurrió un error al intentar buscar el estudiante");
208     });
209
210 };

```

Elaborado por: Los autores

La Figura 2.18 muestra el método que realiza la actualización de los datos del estudiante.

Figura 2.18 Código Fuente para actualizar los datos de un estudiante

```

$scope.actualizarEstudiante = function () {
    var idEstudiante = $scope.estudiante.cedula;
    var estudiante = $scope.estudiante;

    estudianteServicio.update({id: idEstudiante}, estudiante, function () {

        dialogs.notify("Notificación", "Se han actualizado los datos del estudiante");
        $scope.isUpdate = false;
        $scope.valorBotonRegistro = "Registrar Estudiante";
        $scope.estudiante = [];
    });
};

```

Elaborado por: Los autores

La Figura 2.19 muestra el método que realiza la validación de la cédula.

Figura 2.19 Código Fuente para validar cédula

```

$scope.check_cedula = function ($value)
{
    var array = [];
    var num = '';
    var total = '';
    var digito = '';
    var mult = '';
    var decena = '';
    var final = '';

    var cedula = $value;
    if (typeof cedula === 'undefined') {
        cedula = '0';
    }
    array = cedula.split("");
    num = array.length;
    if (num === 10)
    {
        total = 0;
        digito = (array[9] * 1);
        for (var i = 0; i < (num - 1); i++)
        {
            mult = 0;
            if ((i % 2) !== 0) {
                total = total + (array[i] * 1);
            }
        }
    }
}

```

```

    }
    else
    {
        mult = array[i] * 2;
        if (mult > 9)
            total = total + (mult - 9);
        else
            total = total + mult;
    }
}
decena = total / 10;
decena = Math.floor(decena);
decena = (decena + 1) * 10;
final = (decena - total);
if ((final === 10 && digito === 0) || (final === digito)) {

    return true;
}
else
{

    return false;
}
}
else
{

    return false;
}
}

```

Elaborado por: Los autores

La Figura 2.20 muestra el método que realiza la validación del número telefónico.

Figura 2.20 Código Fuente para validar el número telefónico

```

<input ng-model="estudiante.telefonoestudiante" type="tel"
      name="telefonoEstudiante"
      class="form-control" placeholder="Ingrese su teléfono"
      ng-pattern="/^[0-9]*$/ "
      ng-maxlength="9"
      ng-minlength="7"
      title="Solo se aceptan números locales o con código de provincia. Ej: 2222222 o 0222222222 "
      style="width: 200px; margin-left: 88px;"
      required/>

```

Elaborado por: Los autores

La Figura 2.21 muestra el método que permite registrar al estudiante, en caso de que la búsqueda no devuelva ningún resultado.

Figura 2.21 Código Fuente para registrar estudiante

```

estudianteServicio.save({
  cedula: $scope.estudiante.cedula,
  apellidoestudiante: $scope.estudiante.apellidoestudiante,
  correoelectronicoestudiante: $scope.estudiante.correoelectronicoestudiante,
  direccionestudiante: $scope.estudiante.direccionestudiante,
  fechanacimientoestudiante: $scope.estudiante.fechanacimientoestudiante,
  generoestudiante: $scope.estudiante.generoestudiante,
  nombreestudiante: $scope.estudiante.nombreestudiante,
  estadoaprobacionclubes: 'Por Determinar',
  numerounicoestudiante: $scope.estudiante.numerounicoestudiante,
  semestrereferencial: $scope.estudiante.semestrereferencial,
  telefonoestudiante: $scope.estudiante.telefonoestudiante
}, function () {
  dialogs.notify("Notificación", "Estudiante registrado con éxito");
  $scope.estudiante = [];
  $scope.esEstudianteNuevo = false;
});

```

Elaborado por: Los autores

Debido a que en la Tarea 1.5 se generaron los RESTful web services desde la base de datos, para la Tarea 3.2 simplemente se colocaron las llamadas a cada método de acuerdo a la interfaz diseñada en la Tarea 3.1.

El método que se requiere implementar en la Tarea 3.3 ya fue implementado en la Tarea 2.1, por lo que solamente se realizó la llamada al método de acuerdo a la lógica de la interfaz.

La Figura 2.22 muestra el código que se utiliza para registrar el pago que realiza el estudiante.

Figura 2.22 Código Fuente para registrar el pago del aporte

```

pagaAporteServicio.save({
  cedula: $scope.estudiante,
  fechapago: fechaPago,
  idaporte: $scope.jsonDetalleAporte[i]
}, function () {

```

Elaborado por: Los autores

La Figura 2.23 muestra el código que se utiliza para registrar la desafiliación de un estudiante.

Figura 2.23 Código Fuente para registrar la desafiliación de un estudiante

```

$scope.registrarDesafiliacion = function (idEstudiante) {

    if (!$scope.verificarExistenciaMulta()) {
        var dlg = dialogs.confirm();
        dlg.result.then(function (btn) {
            $scope.estudiante.estadoafiliacionestudiante = "Desafiliado";

            estudianteServicio.update({id: idEstudiante}, $scope.estudiante, function () {

                dialogs.notify("Notificación", "Se ha registrado la desafiliación del estudiante");
                $scope.estudiante = [];
            });
        }, function (btn) {

        });
    }
};

```

Elaborado por: Los autores

2.3.1.2.3 Emisión de Mensajes

La Figura 2.24 muestra el código que se utiliza para emitir mensajes de confirmación:

Figura 2.24 Código Fuente para emitir mensajes

```

var dlg = dialogs.confirm();

```

Elaborado por: Los autores

En la tarea 2.6 se desarrolló el método para emitir mensajes de confirmación, para la tarea: 3.9 simplemente se realizó la llamada al método ya implementado.

2.3.2 SEGUNDO SPRINT

2.3.2.1 Diseño

Las tareas que requieren diseño en el segundo Sprint son:

- 4.1 Crear la interfaz para la gestión de proveedores y productos.
- 4.5 Crear la interfaz para registrar las ventas.
- 5.1 Crear la interfaz para la gestión de clubes.
- 5.6 Crear la interfaz para matricularse en un club.
- 6.4 Crear la interfaz para registrar la aprobación del estudiante.

2.3.2.1.1 Diseño de las Interfaces

En la Figura 2.25 se presenta el diseño de la interfaz que permite la gestión de proveedores.

Figura 2.25 Interfaz de Gestión de Proveedores

The screenshot shows a web browser window with the URL `localhost:9000/#/registroProveedor`. The page title is "Registrar Proveedor" and the breadcrumb is "Módulos > Ventas > Registrar Proveedor". On the left, there is a sidebar menu with "Módulos" and several options: Ventas, Estudiantes, Eventos, Clubes, Inventario, Deportes, and Usuarios. The main content area contains a form with the following fields: "Proveedor:" (text input), "Teléfono:" (text input), and "Día de Entrega:" (dropdown menu). A "Registrar Proveedor" button is located to the right of the form. Below the form, there is a section titled "Proveedores Registrados:" with a "Buscar Proveedor:" search input and a table with columns: "Proveedor", "Teléfono", "Día Entrega", and "Opciones".

Elaborado por: Los autores

En la Figura 2.26 se presenta el diseño de la interfaz que permite la gestión de productos.

Figura 2.26 Interfaz de Gestión de Productos

The screenshot shows a web browser window with the URL `localhost:9000/#/registroProducto`. The page title is "Registrar Producto" and the breadcrumb is "Módulos > Ventas > Registrar Producto". On the left, there is a sidebar menu with "Módulos" and several options: Ventas, Estudiantes, Eventos, Clubes, Inventario, Deportes, and Usuarios. The main content area contains a form with the following fields: "Nombre:" (text input), "Descripción:" (text input), "Precio Unitario:" (text input), "Cantidad:" (text input), and "Proveedor:" (dropdown menu). A "Registrar Producto" button is located to the right of the form.

Elaborado por: Los autores

En la Figura 2.27 se presenta el diseño de la interfaz que permite el registro de ventas.

Figura 2.27 Interfaz de Registro de Ventas

Módulos > Ventas > Registrar Producto

Registrar Producto

Nombre:

Descripción:

Precio Unitario:

Cantidad:

Proveedor:

Productos Registrados:

Buscar Producto:

Nombre	Descripción	Precio Uni...	Cantidad	Proveedor	Opciones
Yogurt	Yogurt	0.8	25	Inalecsa	<input type="button" value="Eliminar"/>
Jugo	Jugo	1.1	8	Inalecsa	<input type="button" value="Eliminar"/>

Elaborado por: Los autores

En la Figura 2.28 se presenta el diseño de la interfaz que permite la gestión de clubes.

Figura 2.28 Interfaz de Gestión de Clubes

Módulos > Ventas > Registrar Producto

Registrar Producto

Nombre:

Descripción:

Precio Unitario:

Cantidad:

Proveedor:

Productos Registrados:

Buscar Producto:

Nombre	Descripción	Precio Uni...	Cantidad	Proveedor	Opciones
Yogurt	Yogurt	0.8	25	Inalecsa	<input type="button" value="Eliminar"/>
Jugo	Jugo	1.1	8	Inalecsa	<input type="button" value="Eliminar"/>

Elaborado por: Los autores

En la Figura 2.29 se presenta el diseño de la interfaz para matricularse en un club.

Figura 2.29 Interfaz de Matriculación en Clubes

Módulos > Clubes > Matricular Estudiante en Club

Estudiante:

Cédula:

Nombre Estudiante:

Clubes Disponibles

Semestre:

Clubes:

Registro de Matriculación:

Nombre	Profesor	Día	Hora Inicio	Hora Fin	Opciones

Elaborado por: Los autores

En la Figura 2.30 se presenta el diseño de la interfaz para registrar la aprobación del estudiante.

Figura 2.30 Interfaz de Registro de Aprobación

Módulos > Clubes > Registrar Aprobación en Club

Estudiante:

Cédula:

Estudiante:

Estado Aprobación:

Registrar Aprobación:

Estado Aprobación:

Copyright © 2014 Facultad de Ingeniería de Sistemas
Escuela Politécnica Nacional del Ecuador

Elaborado por: Los autores

2.3.2.2 Implementación

Las tareas que requieren implementación en el segundo Sprint son:

- 4.2 Implementar la lógica de la interfaz para gestión de proveedores y productos.
- 4.3 Crear un método que permita buscar un producto.
- 4.4 Crear un método que permita registrar la venta.
- 4.6 Emitir un mensaje de confirmación al registrar la venta.
- 5.2 Implementar la lógica de la interfaz para gestión de clubes.
- 5.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
- 5.4 Crear un método que muestre la lista de clubes a dictarse durante el semestre.
- 5.5 Crear un método que permita registrar al estudiante en el club seleccionado.
- 5.7 Emitir un mensaje de confirmación al matricular al estudiante en el club.
- 6.1 Crear un método que muestre la lista de clubes que se dictaron durante el semestre.
- 6.2 Crear un método que despliegue la lista de los estudiantes matriculados en el club seleccionado.
- 6.3 Crear un método que permita registrar la aprobación del estudiante.
- 6.5 Emitir un mensaje de confirmación al registrar la aprobación del estudiante.

En la tarea 2.1 implementada en el primer Sprint se creó el método para buscar a un estudiante de acuerdo a su número de cédula, para la tarea 5.3 solamente se llamará al método ya creado.

En la Tarea 1.5 se generaron los RESTful web services desde la base de datos, para las Tareas: 4.2 y 5.2 simplemente se colocaron las llamadas a cada método de acuerdo a las interfaces diseñadas en la tarea 4.1 y 5.1 respectivamente.

La Figura 2.31 muestra el código que se utiliza para buscar un producto.

Figura 2.31 Código Fuente para buscar un producto

```

$scope.filterNephi = function () {
    var filterText = 'nombreproducto:' + $scope.varFilter;

    var longitud = $scope.jsonProductosDisponibles.length;
    // var filterText = 'nombreProducto:' + $scope.nameFilter;
    if (filterText !== 'nombreproducto:') {
        $scope.filterOptions.filterText = filterText;
    } else {
        $scope.filterOptions.filterText = '';
    }
}
};

```

Elaborado por: Los autores

La Figura 2.32 muestra el código que se utiliza para registrar la venta.

Figura 2.32 Código Fuente para registrar la venta

```

$scope.venderProducto = function () {
    var index = this.row.entity.index;
    var stockOriginal = $scope.jsonProductosDisponibles [index] ["cantidadproducto"];
    $scope.jsonProductosDisponibles [index] ["cantidadproducto"]--;
    var stockNuevo = $scope.jsonProductosDisponibles [index] ["cantidadproducto"];
    var cantidadVendida = stockOriginal - stockNuevo;
    var banderaExisteItem = $scope.buscarSiExisteItemEnVenta($scope.jsonProductosDisponibles
        [index] ["idproducto"]);
    if (banderaExisteItem === false) {
        $scope.productoVendido = [];
        $scope.productoVendido = $scope.jsonProductosDisponibles[index];
        $scope.productoVendido.cantidadvendida = cantidadVendida;
        $scope.ventaActual.push($scope.productoVendido);
        if ($scope.jsonProductosDisponibles[index] ["cantidadproducto"] === 0) {
            $scope.jsonProductosDisponibles.splice(index, 1);
        }
    }
    else {
        var idProducto = $scope.jsonProductosDisponibles [index] ["idproducto"];
        for (var k = 0; k < $scope.ventaActual.length; ++k) {
            if (idProducto === $scope.ventaActual[k] ["idproducto"]) {
                $scope.ventaActual[k] ["cantidadvendida"]++;
            }
        }
    }
}

```

Elaborado por: Los autores

La Figura 2.33 muestra el código que permite obtener la lista de clubes que se dictan durante el semestre.

Figura 2.33 Código Fuente para mostrar clubes

```

$scope.buscarClubPorSemestre = function (semestre) {
  clubServicio.findBySemestre({semestreclub: semestre}, function (data) {
    $scope.jsonClubesDisponibles = data;
  }, function () {
    dialogs.error("Error", "Ocurrió un error al intentar obtener la lista de clubes");
  });
};

```

Elaborado por: Los autores

La Figura 2.34 muestra el código que permite registrar a un estudiante en el club seleccionado

Figura 2.34 Código Fuente para registrar a un estudiante en un club

```

$scope.finalizarMatricula = function () {
  var idEstudiante = $scope.estudiante.cedula;
  var idClub = $scope.registroMatricula[0]["idclub"];
  $scope.estudiante.idclub = idClub;
  $scope.estudiante.estadoaprobacionclubes = "Por Determinar";
  estudianteServicio.update({id: idEstudiante}, $scope.estudiante, function () {
    dialogs.notify("Notificación", "Se ha matriculado el estudiante en el club");
    $scope.registroMatricula = [];
    $scope.estudiante = [];
  }, function () {
    dialogs.error("Error", "Ocurrió un error mientras se matriculaba al estudiante en el club");
  });
};

$scope.maticularEstudiante = function () {
  if ($scope.registroMatricula.length < 1) {
    $scope.registroMatricula.push($scope.clubSeleccionado);
  }
  else {
    dialogs.error("Error", "Solo puede matricularse en un club por semestre, si \n\
desea matricularse en otro club elimine el que está matriculado actualmente");
  }
};

```

Elaborado por: Los autores

La Figura 2.35 muestra el código para registrar la aprobación del estudiante.

Figura 2.35 Código Fuente para registrar la aprobación del estudiante

```

$scope.registrarAprobacionEstudiante = function () {
  var idEstudiante = $scope.estudiante.cedula;
  $scope.estudiante.estadoaprobacionclubes = $scope.estadoARegistrarAEstudiante;
  estudianteServicio.update({id: idEstudiante}, $scope.estudiante, function () {

    dialogs.notify("Notificación", "Se ha registro el estado de la aprobación del club al estudiante");
    $scope.estudiante = [];
    $scope.estadoARegistrarAEstudiante = new String();
  }, function () {
    dialogs.error("Error", "Ha ocurrido un error durante el registro del estado de aprobación");
  });
};

```

Elaborado por: Los autores

En la tarea 2.6 del primer Sprint se desarrolló el método para emitir mensajes de confirmación, para las tareas: 4.6, 5.7 y 6.5 se realizó la llamada al método ya implementado.

2.3.3 TERCER SPRINT

2.3.3.1 Diseño

Las tareas que requieren diseño en el tercer Sprint son:

- 7.2 Crear la interfaz para la publicación de eventos.
- 8.1 Crear la interfaz para la gestión de multas.
- 8.7 Crear la interfaz para el registro de inasistencia.
- 9.1 Crear la interfaz para la gestión de disciplinas deportivas.
- 9.4 Crear la interfaz para registrar un equipo.
- 9.6 Crear la interfaz para registrar un jugador.
- 9.8 Crear la interfaz para el registrar a un jugador en las disciplinas.
- 10.1 Crear la interfaz para la gestión de dispositivos.
- 10.7 Crear la interfaz para el registro de préstamo de un dispositivo.

2.3.3.1.1 Diseño de las Interfaces

La figura 2.36 muestra el diseño de la interfaz para publicar eventos.

Figura 2.36 Interfaz para Publicar Eventos

The screenshot shows a web browser window with the URL `localhost9000/#/publicarInformacionEvento`. The page has a sidebar menu on the left with the following items: Módulos, Ventas, Estudiantes, Eventos, Publicar Información de Evento (highlighted), Registrar inasistencia a evento, Clubes, Inventario, Deportes, and Usuarios. The main content area is titled 'Publicar Información Evento' and contains a form with the following fields: 'Nombre del evento' (text input), 'Descripción del Evento' (text input), 'Fecha Evento' (text input with a calendar icon), and 'Obligatoriedad' (dropdown menu with the option 'Elija la opción'). A blue 'Registrar Evento' button is positioned to the right of the form. Below the form, there is a section 'Eventos Registrados' with a table header: 'Nombre', 'Descripción', 'Fecha Realización', and 'Opciones'. The table body is currently empty.

Elaborado por: Los autores

La figura 2.37 muestra el diseño de la interfaz para la gestión de multas

Figura 2.37 Interfaz para Gestión de Multas

Elaborado por: Los autores

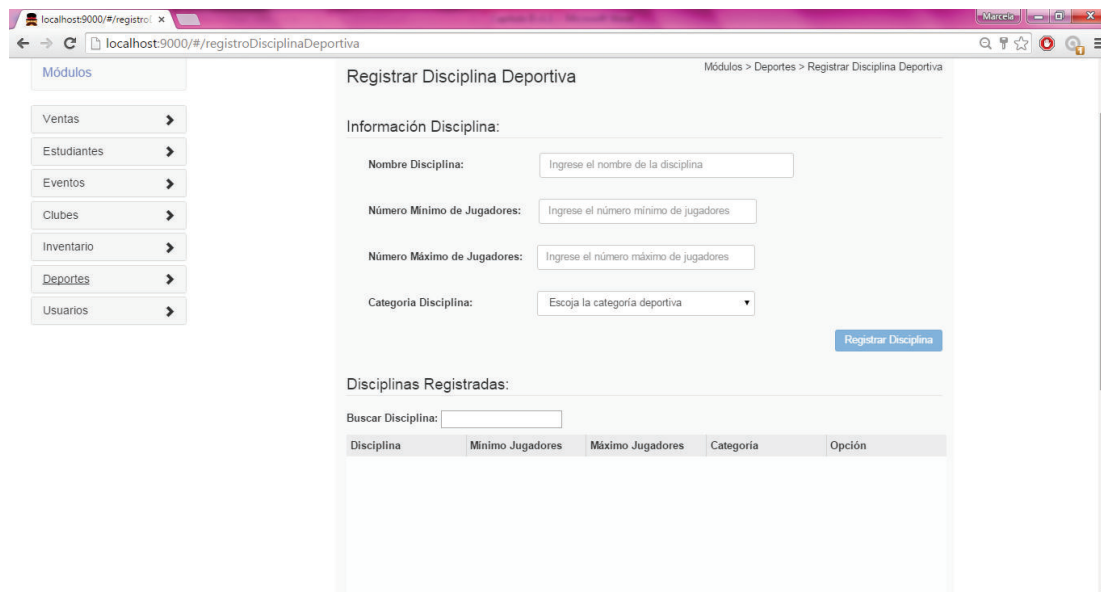
La figura 2.38 muestra el diseño de la interfaz para registrar la inasistencia

Figura 2.38 Interfaz para Registrar Asistencia

Elaborado por: Los autores

La figura 2.39 muestra el prototipo de la interfaz para la gestión de disciplinas deportivas.

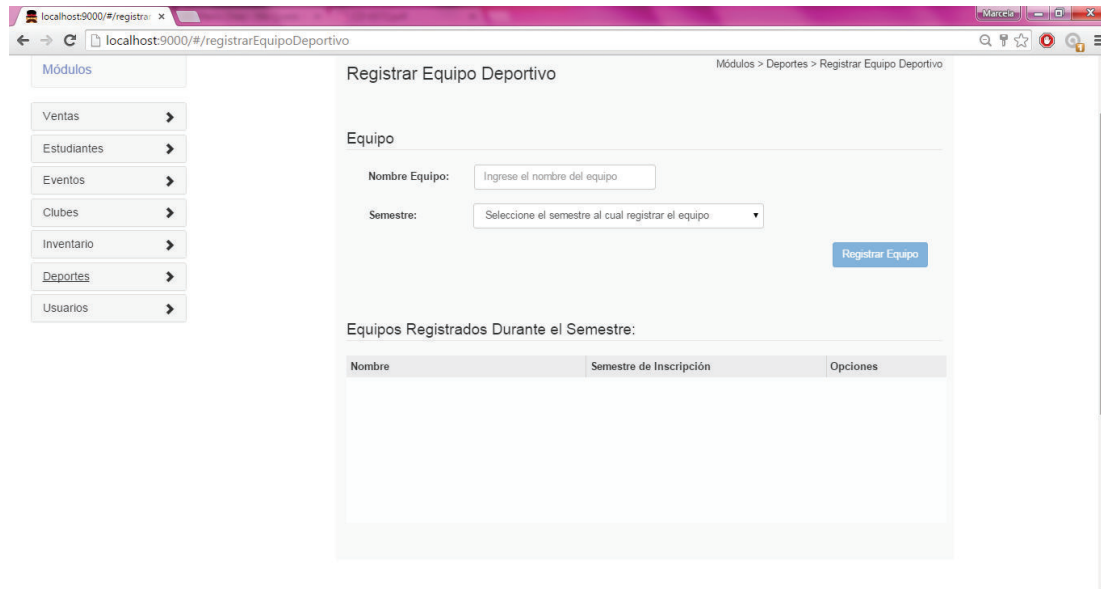
Figura 2.39 Interfaz para Gestión de Disciplinas Deportivas



Elaborado por: Los autores

La figura 2.40 muestra el prototipo de la interfaz para el registro de equipo.

Figura 2.40 Interfaz para Registrar Equipo



Elaborado por: Los autores

La figura 2.41 muestra el prototipo de la interfaz para el registro de un jugador.

Figura 2.41 Interfaz para Registrar Jugador

Elaborado por: Los autores

La figura 2.42 muestra el prototipo de la interfaz para el registro de un jugador en las disciplinas deportivas

Figura 2.42 Interfaz para Registrar Jugador en las Disciplinas Deportivas

Elaborado por: Los autores

La figura 2.43 muestra el diseño de la interfaz para la gestión de dispositivos.

Figura 2.43 Interfaz para Gestión de Dispositivos

Registrar Activo en Inventario

Módulos > Inventario > Registro Inventario

Información Activo:

Código:

Descripción:

Ubicación:

Avalúo:

Estado:

Categoría Activo:

Inventario:

Buscar Activo:

Código	Descripción	Ubicación	Estado	Fecha Adquisi...	Avaluo	Opciones
--------	-------------	-----------	--------	------------------	--------	----------

Elaborado por: Los autores

La figura 2.44 muestra el diseño de la interfaz para el registro de préstamo de un dispositivo.

Figura 2.44 Interfaz para registrar el préstamo de un dispositivo

Registrar Préstamo de Dispositivos

Módulos > Inventario > Registrar Préstamo de Dispositivos

Estudiante:

Buscar Estudiante:

Nombre Estudiante:

Lista de Dispositivos Disponibles:

Buscar Dispositivo:

Código	Descripción	Ubicación	Estado Activo	Opción
--------	-------------	-----------	---------------	--------

Dispositivos Prestados al Estudiante:

Descripción	Ubicación	Estado Activo	Opción
-------------	-----------	---------------	--------

Elaborado por: Los autores

2.3.3.2 Implementación

Las tareas que requieren implementación en el tercer Sprint son:

- 7.1 Crear un método que permita registrar un evento.

- 7.3 Emitir un mensaje de confirmación al publicar un evento.
- 8.2 Implementar la lógica de la interfaz para la gestión de multas.
- 8.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
- 8.4 Crear un método que muestre una lista de eventos realizados en un determinado periodo de tiempo.
- 8.5 Crear un método que permita asignar una multa al estudiante que no asistió a un evento obligatorio.
- 8.6 Emitir mensajes de confirmación.
- 9.2 Implementar la lógica de la interfaz para la gestión de disciplinas.
- 9.3 Crear un método que permita registrar un equipo.
- 9.5 Crear un método que permita registrar al jugador.
- 9.7 Crear un método que muestre las categorías disponibles en las que un jugador pueda participar.
- 10.2 Implementar la lógica de la interfaz para gestión de dispositivos.
- 10.3 Crear un método que permita buscar un dispositivo.
- 10.4 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
- 10.5 Crear un método que permita registrar el préstamo del dispositivo seleccionado a un estudiante.
- 10.6 Crear un método que muestre los dispositivos que se encuentran prestados a un estudiante.
- 10.8 Emitir un mensaje de confirmación al registrar el préstamo de un dispositivo.

El método que se requiere implementar en las Tareas 8.3 y 10.4 ya fue implementado en la Tarea 2.1 del primer Sprint, por lo que solamente se colocaron las llamadas al método de acuerdo a la lógica de cada interfaz, de igual manera debido a que en la Tarea 1.5 se generaron los RESTful web services desde la base de datos, para las Tareas 8.2, 9.2 y 10.2 simplemente se colocaron las llamadas a cada método de acuerdo a las interfaces diseñadas en la Tarea 8.1, 9.1 y 10.1 respectivamente.

La Figura 2.45 muestra el código que se utiliza para registrar un evento.

Figura 2.45 Código Fuente para registrar un evento

```

$scope.registrarEvento = function () {
  if (isUpdate === false) {
    eventoServicio.save({
      idusuario: $scope.usuario,
      nombreevento: $scope.evento.nombreevento,
      descripcionevento: $scope.evento.descripcionevento,
      fechaevento: $scope.evento.fechaevento,
      obligatorio: $scope.evento.obligatorio
    }, function () {
      dialogs.notify("Notificación", "Se ha registrado con éxito el Evento");
      $scope.evento = [];
      $scope.obtenerEventos();
    }, function () {
      dialogs.error("Error", "Ha ocurrido un error mientras se intentaba registrar el evento");
    });
  }
  else {
    var idEvento = $scope.evento.idevento;
    eventoServicio.update({id: idEvento}, $scope.evento, function () {
      dialogs.notify("Notificación", "Se ha actualizado con éxito el evento");
      $scope.evento = [];
      $scope.valorBotonRegistro = 'Registrar Evento';
      $scope.obtenerEventos();
      isUpdate = false;
    }, function () {
      dialogs.error("Error", "Ha ocurrido un error mientras se intentaba actualizar el evento");
    });
  }
}

```

Elaborado por: Los autores

La Figura 2.46 muestra el código que se utiliza para mostrar los eventos en un determinado tiempo.

Figura 2.46 Código Fuente para mostrar eventos

```

$scope.buscarEventos = function () {
  eventoServicio.findByDate({fechadesde: $scope.búsquedaEventos.desdeFecha,
    fechahasta: $scope.búsquedaEventos.hastaFecha}, function (data) {
    $scope.jsonEventos = data;
  }, function () {
    dialogs.error("Error", "Ocurrió un error al intentar buscar los eventos");
  });
};

$scope.gridOptions = {
  data: 'jsonEventos',
  columnDefs: [{field: 'nombreevento', displayName: 'Nombre Evento', width: "30%"},
    {field: 'descripcionevento', displayName: 'Descripción',
      width: "40%"},
    {field: 'fechaevento', displayName: 'Fecha evento',
      cellFilter: 'date:\'MM/dd/yyyy\'' , width: "27%"},
  ],
  jqueryUITheme: false,
  rowHeight: 70,
  filterOptions: $scope.filterOptions,
  afterSelectionChange: function (data) {
    $scope.evento = data.entity;
  }
}

```

Elaborado por: Los autores

La Figura 2.47 muestra el código que se utiliza para registrar la inasistencia de un estudiante a un evento obligatorio.

Figura 2.47 Código Fuente para asignar multas

```

$scope.registrarInasistencia = function (index) {

    if ($scope.jsonEventos[index]["obligatorio"] === true) {
        asisteServicio.save({
            cedula: $scope.estudiante,
            idevento: $scope.jsonEventos[index]
        }, function () {

        }, function () {
            dialogs.error("Error", "Ocurrió un error al intentar registrar la inasistencia");
        });
    } else {
        asisteServicio.save({
            cedula: $scope.estudiante,
            idevento: $scope.jsonEventos[index]
        }, function () {

            dialogs.notify("Notificación", "Se ha registrado con éxito la inasistencia");
        }, function () {
            dialogs.error("Error", "Ocurrió un error al intentar registrar la inasistencia");
        });
    }
};

```

Elaborado por: Los autores

La Figura 2.48 muestra el código que se utiliza para registrar un equipo.

Figura 2.48 Código Fuente para registrar un equipo

```

$scope.registrarEquipo = function () {
    if (isUpdate === false) {
        equipoServicio.save({
            nombreequipo: $scope.equipo.nombreequipo,
            semestreequipo: $scope.equipo.semestreequipo
        }, function () {
            $scope.buscarEquiposPorSemestre();
            $scope.equipo = [];
            dialogs.notify("Notificación", "Se ha registrado con éxito el equipo");
        }, function () {
            dialogs.error("Error", "Ocurrió un error al intentar registrar el equipo");
        });
    } else {
        equipoServicio.update({
            id: $scope.equipo.idequipo
        }, $scope.equipo, function () {
            dialogs.notify("Notificación", "Se ha actualizado con éxito el registro");
            isUpdate = false;
            $scope.buscarEquiposPorSemestre();
        }, function () {
            dialogs.error("Error", "Ha ocurrido un error mientras se intentaba actualizar el registro");
        });
    }
}

```

Elaborado por: Los autores

La Figura 2.49 muestra el código que se utiliza para registrar un jugador.

Figura 2.49 Código Fuente para registrar un jugador

```

$scope.registrarJugador = function () {
  if (isUpdate === false) {
    jugadorServicio.save({
      nombrejugador: $scope.jugador.nombrejugador,
      idequipo: $scope.jugador.idequipo,
      apellidojugador: $scope.jugador.apellidojugador,
      tipojugador: $scope.jugador.tipojugador,
      numerojugador: $scope.jugador.numerojugador,
      generojugador: $scope.jugador.generojugador
    }, function () {

      dialogs.notify("Notificación", "Se ha registrado con éxito el jugador");
      $scope.obtenerJugadoresPorEquipo();

    }, function () {
      dialogs.error("Error", "Ha ocurrido un error durante el registro");
    });
  } else
  {
    jugadorServicio.update({id: $scope.jugador.idjugador}, $scope.jugador, function () {

      dialogs.notify("Notificación", "Se ha actualizado con éxito el registro");
      isUpdate = false;
      $scope.obtenerJugadoresPorEquipo();
    }, function () {
      dialogs.error("Error", "Ocurrió un error al intentar actualizar el registro");
    });
  }
};

```

Elaborado por: Los autores

La Figura 2.50 muestra el código que se utiliza para mostrar las categorías disponibles en las que un jugador puede participar.

Figura 2.50 Código Fuente para mostrar categorías disponibles

```

$scope.obtenerDisciplinasRegistradas = function () {
  disciplinaServicio.query({}, function (data) {
    $scope.disciplinasRegistradas = data;
    var contador = 0;
    for (var i = 0; i < $scope.disciplinasRegistradas.length; i++) {
      $scope.disciplinasRegistradas[i]["index"] = contador++;
    }
  }, function () {
    dialogs.error("Error", "Ocurrió un error mientras intentaba obtener la lista de disciplinas");
  });
};

```

Elaborado por: Los autores

La Figura 2.51 muestra el código que se utiliza para buscar un dispositivo.

Figura 2.51 Código Fuente para buscar un dispositivo

```

$scope.obtenerDispositivosDisponibles = function () {
    activoServicio.findActivosDisponibles({estadoprestamoactivo: 'Disponible'}, function (data) {
        $scope.jsonDispositivosDisponibles = data;
    }, function () {
        dialogs.error("Error", "Hubo un error al buscar los dispositivos disponibles");
    });
};

```

Elaborado por: Los autores

La Figura 2.52 muestra el código que se utiliza para registrar el préstamo del dispositivo seleccionado a un estudiante.

Figura 2.52 Código Fuente para registrar el préstamo de un dispositivo

```

$scope.registrarPrestamo = function () {

    var index = this.row.rowIndex;

    $scope.jsonDispositivosDisponibles[index]["cedula"] = $scope.estudiante;
    $scope.jsonDispositivosDisponibles[index]["estadoprestamoactivo"] = "Prestado";

    activoServicio.update({id: $scope.jsonDispositivosDisponibles[index]["idactivo"]},
    $scope.jsonDispositivosDisponibles[index], function () {

        $scope.obtenerDispositivosPrestados();
        $scope.obtenerDispositivosDisponibles();
        $scope.disablePrestamos = true;

    }, function () {

    });

};

```

Elaborado por: Los autores

La Figura 2.53 muestra el código que se utiliza para mostrar los dispositivos que se encuentran prestados a un estudiante.

Figura 2.53 Código Fuente para mostrar los dispositivos prestados

```

$scope.obtenerDispositivosPrestados = function () {
    activoServicio.findActivosPrestados(
        {estadoprestamoactivo: 'Prestado',
         cedula: $scope.estudiante.cedula
        }, function (data) {
            $scope.prestamoActual = data;
            if ($scope.prestamoActual.length === 0) {
                $scope.disableDevolver = true;
            }
        }, function () {
            dialogs.error("Error", "Ocurrió un error al intentar obtener los dispositivos prestados");
        });
    });
};

```

Elaborado por: Los autores

En la tarea 2.6 del primer Sprint se desarrolló el método para emitir mensajes de confirmación, para las tareas: 7.3, 8.6 y 10.8 se realizó la llamada al método ya implementado.

2.3.4 CUARTO SPRINT

2.3.4.1 Diseño

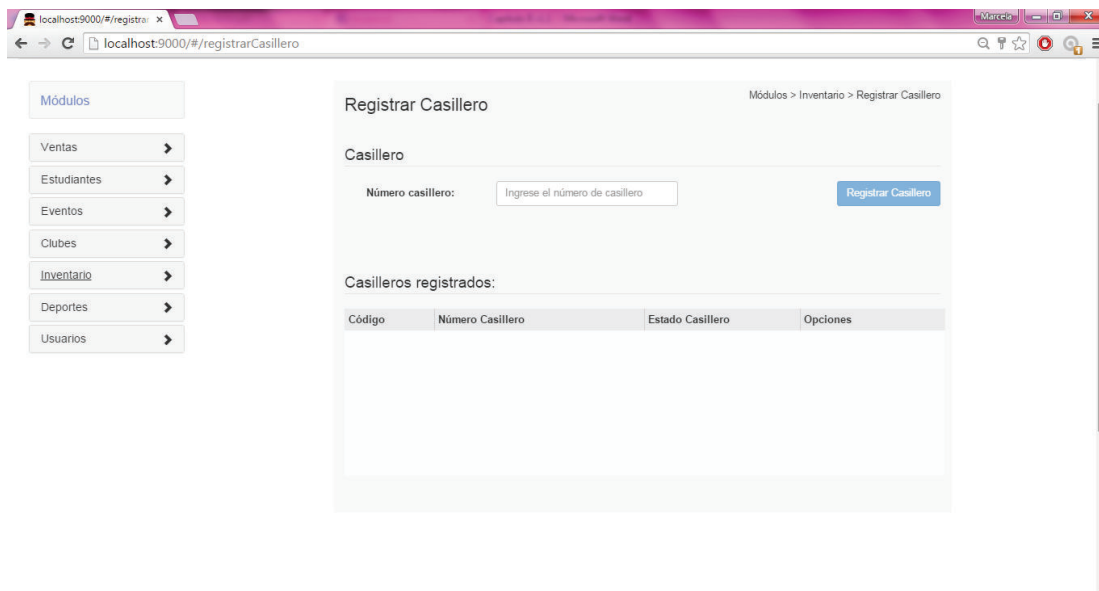
Las tareas que requieren diseño en el cuarto Sprint son:

- 11.1 Crear la interfaz para la gestión de casilleros.
- 11.7 Crear la interfaz para el alquiler de casilleros.
- 12.1 Crear la interfaz para la gestión de usuarios.
- 12.3 Crear la interfaz para el inicio de sesión
- 13.1 Crear la página de inicio e incluir el texto de visión y misión de la AEIS
- 13.2 Colocar enlaces a otras páginas relacionadas con la AEIS, la página oficial de la EPN y de la FIS.

2.3.4.1.1 Diseño de las Interfaces

La Figura 2.54 muestra el prototipo de la interfaz para la gestión de casilleros.

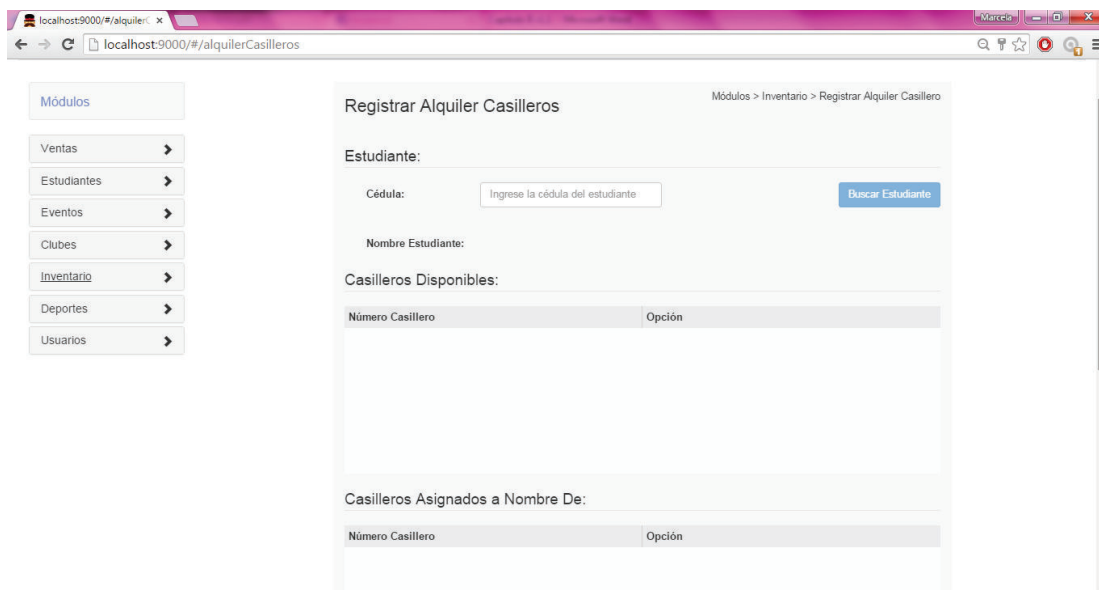
Figura 2.54 Interfaz para Gestión de Casilleros



Elaborado por: Los autores

La Figura 2.55 muestra el prototipo de la interfaz para el alquiler de casilleros.

Figura 2.55 Interfaz para Alquiler de Casilleros



Elaborado por: Los autores

La Figura 2.56 muestra el prototipo de la interfaz para gestión de Usuarios.

Figura 2.56 Interfaz para Gestión de Usuarios

Elaborado por: Los autores

La figura 2.57 muestra el prototipo de la interfaz para el Inicio de Sesión

Figura 2.57 Interfaz para Inicio de Sesión

Elaborado por: Los autores

La figura 2.58 muestra el prototipo de la interfaz que incluye la misión de la AEIS.

Figura 2.58 Interfaz para mostrar la misión de la AEIS



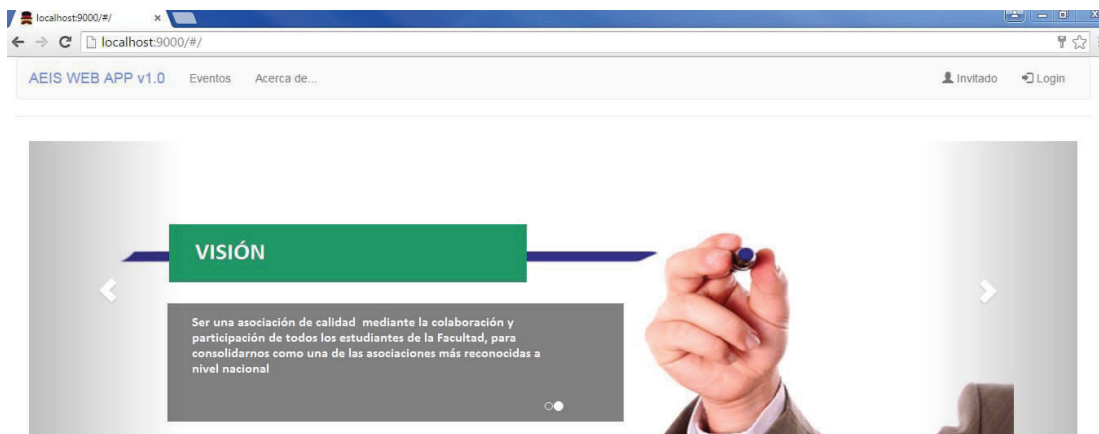
Asociación de Estudiantes de Ingeniería de Sistemas

Bienvenido al portal web de la Asociación de Estudiantes de Ingeniería de Sistemas AEIS.

Elaborado por: Los autores

La Figura 2.59 muestra el prototipo de la interfaz que incluye la visión de la AEIS.

Figura 2.59 Interfaz para mostrar la visión de la AEIS



Asociación de Estudiantes de Ingeniería de Sistemas

Bienvenido al portal web de la Asociación de Estudiantes de Ingeniería de Sistemas AEIS.

Elaborado por: Los autores

2.3.4.2 Implementación

Las tareas que requieren implementación en el cuarto Sprint son:

- 11.2 Implementar la lógica de la interfaz para gestión de casilleros.
- 11.3 Crear un método que permita buscar a un estudiante de acuerdo a su cédula.
- 11.4 Crear un método que muestre la lista de casilleros disponibles.
- 11.5 Crear un método que permita registrar el alquiler de un casillero a un estudiante.
- 11.6 Crear reportes que muestren los casilleros alquilados a los estudiantes.
- 11.8 Emitir un mensaje de confirmación al registrar el alquiler de un casillero
- 12.2 Implementar la lógica para la gestión de usuarios.
- 12.4 Crear un método de autenticación de usuarios.
- 12.5 Crear un método que muestre el menú de acuerdo al tipo de usuario.

El método que se requiere implementar en la tareas 11.3 ya fue implementado en la Tarea 2.1 del primer Sprint, por lo que solamente se colocó la llamada al método de acuerdo a la lógica de la interfaz, de igual manera debido a que en la Tarea 1.5 se generaron los RESTful web services desde la base de datos, para las Tareas 11.2 y 12.2 simplemente se colocaron las llamadas a cada método de acuerdo a las interfaces diseñadas en las tareas 11.1 y 12.1 respectivamente. La Figura 2.60 muestra el código que se utiliza para mostrar la lista de casilleros disponibles.

Figura 2.60 Código Fuente para mostrar casilleros disponibles

```
$scope.obtenerCasillerosDisponibles = function () {
    casilleroServicio.findCasillerosDisponibles({estadocasillero: 'Disponible'}, function (data) {
        $scope.jsonCasillerosDisponibles = data;
    }, function () {
        dialogs.error("Error", "Ha ocurrido un error al intentar obtener el registro de casilleros");
    });
};
```

Elaborado por: Los autores

La Figura 2.61 muestra el código que se utiliza para registrar el alquiler de un casillero a un estudiante.

Figura 2.61 Código Fuente para registrar el alquiler de un casillero

```

$scope.alquilarCasillero = function () {
    var index = this.row.rowIndex;
    $scope.jsonCasillerosDisponibles[index]["cedula"] = $scope.estudiante;
    $scope.jsonCasillerosDisponibles[index]["estadocasillero"] = 'Alquilado';
    casilleroServicio.update({id: $scope.jsonCasillerosDisponibles[index]["idcasillero"]},
    $scope.jsonCasillerosDisponibles[index], function () {
        $scope.obtenerCasillerosDisponibles();
        $scope.obtenerCasillerosAlquilados();
    }, function () {
        dialogs.error("Error", "Ha ocurrido un error al intentar registrar el alquiler del casillero");
    });
};

```

Elaborado por: Los autores

La Figura 2.62 muestra el código que se utiliza para la autenticación de usuarios.

Figura 2.62 Código Fuente para la autenticación de usuario

```

$scope.login = function () {
    $scope.dataLoading = true;
    var password=encryptService.hash($scope.password);

    AuthenticationService.Login($scope.username,password , function (response) {

        if (response.status===200) {
            AuthenticationService.SetCredentials($scope.username, $scope.password);
            $location.path('/');
        } else {
            $scope.error = response.message;
            $scope.dataLoading = false;
        }
    });
};
});

```

Elaborado por: Los autores

La Figura 2.63 muestra el código que se utiliza para mostrar el menú de acuerdo al tipo de usuario.

Figura 2.63 Código Fuente para mostrar el menú de acuerdo al tipo de usuario

```

$scope.verificarValidezGrupos = function () {
    $scope.banderaVentas = false;
    $scope.banderaRegistrarYMatricular = false;
    $scope.banderaEventos = false;
    $scope.banderaUsuarios = false;
    $scope.banderaClubes = false;
    $scope.banderaInventario = false;
    $scope.banderaDeportes = false;
    $scope.banderaRegistrarMultaYPagoAporte = false;
    for (var i = 0; i < $scope.rolesDelUsuarioActual.length; i++) {
        //Condiciones usuarios
        if ($scope.rolesDelUsuarioActual[i] === 'presidente') {
            $scope.banderaUsuarios = true;
        }
        //FIN Condiciones usuarios
        //Condiciones VENTAS
        if (($scope.rolesDelUsuarioActual[i] === 'presidente' ||
            $scope.rolesDelUsuarioActual[i] === 'empleado')) {
            $scope.banderaVentas = true;
        }
        //Condiciones Estudiante
        if ($scope.rolesDelUsuarioActual[i] === 'presidente' ||
            $scope.rolesDelUsuarioActual[i] === 'vicepresidentefinanciero')
        {
            $scope.banderaRegistrarMultaYPagoAporte = true;
        }
        //Fin Condiciones Estudiante
        //Condiciones Eventos
        if ($scope.rolesDelUsuarioActual[i] === 'presidente' ||
            $scope.rolesDelUsuarioActual[i] === 'empleado') {
            $scope.banderaEventos = true;
        }
        if ($scope.rolesDelUsuarioActual[i] === 'vicepresidentecultura') {
            $scope.banderaEventos = true;
        }
        //FIN Condiciones Eventos
        //Condiciones Clubes
        if ($scope.rolesDelUsuarioActual[i] === 'presidente' ||
            $scope.rolesDelUsuarioActual[i] === "vicepresidenteadademico")
            $scope.banderaClubes = true;
            $scope.banderaRegistrarYMatricular = true;
        }
        if ($scope.rolesDelUsuarioActual[i] === "profesor") {
            $scope.banderaClubes = true;
        }
        //FIN condiciones Clubes
        //Condiciones Inventario
        if ($scope.rolesDelUsuarioActual[i] === 'presidente' ||
            $scope.rolesDelUsuarioActual[i] === 'empleado') {
            $scope.banderaInventario = true;
        }
    }
}

```

Elaborado por: Los autores

En la tarea 2.6 del primer Sprint se desarrolló el método para emitir mensajes de confirmación, para la tarea: 11.8 se realizó la llamada al método ya implementado.

2.4 SPRINT REVIEW

De acuerdo a la metodología Scrum, al final de cada Sprint se llevó a cabo el Sprint Review, para esta reunión se realizaron pruebas de aceptación para verificar el cumplimiento de los requisitos detallados en las historias de usuario, usando el formato indicado en la Tabla 2.13, este formato contiene:

- **Número:** El número secuencial de la prueba de aceptación.
- **Nombre Historia:** Texto identificador de la historia de usuario sobre la cual se realizará la prueba.
- **Prerrequisitos:** Son los requisitos previos para realizar lo que describe la prueba de aceptación.
- **Pasos de Ejecución:** Pasos que deben ejecutarse para obtener el resultado.
- **Resultado Esperado:** Describe el resultado exitoso al ejecutar la prueba.
- **Evaluación:** Describe cualitativamente si el resultado de la prueba fue exitoso o fallido.

Tabla 2.13 Formato Pruebas de Aceptación

PRUEBA DE ACEPTACIÓN	PA #
Historia de Usuario:	
Prerrequisitos:	
Pasos de Ejecución:	
Resultado Esperado:	
Evaluación:	

Elaborado por: Los autores

2.4.1 PRIMER SPRINT

Para el Sprint Review correspondiente al primer sprint se elaboraron las pruebas de aceptación para las historias de usuario:

- HU01 Registrar datos personales
 - o 2.5 Interfaz para el registro de estudiantes
- HU02 Registrar el pago del aporte
 - o 3.1 Interfaz para gestión de aportes
 - o 3.5 Interfaz para el registro de pagos

PRUEBA DE ACEPTACIÓN	PA01
Historia de Usuario: HU01 Registrar datos personales del estudiante	
Prerrequisitos:	
Ninguno	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema con el rol: Presidente. - Acceder al módulo: Estudiantes opción Registrar Estudiante. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar. 	
Resultado Esperado:	
Se guardan los datos del estudiante y se muestra un mensaje indicando que los datos se registraron.	
Evaluación: Prueba Exitosa	

PRUEBA DE ACEPTACIÓN	PA02
Historia de Usuario: HU01 Registrar datos personales del estudiante	
Prerrequisitos:	
El estudiante debe estar registrado en el sistema.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema con el rol: Presidente. - Acceder al módulo: Estudiantes opción Registrar Estudiante. - Ingresar el número de cédula. - Clic en el botón Buscar Estudiante. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar. 	
Resultado Esperado:	
Actualización de los datos del estudiante y un mensaje indicando que los datos se actualizaron.	
Evaluación: Prueba Exitosa	

PRUEBA DE ACEPTACIÓN		PA03
Historia de Usuario: HU02 Registrar el pago del aporte		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Tesorero. - Acceder al módulo: Estudiantes opción Registrar aporte. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Aporte. 		
Resultado Esperado:		
Se registra el valor del aporte que tienen que pagar las personas que deseen afiliarse y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA04
Historia de Usuario: HU02 Registrar el pago del aporte		
Prerrequisitos:		
El estudiante que va a realizar el pago debe estar registrado en el sistema.		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Tesorero. - Acceder al módulo: Estudiantes opción Registrar pago de aporte. - Ingresar la cédula del estudiante. - Clic en el botón Buscar. - Seleccionar el semestre. - Clic en el botón Buscar Aportes. - Seleccionar los rubros que el estudiante va a cancelar. - Clic en el botón Registrar Afiliación. 		
Resultado Esperado:		
Se registra la afiliación del estudiante y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA05
Historia de Usuario: HU02 Registrar el pago del aporte		
Prerrequisitos:		
El estudiante que va a realizar el pago debe estar registrado en el sistema.		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Tesorero. - Acceder al módulo: Estudiantes opción Registrar pago de aporte. - Ingresar la cédula del estudiante. - Clic en el botón Buscar. - Clic en el botón Registrar Desafiliación. 		
Resultado Esperado:		

Se registra la desafiliación del estudiante y se muestra un mensaje de información.

Evaluación: Prueba Exitosa

2.4.2 SEGUNDO SPRINT

Para el Sprint Review correspondiente al Segundo Sprint se elaboraron las pruebas de aceptación para las historias de usuario:

- HU03 Matricular estudiante en un Club
 - o 5.1 Interfaz para gestión de club
 - o 5.6 Interfaz para matricularse en un club
- HU04 Registrar la aprobación del estudiante
 - o 6.4 Interfaz para registrar la aprobación del estudiante
- HU05 Registrar ventas
 - o 4.1 Interfaz para registrar proveedores y productos
 - o 4.5 Interfaz para registrar ventas

PRUEBA DE ACEPTACIÓN	PA06
Historia de Usuario: HU03 Matricular estudiante en un club	
Prerrequisitos:	
Ninguno	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente Académico. - Acceder al módulo: Clubes opción Registrar Club. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Club. 	
Resultado Esperado:	
El club queda registrado y se muestra un mensaje de información.	
Evaluación: Prueba Exitosa	

PRUEBA DE ACEPTACIÓN	PA07
Historia de Usuario: HU03 Matricular estudiante en un club	
Prerrequisitos:	
El estudiante debe estar registrado en el sistema.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente Académico - Acceder al módulo: Clubes opción Matricular Estudiante en Club - Ingresar la cédula del estudiante. - Clic en el botón Buscar Estudiante. - Seleccionar el semestre. 	

<ul style="list-style-type: none"> - Seleccionar el club en el que se va a matricular al estudiante. - Clic en el botón Matricularse. - Clic en el botón Finalizar matrícula.
Resultado Esperado:
El estudiante queda matriculado en el club y se muestra un mensaje de información.
Evaluación: Prueba Fallida No se muestran el profesor que dicta el club.

PRUEBA DE ACEPTACIÓN	PA08
Historia de Usuario: HU04 Registrar la aprobación del estudiante	
Prerrequisitos:	
El estudiante debe estar matriculado en un club.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Profesor - Acceder al módulo: Clubes opción Registrar aprobación en Club - Ingresar la cédula del estudiante. - Clic en el botón Buscar Estudiante. - Seleccionar el estado de aprobación. - Clic en el botón Registrar Aprobación. 	
Resultado Esperado:	
Se registra la aprobación del estudiante en un club.	
Evaluación: Prueba Exitosa	

PRUEBA DE ACEPTACIÓN	PA09
Historia de Usuario: HU05 Registrar ventas	
Prerrequisitos:	
Ninguno	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado - Acceder al módulo: Ventas opción Registrar Proveedor. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Proveedor. 	
Resultado Esperado:	
El proveedor queda registrado y se muestra un mensaje de información.	
Evaluación: Prueba Exitosa	

PRUEBA DE ACEPTACIÓN	PA10
Historia de Usuario: HU05 Registrar ventas	
Prerrequisitos:	
Ninguno	
Pasos de Ejecución:	

<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado - Acceder al módulo: Ventas opción Registrar Producto. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Producto.
Resultado Esperado:
El producto queda registrado y se muestra un mensaje de información.
Evaluación: Prueba Exitosa

PRUEBA DE ACEPTACIÓN	PA11
Historia de Usuario: HU05 Registrar ventas	
Prerrequisitos:	
Al menos un producto debe estar registrado en la base.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado - Acceder al módulo: Ventas opción Registrar Ventas - Ingresar el nombre del producto - Clic en el botón Vender - Clic en el botón Registrar Venta 	
Resultado Esperado:	
Se registra la venta y se muestra un mensaje de información.	
Evaluación: Prueba Exitosa	

2.4.3 TERCER SPRINT

Para el Sprint Review correspondiente al Tercer Sprint se elaboraron las pruebas de aceptación para las historias de usuario:

- HU06 Registrar el préstamo o devolución de un dispositivo
 - o 10.1 Interfaz para gestión de dispositivos
 - o 10.7 Interfaz para registro de préstamos
- HU08 Publicar la información de un evento
 - o 7.2 Interfaz para publicar eventos
- HU09 Registrar la inasistencia a un evento
 - o 8.1 Interfaz para gestión de multas
 - o 8.7 Interfaz para registro de inasistencia
- HU10 Inscripción de Equipo Deportivo
 - o 9.1 Interfaz para gestión de disciplinas deportivas
 - o 9.4 Interfaz para registrar un equipo
 - o 9.6 Interfaz para registrar un jugador en equipo

○ 9.8 Interfaz para registrar jugador en disciplinas

PRUEBA DE ACEPTACIÓN		PA12
Historia de Usuario: HU06 Registrar el préstamo o devolución de un dispositivo		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado - Acceder al módulo: Inventario opción Registrar Activo en Inventario. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Activo. 		
Resultado Esperado:		
El dispositivo queda registrado y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA13
Historia de Usuario: HU06 Registrar el préstamo o devolución de un dispositivo		
Prerrequisitos:		
Al menos un dispositivo debe estar disponible		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado - Acceder al módulo: Inventario opción Registrar préstamo de dispositivos. - Ingresar la cédula del estudiante. - Clic en el botón Buscar Estudiante - Seleccionar un dispositivo de la lista de dispositivos disponibles. - Clic en el botón Alquilar. 		
Resultado Esperado:		
Se registra el alquiler del dispositivo y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA14
Historia de Usuario: HU08 Publicar la información de un evento		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente de Cultura - Acceder al módulo: Eventos opción Publicar información de evento. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Evento. 		
Resultado Esperado:		
Se publica la información de un evento.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA15
Historia de Usuario: HU09 Registrar la inasistencia a un evento		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Presidente. - Acceder al módulo: Estudiante opción Registrar Multa. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Multa. 		
Resultado Esperado:		
La multa queda registrada y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA16
Historia de Usuario: HU09 Registrar la inasistencia a un evento		
Prerrequisitos:		
Debe existir al menos un evento registrado.		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente de Cultura. - Acceder al módulo: Eventos opción Registrar inasistencia a evento. - Ingresar la cédula del estudiante. - Ingresar un rango de fechas. - Clic en el botón buscar eventos. - Clic en el botón Registrar Inasistencia. 		
Resultado Esperado:		
Se registra la inasistencia y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA17
Historia de Usuario: HU10 Inscripción de Equipo Deportivo		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente de Deportes. - Acceder al módulo: Deportes opción Registrar Disciplina Deportiva. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Disciplina. 		
Resultado Esperado:		
La disciplina deportiva queda registrada y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA18
Historia de Usuario: HU10 Inscripción de Equipo Deportivo		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente de Deportes. - Acceder al módulo: Deportes opción Registrar equipo deportivo. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Equipo. 		
Resultado Esperado:		
El equipo deportivo queda registrado y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA19
Historia de Usuario: HU10 Inscripción de Equipo Deportivo		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente de Deportes. - Acceder al módulo: Deportes opción Registrar jugador en equipo. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Jugador 		
Resultado Esperado:		
El jugador queda registrado y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

PRUEBA DE ACEPTACIÓN		PA20
Historia de Usuario: HU10 Inscripción de Equipo Deportivo		
Prerrequisitos:		
Ninguno		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Vicepresidente de Deportes. - Acceder al módulo: Deportes opción Registrar Jugador en disciplina deportiva. - Ingresar la información solicitada en la interfaz. - Clic en el botón registrar jugador en disciplina. 		
Resultado Esperado:		
Se registran las disciplinas deportivas en las que participará el jugador y se muestra un mensaje de información.		
Evaluación: Prueba Exitosa		

Adicionalmente en este Sprint recibimos dos peticiones de parte del cliente:

- Colocar la misión y visión de la AEIS como parte de la página de inicio del portal
- Colocar enlaces a los sitios web oficiales de la EPN y de la FIS.

Estas dos tareas se implementarán en el Cuarto Sprint.

La prueba para la historia de usuario HU03 – Matricular estudiante en club, resultó fallida, por lo que se corregirá en el siguiente Sprint.

2.4.4 CUARTO SPRINT

Para el Sprint Review correspondiente al Cuarto Sprint se elaboraron las pruebas de aceptación para las historias de usuario:

- HU07 Registrar el alquiler del casillero
 - o 11.1 Interfaz para gestión de casilleros
 - o 11.7 Interfaz para alquiler de casilleros
- Inicio de Sesión

PRUEBA DE ACEPTACIÓN	PA21
Historia de Usuario: HU11 Registrar el alquiler de un casillero	
Prerrequisitos:	
Al menos un casillero debe estar disponible.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado. - Acceder al módulo: Inventario opción Registrar casillero. - Ingresar la información solicitada en la interfaz. - Clic en el botón Registrar Casillero. 	
Resultado Esperado:	
Se registra el casillero y se muestra un mensaje de información.	
Evaluación: Prueba Exitosa	

PRUEBA DE ACEPTACIÓN	PA22
Historia de Usuario: HU11 Registrar el alquiler de un casillero	
Prerrequisitos:	
Al menos un casillero debe estar disponible.	
Pasos de Ejecución:	
<ul style="list-style-type: none"> - Ingresar al sistema como usuario: Empleado. - Acceder al módulo: Inventario opción Registrar alquiler de casillero. - Ingresar la cédula del estudiante. - Clic en el botón Buscar. - Seleccionar un casillero de la lista de casilleros disponibles. - Clic en el botón Alquilar. 	
Resultado Esperado:	

Se registra el alquiler del casillero y se muestra un mensaje de información.
Evaluación: Prueba Exitosa

PRUEBA DE ACEPTACIÓN		PA23
Historia de Usuario: Inicio de Sesión		
Prerrequisitos:		
El usuario debe estar registrado en el sistema		
Pasos de Ejecución:		
<ul style="list-style-type: none"> - Seleccionar la opción: Login - Ingresar el usuario y contraseña - Clic en el botón Login 		
Resultado Esperado:		
Ingresar al Portal Web		
Evaluación: Prueba Exitosa		

2.4.5 ANÁLISIS DEL PROYECTO

Al terminar el desarrollo del portal web AEIS APP WEB, podemos realizar el siguiente análisis:

Al iniciar el proyecto se planificaron 78 tareas, que se cumplieron de acuerdo a lo planificado en cada Sprint. En el tercer Sprint Review el cliente planteó otros requisitos por lo que se incrementaron 2 tareas más, terminando con un Product Backlog de 80 tareas en total.

2.4.5.1 Gráfico de Esfuerzo

En gráfico de esfuerzo permite visualizar la cantidad de trabajo pendiente que tiene el equipo de desarrollo y la velocidad con la que está avanzando, en donde:

- El eje Y (esfuerzo): representa el trabajo pendiente por realizar
- El eje X (tiempo): representa los días del sprint.

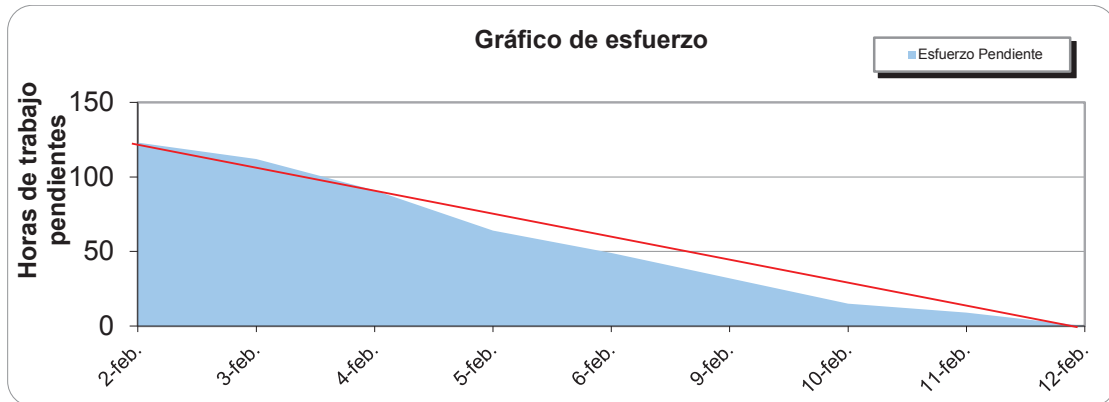
El avance ideal de Sprint se representa por una pendiente que reduce el esfuerzo de forma gradual desde el primer hasta el último día del sprint

2.4.5.1.1 Primer Sprint

La figura 2.64 representa el gráfico del esfuerzo realizado durante el Primer Sprint en el cual se completaron 133 Story Points en 9 días. El avance ideal se representa con la línea roja y el esfuerzo real con el área azul. Se puede observar que al inicio de este Sprint el ritmo de trabajo del equipo era lento, por lo que el

área azul sobrepasa la línea roja del esfuerzo ideal, pese a esto se cumplieron todas las tareas del primer sprint.

Figura 2.64 Gráfico de Esfuerzo del Primer Sprint

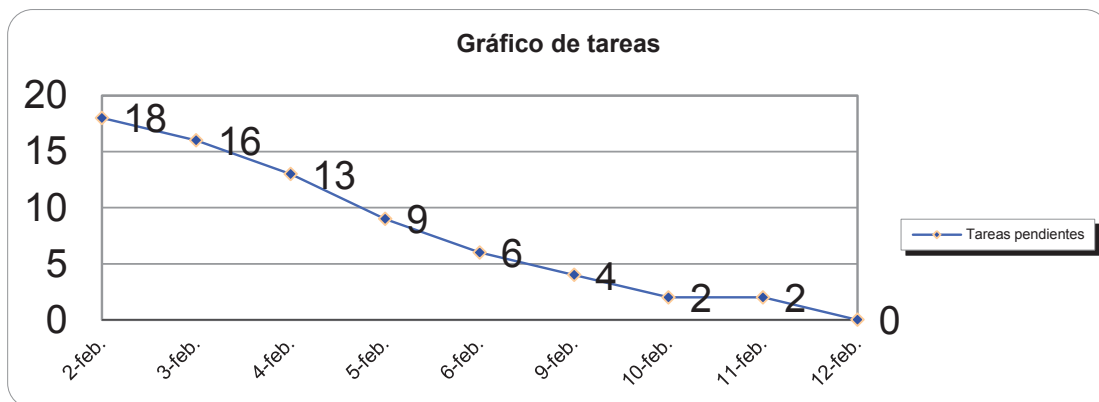


Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

En la Figura 2.65 se observa el avance de las tareas del primer sprint, este gráfico muestra que conforme los días el equipo avanzó sin novedades al cumplir las tareas planificadas.

Figura 2.65 Gráfico de avance de tareas en el Primer Sprint



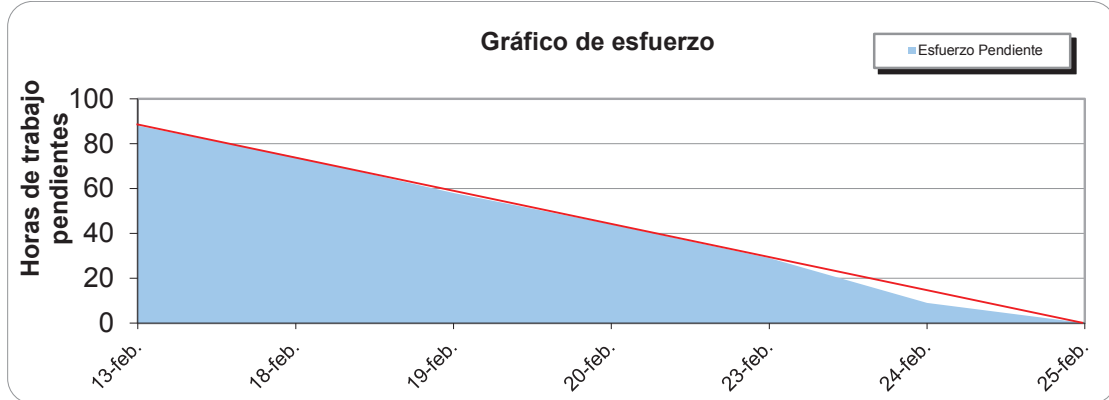
Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.4.5.1.2 Segundo Sprint

La figura 2.66 representa el gráfico del esfuerzo realizado durante el segundo Sprint en el cual se completaron 120 Story Points en 9 días. El avance ideal se representa con la línea roja y el esfuerzo real con el área azul, en este caso el esfuerzo real coincide con el esfuerzo ideal.

Figura 2.66 Gráfico de Esfuerzo del Segundo Sprint

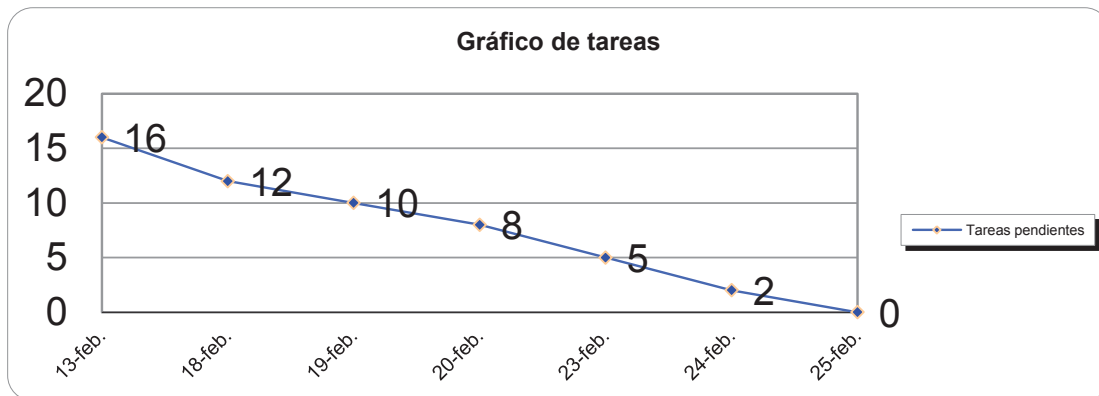


Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

En la Figura 2.67 se observa el avance de las tareas del segundo sprint, este gráfico muestra que conforme los días el equipo avanzó sin novedades al cumplir las tareas planificadas.

Figura 2.67 Gráfico de avance de tareas en el Segundo Sprint



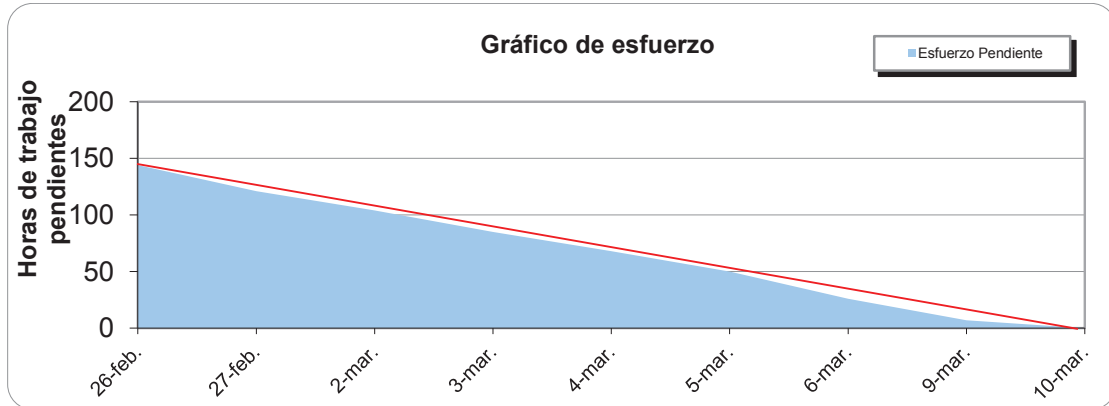
Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.4.5.1.3 Tercer Sprint

La Figura 2.68 representa el gráfico del esfuerzo realizado durante el tercer Sprint en el cual se completaron 138 Story Points en 9 días. El avance ideal se representa con la línea roja y el esfuerzo real con el área azul, en este caso el esfuerzo real coincide con el esfuerzo ideal.

Figura 2.68 Gráfico de Esfuerzo del Tercer Sprint

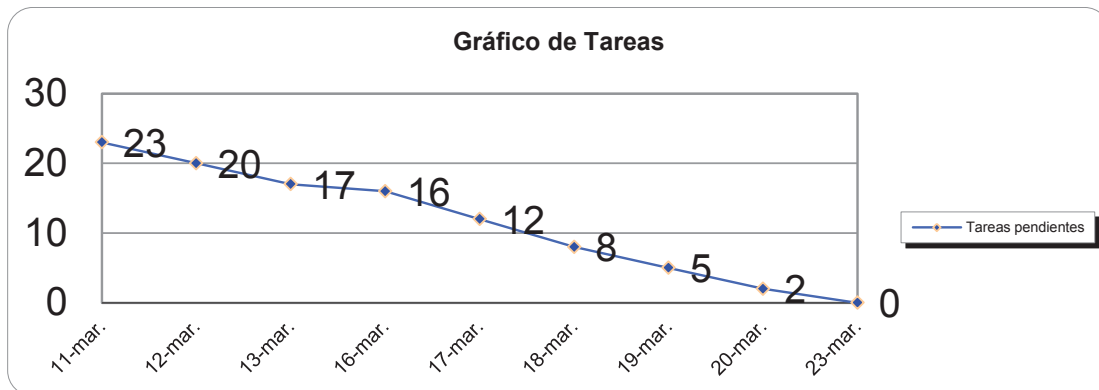


Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

En la Figura 2.69 se observa el avance de las tareas del tercer sprint, este gráfico muestra que conforme los días el equipo avanzó sin novedades al cumplir las tareas planificadas.

Figura 2.69 Gráfico de avance de tareas en el Tercer Sprint



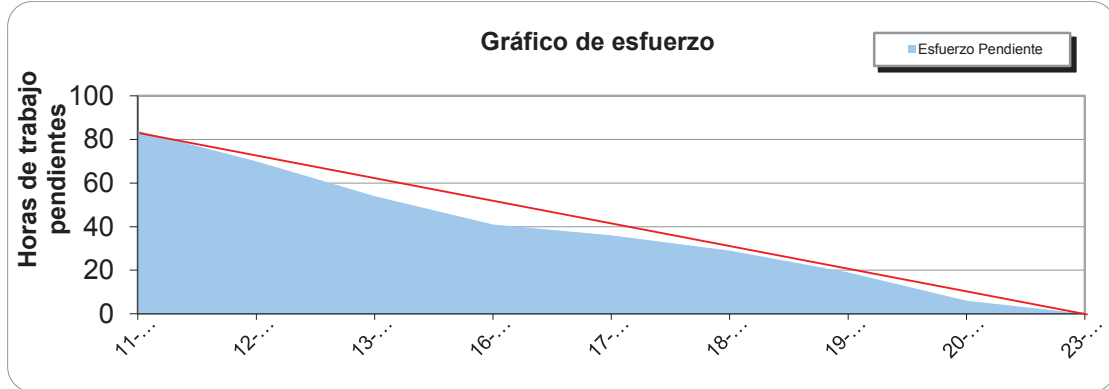
Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

2.4.5.1.4 Cuarto Sprint

La Figura 2.70 representa el gráfico del esfuerzo realizado durante el cuarto Sprint en el cual se completaron 99 Story Points en 9 días. El avance ideal se representa con la línea roja y el esfuerzo real con el área azul, en este caso el esfuerzo real coincide con el esfuerzo ideal.

Figura 2.70 Gráfico de Esfuerzo del Cuarto Sprint

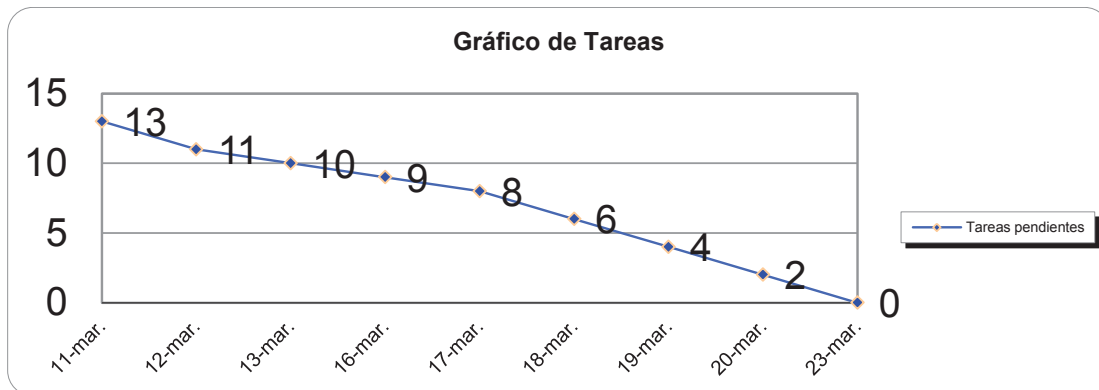


Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

En la Figura 2.71 se observa el avance de las tareas del cuarto Sprint, este gráfico muestra que conforme los días el equipo avanzó sin novedades al cumplir las tareas planificadas.

Figura 2.71 Gráfico de avance de tareas en el Segundo Sprint



Elaborado por: Los Autores

Fuente: Juan Palacio, 2006, <http://www.navegapolis.net/>

3 CAPÍTULO III: EVALUACIÓN DEL PORTAL

Una vez terminado el desarrollo y las pruebas del Portal, se procederá a realizar una demostración del funcionamiento del mismo en un ambiente de producción simulado.

3.1 RECOLECCIÓN DE LOS DATOS DE EVALUACIÓN DE LA AEIS

Debido a que la AEIS no dispone de una base de datos que se pueda usar para migrar la información al Portal Web, se optó por recolectar los datos: estudiantes, productos y proveedores e inventario para generar un script que permita cargar esta información directamente a la base de datos.

3.1.1 INFORMACIÓN DE ESTUDIANTES

Para cargar la información de los estudiantes se solicitó en secretaría de la Facultad el listado de información personal obtenido del sistema SAEW. En la Tabla 3.1 se presenta parte de la información a utilizar en el ambiente de producción simulado.

Tabla 3.1 Recolección de información de estudiantes

cedula	nombre	apellido	numero unico	semestre	telefono	direccion	correo	genero (F=M)	fecha nacimiento
1745698532	Kenny	López	201210232	Tercera	2 752 951	Bolívar y Mariana de Jerár	lopez@hotmail.com	M	05/07/1993
1725489652	Daniela	Ramar	201220454	Cuarta	2 874 632	10 de Agosto y Calón	ramar@gmail.com	F	03/02/1992
1734521453	Jefferson	Paredor	201210123	Quinta	2 654 321	Diego de Almagro y Orellana	paredor.j@gmail.com	M	06/06/1993
1541267851	Maria	Mantora	201020456	Quinta	2 332 452	Venzalloy Cuera y Caiceda	mantora@yahoo.es	M	06/05/1990
1798563412	Katerina	Vela	201120987	Sexta	2 426 426	Juan García y Pedro Fuentes	vela.k@hotmail.com	F	25/09/1991
1745203085	Jarú	Zurita	201011285	Tercera	2 429 615	Pedra Carbay Diego de Almagro	zurita.j@hotmail.com	M	24/03/1989
1702548952	Jarú	Ochoa	200920456	Séptima	2 815 325	Carlar Freilo y Pedro Carba	ochoa.jm@gmail.com	M	14/05/1990

Elaborado por: Los autores

3.1.2 INFORMACIÓN DE PRODUCTOS Y PROVEEDORES

La información de productos y proveedores se obtuvo de las facturas que maneja la AEIS. En las Tablas 3.2 y 3.3 se presenta parte de la información a utilizar en el ambiente de producción simulado.

Tabla 3.2 Recolección de información de proveedores

nombre	telefono	dia entrega
Toni	2 824 657	Lunes
Nestle	2 345 659	Martes
Inalesa	2 458 965	Martes
Banchis	2 338 745	Viernes
Carli	2 745 152	Viernes
Alpina	2 531 845	Miércoles
Van Camps	2 546 324	Miércoles

Elaborado por: Los autores

Tabla 3.3 Recolección de información de productos

nombre producto	descripción	precio unitario	cantidad
Tostitos	Tostitos	0,45	50
Leche Toni	Leche Toni	0,8	20
Panchitos	Panchitos	0,25	30
Mentas	Mentas	0,05	100
Pan de chocolate	Pan de chocolate	0,5	15
Melbas	Melbas	0,5	15
Galletas Amor	Galletas Amor	0,85	20
Gaseosa Pepsi	Gaseosa Pepsi	0,75	30
Agua Cielo persona	Agua Cielo persona	0,4	40
Nachos	Nachos	0,45	25
Agua cielo 1litro	Agua cielo 1litro	0,6	15
Chifles	Chifles	0,4	35
Tocinetas	Tocinetas	0,5	20
Picaditas	Picaditas	0,5	20
Pa'Fritas	Pa'Fritas	0,25	40
Avena alpina (enva	Avena alpina (enva	0,9	20
Avena alpina (fund	Avena alpina (fund	0,5	10
Atún	Atún	1,1	10
Quíntuple	Quíntuple	0,6	35

Elaborado Por: Los autores

3.1.3 INFORMACIÓN DE INVENTARIO

La información del inventario disponible fue proporcionada por la directiva de la AEIS. En la Tabla 3.4 se presenta la información a utilizar en el ambiente de producción simulado.

Tabla 3.4 Recolección de información de inventario

codigoActivo	decripcionActivo	ubicacionActivo	estadoActivo	fechaAdqui	avaluoInici	categoriaActivo
123	Mesa de ping por	Bar AEIS	Bueno	01/02/2014	150	Mueble
1234	Juego de Muebles	Bar AEIS	Bueno	10/03/2014	550	Mueble
122	Mesa de fútbolín	Bar AEIS	Regular	17/05/2013	600	Mueble
165	Televisión	Bar AEIS	Bueno	23/05/2014	1100	Dispositivo Electrónico
124	Proyector	Laboratorio AEIS	Bueno	01/06/2013	850	Dispositivo Electrónico
960	Computador	Laboratorio AEIS	Bueno	01/06/2013	750	Dispositivo Electrónico
961	Computador	Laboratorio AEIS	Bueno	01/06/2013	750	Dispositivo Electrónico
962	Computador	Laboratorio AEIS	Bueno	01/06/2013	750	Dispositivo Electrónico
963	Computador	Laboratorio AEIS	Bueno	01/06/2013	750	Dispositivo Electrónico

Elaborado por: Los autores

3.2 IMPLANTACIÓN DEL SISTEMA

Para la implantación del sistema se requiere de dos servidores un servidor de Backend y otro de Frontend.

3.2.1 REQUISITOS

3.2.1.1 Hardware

En la Tabla 3.5 se describen los requisitos mínimos de hardware para el servidor de Backend.

Tabla 3.5 Requisitos de Hardware para los servidores de Backend

COMPONENTE	MÍNIMO	RECOMENDADO
Procesador	2GHz	2 GHz
Memoria RAM	4 GB	6 GB
Disco Duro	200 GB	500 GB

Elaborado Por: Los autores

En la Tabla 3.6 se describen los requisitos mínimos de hardware para el servidor Frontend.

Tabla 3.6 Requisitos de Hardware para los servidores de Frontend

COMPONENTE	MÍNIMO	RECOMENDADO
Procesador	2 GHz	2 GHz
Memoria RAM	4 GB	6 GB
Disco Duro	100 GB	500 GB

Elaborado Por: Los autores

3.2.1.2 Software

En la Tabla 3.7 se describen los requisitos de software para el servidor de Backend.

Tabla 3.7 Requisitos de Software para el servidor de Backend

SOFTWARE	DESCRIPCIÓN
Sistema Operativo	Windows 7 o superior
Gestor de Base de Datos	PostgreSQL versión 9.4

Elaborado Por: Los autores

En la Tabla 3.8 se describen los requisitos de software para el servidor de Frontend.

Tabla 3.8 Requisitos de Software para el servidor de Frontend

SOFTWARE	DESCRIPCIÓN
Sistema Operativo	Windows 7 o superior
Entorno de Ejecución	Node.js

Elaborado Por: Los autores

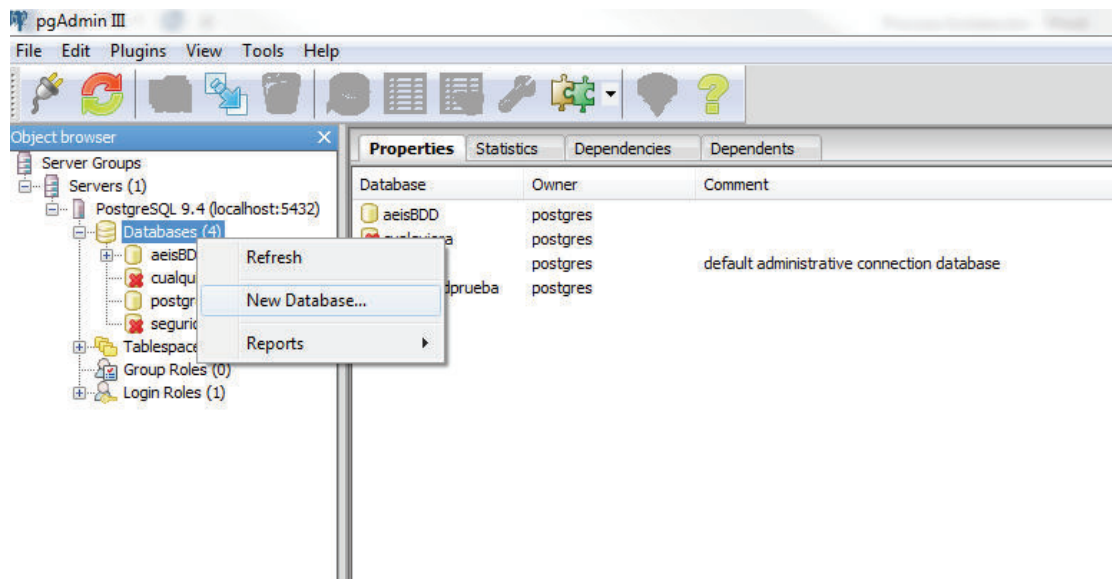
3.2.2 PROCESO DE INSTALACIÓN

3.2.2.1 Instalación Backend

3.2.2.1.1 Instalación de la Base de Datos

- Ejecutar el instalador de PostgreSQL que se ajuste a la arquitectura del computador.
- Instalar la aplicación con la clave de administrador aeis2015.
- Una vez terminada la instalación abrir la interfaz gráfica de PostgreSQL llamada pgAdminIII
- Conectarse al servidor instalado con la clave aeis2015
- Una vez conectado dar clic derecho sobre “Databases” y seleccionar “New Database”

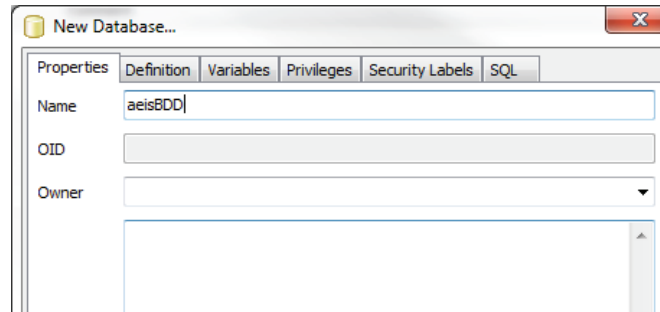
Figura 3.1 Creación de la Base de Datos – Parte I



Elaborado por: Los autores

- Crear una nueva base de datos de nombre: aeisBDD y presionar ok

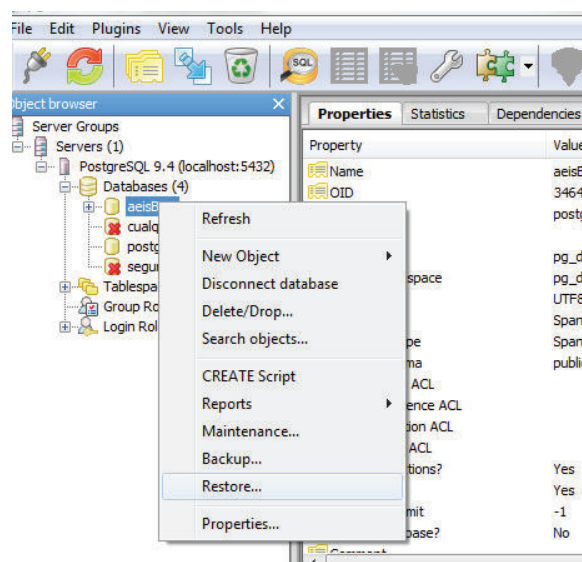
Figura 3.2 Creación de la Base de Datos – Parte II



Elaborado por: Los autores

- Dar clic derecho en la base de datos recién creada y seleccionar la opción “Restore”

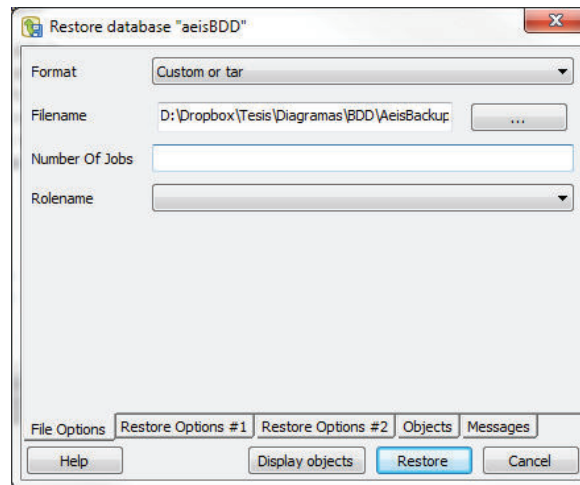
Figura 3.3 Restauración de la Base de Datos – Parte I



Elaborado por: Los autores

- Seleccionar el archivo AeisBackupV3 de la carpeta de instaladores base de datos. Y dar clic en “Restore”

Figura 3.4 Restauración de la Base de Datos – Parte II



3.2.2.1.2 Instalación del Servidor

- Abrir la carpeta “Instaladores Backend”
- Ejecutar el script “Setup Glassfish”. Este script se encargará de registrar los comandos del servidor Glassfish en las variables de entorno del sistema.

Figura 3.5 Ejecución del Script Setup Glassfish

```

C:\Windows\system32\cmd.exe
C:\Users\Administrador\Documents\Instaladores AEIS WEB APP\Instaladores Backend>
ECHO OFF
Instalando comandos glassfish en variable de entorno del sistema

CORRECTO: se guardó el valor especificado.
Instalacion de variables realizada con exito
Presione una tecla para continuar . . . _
  
```

Elaborado por: Los autores

- A continuación ejecutar el script “Setup Backend”. Este script iniciará el servidor de aplicaciones y desplegará la aplicación.

Nota: Esta ventana quedará abierta mientras el servidor esté ejecutándose. En caso de cerrarla el servidor también se detendrá.

Figura 3.6 Ejecución del Script Setup Backend

```

C:\Windows\system32\cmd.exe
C:\Users\Administrador\Documents\Instaladores AEIS WEB APP\Instaladores Backend>
ECHO OFF
Instalacion de variables realizada con exito
Iniciando Servidor Glassfish y desplegando aplicacion.
Para verificar el despliegue dirijase al la direccion: http://localhost:8080/TesisBackend
Para detener este servidor cierre la ventana o presione Ctrl+C
Presione una tecla para continuar . . .

```

Elaborado por: Los autores

- Ingresar a la dirección <http://localhost:8080/TesisBackend> y verificar que la aplicación se haya desplegado con éxito.

Figura 3.7 Despliegue del Servidor de Backend



¡La aplicación AEIS WEB APP Backend se ha iniciado correctamente!

Elaborado por: Los autores

Como iniciar el servidor una vez que se ha instalado Glassfish:

- En caso de que se necesite reiniciar el computador o se ha detenido el servidor por error solo basta volver a iniciarlo ejecutando el script “Setup Backend”.

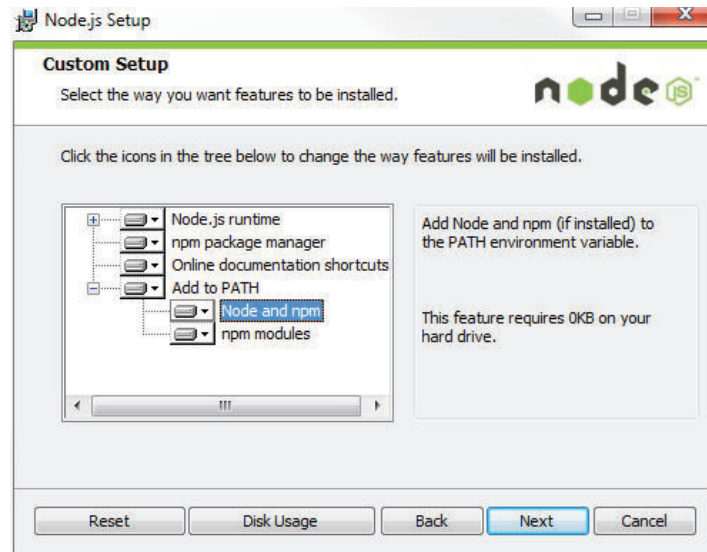
3.2.2.2 Instalación Frontend

3.2.2.2.1 Instalación Node.js

- Proceder a instalar el entorno de ejecución Node.js. Seleccionar la versión compatible con la arquitectura del computador.

- Una vez iniciada la instalación escoger todos los parámetros por defecto y dar clic en siguiente.

Figura 3.8 Instalación de Node.js

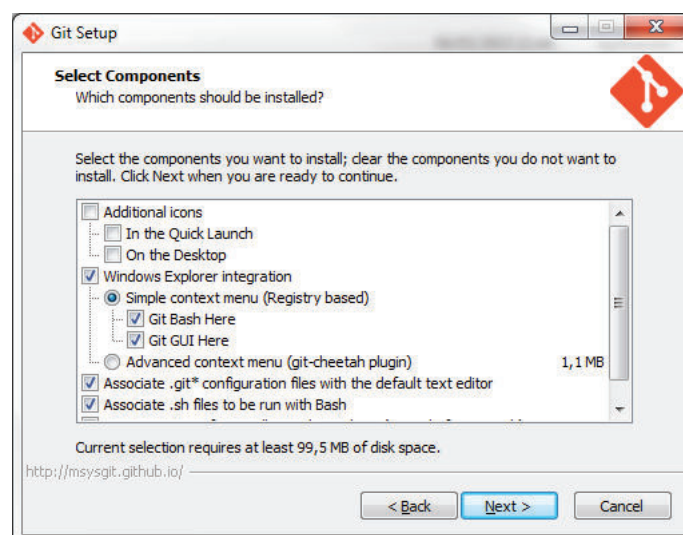


Elaborado por: Los autores

3.2.2.2.2 Instalación GIT

- Iniciar el archivo setup Git-1.9.5-preview20141217
- En la pantalla de selección de componentes seleccionar todo lo que viene por defecto.

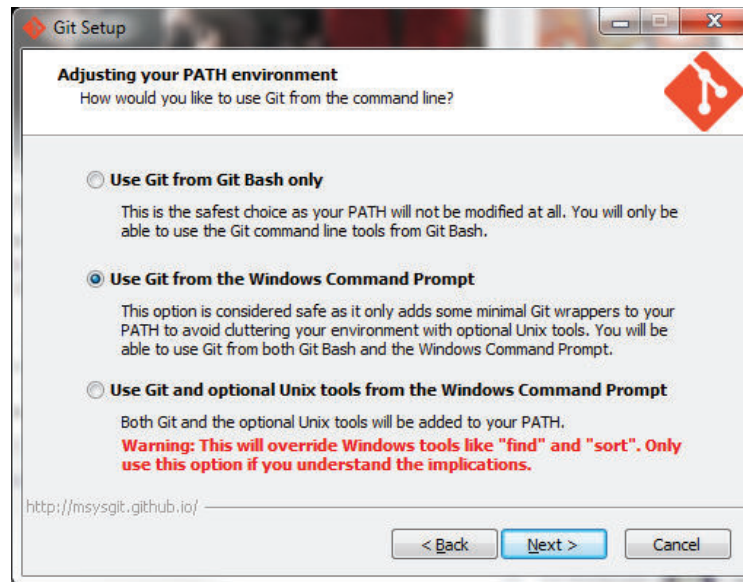
Figura 3.9 Instalación de GIT – Parte I



Elaborado por: Los autores

- En la siguiente pantalla seleccionar la opción “Use GIT from the Windows Command Prompt”. Esta opción sirve para registrar los comandos GIT en las variables de entorno del sistema.

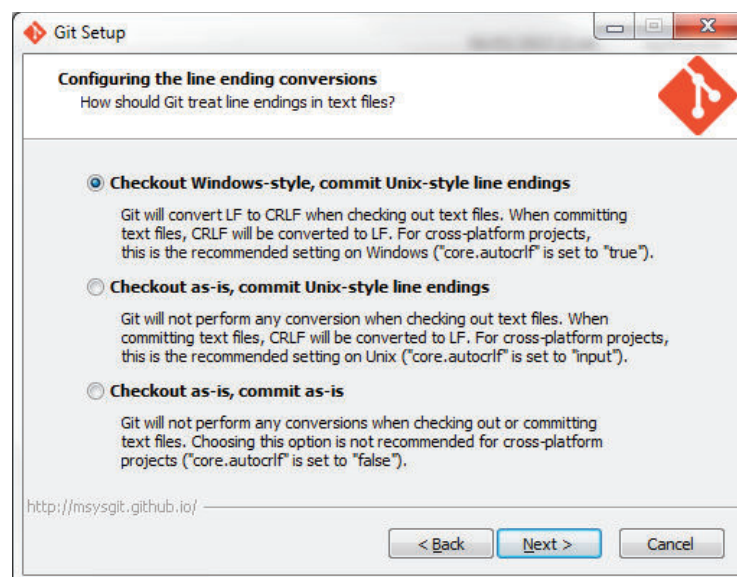
Figura 3.10 Instalación de GIT – Parte II



Elaborado por: Los autores

- En la siguiente pantalla seleccionar la primera opción para que GIT automáticamente transforme los textos de formato Windows a Unix y viceversa.

Figura 3.11 Instalación de GIT – Parte III



Elaborado por: Los autores

- Seleccionar siguiente para terminar la instalación.
- A continuación dar clic derecho en el archivo “Setup Frontend” incluido dentro de la carpeta AEIS WEB APP y seleccionamos ejecutar como administrador.

Ingresamos a la aplicación en la dirección <http://localhost:9000>

3.3 PUESTA EN PRODUCCIÓN DEL PORTAL

Luego de desplegar el portal web AEIS WEB APP y con la carga de datos respectiva, se procedió a evaluar la calidad del producto software.

3.3.1 PRUEBAS DE COMPATIBILIDAD

Se accedió al portal web desde 3 exploradores: Chrome versión 44.0, Internet Explorer versión 11.0, Firefox versión 40.0 con el fin de probar: la apariencia, navegabilidad y tiempo de respuesta.

La Figura 3.12 muestra la página de inicio del portal en el navegador: Chrome.

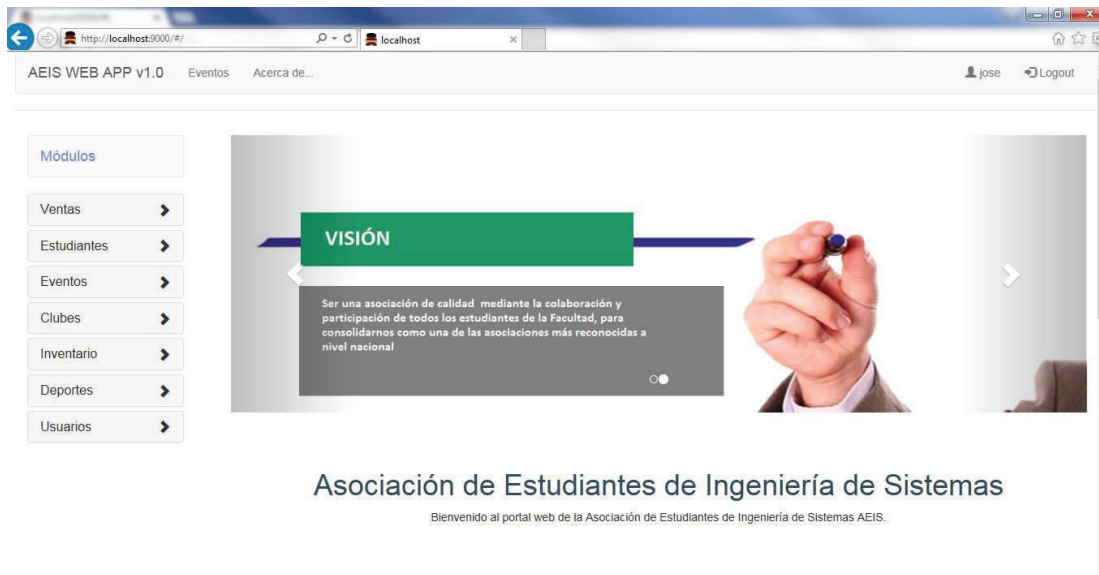
Figura 3.12 Página de Inicio del Portal en el navegador Chrome



Elaborado por: Los autores

La Figura 3.13 muestra la página de inicio del portal en el navegador: Internet Explorer

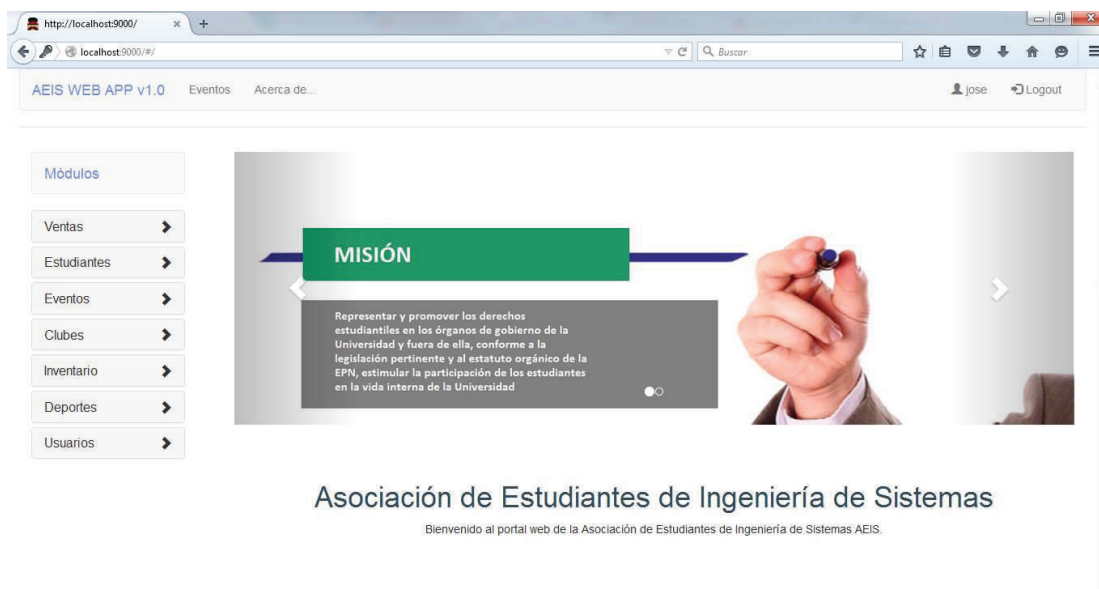
Figura 3.13 Página de Inicio del Portal en el navegador Internet Explorer



Elaborado por: Los autores

La Figura 3.14 muestra la página de inicio del portal en el navegador: Firefox

Figura 3.14 Página de Inicio del Portal en el navegador Firefox



Elaborado por: Los autores

3.3.2 ACCESO A LAS PÁGINAS

Se realizaron mediciones de carga (tiempo y tamaño) de las diferentes páginas usando la herramienta: Network incluida en el navegador Chrome. La Tabla 3.8 presenta las mediciones de las páginas más representativas.

Tabla 3.9 Medición de carga de las páginas del portal

Página	Tiempo (ms)	Tamaño (KB)
Registrar Estudiante	21	14,4
Registrar Pago de aporte	18	6,3
Registrar ventas	20	2,7
Matricular estudiante en un Club	36	7,2
Registrar aprobación en club	18	5,8
Publicar la información de un evento	25	5,4
Registrar la inasistencia a un evento	6	6,4
Inscripción de Equipo Deportivo	17	5,8
Inscripción de Jugador	33	9,3
Inscripción de Jugador en Disciplinas deportivas	18	6,6
Registrar el préstamo o devolución de un dispositivo	24	4,8
Registrar el alquiler del casillero.	17	4,0
Login	6	2.8

Elaborado por: Los autores

Se puede observar que los tiempos de respuesta son bastante pequeños a la vez que la carga total de peticiones que se transfiere entre el cliente y el servidor. Esto demuestra la eficacia de una aplicación construida totalmente sobre JavaScript corriendo sobre un servidor escrito en el mismo lenguaje. Los tiempos de respuesta se minimizan mejorando la navegabilidad y la experiencia del usuario con la aplicación.

3.4 ANÁLISIS DE RESULTADOS

3.4.1 RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN

La Tabla 3.10 muestra los resultados de las pruebas que se realizaron en cada Sprint Review (ver sección 2.4 del Capítulo II). De manera general se concluye que el Product Owner está satisfecho con el funcionamiento del portal web.

Tabla 3.10 Resultados Pruebas de aceptación

Orden	Historia de Usuario	Resultado Obtenido
1	Registrar datos personales del estudiante	Éxito
2	Registrar el pago del aporte	Éxito
3	Registrar ventas	Éxito
4	Matricular estudiante en un Club	Éxito
5	Registrar la aprobación del estudiante	Éxito
6	Publicar la información de un evento	Éxito
7	Registrar la inasistencia a un evento	Éxito

8	Inscripción de Equipo Deportivo	Éxito
9	Registrar el préstamo o devolución de un dispositivo	Éxito
10	Registrar el alquiler del casillero.	Éxito

Elaborado por: Los autores

Después de que el portal fue puesto en producción, los usuarios finales ejecutaron pruebas de aceptación, a continuación un detalle de las pruebas más relevantes:

Tabla 3.11 Prueba Registrar Estudiante

Prueba de Aceptación # 1	
Usuario	Presidente AEIS
Historia de Usuario	Registrar Estudiante
Entradas	Cedula: 1721783213 Nombres: José Luis Apellidos: Escobar Cárdenas Número Único: 201010237 Teléfono: 2635498 Fecha Nacimiento: 31/03/1991 Dirección: Punta Arenas S13.89 y Calle A Correo: jlec1991@hotmail.com Semestre Referencial: Egresado Género: M
Salidas	<ul style="list-style-type: none"> - Todos los campos son obligatorios en caso de no escribirse algún campo se le indicará al usuario que el campo es obligatorio. - El botón: "Registrar Estudiante" no debe activarse hasta que todos los campos obligatorios se encuentren llenos. - En caso de ingresar un número de cédula inválido se mostrará un mensaje de error al usuario. - En caso de ingresar un correo que no se encuentre bajo el formato correcto se mostrará un mensaje de error al usuario. - En caso de no ingresar un número telefónico de longitud correcta se mostrará un mensaje de error al usuario. - Al registrar el estudiante con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo Estudiantes hacer clic en Registrar Estudiante. 2. Registrar los datos del estudiante. 3. Clic en el botón Registrar Estudiante
Resultados Esperados	Se registra el estudiante en la base de datos.
Resultados Obtenidos	Satisfactorios

Casos de excepción	<ul style="list-style-type: none"> - Validación número de cédula. - Validación de campos vacíos. - Validación de número telefónico. - Validación de correo electrónico
---------------------------	--

Elaborado por: Los autores

Tabla 3.12 Prueba Registrar Pago Aporte - Afiliación

Prueba de Aceptación # 2	
Usuario	Vicepresidente Financiero
Historia de Usuario	Registrar el pago del aporte
Entradas	Buscar (Cedula): 1721783213
Salidas	<ul style="list-style-type: none"> - Datos del estudiante que tiene el número de cédula: 1721783213. - Lista de valores a pagar de acuerdo al semestre. - El botón: “Registrar Afiliación” no debe activarse hasta seleccionar los valores del aporte que el estudiante va a pagar. - El botón: “Registrar Desafiliación” no debe activarse hasta ingresar una cédula de un estudiante que se encuentre registrado en el sistema. - En caso de ingresar un número de cédula inválido se mostrará un mensaje de error al usuario. - Al registrar el pago del aporte con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo Estudiante, hacer clic en Registrar pago de Aporte. 2. Ingresar un número de cédula. 3. Clic en el botón Buscar 4. Seleccionar el semestre. 5. Clic en el botón Buscar Aportes. 6. Seleccionar los rubros que el estudiante va a cancelar. 7. Clic en el botón Registrar Afiliación.
Resultados Esperados	Se registra la afiliación del estudiante.
Resultados Obtenidos	Satisfactorios
Casos de excepción	<ul style="list-style-type: none"> - Validación número de cédula.

Elaborado por: Los autores

Tabla 3.13 Prueba Registrar Pago Aporte - Desafiliación

Prueba de Aceptación # 3	
Usuario	Tesorero AEIS
Historia de Usuario	Registrar el pago del aporte
Entradas	Buscar (Cedula): 1724375892
Salidas	<ul style="list-style-type: none"> - Datos del estudiante que tiene el número de cédula: 172.4375892. - Lista de valores a pagar de acuerdo al semestre. - El botón: “Registrar Afiliación” no debe activarse hasta seleccionar los valores del aporte que el estudiante va a pagar. - El botón: “Registrar Desafiliación” no debe activarse hasta ingresar una cédula de un estudiante que se encuentre registrado en el sistema. - En caso de ingresar un número de cédula inválido se mostrará un mensaje de error al usuario. - Al registrar el pago del aporte con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo Estudiante, hacer clic en Registrar pago de Aporte. 2. Ingresar un número de cédula. 3. Clic en el botón Buscar 4. Clic en el botón Registrar Desafiliación.
Resultados Esperados	Se registra la desafiliación del estudiante.
Resultados Obtenidos	Satisfactorios
Casos de excepción	- Validación número de cédula.

Elaborado por: Los autores

Tabla 3.14 Prueba Registrar Ventas

Prueba de Aceptación # 4	
Usuario	Empleado
Historia de Usuario	Registrar ventas
Entradas	- Cantidad del producto a vender.
Salidas	<ul style="list-style-type: none"> - Precio total a pagar por la venta. - El botón: “Registrar Venta” no debe activarse hasta seleccionar al menos un producto. - Al registrar la venta con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo Ventas, hacer clic en Registrar venta. 2. Ingresar el nombre del producto 3. Clic en el botón Vender 4. Clic en el botón Registrar Venta
Resultados Esperados	Se registra la venta.
Resultados Obtenidos	Satisfactorios
Casos de excepción	Ninguno

Elaborado por: Los autores

Tabla 3.15 Prueba Matricular Estudiante en Club

Prueba de Aceptación # 5	
Usuario	Vicepresidente Académico
Historia de Usuario	Matricular Estudiante en Club
Entradas	- Buscar (Cedula): 1724375892
Salidas	<ul style="list-style-type: none"> - Datos del estudiante que tiene el número de cédula: 1724375892. - Lista de clubes disponibles de acuerdo al semestre. - El botón: “Matricular Estudiante” no debe activarse hasta seleccionar un club. - En caso de ingresar un número de cédula inválido se mostrará un mensaje de error al usuario. - Al matricular al estudiante en un club con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo: Clubes opción Matricular Estudiante en Club 2. Ingresar la cédula del estudiante. 3. Clic en el botón Buscar Estudiante. 4. Seleccionar el semestre. 5. Seleccionar el club en el que se va a matricular al estudiante. 6. Clic en el botón Matricularse. 7. Clic en el botón Finalizar matrícula.
Resultados Esperados	Se registra la matrícula del estudiante en el club seleccionado.
Resultados Obtenidos	Satisfactorios
Casos de excepción	- Validación número de cédula.

Elaborado por: Los autores

Tabla 3.16 Prueba Registrar Aprobación del estudiante

Prueba de Aceptación # 6	
Usuario	Profesor
Historia de Usuario	Registrar aprobación de estudiante en club
Entradas	- Buscar (Cedula): 1724375892
Salidas	<ul style="list-style-type: none"> - Datos del estudiante que tiene el número de cédula: 1724375892. - El botón: “Registrar Aprobación” no debe activarse hasta seleccionar un estado de aprobación. - En caso de ingresar un número de cédula inválido se mostrará un mensaje de error al usuario. - Al registrar el estado de aprobación del estudiante con éxito se mostrará una notificación al usuario.

Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo: Clubes opción Registrar aprobación en Club 2. Ingresar la cédula del estudiante. 3. Clic en el botón Buscar Estudiante. 4. Seleccionar el estado de aprobación. 5. Clic en el botón Registrar Aprobación.
Resultados Esperados	Se registra el estado de aprobación del estudiante en el club seleccionado.
Resultados Obtenidos	Satisfactorios
Casos de excepción	- Validación número de cédula.

Elaborado por: Los autores

Tabla 3.17 Prueba Publicar Información Evento

Prueba de Aceptación # 7	
Usuario	Profesor
Historia de Usuario	Registrar aprobación de estudiante en club
Entradas	<ul style="list-style-type: none"> - Nombre del evento: Bautizo - Descripción del Evento: Bautizo de Sistemas, en el complejo de médicos. - Fecha Evento: 19/08/2015 - Obligatoriedad: Obligatorio
Salidas	<ul style="list-style-type: none"> - Todos los campos son obligatorios en caso de no escribirse algún campo se le indicará al usuario que el campo es obligatorio. - El botón: "Registrar Evento" no debe activarse hasta que todos los campos obligatorios se encuentren llenos. - Al registrar el evento con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo: Eventos opción Publicar información de evento. 2. Ingresar la información solicitada en la interfaz. 3. Clic en el botón Registrar Evento.
Resultados Esperados	Se registra el evento.
Resultados Obtenidos	Satisfactorios
Casos de excepción	Ninguno

Elaborado por: Los autores

Tabla 3.18 Prueba Inscripción Equipo Deportivo

Prueba de Aceptación # 8	
Usuario	Vicepresidente de Deportes
Historia de Usuario	Inscripción Equipo Deportivo
Entradas	<ul style="list-style-type: none"> - Nombre equipo: Séptimo de Básica - Semestre: 2015-A

Salidas	<ul style="list-style-type: none"> - Todos los campos son obligatorios en caso de no escribirse algún campo se le indicará al usuario que el campo es obligatorio. - El botón: “Registrar Equipo” no debe activarse hasta que todos los campos obligatorios se encuentren llenos. - Al registrar el equipo con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo: Eventos opción Deportes opción Registrar equipo deportivo. 2. Ingresar la información solicitada en la interfaz. 3. Clic en el botón Registrar Equipo.
Resultados Esperados	Se registra el equipo.
Resultados Obtenidos	Satisfactorios
Casos de excepción	Ninguno

Elaborado por: Los autores

Tabla 3.19 Prueba Inscripción Equipo Deportivo

Prueba de Aceptación # 9	
Usuario	Vicepresidente de Deportes
Historia de Usuario	Inscripción Equipo Deportivo
Entradas	<ul style="list-style-type: none"> - Semestre: 2015-A - Equipos Disponibles: Quitapenas - Nombre Jugador: Marcela - Apellido Jugador: Mosquera - Tipo Jugador: Afiliado - Número Jugador: 20 - Género Jugador F
Salidas	<ul style="list-style-type: none"> - Todos los campos son obligatorios en caso de no escribirse algún campo se le indicará al usuario que el campo es obligatorio. - El botón: “Registrar Jugador” no debe activarse hasta que todos los campos obligatorios se encuentren llenos. - Al registrar el jugador con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo: Eventos opción Deportes opción Registrar jugador en equipo. 2. Ingresar la información solicitada en la interfaz. 3. Clic en el botón Registrar Jugador.
Resultados Esperados	Se registra el jugador en el equipo.
Resultados Obtenidos	Satisfactorios
Casos de excepción	Ninguno

Elaborado por: Los autores

Tabla 3.20 Prueba Inscripción Equipo Deportivo

Prueba de Aceptación # 10	
Usuario	Vicepresidente de Deportes
Historia de Usuario	Inscripción Equipo Deportivo
Entradas	<ul style="list-style-type: none"> - Semestre: 2015-A - Equipos Disponibles: Quitapenas - Jugador: Marcela Mosquera - Seleccione Categoría: Femenino - Seleccione Disciplina: Ping Pong
Salidas	<ul style="list-style-type: none"> - Todos los campos son obligatorios en caso de no escribirse algún campo se le indicará al usuario que el campo es obligatorio. - El botón: “Registrar Jugador en Disciplina” no debe activarse hasta que todos los campos obligatorios se encuentren llenos. - Al registrar el jugador en una disciplina con éxito se mostrará una notificación al usuario.
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el módulo: Eventos opción Deportes opción Registrar Jugador en disciplina deportiva. 2. Ingresar la información solicitada en la interfaz. 3. Clic en el botón registrar jugador en disciplina.
Resultados Esperados	Se registra el jugador en las disciplinas deportivas.
Resultados Obtenidos	Satisfactorios
Casos de excepción	Ninguno

Elaborado por: Los autores

Luego de que los usuarios finales probaran el portal web se concluye que están satisfechos con el funcionamiento del mismo.

3.4.2 EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE

Se realizó una encuesta a 6 usuarios finales del portal AEIS WEB APP, los mismos que realizaron las pruebas de aceptación descritas en la Sección 3.4.1, estos usuarios tienen los roles de: empleado, vicepresidente académico, vicepresidente financiero, vicepresidente de cultura, vicepresidente de deportes y profesor.

La encuesta consistió en 12 preguntas que se enfocaron en obtener información sobre cómo perciben los usuarios las características de: funcionalidad, usabilidad, apariencia del portal y el grado de satisfacción que tiene luego de usar el portal AEIS WEB APP. En la Tabla 3.21 se presentan las preguntas usadas en la encuesta y la característica que evalúa cada pregunta.

Tabla 3.21 Encuesta de Satisfacción para Usuarios Finales

Pregunta	Característica que evalúa
1. ¿El portal web proporciona las funciones que usted necesita?	Funcionalidad
2. ¿El portal web maneja la información de forma segura y protegida, y es capaz de protegerse de accesos no autorizados?	Funcionalidad
3. Los títulos del menú son claros y facilitan el acceso a los contenidos.	Usabilidad
4. ¿El portal web muestra de forma clara todas las actividades que puede realizar en él?	Usabilidad
5. ¿El portal web muestra ayuda contextual sobre cómo llenar formularios?	Usabilidad
6. ¿Considera que el portal web es fácil de usar?	Usabilidad
7. Los distintos elementos de la página se muestran bien.	Apariencia del Portal
8. El tamaño de la fuente es adecuado.	Apariencia del Portal
9. Los colores y gráficos son adecuados.	Apariencia del Portal
10. La visualización del portal web es buena.	Apariencia del Portal
11. El diseño del portal web es atractivo.	Apariencia del Portal
12. En general ¿Indique su satisfacción con el portal web?	Satisfacción del Usuario

Elaborado por: Los autores

A continuación se presentan los resultados obtenidos.

3.4.2.1 Funcionalidad

Tanto en la pregunta 1 como en la 2, se obtuvo un resultado de 100% de aceptación en las características funcionales.

1. ¿El portal web proporciona las funciones que usted necesita?

Figura 3.15 Resultados Primera Pregunta



Elaborado por: Los autores

2. ¿El portal web maneja la información de forma segura y protegida, y es capaz de protegerse de accesos no autorizados?

Figura 3.16 Resultados Segunda Pregunta



Elaborado por: Los autores

3.4.2.2 Usabilidad

En la pregunta 3 se obtuvo que 100% de los usuarios finales consideran que el menú es claro y facilita el acceso a los contenidos.

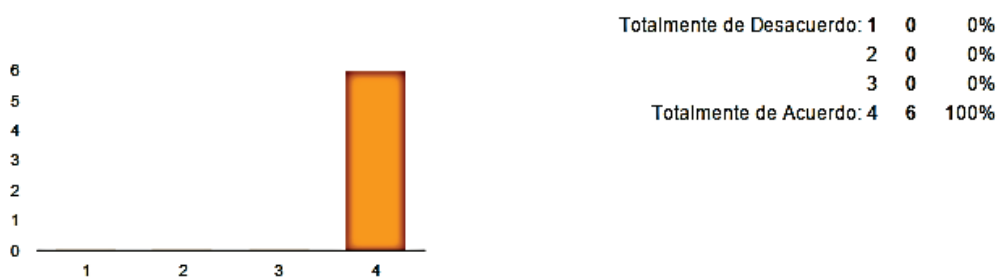
En la pregunta 4 se obtuvo que el 100% de los usuarios finales indican que el portal web muestra de forma clara todas las actividades que pueden realizar en él.

En la pregunta 5 se obtuvo que el 100% indican que se muestra la ayuda conceptual para llenar formularios.

En la pregunta 6 el 100% de usuarios encuestados indican que consideran que el portal es fácil de usar.

- Los títulos del menú son claros y facilitan el acceso a los contenidos.

Figura 3.17 Resultados Tercera Pregunta



Elaborado por: Los autores

- ¿El portal web muestra de forma clara todas las actividades que puede realizar en él?

Figura 3.18 Resultados Cuarta Pregunta



Si	6	100%
No	0	0%

Elaborado por: Los autores

5. ¿El portal web muestra ayuda contextual sobre cómo llenar formularios?

Figura 3.19 Resultados Quinta Pregunta



Si	6	100%
No	0	0%

Elaborado por: Los autores

6. ¿Considera que el portal web es fácil de usar?

Figura 3.20 Resultados Sexta Pregunta



Si	6	100%
No	0	0%

Elaborado por: Los autores

3.4.2.3 Apariencia del Portal

En la pregunta 7 obtuvo que 100% de los usuarios finales indican que todos los elementos de la página se muestran bien.

En la pregunta 8 obtuvo que el 83.3% de los usuarios finales están totalmente de acuerdo con el tamaño de fuente usado en el portal, mientras que el 16.7% está parcialmente de acuerdo con el tamaño de fuente usado.

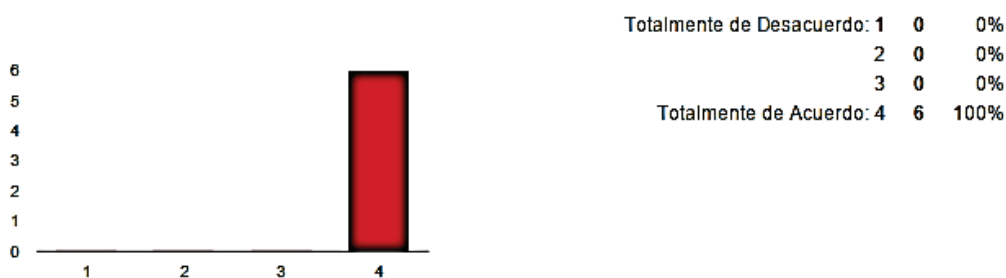
En la pregunta 9 obtuvo que el 66.7% de los usuarios finales están totalmente de acuerdo con los colores y gráficos usados en el portal, mientras que el 33.3% está parcialmente de acuerdo con los colores del portal.

En la pregunta 10 obtuvo que el 83.3% de los usuarios finales están totalmente de acuerdo con que la visualización del portal es buena, mientras que el 16.7% está parcialmente de acuerdo con la visualización del portal.

En la pregunta 11 obtuvo que 100% de los usuarios finales indican que el diseño del portal web es atractivo.

7. Los distintos elementos de la página se muestran bien.

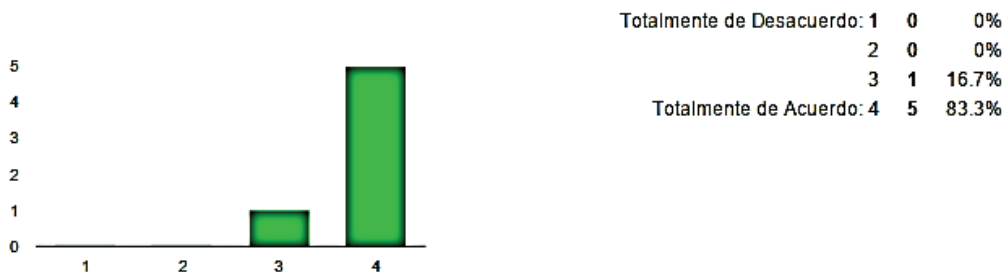
Figura 3.21 Resultados Séptima Pregunta



Elaborado por: Los autores

8. El tamaño de la fuente es adecuado.

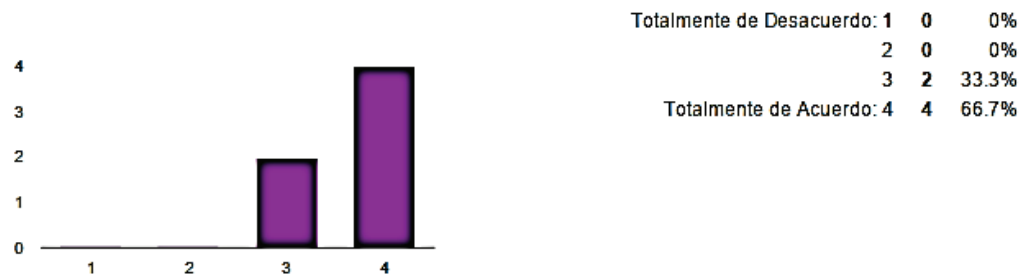
Figura 3.22 Resultados Octava Pregunta



Elaborado por: Los autores

9. Los colores y gráficos son adecuados.

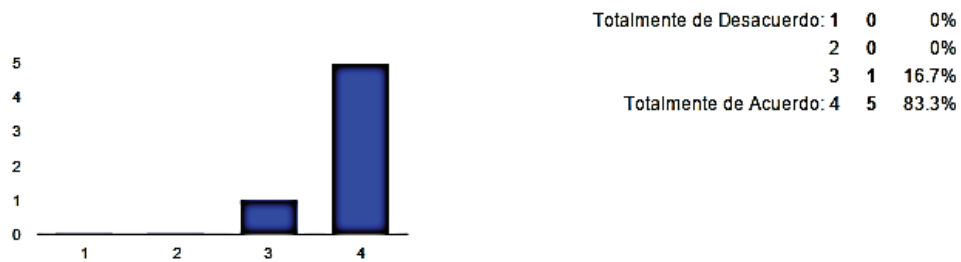
Figura 3.23 Resultados Novena Pregunta



Elaborado por: Los autores

10. La visualización del portal web es buena.

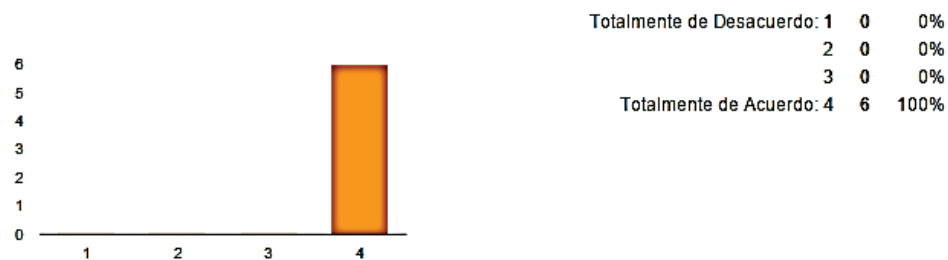
Figura 3.24 Resultados Décima Pregunta



Elaborado por: Los autores

11. El diseño del portal web es atractivo.

Figura 3.25 Resultados Undécima Pregunta



Elaborado por: Los autores

3.4.2.4 Satisfacción del Usuario

En la pregunta 12 obtuvo que 100% de los usuarios finales indican que se encuentran satisfechos con el portal web.

12. En general ¿Indique su satisfacción con el portal web?

Figura 3.26 Resultados Duodécima Pregunta



Elaborado por: Los autores

Luego de aplicar la encuesta se puede concluir que los usuarios finales se encuentran satisfechos con la funcionalidad del portal AEIS WEB APP, consideran que tiene un diseño atractivo con un menú que facilita el acceso a los contenidos y que muestra de manera clara todas las actividades que se pueden realizar en él además de proporcionar la ayuda suficiente para llenar los formularios.

4 CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

El objetivo del proyecto de titulación fue desarrollar e implementar un Portal Web para la Gestión de la Asociación de Estudiantes de Ingeniería en Sistemas, al finalizar el trabajo se presenta el portal AEIS WEB APP, que fue desplegado en el servidor del Laboratorio de Informática y Ciencias de la Computación (LDICC).

De la ejecución del proyecto se obtuvieron las siguientes conclusiones y recomendaciones:

4.1 CONCLUSIONES

- Se analizó el entorno de la Asociación de Estudiantes de Ingeniería en Sistemas para determinar los procesos que realiza, obteniendo el modelo de procesos determinado en el Capítulo I en la Figura 1.2. Del modelo de procesos se determinó que son importantes y prioritarios los procesos de Afiliación de Estudiantes, Matriculación de Estudiantes en Clubes y Venta de Snacks y Bebidas, seguido por los procesos de: Préstamo de Dispositivos, Alquiler de Casilleros, Organización de Eventos Sociales e Inscripción en el Campeonato Interno de Deportes.
- Luego de desarrollar el proyecto siguiendo la metodología Scrum podemos resaltar lo siguiente: el uso de historias de usuario permitió comprender de mejor manera los procesos que posteriormente tenían que implementarse, estas historias priorizadas pasaron a formar el Product Backlog, en cada Sprint se desglosó cada historias de usuario en tareas específicas a las que se les asignó una estimación expresada en Story Points, estas tareas fueron implementadas por el equipo de desarrollo y presentadas en cada Sprint Review el Product Owner.
- El uso de la metodología SCRUM permitió obtener incrementos en el producto de software en tiempos cortos, de esta manera el cliente y el equipo de desarrollo percibieron el avance en el producto durante el proceso de desarrollo.

- La participación del cliente fue de gran importancia para despejar dudas e inquietudes sobre los requisitos y obtener una retroalimentación para mejorar en el proceso de desarrollo del portal web.
- Se logró la automatización de todos los procesos descritos durante el análisis realizado en el Capítulo I de acuerdo a las características solicitadas por el cliente.
- En cada iteración se realizaron pruebas de software lo que permite asegurar que el Portal Web funciona correctamente y cumple todos los requisitos del cliente.
- Los resultados obtenidos en el Capítulo III indican que existe satisfacción de los usuarios finales pues perciben que el portal es intuitivo de usar y cumple con los requisitos para los que fue creado.
- Se podría usar una variación de la metodología aplicada durante el desarrollo para modificar los requisitos y poder adaptar la aplicación para su uso no solamente dentro de la AEIS sino también dentro de otras asociaciones dentro de la Escuela Politécnica Nacional e incluso fuera de ella.

4.2 RECOMENDACIONES

- Es importante entender los procesos del negocio que se van a automatizar ya que con esto es más fácil establecer los requerimientos.
- Se recomienda el uso de la metodología SCRUM debido a que se pueden entregar resultados en tiempos cortos, las reuniones diarias permiten visualizar el avance y las dificultades que tiene el equipo de desarrollo y en el Sprint Review se recibe la retroalimentación del cliente.

- Se recomienda la continua participación del cliente en el desarrollo de Software para poder obtener retroalimentación sobre la calidad del producto de software en cada iteración.
- Se recomienda seguir buenos estándares de seguridad y diseño de páginas web para que la misma pueda ser accesible desde distintos medios tanto desde PC, Laptops y dispositivos móviles.
- Se recomienda cumplir a cabalidad con los tiempos establecidos para cada sprint con el fin de no alargar el proceso de desarrollo del producto de software respetando siempre el orden de prioridad de cada funcionalidad establecido durante el proceso de análisis.
- Es recomendable llevar un reporte de incidencias (bugs) que puedan ocurrir durante el desarrollo de la aplicación, ya que esto permite tener un historial de todos los errores que se deben corregir al finalizar el Sprint.
- Una versión posterior del portal AEIS WEB APP puede incluir la organización total del campeonato interno de deportes.

BIBLIOGRAFÍA

- [1] Asociación de Estudiantes de Ingeniería de Sistemas, «Estatuto de la Asociación de Estudiantes de Ingeniería de Sistemas (AEIS),» de *Reunión de Conformación de la AEIS*, Quito, 2009.
- [2] Palacios y Juan, *Flexibilidad con Scrum: Principios de Diseño e Implantación de Campos de Scrum*, 2da ed., 2008.
- [3] Isla Visual, «Isla Visual,» 11 2012. [En línea]. Disponible: http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php. [Último acceso: Diciembre 2014].
- [4] K. Schwaver y J. Sutherland, «Scrum Guides,» Julio 2013. [En línea]. Disponible: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>. [Último acceso: Octubre 2014].
- [5] X. Quesada Allue, A. Cymment y M. Alaimo , «Scrum Alliance,» Marzo 2013. [En línea]. Disponible: <https://www.scrumalliance.org/scrum/media/ScrumAllianceMedia/Files%20and%20PDFs/Why%20Scrum/Core%20Scrum%20Translations/Core-Scrum-Spanish.pdf>. [Último acceso: Octubre 2014].
- [6] O. E. Guzmán Matus. [En línea]. Disponible: <https://sites.google.com/site/oeguzman/8artefactosdescrum>. [Último acceso: Octubre 2014].
- [7] V. Nevado Cabello, *Introducción a las Bases de Datos Relacionales*, Madrid: Vision Libros, 2010.
- [8] R. Martínez, «PostgreSQL-es,» 10 2010. [En línea]. Disponible: http://www.postgresql.org.es/sobre_postgresql. [Último acceso: Diciembre 2014].
- [9] «Wikipedia,» [En línea]. Disponible: http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n.
- [10] Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, «Experto Java,» 2004. [En línea]. Disponible:

<http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/sesion1-apuntes.htm>. [Último acceso: Diciembre 2014].

- [11] Google. Inc, «AngularJS Developers Guide,» 2015. [En línea]. Disponible: <https://docs.angularjs.org/guide/introduction>. [Último acceso: Agosto 2015].
- [12] NodeJS Foundation, «NodeJs,» 2015. [En línea]. Disponible: <https://nodejs.org/>. [Último acceso: Agosto 2015].

ANEXOS

A continuación se lista los archivos que se encuentran en el CD:

Anexo A: Manual de Instalación

Anexo B: Manual de Usuario

Anexo C: Backup de la base de datos: aeisBDD

Anexo D: Scripts para instalación de Backend y Frontend