

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**DESARROLLO DE UN SISTEMA DE CONTROL ESCOLAR
PARA LA ESCUELA BILINGÜE SANGAY COMBINANDO
LAS METODOLOGÍAS SCRUM Y XTREME
PROGRAMMING**

Previa a la obtención del Título de:

INGENIERÍA EN SISTEMAS E INFORMÁTICA

POR:

**ANA LUCÍA CHÁVEZ HIDALGO
JOSÉ HERNÁN TENORIO CHICAIZA**

Sangolquí, julio de 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por la Srta. Ana Lucía Chávez Hidalgo y el Sr. José Hernán Tenorio Chicaiza, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Julio de 2012

**ING. CÉSAR VILLACÍS
DIRECTOR**

AGRADECIMIENTO

Agradezco a Dios por darme la sabiduría, la fortaleza para seguir día a día por el sendero de la luz.

A mis padres les expreso mis agradecimientos por toda la confianza depositada en mí y todas las horas de enseñanza, amor y cariño que supieron otorgarme para mi formación.

A mis hermanos por apoyarme en cada etapa de mi vida.

A mis profesores, quienes han impartido sus enseñanzas y conocimientos a los largo de mi carrera.

A mis amigos y compañeros con quienes compartimos gratos momentos, penas y alegrías a lo largo del tiempo de estudios.

ANA LUCÍA CHÁVEZ HIDALGO

AGRADECIMIENTO

A Dios por ser mi guía y mi luz por el camino de la vida.

A mis padres, hermanos y sobrinos, quienes con su amor, apoyo incondicional y paciencia, me dieron un enorme impulso e inspiración.

A mis amigos de la Universidad por su continuo y afectuoso aliento y por todos los momentos agradables que compartimos durante nuestro paso por la universidad.

A Santiago Pazmiño, que con sus consejos y aliento desinteresado, me ayudaron e incentivaron para cumplir esta meta.

JOSÉ HERNÁN TENORIO CHICAIZA

DEDICATORIA

A Dios, a mis padres Guillermo y Lolita, quienes me brindaron desinteresadamente la oportunidad para que pueda adquirir esta noble profesión.

A mis hermanos Fabrizio, Paulina, y Marcelo por apoyarme siempre y brindarme la ayuda que he necesitado.

A mis sobrinas Paula y Sofía que son la alegría de la casa.

ANA LUCÍA CHÁVEZ HIDALGO

DEDICATORIA

Dedico este proyecto de tesis a mi Dios por estar conmigo en cada paso de mi vida, cuidándome y dándome fortaleza para continuar.

A mis padres José y Olga quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento.

A mis hermanos Susana, Juan, Fanny y María, por su apoyo constante durante este proceso.

A mis sobrinos Javier, Tania y Nicolás por brindarme su alegría. Es por ellos que soy lo que soy ahora. Los amo con toda mi vida.

JOSÉ HERNÁN TENORIO CHICAIZA

LISTADO DE TABLAS

Tabla 2.1:	Fases de Scrum	12
Tabla 2.2:	Roles y responsabilidades de Scrum	13
Tabla 2.3:	Prácticas y herramientas de Scrum	14
Tabla 2.4:	Fases de Xtreme Programming.....	16
Tabla 2.5:	Roles y responsabilidades de XP	17
Tabla 2.6:	Técnicas y prácticas de XP	19
Tabla 2.7:	Capas de la arquitectura J2EE	25
Tabla 2.8:	Tipos de contenedores	27
Tabla 3.1	Funciones del sistema SICOES	36
Tabla 3.2	Definición de problema general.....	39
Tabla 3.3	Necesidades principales.....	42
Tabla 3.4	Características de los usuarios	44
Tabla 4.1	Pila del Producto Inicial (Product Backlog).....	57
Tabla 4.2	Descripción de las etiquetas HTML	63
Tabla 4.3	Estandarización de código Fuente.....	64
Tabla 4.4	Pila del producto de la Iteración 1	66
Tabla 4.5	Pila de actividades de la Iteración 1	68
Tabla 4.6	Tareas completadas de la Iteración 1	69
Tabla 4.7	Pila del producto de la Iteración 2.....	74
Tabla 4.8	Pila de actividades de la Iteración 2	75
Tabla 4.9	Tabla tareas completadas de la Iteración 2.....	81
Tabla 4.10	Pila del producto de la Iteración 3.....	86
Tabla 4.11	Pila de actividades de la Iteración 3	87
Tabla 4.12	Tareas completadas de la Iteración 3	89
Tabla 4.13	Pila del producto de la Iteración 4.....	95
Tabla 4.14	Pila de requerimientos de la Iteración 4.....	96
Tabla 4.15	Tareas completadas de la Iteración 4	97

Tabla 4.16	Pila del producto de la Iteración 5.....	102
Tabla 4.17	Pila de requerimiento de la Iteración 5.....	103
Tabla 4.18	Tareas completadas de la Iteración 5	106
Tabla 4.19	Pila del producto de la Iteración 6.....	113
Tabla 4.20	Pila de actividades de la Iteración 6	114
Tabla 4.21	Tareas completadas de la Iteración 6	115
Tabla 4.22	Pila del producto final	118
Tabla 5.1	Estándar para el desarrollo de modelo de datos.....	120

LISTADO FIGURAS

Figura 2.1. Proceso Scrum	11
Figura 2.2. Ciclo de Vida XP	15
Figura 2.3. Arquitectura J2EE	24
Figura 3.1. Equipo que conforma el talento humano de SICOES	37
Figura 3.2. Diagrama de la arquitectura de SICOES	40
Figura 3.3. Funciones SICOES	42
Figura 4.1 Gráfico Burn Down Chart de la Iteración 1	70
Figura 4.2 Pruebas de la clase Usuario	71
Figura 4.3 Demo de la Iteración 1	72
Figura 4.4 Pruebas de la clase tipo docente	78
Figura 4.5 Pruebas de la clase tipo baja	79
Figura 4.6 Pruebas de la clase ciclo lectivo	79
Figura 4.7 Gráfico Burn Down Chart de la Iteración 2	80
Figura 4.8 Demo de la Iteración 2	84
Figura 4.9 Gráfico Burn Down Chart de la Iteración 3	91
Figura 4.10 Pruebas de la clase Alumno	92
Figura 4.11 Pruebas de la clase inscripción.	92
Figura 4.12 Pruebas usando de la clase matriculación.	93
Figura 4.13 Demo de la Iteración 3	93
Figura 4.14 Gráfico Burn Down Chart de la Iteración 4	98
Figura 4.15 Pruebas de la clase docente	99
Figura 4.16 Pruebas de la clase asignar asignatura docente.	99
Figura 4.17 Demo de la Iteración 4	100
Figura 4.18 Gráfico Burn Down Chart del la Iteración 5	110
Figura 4.19 Pruebas de la clase horarios	111
Figura 4.20 Demo del la Iteración 5	111
Figura 4.21 Gráfico Burn Down Chart de la Iteración 6	116
Figura 4.22 Demo de la Iteración 6	117
Figura 5.1 Modelo Entidad Relación	128

Figura 5.2	Diagrama de Clase	129
Figura 5.3	Diagrama de arquitectura	130

LISTADO DE ANEXOS

ANEXO A – Manual del Usuario	131
1 OBJETIVO GENERAL	131
2 DESCRIPCIÓN GENERAL DEL SISTEMA.....	131
3 BENEFICIOS	131
4 ESPECIFICACIONES INICIALES.....	132
5 MENÚ DEL SISTEMA	134
ANEXO B – CARTAS.....	163

ÍNDICE DE CONTENIDOS

CAPÍTULO 1.....	3
EL PROBLEMA.....	3
1.1 Tema.....	3
1.2 Definición del problema.....	3
1.2.1 Contextualización del problema.....	3
1.2.2 Formulación del Problema.....	5
1.3 Justificación del problema.....	6
1.4 Objetivos.....	7
1.4.1 Objetivo General.....	7
1.4.2 Objetivos Específicos.....	7
1.5 Alcance del Proyecto.....	8
CAPÍTULO 2.....	10
2.1 Metodología.....	10
2.2 SCRUM.....	10
2.2.1 Procesos.....	11
2.2.2 Roles y Responsabilidades.....	13
2.2.3 Prácticas.....	14
2.3 Xtreme Programming.....	15
2.3.1 Procesos.....	15
2.3.2 Roles y responsabilidades.....	17
2.3.3 Prácticas.....	19
2.4 Combinación de Xtreme Programming y Scrum.....	21
2.5 Métodos y técnicas de SCRUM.....	21
2.6 Métodos y técnicas de Xtreme Programming.....	22
2.6.1 Herramientas Empleadas.....	23
2.6.2 Capas del Sistema.....	29
2.6.3 Herramientas de Desarrollo.....	32
CAPÍTULO 3.....	35
3.1 Introducción.....	35
3.2 Propósito.....	35
3.3 Alcance.....	36
3.4 Limitaciones del sistema.....	37
3.5 Personal involucrado.....	37

3.6	Definiciones.....	37
3.7	Acrónimos	38
3.8	Referencias	38
3.9	Descripción general.....	39
3.9.1	Perspectiva del producto.....	39
3.9.2	Definición general del problema.....	39
3.9.3	Estructura técnica del sistema	40
3.9.4	Funciones del producto.....	41
3.9.5	Necesidades del sistema	42
3.9.6	Características de los usuarios	44
3.9.7	Herramientas de desarrollo.....	46
3.9.8	Especificaciones de requerimientos.....	46
3.9.9	Requisitos comunes de las interfaces.....	48
3.9.10	Requerimientos funcionales	49
3.9.11	Requerimientos no funcionales.....	52
CAPÍTULO 4.....		55
4.1	Planificación del proyecto utilizando la metodología SCRUM	55
4.2	Desarrollo del Sistema	60
4.3	Estándares de Programación.....	62
4.3.1	Análisis y Desarrollo de la Iteración 1	65
4.3.2	Análisis y desarrollo del requerimiento de la Iteración 2	73
4.3.3	Análisis y Desarrollo del requerimiento de la Iteración 3.....	84
4.3.4	Análisis y Desarrollo del requerimiento de la Iteración 4.....	94
4.3.5	Análisis y Desarrollo del requerimiento de la Iteración 5.....	100
4.3.6	Análisis y Desarrollo del requerimiento de la Iteración 6.....	112
4.3.7	Product Backlog Final	118
CAPÍTULO 5.....		119
5.1	Modelo de Datos	119
5.2	Diseño de la base de datos.....	120
5.3	Modelo entidad relación	121
5.4	Diagrama de Clases.....	121
5.5	Diagrama de Arquitectura	121
5.6	Manual de usuario.....	121
CAPÍTULO 6.....		125
6.1	CONCLUSIONES.....	125
6.2	RECOMENDACIONES	126

6.3	BIBLIOGRAFÍA	128
6.3.1	SITIOS DE REFERENCIA EN INTERNET	129

RESUMEN

Los avances tecnológicos que se han generado a través de la informática han causado un gran impacto social en las instituciones educativas; es por ello que muchas de estas, se han adaptado a los cambios y han visto la necesidad de automatizar sus procesos.

El manejo de un sistema de control escolar es importante para los planes de cualquier institución educativa, actualmente representan el medio eficaz para agilizar los procedimientos que en ella se desarrollan, generan mayor productividad en las instituciones que disponen de los beneficios de este tipo de sistemas.

SICOES SANGAY (Sistema de Control Escolar), es una solución integral que permite a la institución optimizar las operaciones, con la finalidad de simplificar procesos para su mejor labor dentro del plantel, cubriendo las necesidades de información académica y administrativa. Su principal objetivo es centralizar y automatizar los procesos de matriculación, inscripción, registro de asistencia, registro de notas, etc.

En el proyecto se implementaron las metodologías Scrum y Xtreme Programming para automatizar todos los procesos de la escuela Sangay como la matriculación, inscripción, generación de libretas de calificaciones, etc. Se logró mejorar la calidad del servicio que presta a los estudiantes y padres de familia; reducir el tiempo en la ejecución de los procesos académicos y eliminar errores de cálculos.

ABSTRACT

Technological advances that have been generated through the computer have caused great social impact in educational institutions, which is why many of these have adapted to changes and have seen the need to automate their processes.

Managing a school control system is important for any educational institution plans now represent an effective means to expedite the procedures developed in it, higher productivity in the institutions that have the benefits of such systems.

SICOES SANGAY (School Management System) is a solution that allows the institution to optimize operations, in order to simplify processes for their best work in school, meeting the needs of academic and administrative information. Its main objective is to centralize and automate the processes of enrollment, enrollment, attendance record, record notes, etc.

This project was implemented using Scrum and Xtreme Programming methodologies to automate all processes of Sangay School such as enrollment, registration, generation of report cards, etc. The purpose of this work is to improve the quality of service provided to students and parents and reduce the time it took for all processes performed manually, is also considered an improvement in the elimination of calculation errors.

CAPÍTULO 1

EL PROBLEMA

1.1 Tema

Desarrollo de un Sistema de Control Escolar para la Escuela Bilingüe Sangay combinando las metodologías Scrum y Xtreme Programming.

1.2 Definición del problema.

En la escuela Bilingüe Sangay no existe un registro digitalizado de notas, lista de estudiantes, evaluaciones, exámenes finales, lo que ha generado un problema en la obtención de la información, pues el manejo manual provoca errores y malestar en los estudiantes y padres de familia.

1.2.1 Contextualización del problema

Los avances tecnológicos que se han generado a través de la informática han causado un gran impacto social en las instituciones educativas; es por ello que muchas de estas, se han adaptado a los cambios y han visto la necesidad de automatizar sus procesos.

La Escuela Bilingüe Sangay, ubicada en la ciudad de Quito en la Av. Pedro Vicente Maldonado S20-293 y Ayapamba. Fundada en Agosto de 1984, en su

calidad de institución educativa, pretende cumplir con los objetivos propuestos mediante la permanente actividad y el esfuerzo innovador que permita al educando encaminarse hacia la cúspide de su realización para enfrentar eficazmente los retos del nuevo milenio.

La institución educativa, tiene alrededor de 120 niños, distribuidos en los diferentes años básicos (de primero a séptimo), a su vez cuenta con 15 profesores y 2 personas que trabajan en el área administrativa, que deben llevar toda la información académica en forma simultánea, lo que da como resultado lo que considera “un cuello de botella” por la presión ejercida por los alumnos y padres de familia.

Quienes hacen la Escuela Bilingüe Sangay están convencidos que la educación no debe ser solamente cuantitativa, sino también cualitativa, ya que el bienestar humano es el fruto de la cooperación entre la inteligencia y la fuerza.

El manejo de un sistema de control escolar es importante para los planes de cualquier institución educativa, actualmente representan el medio eficaz para agilizar los procedimientos que en ella se desarrollan, generan mayor productividad en las instituciones que disponen de los beneficios de este tipo de sistemas.

Los procesos administrativos académicos de la escuela Bilingüe Sangay no se encuentran automatizados, por lo que se dificulta una mejor atención interna al personal docente y administrativo, y externa a los estudiantes y padres de familia,

ya que no existe una interacción completa con todos los usuarios que participan en la formación de los estudiantes.

La falta de integración de los procesos que se manejan en las diferentes áreas de la escuela, no permite tener un completo control, que ayude a incrementar la competitividad del espacio académico de la institución, ya que al no tener sus procesos complementados con la tecnología, no dan a sus estudiantes un valor agregado, que integre la calidad académica que brinda la institución, con la utilización de tecnología acorde con las necesidades de la escuela.

1.2.2 Formulación del Problema

Actualmente la escuela Bilingüe Sangay no posee un sistema que automatice sus tareas académicas administrativas, lo que ocasiona que cada año lectivo la planificación académica se vuelva una tarea sumamente compleja, que frecuentemente ocasiona gran pérdida de tiempo y esfuerzo, iniciando con la inscripción de alumnos en cada año lectivo, registro de notas y faltas, seguido con la dificultad que involucra manejar un amplio número de registros en hojas de cálculo, y junto a la generación de reportes que ocupan una gran cantidad de recursos materiales y humanos , que podrían ser canalizados en otras actividades.

Sin duda, entre los principales problemas que la escuela mantiene en su gestión académica es la falta de un medio apropiado para el ingreso de calificaciones de los estudiantes; para cada año básico existe una serie de datos

correspondientes a las asignaturas, esto lo hace muy cuidadosamente cada docente porque tiene que realizar varias operaciones para sacar el promedio por alumno.

Este proceso se amplía por la cantidad de educandos con que cuenta la institución por lo tanto esto hace que a los estudiantes y padres de familia no se les proporcione a tiempo las calificaciones o cualquier otra información que conforma el registro académico.

Debido a la acumulación y difícil control del registro académico de cada uno de los estudiantes, es necesario optimizar dicha situación mediante un sistema informático como una alternativa para solucionar la dificultad de llevar un control de registro académico de los alumnos.

1.3 Justificación del problema

El proyecto está enfocado en automatizar los procesos mediante el desarrollo de un sistema informático, el cual debe permitir a este centro educativo manejar la información de sus alumnos de nivel básico (de primero a séptimo de básica) de forma óptima, mejorando el registro académico en:

- Proceso de matrícula al inicio de cada año lectivo.
- Actualización de información general de los alumnos
- Actualización de notas y promedios.
- Respuesta inmediata en consultas de información académica y datos generales de los alumnos.

- Manejo digitalizado de archivos de información como: el ingreso de calificaciones, lista de alumnos, lista de docentes, libretas de calificaciones, entre otras.
- Mejoramiento en la elaboración de reportes, y entrega oportuna de la información académica requeridos por padres de familia.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un Sistema de Control Escolar para la Escuela Bilingüe Sangay combinando las metodologías SCRUM con Xtreme Programming, para administrar y controlar la información de los estudiantes de forma automatizada por el usuario final.

1.4.2 Objetivos Específicos

- Revisar el marco teórico referente a las metodologías Scrum y Xtreme Programming para el desarrollo de sistemas de información.
- Realizar la especificación de requerimientos del sistema de control escolar aplicando la norma IEEE830.
- Realizar la planificación del sistema utilizando la metodología Scrum.
- Aplicar las metodologías Scrum y Xtreme Programming para el desarrollo e implementación del sistema.

1.5 Alcance del Proyecto

Se realizará las siguientes tareas:

- El registro académico debe cubrir las necesidades de información del centro educativo y la población estudiantil para desarrollar habilidades en la consulta administrativa de la información académica.
- Contar con un sistema automatizado para el control de registro académico en la Escuela Bilingüe Sangay.

El Sistema comprenderá de los siguientes módulos:

- **Módulo administrativo – seguridad**

El sistema incluirá un módulo administrativo que se encargará de todas las personalizaciones del sistema de acuerdo a la necesidad del centro educativo, mientras que el módulo de seguridad permitirá el manejo de usuario, perfiles y roles para garantizar el acceso, seguridad y veracidad de toda la información del sistema

- **Módulo alumnos**

Este módulo controlará la información de los alumnos, por tal motivo contará con opciones para ingresar nuevos alumnos con los datos personales del mismo.

- **Módulo docentes**

Se registrará la información general de los docentes, asignará materias a cada profesor y registrará el motivo de entrada y salida de institución.

- **Módulo calendario y horarios**

Es el encargado de generar y administrar los años básicos con sus horarios durante los periodos académicos.

- **Módulo gestión escolar**

Es uno de los módulos más importantes del sistema, pues permitirá realizar las inscripciones de alumnos que desean ingresar a un nuevo año lectivo. Posteriormente gestionará la matriculación de dichos alumnos.

- **Modulo informes y consultas**

Se encargará de la generación de reportes y consultas solicitados por la escuela.

CAPÍTULO 2

MARCO TEÓRICO

Para la elaboración de un sistema informático flexible y escalable, es necesaria una metodología de desarrollo de software así como también herramientas que permitan la culminación del trabajo de acuerdo al tiempo planificado.

De acuerdo a varias investigaciones y análisis realizados, un sistema puede desarrollarse mediante la combinación de metodologías ágiles, la cual admita reducir tiempo y elaborar un software de calidad con un equipo de desarrollo pequeño.

2.1 Metodología

2.2 SCRUM

La primera vez que se asoció el término Scrum a los procesos de desarrollo fue en 1986, cuando Nonaka y Takeuchi presentaron su artículo *The New Product Development Game*¹. Nonaka y Takeuchi presentaban en este artículo un proceso adaptativo, rápido y auto-organizado de desarrollo de productos. El término Scrum deriva del mismo término en rugby, que hace referencia a como se devuelve un

¹ Takeuchi, H y Nonaka, I. *The New Product Development Game*. Harvard Business Review Jan/ Feb 1986.

balón que ha salido fuera del campo, al terreno de juego de una manera colectiva, la traducción al castellano sería melé.

Scrum surgió como práctica en el desarrollo de productos tecnológicos y no sería hasta 1993 que Jeff Sutherland aplicara el modelo al desarrollo de software en la Easel Corporation. En 1996 Sutherland presentó junto con Ken Schwaber las prácticas que empleaba como proceso formal para la gestión del desarrollo de software en OOPSLA 96. Estas prácticas de gestión pasarían a incluirse junto con otras muchas en la lista de modelos ágiles de Agile Alliance en el año 2001.²

2.2.1 Procesos

En Scrum se puede identificar tres fases: planificación del sprint, seguimiento del Sprint y revisión del sprint, como se puede apreciar en la Figura 2.1.

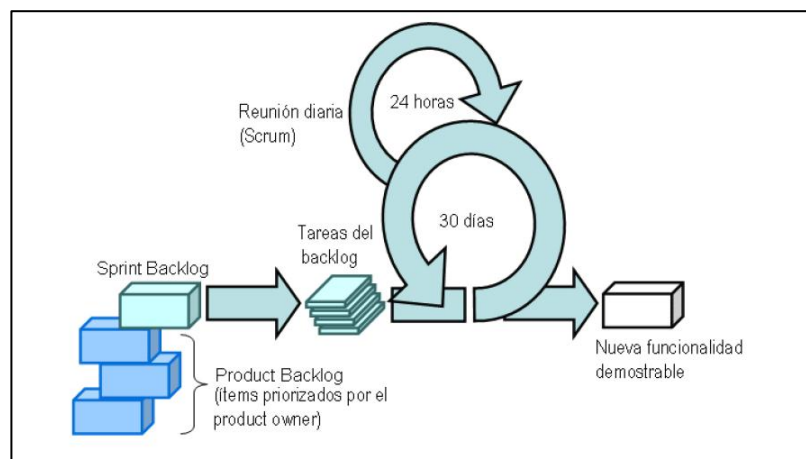


Figura 2.1. Proceso Scrum³

² <http://upcommons.upc.edu/pfc/bitstream/2099.1/5608/1/50015.pdf>

³ Figura obtenida de Flexibilidad con Scrum

En la Tabla 2.1 se puede observar las principales tareas que se llevan a cabo en cada fase.

Tabla 2.1: Fases de Scrum

Fase	Descripción
Planificación del Sprint	En esta fase se define el Product Backlog. Si todavía no ha sido definido, consiste en una lista priorizada de requisitos del sistema y es un documento vivo, que puede ser continuamente actualizado. En cada iteración el Product Backlog es revisado por el equipo. También se lleva a cabo la planificación del primer Sprint ⁴ . La planificación de cualquier sprint es la jornada de trabajo previa al inicio de cualquier sprint y en la cual se determinan cuáles son los objetivos y el trabajo que se deben cubrir en esa iteración. En esta reunión se obtiene una lista de tareas que se denomina Sprint Backlog, y el lema u objetivo principal del sprint. ⁵
Seguimiento del Sprint	A lo largo de esta fase se llevan a cabo breves reuniones diarias, para ver el avance de las tareas y el trabajo que está previsto para la jornada. En estas reuniones solo están presentes el Scrum Master y el equipo, las preguntas que se realizan suelen ser tres: <ol style="list-style-type: none"> 1. Qué trabajo se ha realizado desde la reunión anterior. 2. Qué trabajo que se va a hacer hasta la próxima reunión. 3. Qué impedimentos que deben solventarse para proseguir con el trabajo.⁶
Revisión del Sprint	Una vez finalizado el Sprint, se realiza un análisis y revisión del incremento generado. En esta reunión se presentan los resultados finales y se recomienda siempre tener preparada una demo. Existen múltiples razones para recomendar tener una demo al final de cada sprint, entre ellas la mejora del feedback con los interesados, reconocimiento del trabajo, un esfuerzo por finalizar las cosas o un correctivo en caso de tener una demo mal desarrollada. ⁷

⁴ En Scrum a las iteraciones se les llama Sprints

⁵ <http://upcommons.upc.edu/pfc/bitstream/2099.1/5608/1/50015.pdf>

⁶ <http://sedici.unlp.edu.ar/bitstream/handle/10915/5519/Documento%20completo.pdf?sequence=1>

⁷ <http://upcommons.upc.edu/pfc/bitstream/2099.1/5608/1/50015.pdf>

2.2.2 Roles y Responsabilidades

En la Tabla 2.2 se describe los roles y responsabilidades de cada persona en Scrum.

Tabla 2.2: Roles y responsabilidades de Scrum

Rol	Descripción
Scrum Master (Líder del Proyecto)	Es el encargado de garantizar el funcionamiento de los procesos y de la metodología. Es importante darse cuenta que Scrum Master es más que un rol, es la responsabilidad de funcionamiento de modelo, por tanto muchas veces es aconsejable utilizar a personas y puestos más adecuados según la organización. Un Scrum master debe interactuar tanto con el equipo como con el cliente y con los gestores.
Propietario del producto (Product Owner):	Es la única persona del proyecto conocedora del entorno de negocio del cliente y de la visión del producto y es el responsable de obtener el resultado de mayor valor posible para el cliente. También es el responsable de la financiación necesaria para el proyecto, de tomar las decisiones que afecten a cómo va a ser el resultado final, fechas de lanzamiento y el retorno de inversión. Por regla general y si no se trata de proyectos internos, el propietario del producto suele ser el responsable del proceso de adquisición del cliente. El equipo está involucrado en la estimación del esfuerzo de las tareas del product backlog, en la creación del sprint backlog, etc. ⁸
Equipo de desarrollo (Scrum Team)	Es el equipo del proyecto y tiene la autoridad para decidir en las acciones necesarias y para auto-organizarse con la finalidad de alcanzar los objetivos del sprint. ⁹

⁸ <http://es.wikipedia.org/wiki/Scrum>

⁹ http://assets.scrumfoundation.com/downloads/3/scrumprimer_es.pdf?

2.2.3 Prácticas

Scrum no requiere y/o provee de ninguna práctica concreta para el desarrollo del software, sin embargo sí que dispone de prácticas y herramientas para la gestión de las diferentes fases de Scrum. En la Tabla 2.3 se muestra las principales prácticas y herramientas de la metodología Scrum.

Tabla 2.3: Prácticas y herramientas de Scrum

Prácticas	Descripción
Product Backlog (Pila del producto)	Define los requisitos del sistema o el trabajo a hacer a lo largo del proyecto. Está compuesto por una lista de requisitos de negocios y técnicos, actualizados y priorizados. ¹⁰ El responsable de mantener el product backlog es el propietario del producto.
Sprint Backlog (Pila de tareas)	Es una lista de trabajos que el equipo se compromete a realizar para generar el incremento previsto. Las tareas están asignadas a personas y tienen estimados el tiempo y los recursos necesarios.
Estimación de esfuerzo	Es un proceso iterativo en el cual las estimaciones de los ítems del product backlog son reajustadas acorde a la información obtenida en la última iteración. Este reajuste lo llevan a cabo el equipo de desarrollo y el propietario del producto. ¹¹
Gráfico Burn-down.	Es una herramienta para gestionar y seguir el trabajo de cada sprint y representa gráficamente el avance del sprint. ¹²
Gráfico Burn-up.	Herramienta de gestión y seguimiento que sirve al propietario del producto para controlar las versiones de producto previstas, las funcionalidades de cada una, la velocidad estimada, fechas probables de cada versión, margen de error previsto en las estimaciones y avance real. ¹³

¹⁰ <http://www.navegapolis.net/files/presentaciones/scrum.ppt>

¹¹ http://www.dosideas.com/wiki/Backlog_Del_Producto

¹² http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

¹³ http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

2.3 Xtreme Programming

Xtreme Programming es sin duda es el abanderado de las metodologías ágiles. Nació como un intento, bastante exitoso, de establecer un conjunto de prácticas que facilitasen la finalización de los proyectos.¹⁴ Después de unas cuantas exitosas pruebas, estas prácticas se plasmaron de forma teórica, dando lugar a una metodología, que mantenía sus principales principios y prácticas. El término Xtreme es debido a que las prácticas que se utilizaron, fueron llevadas hasta el extremo.

2.3.1 Procesos

El ciclo de vida de Xtreme Programming está compuesto de seis fases, ver Figura 2.2.

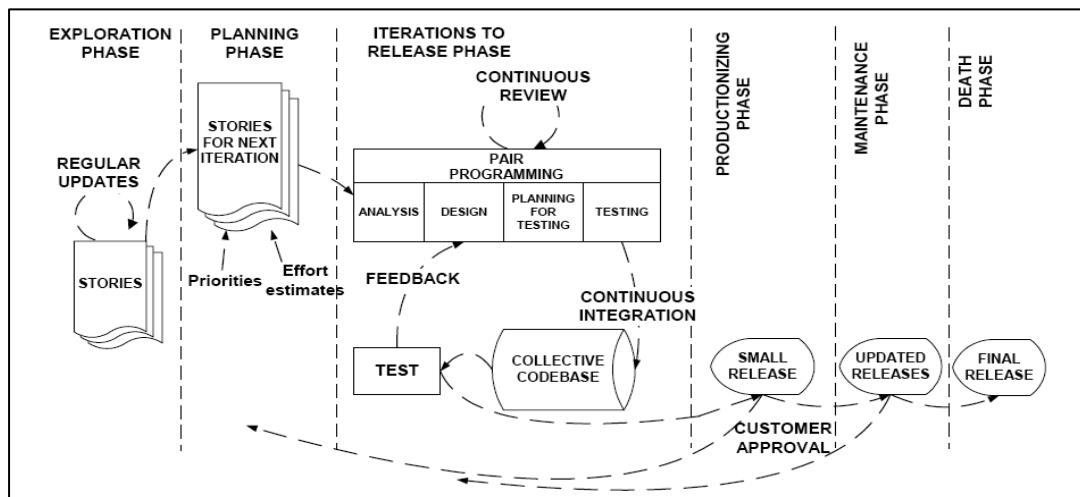


Figura 2.2. Ciclo de Vida XP¹⁵

¹⁴ Kent Beck. Extreme Programming explained: Embrace Change. Reading, Mass. Addison Wesley 1999.

¹⁵ Gerardo Fernandez Escribano, Introducción a Xtreme Programming

En la Tabla 2.4 se explica de una manera más detallada las fases de Xtreme Programming.

Tabla 2.4: Fases de Xtreme Programming

Fases	Descripción
Fase de exploración	<p>Los usuarios escriben las tarjetas de historia que ellos quieren que sean incluidas en la primera versión. Cada una de las tarjetas de historia describe una funcionalidad que será añadida al programa. El equipo de proyecto durante este tiempo se dedica a familiarizarse con las tecnologías y herramientas que utilizará a lo largo del proyecto, probando las herramientas y construyendo un prototipo simple para comprobar la arquitectura. El periodo de tiempo de esta fase puede variar desde unas pocas semanas hasta unos pocos meses, dependiendo de la familiaridad del equipo con las tecnologías.¹⁶</p>
Fase de planificación	<p>Se establece la prioridad de las diferentes historias y se acuerda el contenido de la primera entrega del proyecto.</p> <p>La estimación temporal se basa en un cálculo por parte de los desarrolladores de cada una de las historias, la primera entrega no suele tardar más de dos meses en realizarse.</p> <p>La duración de esta fase no suele exceder el plazo de unos pocos días.¹⁷</p>
Fase de iteraciones	<p>Esta fase incluye la realización de diferentes fases antes de liberar la primera versión del producto. La planificación realizada en la etapa anterior se divide en diferentes iteraciones, de una duración variable entre una semana y cuatro. Los usuarios son los que deciden que historias se van a realizar en cada iteración, sabiendo que en la primera se suele realizar un sistema con la arquitectura de todo el sistema, seleccionando aquellas historias que ayuden a construirla. Las pruebas funcionales creadas por el cliente son ejecutadas al final de cada iteración, de tal manera que al final de esta fase se obtiene una versión lista para producción.¹⁸</p>

¹⁶ <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node61.html>

¹⁷ <http://www.slideshare.net/JoseYZF/metodologas-giles-xp-y-scrum>

¹⁸ <http://tallerinf281.wikispaces.com/file/view/Tabla+comparativa+de+Metodolog%C3%ADas.pdf>

Fases	Descripción
Fase de mantenimiento	Una vez que se ha liberado la primera versión a los usuarios, el proyecto se debe mantener en el entorno de producción siempre y cuando aún haya iteraciones en fase de producción. Esto supone un esfuerzo considerable en la fase de mantenimiento e incluso se sugiere la contratación de nuevo personal para dar soporte a los clientes hasta puede cambiar la estructura del equipo. ¹⁹
Fase de cierre del proyecto.	Los clientes ya no tienen más historias que deban ser implementadas. Es necesario para que se considere que en esta fase se satisfagan todas las necesidades de los clientes y otros aspectos como fiabilidad, rendimiento, etc. La documentación del proyecto se realiza en esta fase, ya que ni la arquitectura, ni el diseño, ni el código sufrirán cambio alguno. Cuando el coste del producto es demasiado elevado para su desarrollo también se puede entrar en esta fase. ²⁰

2.3.2 Roles y responsabilidades.

En la Tabla 2.5 se nombra los diferentes roles que se puede observar en un proyecto XP, según las tareas que desarrollan y los propósitos que persiguen.

Tabla 2.5: Roles y responsabilidades de XP

Rol	Descripción
Cliente	El cliente es el encargado de escribir las historias y las pruebas funcionales, y es el que decide cuando un requisito es satisfecho. También es el encargado de establecer la prioridad de las funcionalidades a implementar.

¹⁹ <http://www.slideshare.net/JoseYZF/metodologas-giles-xp-y-scrum>

²⁰ <http://www.slideshare.net/JoseYZF/metodologas-giles-xp-y-scrum>

Rol	Descripción
Programador	Escribe tanto las pruebas como el código de la aplicación y se debe comunicar fluidamente y frecuentemente con sus compañeros. ²¹
Probador	Ayuda a escribir las pruebas funcionales al cliente, las ejecuta regularmente, transmite los resultados de las pruebas y mantiene las herramientas.
Rastreador	Es el encargado dar el “feedback” en XP. Se encarga de seguir las estimaciones hechas por los clientes y de ir avisando de las desviaciones posibles y como cuanto se ajustan a la realidad, con tal de mejorar las futuras estimaciones. También se encarga de realizar el seguimiento de las iteraciones y valorar si los objetivos se pueden alcanzar con los recursos disponibles (tiempo, personal) o si es necesario hacer algún cambio.
Coach o tutor	Es la persona responsable de todo el proceso. Es importante que tenga conocimientos y experiencia en otros proyectos de XP y de este modo guiar y ayudar al equipo a adaptarse a XP.
Consultor	Es un miembro externo que tiene los conocimientos técnicos necesarios. El consultor guía al equipo para solucionar los problemas específicos. ²²
Gestor o Manager	Más conocido como el jefe. Es el que toma las decisiones críticas y está en permanente contacto con el equipo para poder discernir las diferentes situaciones críticas. ²³

²¹ <http://es.scribd.com/doc/83342499/practica2PPDF>

²² <http://es.scribd.com/doc/83342499/practica2PPDF>

²³ <http://es.scribd.com/doc/83342499/practica2PPDF>

2.3.3 Prácticas.

Xtreme Programming dispone de un gran abanico de técnicas y prácticas, muchas de ellas que han sido escogidas de diferentes metodologías existentes que tenían suficientemente probada su eficiencia.

En la Tabla 2.6 se lista las principales técnicas y prácticas de XP.

Tabla 2.6: Técnicas y prácticas de XP

Prácticas	Descripción
El juego de la planificación	El equipo de desarrolladores estima el esfuerzo necesario para implementar las historias y los clientes determinan los objetivos y tiempos de entrega.
Historias de usuario	Son los requisitos del sistema formulados en una o dos sentencias, en el lenguaje común del cliente. ²⁴
Cortas y pequeñas iteraciones	Un sistema simple se libera cada dos o tres meses y las diferentes versiones del mismo se suceden en periodos no superiores al mes.
Metáforas	El sistema se define utilizando un conjunto de metáforas acordadas entre el cliente y los programadores. Esta historia compartida guiará todo el proceso describiendo cómo funciona el sistema.

²⁴ <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Prácticas	Descripción
Diseño simple	Se da gran importancia a la obtención de diseños simples que se puedan implementar rápidamente, evitando diseños complejos y código extra.
Pruebas	El desarrollo del software es orientado a pruebas (Test Driven Development). Las pruebas unitarias se escriben antes que el código y están funcionando continuamente. Las pruebas funcionales las escriben los clientes. ²⁵
Refactorizar	Reestructurar el sistema eliminando información duplicada, mejorando la comunicación, simplificando y añadiendo flexibilidad.
Programación por pares:	Es la técnica que promulga que dos personas escriban código en el mismo ordenador.
Propiedad colectiva	Cualquiera puede compartir cualquier parte de código con cualquier otro componente del equipo.
Integración continua	Una nueva porción de código es integrada en el código fuente, tan pronto como este lista. El sistema es integrado y construido muchas veces a lo largo del día, todas las pruebas son ejecutadas y deben ser pasadas para aceptar la nueva porción de código.
40 horas a la semana	Se establece un máximo de 40 horas de trabajo semanales. ²⁶

²⁵ <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

²⁶ <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Prácticas	Descripción
Disponibilidad del cliente	Los clientes deben estar disponibles y presentes cuando sean requeridos por el equipo de desarrollo.
Estándares de codificación	Las reglas de codificación son establecidas y seguidas por los programadores. Se enfatiza la comunicación a través del código.
Espacios de trabajo abiertos	Un gran espacio, con cubículos es lo recomendable. ²⁷

2.4 Combinación de Xtreme Programming y Scrum

La elección de las metodologías ágiles a utilizar en este proyecto se ha sustentado fundamental y principalmente, en la popularidad y la documentación, siendo Scrum y Xtreme Programming las dos metodologías que cumplen estos requisitos.

2.5 Métodos y técnicas de SCRUM

Se ha seguido con las indicaciones de Henrik Kniberg para la realización de proyectos con Scrum y Xtreme Programming, que presenta en su libro Scrum y

²⁷ <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Xp desde las Trincheras²⁸. De tal manera que los métodos y técnicas que se utilizará para Scrum son:

- **La pila del producto o product backlog:** Es el documento a través del cual se recogen los requisitos de los clientes.
- **La pila de tareas o sprint backlog:** Es el conjunto de historias, que pertenecen a la pila del producto, que se va a realizar en el sprint.
- **Estimación del esfuerzo:** Cada historia de la pila del producto es estimada con lo que se llama puntos de historia y que se corresponde a días-persona ideales.
- **Gráfico Burn-down:** Este gráfico ayuda a medir la productividad y detectar riesgos de mala distribución del trabajo y desviaciones temporales.
- Reuniones para cada sprint.
- Reuniones de seguimiento diarias.
- **Planning Poker:** Esta técnica se utiliza a la hora de realizar la estimación de las historias. Simula el juego de Poker para obtener una estimación colectiva y cooperativa de cada historia.

2.6 Métodos y técnicas de Xtreme Programming

Los métodos y técnicas que se tomará de Xtreme Programming son las siguientes:

²⁸ <http://infoq.com/minibooks/scrum-xp-fromthetrenches>

- **Test Driven Development:** Diseñar las pruebas antes que implementar las funcionalidades. Todo código que pasa a producción tiene sus pruebas asociadas.
- **Refactorizar:** Rescribir aquellos métodos y clases, para mejorar la legibilidad del código. Esta es una técnica fundamental cuando se genera software de manera incremental y a partir de las pruebas.
- **Diseño incremental:** Mantener un diseño simple desde el principio e ir mejorándolo continuamente.
- **Integración continua:** Esta técnica permite compilar el código cada vez que se sube al repositorio de código, ejecutar las pruebas y desplegarlo en un entorno de producción. A su vez, puede realizar este proceso de forma automática y cada cierto tiempo, de tal manera que se puede observar los diferentes resultados de las pruebas en diferentes condiciones y horas.

2.6.1 Herramientas Empleadas

2.6.1.1 Plataforma J2EE

La plataforma Java EE representa un estándar para la implementación y despliegue de aplicaciones empresariales. Fue diseñada a través de un proceso abierto y se entablaron conversaciones con un amplio abanico de empresas, lo que permitió que se cubriesen la gran mayoría de los requisitos de las aplicaciones empresariales.

Las principales características son:

2.6.1.1.1 Modelo Multicapa

La plataforma Java EE provee un modelo multicapa y distribuido como se puede ver en la Figura 2.3, esto quiere decir que diferentes partes de una aplicación pueden estar corriendo en diferentes dispositivos. La arquitectura define una capa cliente, una capa intermedia (compuestas por una o más subcapas) y una capa de datos, también conocida como la capa de información empresarial (IE).²⁹

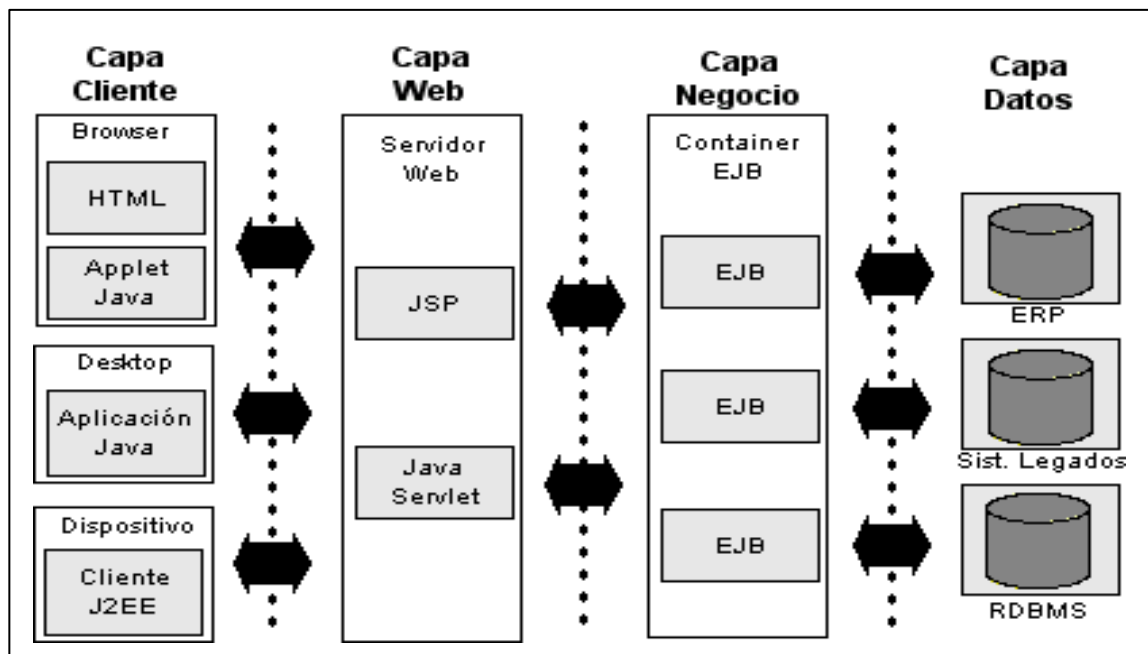


Figura 2.3. Arquitectura J2EE ³⁰

En la Tabla 2. 7 se puede ver de manera más detallada las diferentes capas de la arquitectura J2EE.

²⁹ <http://www.dcc.uchile.cl/~jbarrios/J2EE/>

³⁰ <http://www.dcc.uchile.cl/~jbarrios/J2EE/arq.gif>

Tabla 2.7: Capas de la arquitectura J2EE

Capa Cliente	<p>Está integrada por aplicaciones clientes que acceden al servidor Java EE y normalmente se encuentra en una máquina diferente a la del servidor. Los clientes hacen peticiones al servidor, este procesa las peticiones y las responde. Hay muchos tipos de aplicaciones diferentes que pueden ser clientes Java EE, y no tienen por qué ser aplicaciones Java EE, es más, a menudo no lo suelen ser. Los clientes pueden ser un navegador web, una aplicación autónoma³¹. En esta capa el cliente ejecuta HTML, Applet, aplicaciones Java, etc.</p>
Capa Web	<p>La capa de presentación, consiste en el conjunto de componentes que capturan la interacción entre los clientes y la capa de negocio.</p> <p>Sus principales tareas son las siguientes:</p> <ul style="list-style-type: none"> - Generación dinámica del contenido, en varios formatos, para el cliente. - Recoger las entradas de los usuarios de la interfaz de las aplicaciones cliente y devolver los resultados apropiados desde los componentes de la capa de negocio. - Controlar el flujo de pantallas o páginas en el cliente. - Mantener el estado de los datos para una sesión de un usuario. - Realizar algunas operaciones básicas pertenecientes a la lógica de la aplicación y guardar temporalmente alguna información (usando JavaBeans). <p>Las tecnologías más comunes de la capa de presentación o Web son:</p> <ul style="list-style-type: none"> - Servlets: Procesan dinámicamente las peticiones y construyen las respuestas de los clientes, comúnmente utilizado con páginas HTML. - JavaServerPages: Se definen como el contenido dinámico puede ser añadido a las páginas estáticas.³² - JavaServerPages Standar Tag Library: Es una librería de tags que encapsula las funcionalidades principales más comunes de las páginas JSP. - JavaBeans Components: Objetos que temporalmente almacenan lo datos de las páginas de una aplicación.

³¹ Se utiliza los términos stand-alone, standalone, autónoma e independiente para referirnos a las aplicaciones que se ejecutan en un equipo local y que no necesitan de otros recursos externos, como por ejemplo la red.

³² <http://www.dcc.uchile.cl/~jbarrios/J2EE/>

Capa Negocio	<p>Está compuesta por las clases que proveen la lógica de negocio para una aplicación. La lógica de negocio es el código que provee las funcionalidades para un particular dominio del negocio, como puede ser finanzas.</p> <p>La capa de Negocio se comunica directamente con la capa de presentación para recibir las solicitudes y mostrar resultados después de realizar el proceso.</p> <p>Las tecnologías más comunes de una Capa de Negocio son:</p> <ul style="list-style-type: none"> - Enterprise JavaBeans (EJB). - JAX-WS, web service endpoints. - Java Persistence API entities.
Capa de Datos	<p>La capa de datos está integrada por los servidores de bases de datos. Habitualmente, estos recursos están situados en una máquina independiente de en la que se encuentra el servidor Java EE, y se acceden a ellos a través de la capa de negocio.</p> <p>Las tecnologías más comunes de la capa de datos son:</p> <ul style="list-style-type: none"> - Java Database Connectivity API (JDBC). - Java Persistence API. - J2EE Connector Architecture. - Java Transaction API (JTA).³³

2.6.1.1.2 Servidor de aplicaciones Java EE

Una característica fundamental de la plataforma Java EE es la existencia y la necesidad de utilizar servidores Java EE.

Este es un servidor de aplicaciones que implementa las APIs de la plataforma Java EE y provee de los servicios estándar de Java EE.³⁴ Los servidores Java EE son llamados, a veces, servidores de aplicaciones, porque permiten obtener datos de las aplicaciones cliente, del mismo modo que un servidor web sirve páginas web a un navegador. Un servidor Java EE contiene diferentes tipos de

³³ <http://www.dcc.uchile.cl/~jbarrios/J2EE/>

³⁴ <http://www.proactiva-calidad.com/java/arquitectura/index.html>

componentes, los cuales se corresponden con cada una de las capas de una aplicación multicapa.³⁵

2.6.1.1.3 Gestión de los componentes mediante contenedores

Un contenedor de Java EE es una interfaz entre los componentes y las funcionalidades de más bajo nivel que provee la plataforma Java EE para soportar ese componente. La funcionalidad del contenedor está definida por la plataforma Java EE, y es diferente para cada tipo de componente. No obstante, los servidores Java EE permiten que los diferentes tipos de componentes trabajen juntos para proveer las funcionalidades de una aplicación empresarial. En la tabla 2.8 se nombra los tres tipos de contenedores existentes.

Tabla 2.8: Tipos de contenedores

Contenedor	Descripción
Los contenedores Web	Es la interfaz entre los componentes web y el servidor web. Un componente web puede ser un servlet, una página JSP o una página JavaServer Face. El contenedor gestiona el ciclo de vida de los componentes, envías las peticiones a los componentes de la aplicación, y provee de una interfaz a los datos del contexto, como puede ser la información sobre la petición en curso. ³⁶

³⁵ http://es.wikipedia.org/wiki/Servidor_de_aplicaciones

³⁶ <http://www.proactiva-calidad.com/java/arquitectura/index.html>

Contenedor	Descripción
Los contenedores de la aplicación cliente:	Se comunica entre las aplicaciones cliente Java EE, las cuales son aplicaciones Java que usan componentes del servidor Java EE, y el servidor Java EE. El contenedor de la aplicación cliente corre en la máquina cliente, y es la puerta de enlace entre la aplicación cliente y el servidor Java EE.
Los contenedores EJB:	Los enterprise beans proveen la lógica de negocio en una aplicación Java EE, y el servidor Java EE. El contenedor EJB corre en el servidor Java EE y gestiona la ejecución de los enterprise beans de las aplicaciones. ³⁷

2.6.1.2 Servidor Web

Un servidor web o también conocido como servidor HTTP, se encarga de procesar la aplicación del lado del servidor. El código recibido por el cliente suele ser compilado y ejecutado por un navegador.³⁸

2.6.1.2.1 Funcionamiento

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo

³⁷ <http://www.proactiva-calidad.com/java/arquitectura/index.html>

³⁸ http://es.wikipedia.org/wiki/Servidor_web

de ejemplo, al teclear www.wikipedia.org en el navegador, éste realiza una petición HTTP al servidor de dicha dirección.³⁹

2.6.2 Capas del Sistema

2.6.2.1 JPA (JAVA PERSISTENCE API)

Los Enterprise Java Beans (EJB's) en la especificación 3.0 se fundamentan tanto en el modelo propietario como en el open source al hacer uso de características existentes en alternativas como TopLink, Hibernate y Spring las cuales sirvieron de influencia para el desarrollo de la especificación de JPA.

El modelo de entidades introducido en la especificación de EJB3.0 define el uso de JPA como mecanismo de persistencia, el cual permite persistir los datos en un sistema de manejo de base de datos relacional utilizando los objetos persistentes como tablas en la BDD. JPA reemplaza a los Entity Beans en Java JEE 5, los objetos persistentes en esta especificación son tratados como POJOS (Plain Old Java Object) al igual que en la mayoría de capas de persistencia.⁴⁰

2.6.2.2 EJB 3.0 (ENTERPRISE JAVA BEANS)

La especificación EJB 3.0 introduce un modelo de desarrollo simplificado que reduce la complejidad de los enterprise beans al mismo tiempo que incorpora

³⁹ http://es.wikipedia.org/wiki/Servidor_web

⁴⁰ <http://es.scribd.com/doc/39489671/MAT-JPA>

muchas de las ideas encontradas en tecnologías afines. Las características más relevantes del modelo son:

- XML y Anotaciones
- Inyección de Dependencia
- Interceptores
- Implementación de POJOS
- Uso inteligente de Defaults
- Lenguaje de Consulta Simplificado⁴¹

2.6.2.3 Interfaz Grafica

2.6.2.3.1 Java Server Faces

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF es un framework de interfaz gráfica de usuario para aplicaciones J2EE, está basado en el patrón MVC (Modelo Vista Controlador) el cual ha sido diseñado para mejorar la productividad de los desarrolladores, al utilizar un conjunto de componentes reutilizables en la construcción de un interfaz de usuario.⁴²

⁴¹ <http://matoosfe.com/blog/>

⁴² <http://www.programacion.com/articulos/J2EE>

2.6.2.4 XHTML

XHTML (eXtensible HyperText Markup Language) XHTML es la mezcla de HTML con un XML valido. Es un lenguaje de programación más técnico, es decir este no soporta errores, se puede considerar una versión más estricta y limpia de HTML, pero la ventaja del mismo es que se hace más fácil su mantenimiento debido a que tiene un estándar de programación.⁴³

2.6.2.5 Encriptación Md5 base64

2.6.2.5.1 MD5

MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.⁴⁴

2.6.2.5.2 BASE64

En método de codificar binario dentro de textos. Cada dato binario son 8 bits por byte, considerando que el texto usa poco más de 6 bits por byte. Un número de 6 bits puede hacer 64 combinaciones, por eso también se le llama BASE 64.

La manera de trabajar de este sistema, la codificación, es que cada 3 bloques de 8 bits son guardados en 4 caracteres de 6 bits, donde los caracteres se

⁴³ http://www.despazio.net/activos/textos/tecno_info_medios_audiovisuales.pdf

⁴⁴ <http://es.wikipedia.org/wiki/MD5>

encuentran en el rango [A-Z] [a-z] [0-9] [+/>. Como no se alinean correctamente, los caracteres de [=] se usan al final.⁴⁵

2.6.3 Herramientas de Desarrollo

2.6.3.1 Eclipse

Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).⁴⁶

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.⁴⁷

⁴⁵ <http://www.elhacker.net/base64.html>

⁴⁶ [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

⁴⁷ [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

2.6.3.2 JBOSS

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, se puede utilizar en cualquier sistema operativo para el que esté disponible Java. Los principales desarrolladores trabajan para una empresa de servicios.⁴⁸

2.6.3.3 Jasper Reports

JasperReports es una herramienta de creación de informes que tiene la habilidad de crear documentos de tipo páginas, preparados para imprimir información en archivos de formato PDF, HTML, XLS, CVS y XML. Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.⁴⁹

2.6.3.4 Power Designer

PowerDesigner, es una herramienta de modelamiento que permite visualizar, analizar y manipular metadatos de una manera más fácil, mediante la combinación de técnicas de modelamiento con herramientas de desarrollo como: .NET, Sybase WorkSpace, Sybase Powerbuilder, Java y Eclipse.

⁴⁸ <http://es.wikipedia.org/wiki/JBoss>

⁴⁹ <http://es.scribd.com/doc/90192400/29/Reportes>

Esta herramienta cuenta con una interfaz amigable que ayuda a crear de una manera muy fácil la base de datos; además permite ver de manera gráfica las clases y relaciones.⁵⁰

2.6.3.5 MySQL

MySQL es un gestor de base de datos, en el cual se crea la base de datos y se puede agregar, acceder y procesar información en la misma. Se ha considerado que el MySQL es un sistema de gestión de base de datos relacional.

MySQL se puede descargar desde internet y usarlo sin hacer ningún tipo de pago ya que usa el GPL (Licencia Publica General) y de esta manera definir que se puede y no hacer con el software en diferentes situaciones, por esta razón cualquier persona puede estudiar el código fuente y de este modo cambiarlo y adecuarlo a sus principales necesidades. Hoy en día MySQL es muy rápido, confiable, robusto y fácil de usar tanto para información muy grande o pequeña. MySQL ofrece un rico y muy útil conjunto de funciones, la conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos.⁵¹

⁵⁰ <http://www.mtbase.com/productos/modelamientometadatos/powerdesigner>

⁵¹ <http://es.scribd.com/doc/15285090/residencia-de-control-escolar>

CAPÍTULO 3

ESPECIFICACIÓN DE REQUERIMIENTOS

3.1 Introducción

En esta especificación se identificarán los requerimientos que ha de satisfacer el nuevo sistema SICOES mediante el estudio de las necesidades actuales del cliente.

3.2 Propósito

- Permitir establecer acuerdos entre usuarios en lo que al proyecto de software se refiere.
- Ayudar a los usuarios finales del software a entender exactamente que es lo que realiza el sistema.

Se busca definir en forma clara, precisa, completa y verificable todas las funcionalidades y restricciones del sistema, siendo este un documento sujeto a revisiones por las partes, especialmente por el usuario final (administrador, directivo, administrativo y docente).

Esta especificación se ha realizado de acuerdo al estándar "IEEE Recommended Practice for Software Requirements Specifications (IEEE/ANSI 830-

1998)”, y se basa en las entrevistas realizadas a los usuarios y el estudio de la documentación existente.

3.3 Alcance

La aplicación se encargará de automatizar y optimizar los procesos administrativos y/o escolásticos de la institución.

El sistema realizará las funciones que se definen en la Tabla 3.1:

Tabla 3.1 Funciones del sistema SICOES

FUNCIONES	DESCRIPCIÓN
Módulo de administración y seguridad	El objetivo de este requerimiento es administrar los permisos que tendrán los usuarios y controlar su acceso a cada una de las ventanas del sistema.
Módulo alumnos	Se ingresarán los datos personales de los estudiantes, para posterior asignación de grados y distribución de paralelos.
Módulo docentes	En el módulo docente se ingresarán los datos personales de los profesores, para posterior asignación de grados.
Módulo calendarios y horarios	Se encargará de generar y administrar los grados con sus horarios, durante los periodos académicos.
Módulo gestión escolar	Dentro de este módulo el sistema permitirá realizar las inscripciones de los alumnos.
Módulo informes y consultas	Generará la impresión de reportes solicitados por la institución.

3.4 Limitaciones del sistema

Tomando en cuenta que un sistema de control escolar completo es difícil realizar en el tiempo que se establece para el proyecto de tesis, se han determinado algunas limitantes que posee el sistema y se detallan a continuación:

- No se automatizará la gestión de recursos financieros de la institución.
- No se automatizará la gestión de recursos humanos del plantel.
- No se ingresará en el sistema información histórica de anteriores años lectivos.

3.5 Personal involucrado

El equipo de personas necesario para la realización de este proyecto se detalla a continuación:

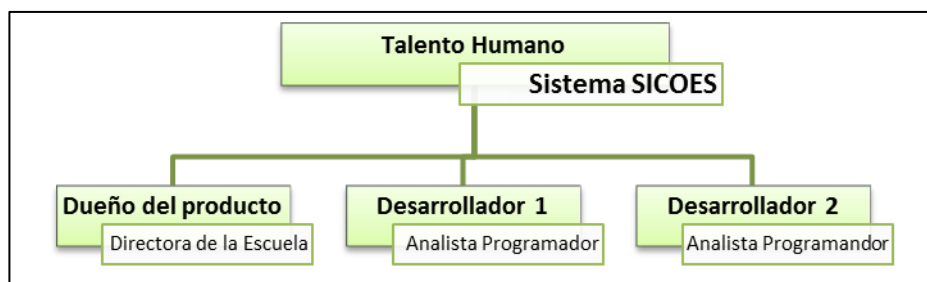


Figura 3.1. Equipo que conforma el talento humano de SICOES

3.6 Definiciones

Calificación: Valor obtenido por el estudiante en una asignatura.

Infracción: Ausencia, falta o atraso cometido por un estudiante o profesor.

Materia: Conjunto de temas relacionados entre sí.

Año básico: Consideración realizada a un grado de la educación escolar.

Paralelo: Versión de un mismo año escolar considerando un mismo año lectivo.

Plantel: Se denomina así a la escuela bilingüe Sangay

Programa curricular: Conjunto de materias.

Representante: Es quien ejerce la responsabilidad legal ante los actos o hechos de un estudiante ante el plantel.

3.7 Acrónimos

BDD: Base de Datos.

IEEE: Instituto de ingenieros eléctricos y electrónicos.

SICOES: Sistema de control escolar

3.8 Referencias

[1] Estándar IEEE830

-IEEE Recommended Practice for Software Requirements Specification.
ANSI/IEEE std. 830, 1998

[2] Estructura de sistemas de control escolar

<http://www.tesoem.edu.mx/tesoem/sa/ce.html>

[3] Procesos de un sistema de control escolar

http://issuu.com/tlmatini/docs/libro_01

[4] Reglamentos educación primaria

<http://www.liceonaval-quito.mil.ec/index.php/reglamento-de-disciplina>

3.9 Descripción general

3.9.1 Perspectiva del producto

Este producto de software, diseñado para la escuela bilingüe Sangay, tiene como propósito automatizar y optimizar los procesos administrativos y/o escolásticos de la institución, siendo un sistema autónomo, independiente que no interactuará con otro software.

3.9.2 Definición general del problema

A continuación se detalla a nivel general el ámbito en el que se basa el aplicativo:

Tabla 3.2 Definición de problema general

El problema de	Al no disponer de un software que automatice los procesos administrativos y/o escolásticos de la institución, ocasiona gran pérdida de tiempo en la realización de actividades repetitivas y constantes en cada año lectivo.
El impacto de ello es	Al no tener un sistema de control escolar ocasiona realizar los procesos administrativos y/o escolásticos de forma manual. Pérdida de tiempo, esfuerzo de recursos y desaprovechamiento de las nuevas tecnologías.
Una solución exitosa debería	Desarrollar un sistema de control escolar que permita administrar y controlar la información de los estudiantes de forma automatizada por el usuario final.

3.9.3 Estructura técnica del sistema

3.9.3.1 Arquitectura

El sistema SICOES es una herramienta destinada a mejorar los procesos administrativos y/o escolásticos de la institución.

La Figura 3.2 muestra como el sistema SICOES va a ser implementado en cuanto a su arquitectura hardware:

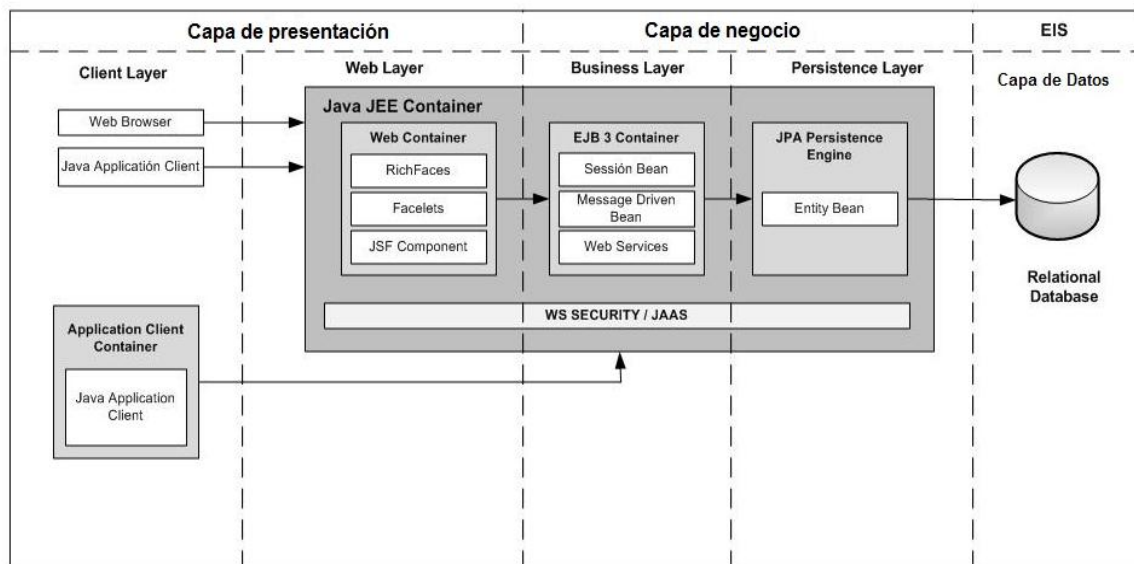


Figura 3.2. Diagrama de la arquitectura de SICOES

La arquitectura que se va a utilizar se basa en un modelo 3 capas donde el proyecto Web o capa de presentación es la parte visual que se presenta al usuario final será desarrollado utilizando herramientas como RICHFACES, FACELETS y hojas de estilo CSS, la misma que hace referencia a la librería de negocios.

La capa de negocio se comunicará con la capa de presentación para recibir las solicitudes y presentar resultados.

Para el análisis, diseño y desarrollo de este proyecto se utilizará la combinación de las metodologías Scrum y Xtreme Programming.

La aplicación podrá ser publicada en un servidor que en este caso, puede ser utilizado también como servidor de base de datos, e incluso como servidor para los servicios Web.

3.9.4 Funciones del producto

Este sistema busca solventar las necesidades de la institución, optimizando los procesos administrativos y/o escolásticos. Esta aplicación web contará con un diseño agradable de fácil utilización y con una gama de herramientas que permitirá realizar las tareas rápidamente.

El sistema contará con la implementación de seis módulos principales como se puede ver en la figura 3.3.

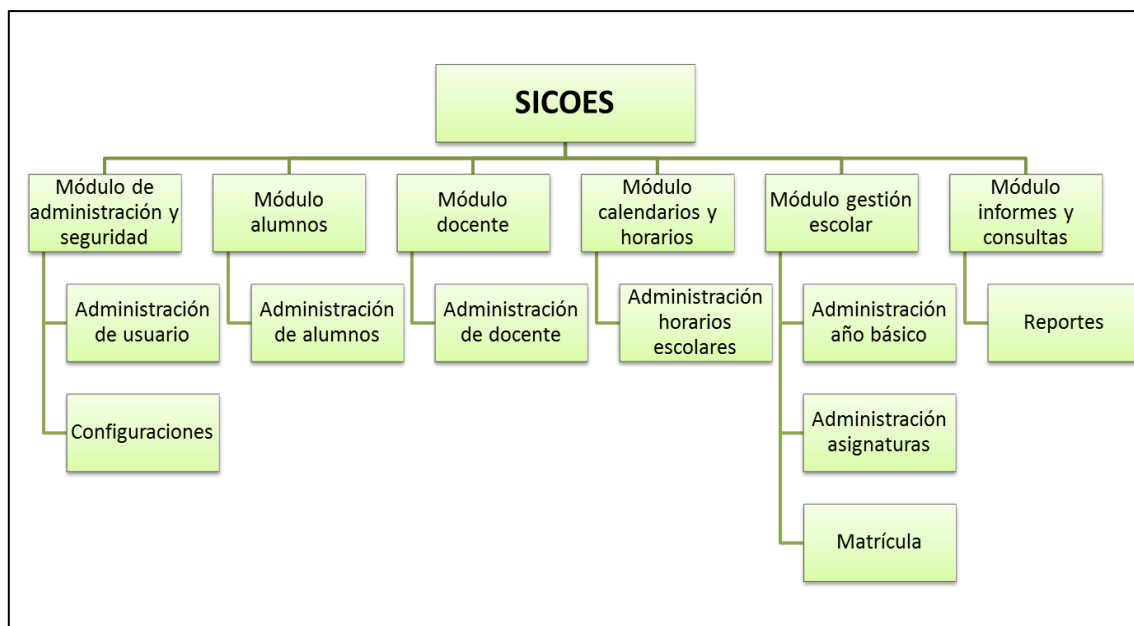


Figura 3.3. Funciones SICOES

3.9.5 Necesidades del sistema

Las necesidades principales que se reconocieron se detallan en la Tabla 3.3:

Tabla 3.3 Necesidades principales

NECESIDAD / PRIORIDAD	EXPECTATIVA	SOLUCIÓN ACTUAL	SOLUCIÓN PROPUESTA
El docente desea tener mejor control de las calificaciones y facilidad de acceso a la información de los alumnos./ IMPORTANTE	Que el sistema sea un sistema web el cual le permita obtener información de los estudiantes, además que el ingreso de notas sea mucho más fácil.	El proceso de ingreso de notas se la realiza manualmente por medio de formularios.	El sistema está siendo trabajado con tecnologías nuevas y cumpliendo con el requerimiento propuesto.

NECESIDAD / PRIORIDAD	EXPECTATIVA	SOLUCIÓN ACTUAL	SOLUCIÓN PROPUESTA
<p>La secretaria necesita agilizar los procesos de matriculación y entrega de libretas de calificaciones. /IMPORTANTE</p>	<p>Minimizar el tiempo de respuesta en la matriculación de los alumnos, así como en la generación de libretas de calificaciones.</p>	<p>El procedimiento que se lleva en el manejo de las fichas de matriculación, se presentaba muy inadecuado, porque al momento de matricular a un alumno la secretaria tenía que borrar los datos para volver a utilizar el mismo formato, todo esto se hacía de forma manual en una computadora.</p>	<p>El sistema automatizara el proceso de matriculación de los alumnos para que el tiempo de respuesta sea mínimo.</p>
<p>Los docentes necesitan que el proceso de ingreso de calificaciones sea mucho más rápido y menos tedioso. /IMPORTANTE</p>	<p>El sistema permita el ingreso de las calificaciones de una manera más rápida y que el cálculo de los promedios se realice automáticamente.</p>	<p>El procedimiento en el ingreso de las calificaciones es muy tedioso, ya que los profesores tienen que llenar un registro con las calificaciones de todos los alumnos manualmente. Esto les ocasiona mucha pérdida de tiempo y esfuerzo.</p>	<p>El sistema automatizara el proceso de ingreso de calificaciones y realizar el cálculo de promedios automáticamente.</p>

NECESIDAD / PRIORIDAD	EXPECTATIVA	SOLUCIÓN ACTUAL	SOLUCIÓN PROPUESTA
Cálculos automáticos /CRÍTICO	El sistema realice procesos automáticos sin tener que realizarlos manualmente en otras aplicaciones.	El proceso de cálculo de notas se la realiza manualmente.	El sistema realizará los procesos de cálculo automáticamente basándose en las reglas de negocio de la institución maneja para realizar el promedio de las notas de cada alumno.

3.9.6 Características de los usuarios

Los usuarios deben poseer las características que se define en la Tabla 3.4:

Tabla 3.4 Características de los usuarios

NOMBRE	DESCRIPCIÓN	RESPONSABILIDADES
Docentes	Usuario que maneja el sistema, realiza operaciones y genera información.	<ul style="list-style-type: none"> • Ingreso de calificaciones. • Ingreso de penalizaciones de los estudiantes. • Consultar información • Visualizar reportes

NOMBRE	DESCRIPCIÓN	RESPONSABILIDADES
Administrador	Usuario que gestiona el sistema en general, prueba su funcionamiento y establece reglas de negocio.	<ul style="list-style-type: none"> • Gestión de usuarios • Gestión de menú • Administración del sistema • Administración de perfiles • Generar reglas de negocio a ser aplicadas en el sistema SICOES. • Creación de horarios.
Director	Usuario que maneja el sistema, realiza operaciones y genera información de cada año lectivo.	<ul style="list-style-type: none"> • Planificación de programas escolares. • Gestión de ediciones de niveles. • Gestión de programas curriculares. • Gestión de asignaturas. • Consultar los reportes generados por el sistema, como son las notas de los alumnos, el listado de alumnos por año básico y todas las materias dictadas en cada uno de los periodos académicos.
Administrativos	Usuario que maneja el sistema, realiza operaciones y genera información.	<ul style="list-style-type: none"> • Matriculación de estudiantes. • Gestión de información de los docentes. • Generación de información estudiantil y representantes. • Consultar información. • Gestionar y generar reportes.

3.9.7 Herramientas de desarrollo

El sistema será desarrollado en lenguaje JAVA, en el IDE Eclipse 1.2.1, con la base datos MySQL y servidor de aplicaciones JBoss 5.1.

Se utilizarán las metodologías SCRUM y Xtreme Programming. Con Scrum se realizará la planificación y seguimiento del proyecto, mientras que con Xtreme Programming se gestionará la parte de Desarrollo del software.

Por ser un sistema web, se necesita de conexión a una intranet para que el sistema pueda correr y visualizarse de forma adecuada. El aplicativo podrá correr exclusivamente en Mozilla Firefox.

Para el funcionamiento de esta aplicación por parte del cliente no será necesario un gran uso de memoria. Sin embargo, en el servidor será necesaria una mayor capacidad de almacenamiento para que se pueda guardar toda la información requerida.

3.9.8 Especificaciones de requerimientos

- **Módulo de Administración:** El administrador tiene la capacidad de ingresar nuevos usuarios al sistema, determinar el perfil de cada uno, ver la información que no es confidencial de los mismo y verificar los menús y pantallas a las que tendrá acceso.

- **Módulo Alumnos:** Para la administración de los datos básicos del alumno, se necesita de una página que realice las operaciones de ingreso, eliminación y edición de los estudiantes. También se podrá realizar el ingreso de calificaciones de cada uno de los estudiantes y realizar la generación de los reportes de calificaciones.
- **Módulo Docentes:** Se necesita tener los datos básicos de los profesores que imparten las materias en la institución, por lo que se requiere de una pantalla que permita administrar esta información. Toda esta información puede ser actualizada y el docente puede ser dado de baja en el caso que ya no labore en la institución.
- **Módulo calendarios y horarios:** El administrado será el encargado de ingresar información de calendarios y horarios de acuerdo a la planificación para cada año lectivo.
- **Módulo gestión escolar:** Una vez que un estudiante este creado e inscrito, el usuario podrá realizar la matriculación de los estudiantes en un nivel determinado.
- **Módulo informes y consultas:** El usuario podrá construir diversos reportes con información personalizada de acuerdo a las necesidades propias de la institución.

3.9.9 Requisitos comunes de las interfaces

3.9.9.1 Interfaz de usuario

- Las interfaces gráficas de usuario se visualizan mediante pantallas de un explorador donde los colores predominantes son el azul y blanco. Los tipos de letras que se utilizan son verdana en un tamaño que varía entre 10 y 14, de acuerdo al tipo de texto, para dar facilidad a la lectura. Los gráficos e imágenes tienen dimensiones apropiadas de acuerdo al impacto y lo que requiera resaltar en una pantalla determinada.

3.9.9.2 Interfaces de software

- Las interfaces de software son Windows, MySQL.

3.9.9.3 Interfaces de comunicación

Las interfaces de comunicaciones son los protocolos TCP/IP, HTTP.

3.9.10 Requerimientos funcionales

NEC.	RF.	DESCRIPCIÓN	MÓDULO	FASE
N0	RF 0.0	Desarrollar un sistema de control escolar que permita de manera fácil, confiable y segura, administrar los datos escolares de forma automatizada por el usuario final.	N/A	Fase I
N1	RF 1.0	<p>El usuario deberá ingresar al sistema mediante un nombre de usuario y una contraseña previamente asignada de acuerdo a su rol.</p> <ul style="list-style-type: none"> • Al momento de ingresar el nombre de usuario y la contraseña, estos datos serán validados por el sistema. • En caso de no ser los datos correctos, se enviará un mensaje de error y se pedirá que se vuelva a ingresar la información. 	N/A	Fase I
N2	RF 2.0	Una vez ingresado al sistema se desplegará una pantalla de acuerdo al rol del usuario, con las opciones de módulos al que se desea ingresar.	N/A	Fase Incepción
N3	RF 3.0	Es necesario que exista una opción en donde se pueda crear, modificar y eliminar las configuraciones previas que van a ser utilizadas dentro del sistema	Módulo de administración y seguridad	Fase Incepción

NEC.	RF.	DESCRIPCIÓN	MÓDULO	FASE
N4	RF 4.0	Debe existir una pantalla en donde se pueda administrar las opciones del menú, en donde se especifique la ubicación de las ventanas.	Módulo de administración y seguridad	Fase Incepción
N5	RF 5.0	Es necesario que se especifiquen los tipos de usuario que van a manejar el sistema.	Módulo de administración y seguridad	Fase Incepción
N6	RF 6.0	Debe existir una pantalla en donde se pueda ingresar los usuarios que van a visualizar el sistema, con su rol y permiso definido.	Módulo de administración y seguridad	Fase Incepción
N7	RF 7.0	Es necesaria la creación de un módulo donde se registren parámetros generales del sistema. Se necesita una pantalla donde se gestione los parámetros generales del sistema como: tipos de nivel, tipos de documentos, escalas de notas, etc.	Módulo de administración y seguridad	Fase Incepción
N8	RF 8.0	Para ingresar un nuevo alumno, debe existir una pantalla en la que se permita ingresar la información básica de los estudiantes, de los padres de familia y de los representantes.	Módulo alumnos	Fase Elaboración
N9	RF 9.0	Para la inscripción de los alumnos debe existir una pantalla donde se gestione este proceso por cada año lectivo.	Módulo alumnos	Fase Elaboración

NEC.	RF.	DESCRIPCIÓN	MÓDULO	FASE
N10	R 10.0	Este módulo debe contar con una pantalla para el ingreso de calificaciones de cada alumno por cada periodo académico. El ingreso de calificaciones se lo hará cada docente de acuerdo a las materias que dicte y a cada año básico que imparta.	Módulo alumnos	Fase Elaboración
N11	RF 11.0	Para la generación de libretas este módulo debe tener una pantalla donde se pueda generar un reporte académico por periodo académico de cada alumno.	Módulo alumnos	Fase Elaboración
N12	RF 12.0	Es necesario tener los datos básicos de todos los docentes que laboran en la institución para la creación del programa curricular de cada año lectivo. Por lo que este módulo debe tener una pantalla donde se ingrese la información básica de cada docente.	Módulo Docentes	Fase Elaboración
N13	RF 13.0	Para generar los horarios académicos de cada nivel escolar, el modulo debe tener una pantalla para la generación de horarios por año básico y docente.	Módulo calendarios y horarios	Fase Elaboración

NEC.	RF.	DESCRIPCIÓN	MÓDULO	FASE
N14	RF 14.0	Es preciso crear años básicos, paralelos, tipos de asignaturas y asignaturas por cada año básico, por lo que este módulo tendrá una pantalla que permitirá administrar este tipo de información.	Módulo gestión escolar	Fase Elaboración
N15	RF1 5.0	Debe existir una pantalla que automatice la matriculación de los estudiantes del plantel.	Módulo gestión escolar	Fase Elaboración
N16	RF1 6.0	Debe existir una pantalla donde se emitirá todos los reportes generados por el sistema solicitados por el cliente.	Módulo consultas y reportes	Fase Elaboración

3.9.11 Requerimientos no funcionales

3.9.11.1 Desempeño:

Se garantiza el buen funcionamiento de la herramienta al contar con un tiempo de respuesta de máximo 10 segundos al mostrar cada pantalla y devolver los resultados, por ejemplo la actualización de una transacción.

El desempeño del sistema será establecido según los siguientes parámetros:

- Tiempo de respuesta aceptable por el usuario.
- Validación de usuarios para el ingreso al sistema

3.9.11.2 Disponibilidad:

La disponibilidad del sistema dependerá del empleo del servidor.

3.9.11.3 Escalabilidad:

El sistema será construido sobre la base de un desarrollo evolutivo, ya que será desarrollado por módulos, de tal forma que estos módulos puedan ser modificados a futuro dependiendo de la necesidad.

3.9.11.4 Facilidad de uso:

El sistema será de fácil uso, intuitivo y sencillo, además se deberá ambientar a las necesidades de la entidad.

El ingreso de los datos al sistema deberá ser diseñado con un límite de transacciones permitidas, es por esto que se ingresarán datos de forma parcial, es decir que el tamaño de los registros o formularios de información que se necesite ingresar, se ajusten a una determinada estabilidad de la red.

El sistema presentará mensajes de error al usuario, identificando el tipo de error, de esta manera el usuario se podrá comunicar con el administrador del programa en caso de no poder solucionar el inconveniente.

El sistema no deberá permitir al usuario cerrar una operación hasta que todos sus procesos, subprocesos y tareas, hayan sido terminados y cerrados satisfactoriamente.

3.9.11.5 Seguridad:

Se garantizará la seguridad, haciendo uso de nombres de usuario y contraseña que definan los diferentes roles de cada uno de los usuarios.

El sistema deberá contar con una interfaz de administración de seguridad que incluya:

- Administración de usuarios.
- Administración de módulos.
- Administración de menú.

El control de acceso implementado debe permitir asignar los perfiles para cada uno de los roles identificados.

El sistema debe validar la información contenida en los formularios de ingreso. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad de campos, longitud de caracteres permitida por campo, manejo de tipos de datos entre otros.

CAPÍTULO 4

PLANIFICACIÓN Y DESARROLLO DEL SISTEMA

Una vez que se han definido todos los requerimientos que conforman la documentación de la pila del producto (product backlog) para el presente proyecto, se empezará con el desarrollo del sistema de control escolar para la escuela bilingüe Sangay.

4.1 Planificación del proyecto utilizando la metodología SCRUM

Como primer punto de partida en esta fase, se tiene la especificación de la pila de producto, que es el documento más relevante en el desarrollo de proyectos con Scrum. Básicamente es una lista priorizada de los requerimientos fundamentales implicados en el presente proyecto. A cada ítem de la pila se llama historias de usuario.

De los diferentes campos recomendados para identificar cada historia de usuario, se ha seleccionado los siguientes:

- **ID:** Identificado único de la historia de usuario.
- **Nombre:** Es una descripción corta de la historia.
- **Importancia:** Definido por el propietario del producto, es el grado de prioridad para el desarrollo de cada historia de usuario.

- **Estimación Inicial:** Es la estimación inicial del equipo sobre la cantidad de trabajo que es necesario para implementar la historia, comparada con otras historias.
- **Como Probarlo:** Una descripción a alto nivel de cómo se demostrará esta historia en la Demo al final del Sprint
- **Notas:** Cualquier otra información, clarificación, referencia a otras fuentes de información, etc.

En la Tabla 4.1, se puede observar que se tiene únicamente 15 requerimientos iniciales que englobarán las diferentes tareas o actividades, las cuales se las detalla en una pila de tareas (sprint backlog). De acuerdo a la metodología, el sprint backlog contiene todas las tareas a realizar y las cuales tienen un peso así como un tiempo estimado de desarrollo.

Para el desarrollo del presente proyecto, se irá especificando cada tarea del sprint backlog de su correspondiente requerimiento, así como toda la documentación de implantación o desarrollo de la misma.

Tabla 4.1 Pila del Producto Inicial (Product Backlog)

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
1	Ingreso al sistema	15	4	1	Entrar a la aplicación, ingresar usuario y contraseña, si el usuario es correcto ingresa al menú inicial caso contrario se presentara un mensaje de error	El usuario debe estar ingresado en la base de datos.
2	Creación de usuario en el sistema	14	4	1	Entrar a la aplicación, ir a seguridad, ir a usuario, ingresar los datos solicitados, realizar el ingreso de la información, verificar en el tracking que se haya creado en nuevo usuario.	Para la creación de usuario previamente deben estar creados los respectivos perfiles de acuerdo a los roles que maneja la institución.
3	Ingreso de información del sistema	13	4	2	Entrar a la aplicación, ir a configuraciones previas ingresar datos de la escuela, verificar la información ingresada en el tracking de la pantalla.	Esta información puede ser eliminada y modificada por el administrador.
4	Ingreso de parámetros generales de alumnos en el sistema	12	4	2	Entrar a la aplicación, ir a configuraciones previas, ir a alumnos, ingresar la información solicitada como: tipo de bajas, tipo de asistencia, escala de notas. Verificar la información ingresada en el tracking de cada pantalla.	Esta información puede ser eliminada y modificada por el administrador.
5	Ingreso de parámetros generales de docentes en el sistema	11	3	2	Entrar a la aplicación, ir a configuraciones previas, ir a docente, ingresar la información solicitada como: tipo docente, motivo de entrada y salida. Verificar la información ingresada en el tracking de cada pantalla.	Esta información puede ser eliminada y modificada por el administrador.

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
6	Ingreso de parámetros generales de calendarios y horarios en el sistema	10	3	2	Entrar a la aplicación, ir a configuraciones previas, ir a calendarios y horarios, ingresar la información solicitada como: ciclo lectivo, periodos, tipo de hora, feriados del año. Verificar la información ingresada en el tracking de cada pantalla.	Esta información puede ser eliminada y modificada por el administrador.
7	Administración Alumnos	9	4	3	Entrar aplicación, ir a docentes, ir a alumnos, ir ingresar nuevo, ingresar información del nuevo alumno, realizar el ingreso de información, verificar en el tracking de la pantalla el nuevo alumno ingresado.	Esta información puede ser eliminada y modificada por el administrador.
8	Inscripción Alumnos	8	4	3	Entrar aplicación, ir a docentes, ir a alumnos, ir ingresar inscripción, ingresar información solicitada, realizar la inscripción, verificar en el tracking de la pantalla el nuevo alumno inscrito.	Esta información puede ser eliminada y modificada por el personal administrativo.
9	Matriculación Alumnos	7	3	3	Entrar a la aplicación, ir a alumnos, ir a ingresar nuevo, ingresar la información del alumno, ingresar alumno, verificar en el tracking el nuevo alumno ingresado. Ir a alumnos, ir a inscripción, ingresar la información del alumno (datos alumno, ciclo lectivo, año básico), realizar inscripción, verificar en el tracking el nuevo alumno inscrito. Ir a gestión escolar, buscar alumno, ingresar información de matriculación, realizar la matricula, verificar en el tracking el nuevo alumno matriculado.	La información de los alumnos y de la inscripción puede ser eliminada y modificada por el administrador. Para el proceso de matriculación solo se puede dar de baja al alumno.

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
10	Administración Docentes	6	4	4	Entrar aplicación, ir a docentes, ir a administrar, ingresar información del docente, realizar el ingreso de información, verificar en el tracking el nuevo docente ingresado.	Esta información puede ser eliminada y modificada por el administrador. Para la generación de horarios se debe relacionar a los docentes con cada materia que estos imparten.
11	Gestión Escolar	5	4	5	Entrar a la aplicación, ir a configuraciones previas ingresar datos de la escuela, verificar la información en el tracking	Esta información puede ser eliminada y modificada por el administrador.
12	Ingreso de asistencia alumnos	4	4	5	Entrar a la aplicación, ir a alumnos, buscar al alumno, ingresar la asistencia del estudiante, verificar la información en el tracking de la pantalla.	Esta información puede ser eliminada y modificada por el docente.
13	Generar horarios escolares	3	3	5	Entrar a la aplicación, ir a horarios y calendarios, ir a horario escolar, ingresar la información que se solicita, ingresar el nuevo horario escolar, verificar la información en el tracking de la pantalla.	Esta información puede ser eliminada y modificada por el personal administrador.
14	Generación de libretas	2	4	5	Entrar a la aplicación, ir a alumnos, ir a generación de libretas, ir a libretas, seleccionar al estudiante que se desea generar la libreta escolar, verificar la libreta de calificaciones en otra pantalla en formato PDF.	Todo reporte se mostrará en formato PDF.
15	Reportes y Consultas	1	4	6	Entrar a la aplicación, ir a reportes y consultas, seleccionar el tipo de reporte, verificar el reporte en el archivo PDF.	Todo reporte se mostrará en formato PDF.

4.2 Desarrollo del Sistema

Una vez que se han definido todos los requerimientos que conforman la documentación de la pila del producto inicial para el presente proyecto, se empezará con el desarrollo del sistema de control escolar.

Tomando en cuenta que la metodología Scrum hace referencia a todas las actividades que el grupo de desarrollo realizará para alcanzar la solución a los requerimientos obtenidos, por lo tanto, adicionalmente a los requerimientos de la escuela bilingüe Sangay, se deberán incluir los requerimientos que implican la utilización de las herramientas seleccionadas.

En este punto se abordará cada uno de los elementos de la pila del producto y su respectivo sprint backlog (pila de tareas), así como la documentación del proceso de desarrollo inmerso en cada actividad del Sprint.

Se iniciará el proceso con la pila de tareas priorizadas, detalladas en la Tabla 4.1 (pila del producto inicial). De acuerdo a la metodología, por cada uno de los elementos de la pila de producto, se debe tener una estimación de la importancia y del esfuerzo inicial. De la estimación de la importancia deberá encargarse el dueño del producto (product owner), quien a su vez coloca una valoración a cada uno de dichos elementos de acuerdo al peso que observa dentro del proyecto.

Por otro lado, para la estimación del esfuerzo inicial, se debe encargar el scrum manager, de acuerdo a las reuniones con el grupo de desarrolladores

(scrum team), para delimitar el esfuerzo requerido. Así mismo en la Tabla 4.1, se puede apreciar la asignación del esfuerzo inicial requerido para cada uno de los 15 requerimientos señalados. Esto es especialmente importante, para poder tener una idea clara de cuáles actividades serán las que más recursos humanos demandarán.

La metodología señala que estas estimaciones se las cuantifica de acuerdo a un criterio del scrum master, por lo que para la presente tesis se tiene que las escalas para dichas estimaciones son:

- Estimación: Escala de 1 - 4.
- Importancia : Escala de 1 - 15

Una vez definida la pila del producto con sus respectivas valoraciones, se debe especificar por cada uno de los elementos (requerimientos), las correspondientes acciones o tareas a realizar o lo que en Scrum se denomina pila de tareas.

Definidas la pila de tareas y las actividades a ejecutar, se procederá a realizar la documentación referente a la aplicación de la metodología, por cada uno de las Iteraciones (Sprints).

Scrum define cada una de las iteraciones como Sprints, para los cuales se escogen un conjunto de historias de usuario de la pila del producto, creando la pila de tareas. Las historias escogidas para cada sprint serán las que se desarrollarán a lo largo de esa iteración, por lo tanto es importante acertar en la estimación de esfuerzo o el tiempo que se dedicará a cada historia de usuario, así como la velocidad del equipo y de este modo asignar el tiempo a cada Sprint. Los Sprints suelen tener una duración bastante reducida y que no suele superar el mes o mes y medio, lo más normal son tres o cuatro semanas. Para el desarrollo de este proyecto se ha fijado una duración de 19 semanas.

4.3 Estándares de Programación

Los estándares de programación permiten normalizar la estructura de codificación de variables, tablas, vistas, ayudando a escribir código de calidad que pueda ser comprendido fácilmente en un futuro por cualquier programador que retome el desarrollo del sistema, para implementar nuevos requerimientos o solo necesite dar mantenimiento a los módulos ya implementados.

La Tabla 4.2 muestra las etiquetas HTML usadas en el desarrollo del proyecto.

Tabla 4.2 Descripción de las etiquetas HTML⁵²

Etiqueta	Descripción
inputText	Simple línea de texto de entrada
form	Formulario HTML
inputSecret	Contraseña de entrada
outputLink	Enlace HTML
outputText	Simple línea de texto de salida
commandButton	Botón: submit, reset o push Button
commandLink	Enlace asociado a un botón pushbutton
message	Muestra en mensaje más reciente para un componente
graphicImage	Muestra una imagen
selectOneMenu	Selección simple para menú
selectOneRadio	Conjunto de botones radio
selectBooleanCheckbox	Checkbox
slectManyMenu	selección múltiple de menú
panelGroup	Dos o más componentes que son mostrados como uno
dataTable	tabla de datos

La codificación del sistema también con lleva a una estandarización para definir a las clases, funciones, esto hace los programas más entendibles haciéndolos más fáciles de leer, además de dar información sobre la función de un identificador, por ejemplo cuando es una clase, constante, paquete, puede ser útil para entender el código fuente del sistema. La Tabla 4.3 muestra la estandarización de código fuente que se usará en el desarrollo del proyecto.

⁵² http://es.scribd.com/doc/52009826/46/Etiquetas-JSF-estandares#outer_page_61

Tabla 4.3 Estandarización de código Fuente

Descripción	Reglas para nombres	Ejemplo
Paquetes	El prefijo del nombre de un paquete se escribe siempre con letras ASCII en minúsculas, y debe ser uno de los nombres de dominio de alto nivel, actualmente com, edu, gov, mil, net, org, entre otros. Los subsecuentes componentes del nombre del paquete variarán de acuerdo a las convenciones de nombres internas de cada organización.	edu.ec.crud.alumno edu.ec.entity
Clases	Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Intentar mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas	public class AsignaturaImpl public class AlumnoImpl
Interfaces	Los nombres de las interfaces siguen la misma regla que las clases.	public interface AlumnoLocal public interface AsignaturaLocal
Métodos	Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.	create(Alumno alumno) create(Asistencia asistencia)
Variables	Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula. Los nombres de las variables deben ser cortos pero con significado.	private int idPais; private int idProvincia; private String message; private boolean visibleMessage;
Constantes	Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("_").	final static String REGISTRO_NUEVO = "El nuevo registro ha sido guardado";

4.3.1 Análisis y Desarrollo de la Iteración 1

Una vez que se han definido todos los requerimientos globales que conforman la documentación de la pila del producto inicial para el presente tema de tesis, se empezará como primera Iteración el desarrollo del requerimiento de las historias de usuario número 1 y 2 que consta de elaboración de la página de logueo y creación de usuarios para el ingreso del usuario final al sistema de control escolar.

Para el desarrollo de este requerimiento, se realizará una especificación detallada de cada una de los requerimientos iniciales que se definieron en la Tabla 4.1.

4.3.1.1 ITERACIÓN (SPRINT) 1

Para la Iteración 1 se fijó una duración de 20 días laborables es decir 4 semanas.

4.3.1.2 Pila de producto de la Interacción 1

Como se ha mencionado anteriormente la pila del producto es la lista de todos los requerimientos del cliente, los mismos se obtienen desde la lista de historias de usuario. En la Tabla 4.4 se detalla la pila del producto que se definió para el desarrollo de la Iteración 1.

Tabla 4.4 Pila del producto de la Iteración 1

PILA DE PRODUCTO SPRINT 1						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
1	Ingreso al sistema	15	4	1	Entrar a la aplicación, ingresar usuario y contraseña, si el usuario es correcto ingresa al menú inicial caso contrario se presentara un mensaje de error.	El usuario debe estar ingresado en la base de datos.
2	Creación de usuario en el sistema	14	4	1	Entrar a la aplicación, ir a seguridad, ir a usuario, ingresar los datos solicitados, realizar el ingreso de la información, verificar en el tracking que se haya creado en nuevo usuario.	Para la creación de usuario previamente deben estar creados los respectivos perfiles de acuerdo a los roles que maneja la institución.

4.3.1.3 Sprint Backlog Iteración 1

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la Iteración 1 que engloba la historia de usuario ingreso al sistema y creación de usuario en el sistema. En la Tabla 4.5 se detalla la pila de tareas para la Iteración 1.

A la hora de organizar, repartir y trabajar sobre las historias de usuario de la pila de tareas, se suele realizar una división de cada historia en diferentes actividades como se ve en la Tabla 4.5. Es importante observar la diferencia entre tarea e historia, mientras que una historia es un entregable y es responsabilidad del dueño del producto, una tarea no es entregable y la responsabilidad es del equipo de desarrollo. Una historia suele dividirse en diferentes tareas.

4.3.1.3.1 Revisión del Sprint

Se presentan las listas de tareas realizadas y pendientes de cada desarrollador del sistema para seguir midiendo su avance para llegar a la fecha límite planteada.

4.3.1.3.2 Tareas completadas

Se en listan las tareas que los desarrolladores ha terminado hasta el momento de la revisión del sprint en la reunión. En la Tabla 4.6 se detalla las tareas completadas al finalizar la Iteración 1.

Tabla 4.5 Pila de actividades de la Iteración 1

Sprint 1		Inicio 6-Junio-2011	Duración (días) 20	Elemento del Product Backlog Ingreso al Sistema		Días 6 -10 Junio 13-17 Junio 20-24 Junio 27-1 Julio				
Id	Tarea	Delegado	Estado	Hora	20	20	20	20		
1	Elaboración del contexto navegacional	A & J	Completado		X					
1	Diseño de interfaces gráficas de usuario	A & J	Completado		X					
1	Elaboración del modelado lógico y físico de la base de datos	A & J	Completado		X					
1	Codificación de la persistencia Entity Beans (JPA) de la página de logueo	A & J	Completado		X					
1	Elaboración de consultas JPQL para la página de logueo	A & J	Completado		X					
1	Codificación de Interface y Session Bean (CRUD) de la página de logueo	A & J	Completado			X				
1	Codificación de la lógica del negocio para la página de logueo	A & J	Completado			X				
1	Configuración de las seguridades con JAAS para la página de logueo	A & J	Completado			X				
1	Configuración de los JSF y Baking Bean de la página de logueo	A & J	Completado			X				
2	Codificación de la persistencia Entity Beans (JPA) para la creación de usuarios	A & J	Completado				X			
2	Elaboración de consultas JPQL para la creación de usuario	A & J	Completado				X			
2	Codificación de Interface y Session Bean (CRUD) para la creación de usuario	A & J	Completado				X			
2	Codificación de la lógica del negocio para la creación de usuarios	A & J	Completado					X		
2	Configuración de las seguridades con JAAS para la creación de usuarios	A & J	Completado					X		
2	Configuración de los JSF y Baking Bean para la creación de usuarios	A & J	Completado					X		

Tabla 4.6 Tareas completadas de la Iteración 1

Tarea asignada a: Ana Lucia Chávez - José Tenorio		Estado Tareas: Completado		Numero de Tareas: 15	
Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación	
1	Elaboración del contexto navegacional	Sistema de Control Escolar	1	09-Junio-2011	
1	Diseño de interfaces gráficas de usuario	Sistema de Control Escolar	1	09-Junio-2011	
1	Elaboración del modelado lógico y físico de la base de datos	Sistema de Control Escolar	1	09-Junio-2011	
1	Codificación de la persistencia Entity Beans (JPA) de la página de logueo	Sistema de Control Escolar	1	10-Junio-2011	
1	Elaboración de consultas JPQL para la página de logueo	Sistema de Control Escolar	1	10-Junio-2011	
1	Codificación de Interface y Session Bean (CRUD) de la página de logueo	Sistema de Control Escolar	1	13-Junio-2011	
1	Codificación de la lógica del negocio para la página de logueo	Sistema de Control Escolar	1	16-Junio-2011	
1	Configuración de las seguridades con JAAS para la página de logueo	Sistema de Control Escolar	1	17-Junio-2011	
1	Configuración de los JSF y Baking Bean de la página de logueo	Sistema de Control Escolar	1	17-Junio-2011	
2	Codificación de la persistencia Entity Beans (JPA) para la creación de usuarios	Sistema de Control Escolar	1	21-Junio-2011	
2	Elaboración de consultas JPQL para la creación de usuario	Sistema de Control Escolar	1	24-Junio-2011	
2	Codificación de Interface y Session Bean (CRUD) para la creación de usuario	Sistema de Control Escolar	1	23-Junio-2011	
2	Codificación de la lógica del negocio para la creación de usuarios	Sistema de Control Escolar	1	28-Junio-2011	
2	Configuración de las seguridades con JAAS para la creación de usuarios	Sistema de Control Escolar	1	01-Julio-2011	
2	Configuración de los JSF y Baking Bean para la creación de usuarios	Sistema de Control Escolar	1	01-Julio-2011	

4.3.1.3.3 Burn Down Chart

En el proceso de desarrollo de un proyecto con Scrum se puede seguir el avance del mismo durante el Sprint por medio de una gráfica llamada Burn Down Chart. La Figura 4.1 muestra de qué manera se están cumpliendo con los tiempos de entrega establecidos para el Sprint en curso, esta gráfica se presenta en relación a las tareas por cumplir y el tiempo de la última entrega del sistema.

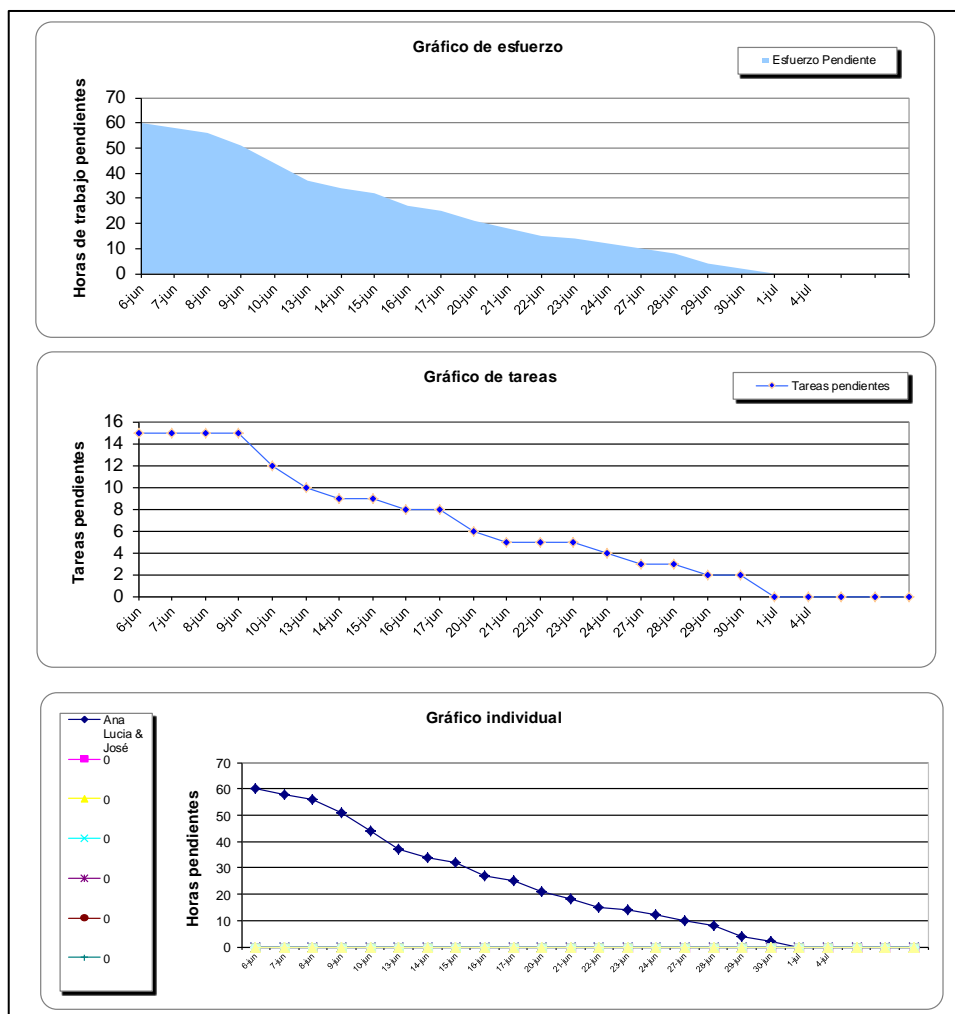


Figura 4.1 Gráfico Burn Down Chart de la Iteración 1

4.3.1.3.4 Pruebas de la Iteración 1

En el Capítulo 2 se especificaron las técnicas escogidas de cada una de las metodologías seleccionadas. Para Xtreme Programming se seleccionó el Test Driven Development, diseño incremental, refactorización e integración continua.

Una de las tareas recurrentes en el desarrollo de este proyecto es la creación de las pruebas, esto es debido a que se practica Test Driven Development y es una condición fundamental para su desarrollo. En la Figura 4.2 se puede ver los resultados exitosos de la clase Usuario.

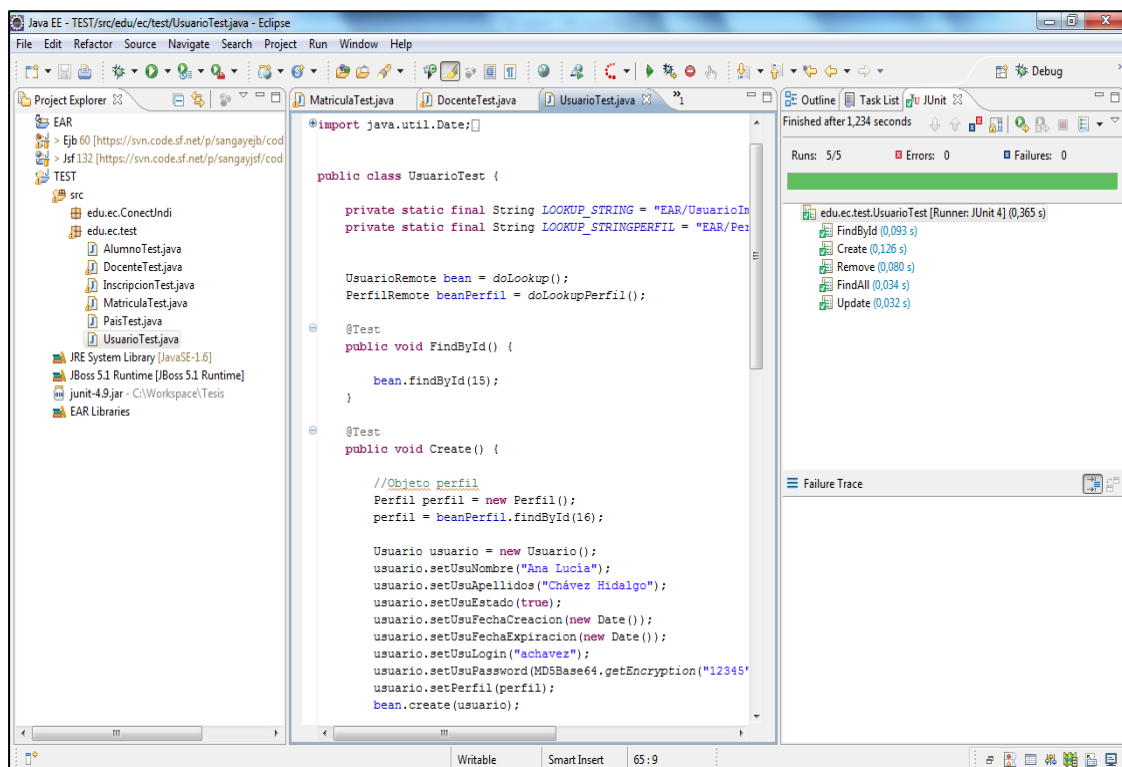


Figura 4.2 Pruebas de la clase Usuario

4.3.1.3.5 Demo de la Iteración 1

En la Figura 4.3 se puede ver el demo de la Iteración 1 que es la página de logueo y la página de administración de usuarios del sistema de control escolar.



Figura 4.3 Demo de la Iteración 1

4.3.2 Análisis y desarrollo del requerimiento de la Iteración 2

En la iteración 2 se continúa con el análisis y desarrollo de las historias de usuario número 2, 3, 4, 5 y 6 que corresponde al ingreso de la información al sistema referente a las configuraciones previas de la organización, alumnos, docentes, horarios y calendarios.

Para el desarrollo de estos requerimientos, se realizará una especificación detallada de cada una de los requerimientos iniciales que se definieron en la Tabla 4.1.

4.3.2.1 ITERACIÓN (SPRINT) 2

Para el desarrollo de la Iteración 2 se fija una duración de 20 días laborables es decir 4 semanas.

4.3.2.2 Pila de Producto Interacción 2

En la Tabla 4.7 se detalla la pila del producto que se definió para la Iteración 2.

4.3.2.3 Sprint Backlog

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la Iteración 2 que engloba las historias de usuarios de ingreso de información al sistema. En la Tabla 4.8 se detalla la pila de actividades para la Iteración 2.

Tabla 4.7 Pila del producto de la Iteración 2

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
3	Ingreso de información del sistema	13	4	2	Entrar a la aplicación, ir a configuraciones previas ingresar datos de la escuela, verificar la información ingresada en el tracking de la pantalla.	Esta información puede ser eliminada y modificada por el administrador.
4	Ingreso de parámetros generales de alumnos en el sistema	12	4	2	Entrar a la aplicación, ir a configuraciones previas, ir a alumnos, ingresar la información solicitada como: tipo de bajas, tipo de asistencia, escala de notas. Verificar la información ingresada en el tracking de cada pantalla.	Esta información puede ser eliminada y modificada por el administrador.
5	Ingreso de parámetros generales de docentes en el sistema	11	3	2	Entrar a la aplicación, ir a configuraciones previas, ir a docente, ingresar la información solicitada como: tipo docente, motivo de entrada y salida. Verificar la información ingresada en el tracking de cada pantalla.	Esta información puede ser eliminada y modificada por el administrador.
6	Ingreso de parámetros generales de calendarios y horarios en el sistema	10	3	2	Entrar a la aplicación, ir a configuraciones previas, ir a calendarios y horarios, ingresar la información solicitada como: ciclo lectivo, periodos, tipo de hora, feriados del año. Verificar la información ingresada en el tracking de cada pantalla.	Esta información puede ser eliminada y modificada por el administrador.

Tabla 4.8 Pila de actividades de la Iteración 2

Sprint		Inicio	Duración (días)	Elemento del Product Backlog	Días	4 - 8 Julio	11 - 15 Julio	18 - 22 Julio	25 - 29 Julio
2		4-Julio-2011	20	Ingreso de información del sistema					
Id	Tarea	Delegado	Estado	Horas	20	20	20	20	
3	Codificación de la persistencia Entity Beans (JPA) para el ingreso de datos de la Organización	A & J	Completado		X				
3	Elaboración de consultas JPQL para el ingreso de datos de la Organización	A & J	Completado		X				
3	Codificación de Interface y Session Bean (CRUD) para el ingreso de datos de la Organización	A & J	Completado		X				
3	Codificación de la lógica del negocio para el ingreso de datos de la Organización	A & J	Completado		X				
3	Configuración de las seguridades con JAAS para el ingreso de datos de la Organización	A & J	Completado		X				
3	Configuración de los JSF y Baking Bean para el ingreso de datos de la Organización	A & J	Completado		X				
3	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Generales	A & J	Completado		X				
3	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Generales	A & J	Completado		X				
3	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Generales	A & J	Completado		X				
3	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Generales	A & J	Completado		X				

Id	Tarea	Delegado	Estado	Horas	20	20	20	20
3	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Generales	A & J	Completado		X			
3	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Generales	A & J	Completado		X			
4	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Alumnos	A & J	Completado			X		
4	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Alumnos	A & J	Completado			X		
4	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Alumnos	A & J	Completado			X		
4	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Alumnos	A & J	Completado			X		
4	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Alumnos	A & J	Completado			X		
4	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Alumnos	A & J	Completado			X		
5	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Docentes	A & J	Completado				X	
5	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Docentes	A & J	Completado				X	
5	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Docentes	A & J	Completado				X	
5	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Docentes	A & J	Completado				X	

Id	Tarea	Delegado	Estado	Horas	20	20	20	20
5	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Docentes	A & J	Completado				X	
5	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Docentes	A & J	Completado				X	
6	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Calendarios y Horarios	A & J	Completado					X
6	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Calendarios y Horarios	A & J	Completado					X
6	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Calendarios y Horarios	A & J	Completado					X
6	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Calendarios y Horarios	A & J	Completado					X
6	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Calendarios y Horarios	A & J	Completado					X
6	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Calendarios y Horarios	A & J	Completado					X

4.3.2.4 Revisión del Sprint

4.3.2.4.1 Tareas completadas

La Tabla 4.9 muestra las tareas que los desarrolladores han terminado hasta el momento de la finalización del Sprint 2.

4.3.2.4.2 Pruebas del Sprint 2

En la Figura 4.4, Figura 4.5, Figura 4.6, se puede ver los resultados exitosos realizados en la Iteración 2.

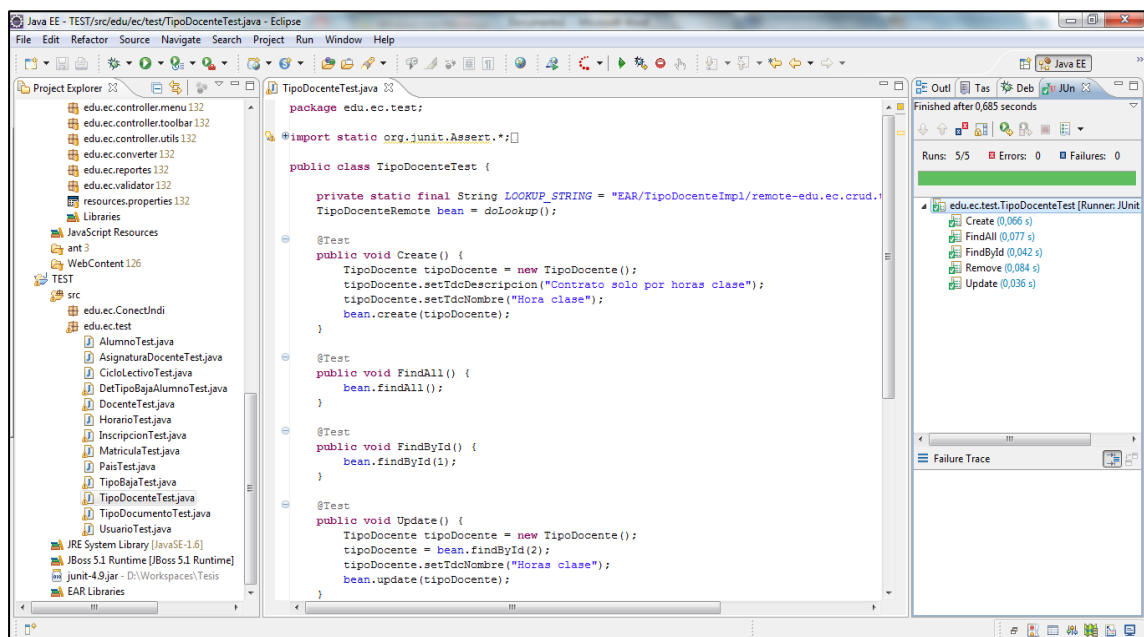


Figura 4.4 Pruebas de la clase tipo docente

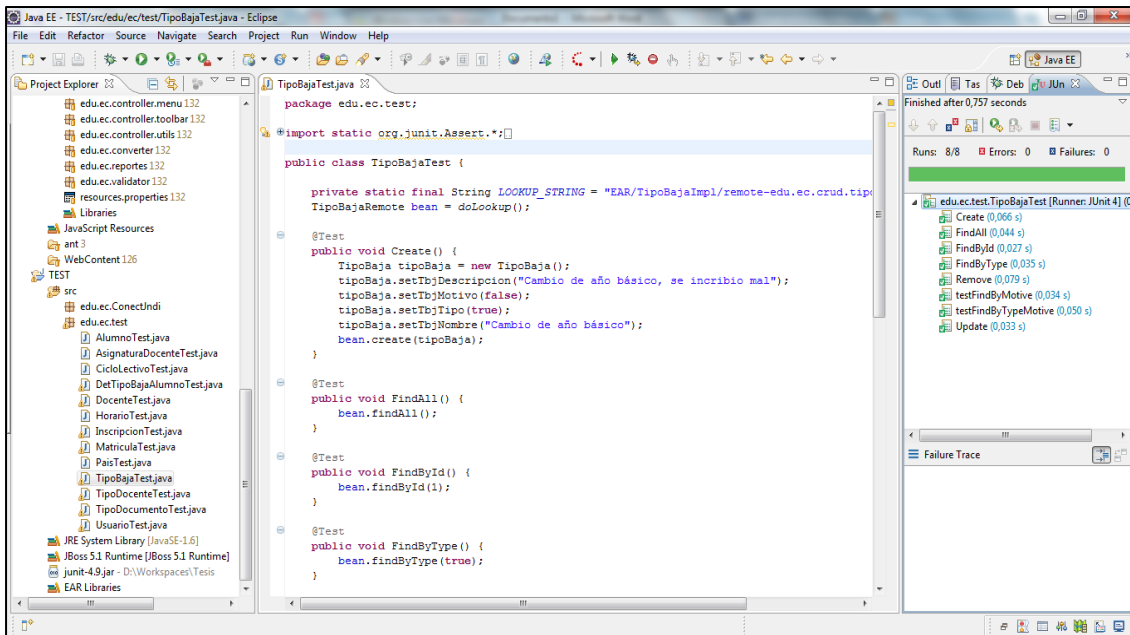


Figura 4.5 Pruebas de la clase tipo baja

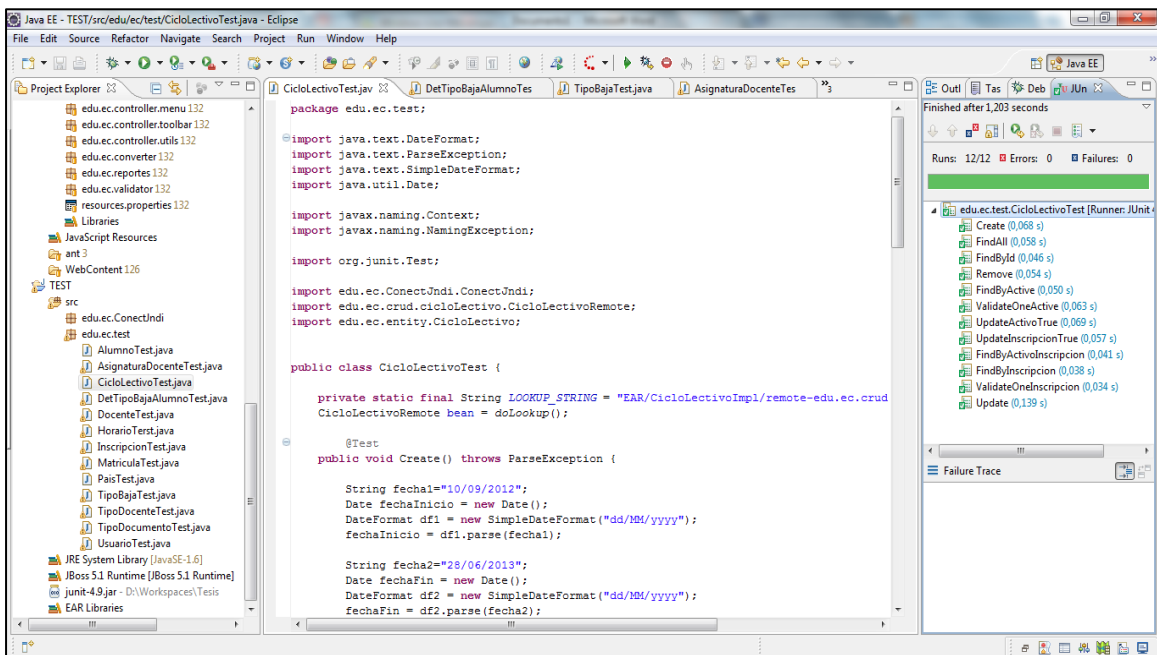


Figura 4.6 Pruebas de la clase ciclo lectivo

4.3.2.4.3 Burn Down Chart

El Figura 4.7 presenta el avance del proyecto hasta el momento de la finalización de la Iteración 2.

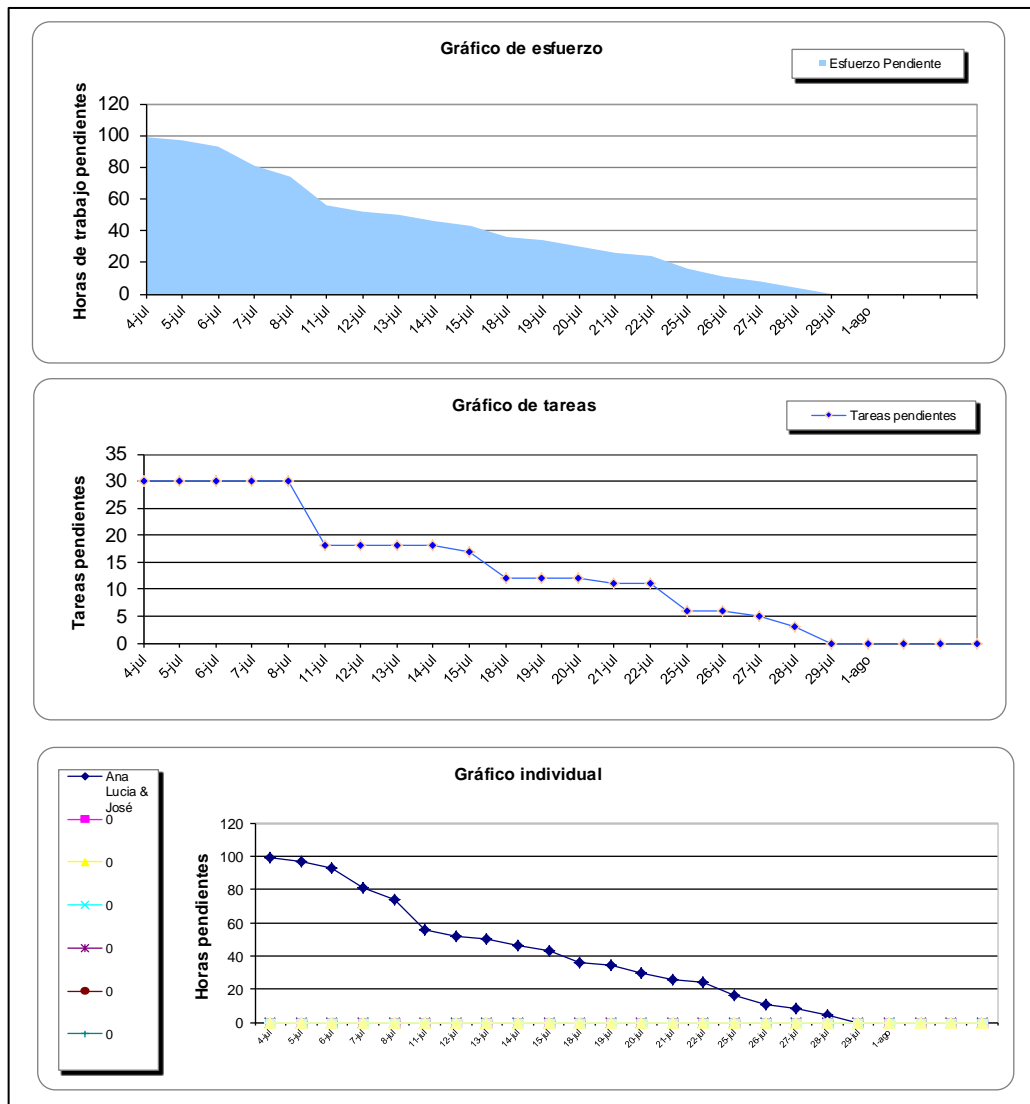


Figura 4.7 Gráfico Burn Down Chart de la Iteración 2

Tabla 4.9 Tabla tareas completadas de la Iteración 2

Tarea asignada a: Ana Lucia Chávez - José Tenorio		Estado Tareas: Completado	Numero de Tareas: 30	
Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
3	Codificación de la persistencia Entity Beans (JPA) para el ingreso de datos de la Organización	Sistema de Control Escolar	2	8 – Julio- 2011
3	Elaboración de consultas JPQL para el ingreso de datos de la Organización	Sistema de Control Escolar	2	8 – Julio - 2011
3	Codificación de Interface y Session Bean (CRUD) para el ingreso de datos de la Organización	Sistema de Control Escolar	2	8 – Julio - 2011
3	Codificación de la lógica del negocio para el ingreso de datos de la Organización	Sistema de Control Escolar	2	8 – Julio - 2011
3	Configuración de las seguridades con JAAS para el ingreso de datos de la Organización	Sistema de Control Escolar	2	8 – Julio - 2011
3	Configuración de los JSF y Baking Bean para el ingreso de datos de la Organización	Sistema de Control Escolar	2	8 – Julio - 2011
3	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Generales	Sistema de Control Escolar	2	8 – Julio - 2011
3	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Generales	Sistema de Control Escolar	2	8 – Julio - 2011
3	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Generales	Sistema de Control Escolar	2	8 – Julio - 2011
3	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Generales	Sistema de Control Escolar	2	8 – Julio - 2011
3	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Generales	Sistema de Control Escolar	2	8 – Julio - 2011

Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
3	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Generales	Sistema de Control Escolar	2	8 – Julio - 2011
4	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Alumnos	Sistema de Control Escolar	2	15 – Julio - 2011
4	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Alumnos	Sistema de Control Escolar	2	15 – Julio - 2011
4	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Alumnos	Sistema de Control Escolar	2	15 – Julio - 2011
4	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Alumnos	Sistema de Control Escolar	2	15 – Julio - 2011
4	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Alumnos	Sistema de Control Escolar	2	15 – Julio - 2011
4	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Alumnos	Sistema de Control Escolar	2	14 – Julio - 2011
5	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Docentes	Sistema de Control Escolar	2	22 – Julio - 2011
5	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Docentes	Sistema de Control Escolar	2	22 – Julio - 2011
5	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Docentes	Sistema de Control Escolar	2	22 – Julio - 2011
5	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Docentes	Sistema de Control Escolar	2	22 – Julio - 2011

Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
5	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Docentes	Sistema de Control Escolar	2	20 – Julio - 2011
5	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Docentes	Sistema de Control Escolar	2	22 – Julio - 2011
6	Codificación de la persistencia Entity Beans (JPA) para el ingreso de las Configuraciones Previas – Calendarios y Horarios	Sistema de Control Escolar	2	28 – Julio - 2011
6	Elaboración de consultas JPQL para el ingreso de las Configuraciones Previas – Calendarios y Horarios	Sistema de Control Escolar	2	27 – Julio - 2011
6	Codificación de Interface y Session Bean (CRUD) para el de las Configuraciones Previas – Calendarios y Horarios	Sistema de Control Escolar	2	28 – Julio - 2011
6	Codificación de la lógica del negocio para el ingreso de las Configuraciones Previas – Calendarios y Horarios	Sistema de Control Escolar	2	27 – Julio - 2011
6	Configuración de las seguridades con JAAS para el ingreso de las Configuraciones Previas – Calendarios y Horarios	Sistema de Control Escolar	2	26 – Julio - 2011
6	Configuración de los JSF y Baking Bean para el ingreso de las Configuraciones Previas – Calendarios y Horarios	Sistema de Control Escolar	2	28 – Julio - 2011

4.3.2.4.4 Demo de la Iteración 2

En la Figura 4.8 se puede ver el demo de la Iteración 2 que es la página administración del sistema de control escolar.

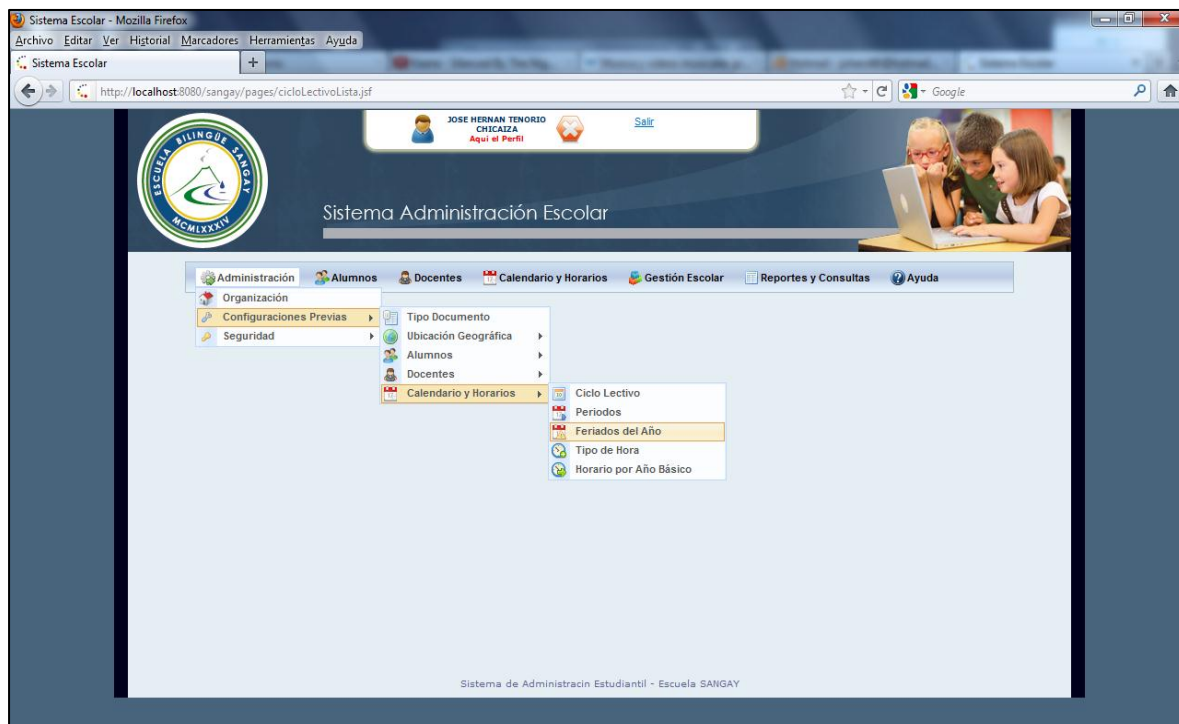


Figura 4.8 Demo de la Iteración 2

4.3.3 Análisis y Desarrollo del requerimiento de la Iteración 3

Para el desarrollo de este requerimiento, se realizará una especificación detallada de cada uno de los requerimientos iniciales que se definieron en la Tabla 4.1.

4.3.3.1 ITERACIÓN (SPRINT) 3

Para el desarrollo de la Iteración 3 se fijó una duración de 15 días laborables es decir 3 semanas.

4.3.3.2 Pila de producto de la Iteración 3

La Tabla 4.10 muestra la pila del producto para la Iteración 3.

4.3.3.3 Sprint Backlog

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la Iteración 3 que engloba las historias de usuarios relacionadas con la administración de estudiantes. En la Tabla 4.11 se detalla la pila de tareas de la Iteración 3.

4.3.3.4 Revisión del Sprint

4.3.3.4.1 Tareas completadas

La Tabla 4.12 muestra las tareas que los desarrolladores ha terminado hasta el momento de la finalización de la Iteración 3.

Tabla 4.10 Pila del producto de la Iteración 3

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
7	Administración Alumnos	9	4	3	Entrar aplicación, ir a docentes, ir a alumnos, ir ingresar nuevo, ingresar información del nuevo alumno, realizar el ingreso de información, verificar en el tracking de la pantalla el nuevo alumno ingresado.	Esta información puede ser eliminada y modificada por el administrador.
8	Inscripción Alumnos	8	4	3	Entrar aplicación, ir a docentes, ir a alumnos, ir ingresar inscripción, ingresar información solicitada, realizar la inscripción, verificar en el tracking de la pantalla el nuevo alumno inscrito.	Esta información puede ser eliminada y modificada por el personal administrativo.
9	Matriculación Alumnos	7	3	3	Entrar a la aplicación, ir a alumnos, ir a ingresar nuevo, ingresar la información del alumno, ingresar alumno, verificar en el tracking el nuevo alumno ingresado. Ir a alumnos, ir a inscripción, ingresar la información del alumno (datos alumno, ciclo lectivo, año básico), realizar inscripción, verificar en el tracking el nuevo alumno inscrito. Ir a gestión escolar, buscar alumno, ingresar información de matriculación, realizar la matricula, verificar en el tracking el nuevo alumno matriculado.	La información de los alumnos y de la inscripción puede ser eliminada y modificada por el administrador. Para el proceso de matriculación solo se puede dar de baja al alumno.

Tabla 4.11 Pila de actividades de la Iteración 3

Sprint 3 Inicio 1-Agosto-2011 Duración (días) 15 Elemento del Product Backlog Matriculación					Días	01 - 05 - Agosto	08 - 12 - Agosto	15 - 19 - Agosto
Id	Tarea	Delegado	Estado	Horas	20	20	20	
7	Codificación de la persistencia Entity Beans (JPA) para ingresar un nuevo alumno	A & J	Completado		X			
7	Elaboración de consultas JPQL para ingresar un nuevo alumno	A & J	Completado		X			
7	Codificación de Interface y Session Bean (CRUD) para ingresar un nuevo alumno	A & J	Completado		X			
7	Codificación de la lógica del negocio para ingresar un nuevo alumno	A & J	Completado		X			
7	Configuración de las seguridades con JAAS para ingresar un nuevo alumno	A & J	Completado		X			
7	Configuración de los JSF y Baking Bean para ingresar un nuevo alumno	A & J	Completado		X			
8	Codificación de la persistencia Entity Beans (JPA) para la inscripción de un alumno	A & J	Completado			X		
8	Elaboración de consultas JPQL para la inscripción de un alumno	A & J	Completado			X		
8	Codificación de Interface y Session Bean (CRUD) para la inscripción de un alumno	A & J	Completado			X		
8	Codificación de la lógica del negocio para la inscripción de un alumno	A & J	Completado			X		

Id	Tarea	Delegado	Estado	Horas	20	20	20
8	Configuración de los JSF y Baking Bean para la inscripción de un alumno	A & J	Completado			X	
9	Codificación de la persistencia Entity Beans (JPA) para realizar el proceso de matriculación	A & J	Completado				X
9	Elaboración de consultas JPQL para realizar el proceso de matriculación	A & J	Completado				X
9	Codificación de Interface y Session Bean (CRUD) para realizar el proceso de matriculación	A & J	Completado				X
9	Codificación de la lógica del negocio para realizar el proceso de matriculación	A & J	Completado				X
9	Configuración de las seguridades con JAAS para realizar el proceso de matriculación	A & J	Completado				X
9	Configuración de los JSF y Baking Bean para realizar el proceso de matriculación	A & J	Completado				X

Tabla 4.12 Tareas completadas de la Iteración 3

Tarea asignada a: Ana Lucia Chávez - José Tenorio				
Estado Tareas: Completado				
Numero de Tareas: 30				
Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
7	Codificación de la persistencia Entity Beans (JPA) para ingresar un nuevo alumno	Sistema de Control Escolar	3	05 – Agosto - 2011
7	Elaboración de consultas JPQL para ingresar un nuevo alumno	Sistema de Control Escolar	3	04 – Agosto – 2011
7	Codificación de Interface y Session Bean (CRUD) para ingresar un nuevo alumno	Sistema de Control Escolar	3	05 – Agosto - 2011
7	Codificación de la lógica del negocio para ingresar un nuevo alumno	Sistema de Control Escolar	3	04 – Agosto – 2011
7	Configuración de las seguridades con JAAS para ingresar un nuevo alumno	Sistema de Control Escolar	3	05 – Agosto – 2011
7	Configuración de los JSF y Baking Bean para ingresar un nuevo alumno	Sistema de Control Escolar	3	05 – Agosto - 2011
8	Codificación de la persistencia Entity Beans (JPA) para la inscripción de un alumno	Sistema de Control Escolar	3	12 – Agosto - 2011
8	Elaboración de consultas JPQL para la inscripción de un alumno	Sistema de Control Escolar	3	11 – Agosto - 2011
8	Codificación de Interface y Session Bean (CRUD) para la inscripción de un alumno	Sistema de Control Escolar	3	10 – Agosto - 2011
8	Codificación de la lógica del negocio para la inscripción de un alumno	Sistema de Control Escolar	3	12 – Agosto - 2011

Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
8	Configuración de las seguridades con JAAS para la inscripción de un alumno	Sistema de Control Escolar	3	12 – Agosto – 2011
8	Configuración de los JSF y Baking Bean para la inscripción de un alumno	Sistema de Control Escolar	3	12 – Agosto - 2011
9	Codificación de la persistencia Entity Beans (JPA) para realizar el proceso de matriculación	Sistema de Control Escolar	3	18 – Agosto - 2011
9	Elaboración de consultas JPQL para realizar el proceso de matriculación	Sistema de Control Escolar	3	18 – Agosto - 2011
9	Codificación de Interface y Session Bean (CRUD) para realizar el proceso de matriculación	Sistema de Control Escolar	3	17 – Agosto - 2011
9	Codificación de la lógica del negocio para realizar el proceso de matriculación	Sistema de Control Escolar	3	17 – Agosto - 2011
9	Configuración de las seguridades con JAAS para realizar el proceso de matriculación	Sistema de Control Escolar	3	18 – Agosto - 2011
9	Configuración de los JSF y Baking Bean para realizar el proceso de matriculación	Sistema de Control Escolar	3	17 – Agosto - 2011

4.3.3.4.2 Burn Down Chart

La Figura 4.9 presenta el avance del proyecto hasta el momento de la finalización de la Iteración 3.

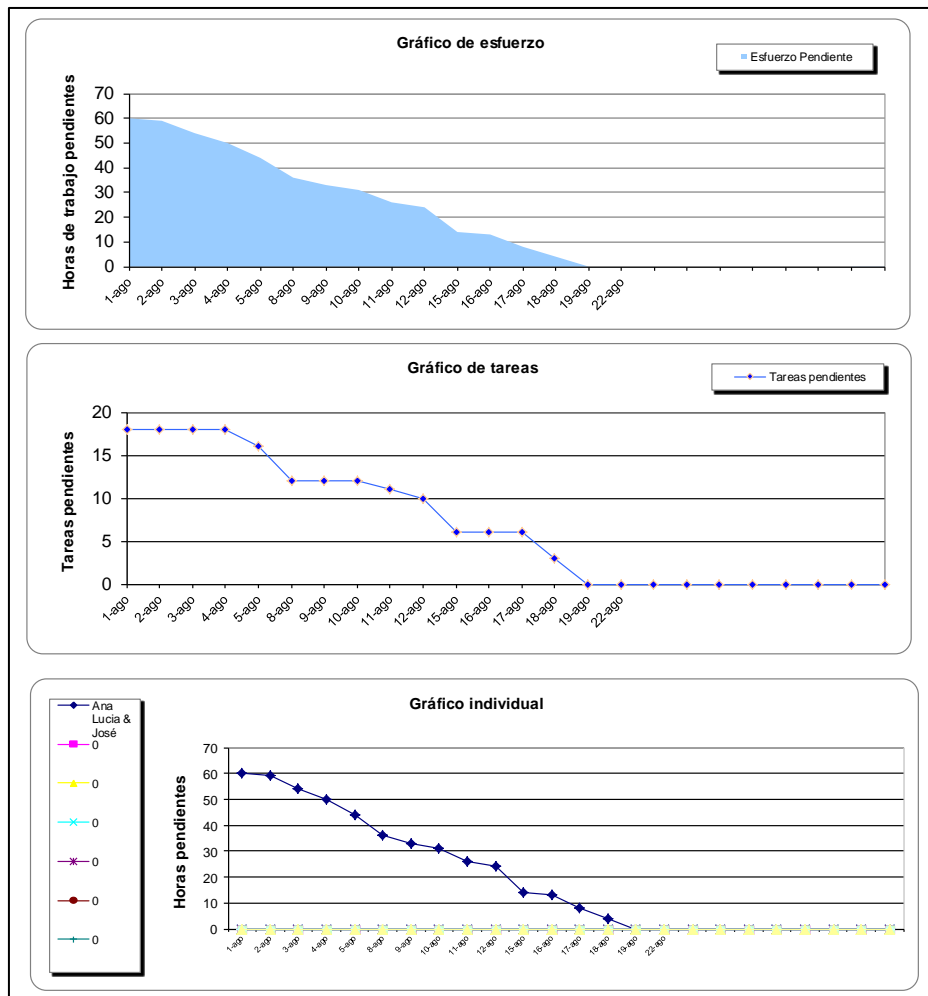


Figura 4.9 Gráfico Burn Down Chart de la Iteración 3

4.3.3.4.3 Pruebas de la Iteración 3

En las Figura 4.10, Figura 4.11, Figura 4.12, se puede ver los resultados exitosos de las clases alumno, inscripción y matriculación correspondientes a la Iteración 3.

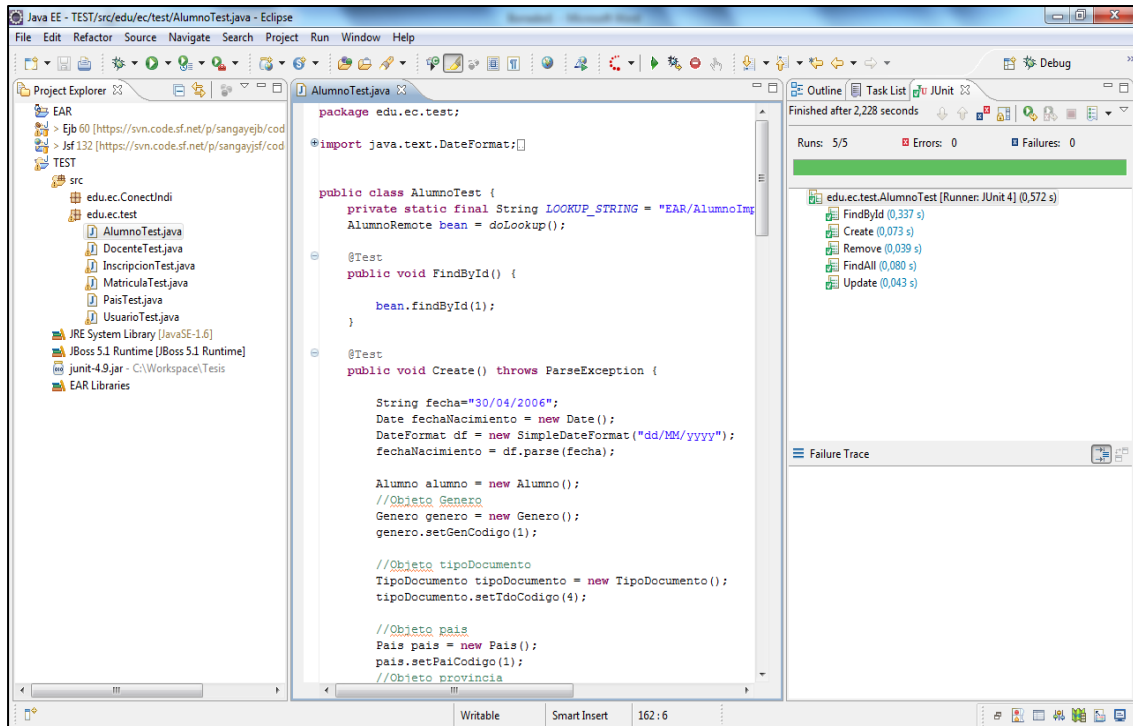


Figura 4.10 Pruebas de la clase Alumno

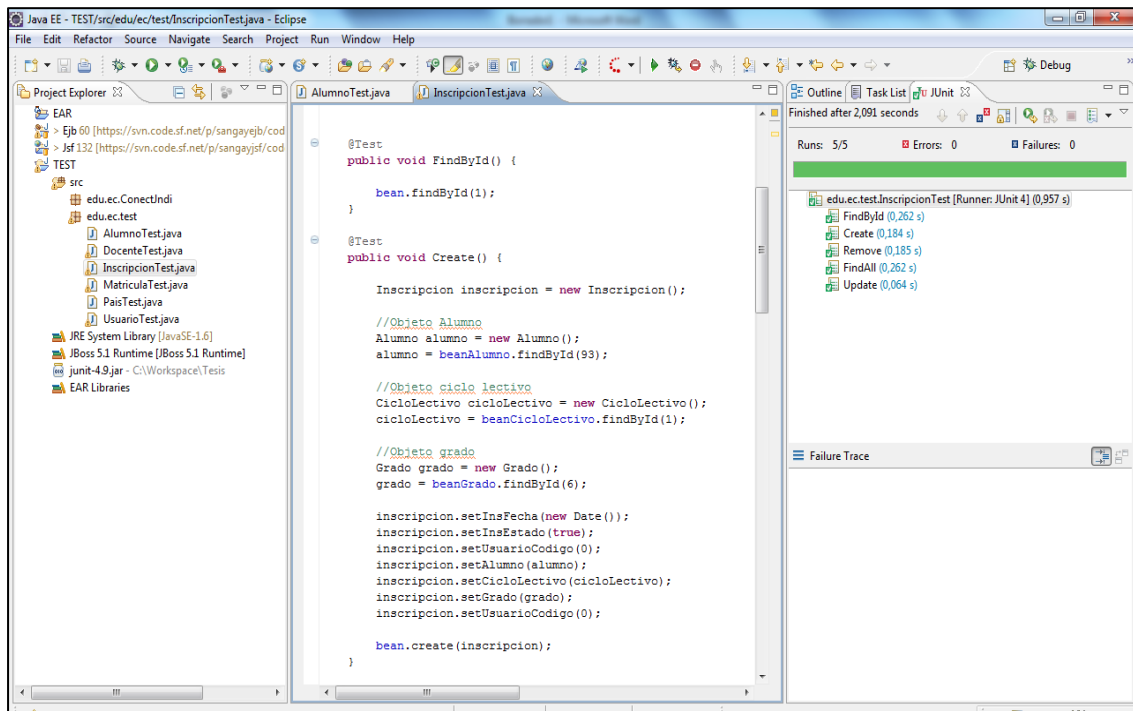


Figura 4.11 Pruebas de la clase inscripción.

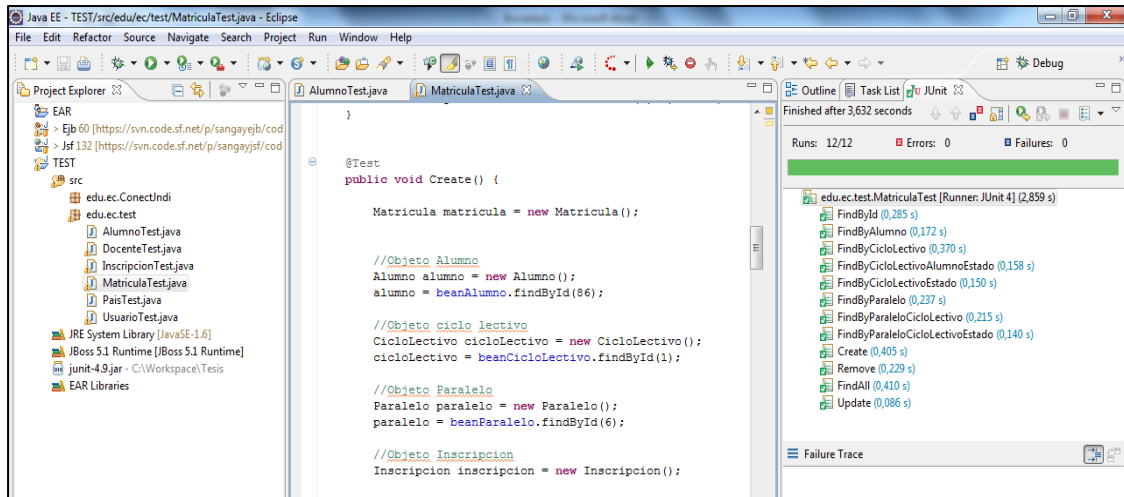


Figura 4.12 Pruebas usando de la clase matriculación.

4.3.3.4 Demo de la Iteración 3

En la Figura 4.13 se ve el demo de la Iteración 3 que es la página de administración de los alumnos del sistema de control escolar.

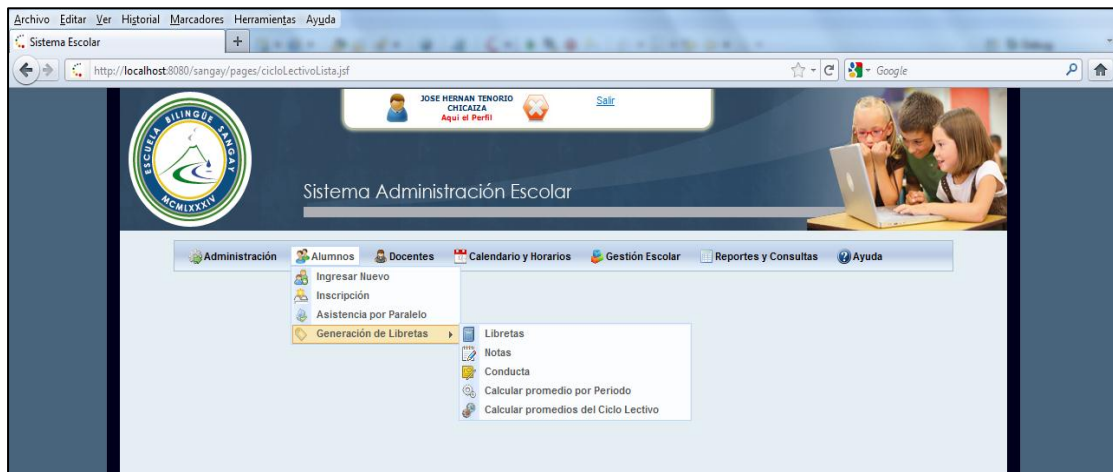


Figura 4.13 Demo dela Iteración 3

4.3.4 Análisis y Desarrollo del requerimiento de la Iteración 4

Para el desarrollo de este requerimiento, se realizará una especificación detallada de cada uno de los requerimientos iniciales que se definieron en la Tabla 4.1.

4.3.4.1 ITERACIÓN (SPRINT) 4

Para el desarrollo de la Iteración 4 se fijó una duración de 5 días laborables es decir 1 semana.

4.3.4.2 Pila de producto de la Iteración 4

La Tabla 4.13 muestra la pila del producto para la Iteración 4.

4.3.4.3 Sprint Backlog

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la Iteración 4 que engloba la historia de usuario administración de docentes. En la Tabla 4.14 se puede ver la pila de tareas para la Iteración 4.

4.3.4.4 Revisión del Sprint

4.3.4.4.1 Tareas completadas

La Tabla 4.15 muestra las tareas que los desarrolladores ha terminado hasta el momento de la finalización de la Iteración 4.

Tabla 4.13 Pila del producto de la Iteración 4

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
10	Administración Docentes	6	4	4	Entrar aplicación, ir a docentes, ir a administrar, ingresar información del docente, realizar el ingreso de información, verificar en el tracking el nuevo docente ingresado.	Esta información puede ser eliminada y modificada por el administrador. Para la generación de horarios se debe relacionar a los docentes con cada materia que estos imparten.

Tabla 4.14 Pila de requerimientos de la Iteración 4

				Días	22 - 26 - Agosto
Sprint	Inicio	Duración (días)	Elemento del Product Backlog		
4	22-Agosto-2011	5	Administración Docentes		
Id	Tarea	Delegado	Estado	Horas	20
10	Codificación de la persistencia Entity Beans (JPA) para administrar Docentes	A & J	Completado		X
10	Elaboración de consultas JPQL para administrar Docentes	A & J	Completado		X
10	Codificación de Interface y Session Bean (CRUD) para administrar Docentes	A & J	Completado		X
10	Codificación de la lógica del negocio para administrar Docentes	A & J	Completado		X
10	Configuración de las seguridades con JAAS para administrar Docentes	A & J	Completado		X
10	Configuración de los JSF y Baking Bean para administrar Docentes	A & J	Completado		X

Tabla 4.15 Tareas completadas de la Iteración 4

Tarea asignada a: Ana Lucia Chávez - José Tenorio		Estado Tareas: Completado	Numero de Tareas: 6	
Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
10	Codificación de la persistencia Entity Beans (JPA) para administrar Docentes	Sistema de Control Escolar	4	25 – Agosto - 2011
10	Elaboración de consultas JPQL para administrar Docentes	Sistema de Control Escolar	4	25 – Agosto – 2011
10	Codificación de Interface y Session Bean (CRUD) para administrar Docentes	Sistema de Control Escolar	4	25 – Agosto - 2011
10	Codificación de la lógica del negocio para administrar Docentes	Sistema de Control Escolar	4	24 – Agosto – 2011
10	Configuración de las seguridades con JAAS para administrar Docentes	Sistema de Control Escolar	4	23 – Agosto – 2011
10	Configuración de los JSF y Baking Bean para administrar Docentes	Sistema de Control Escolar	4	24 – Agosto - 2011

4.3.4.4.2 Burn Down Chart

La Figura 4.14 presenta el avance del proyecto hasta el momento de la finalización de la Iteración 4.

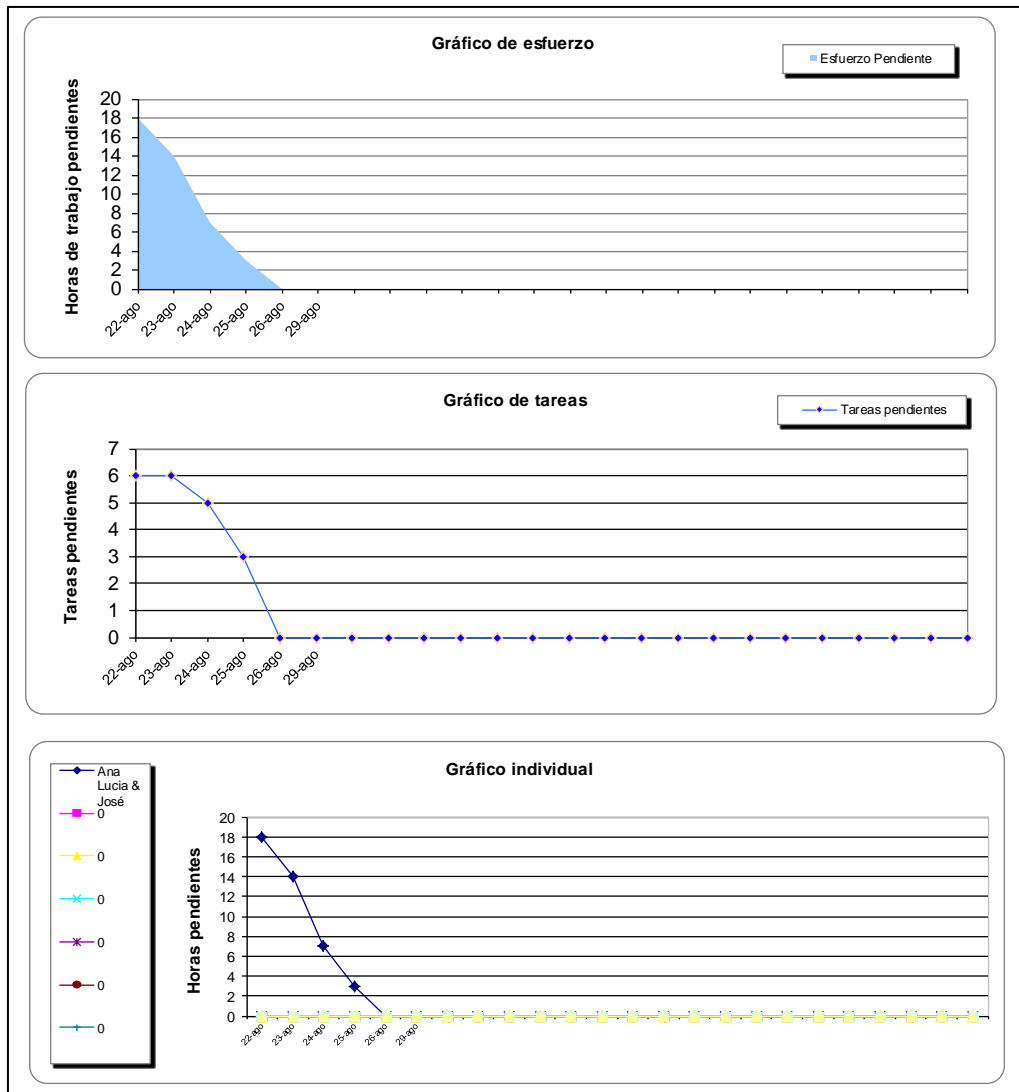


Figura 4.14 Gráfico Burn Down Chart de la Iteración 4

4.3.4.4.3 Pruebas de la Iteración 4.

En las Figuras 4.15, 4.16, se puede ver los resultados exitosos de la Iteración

4.

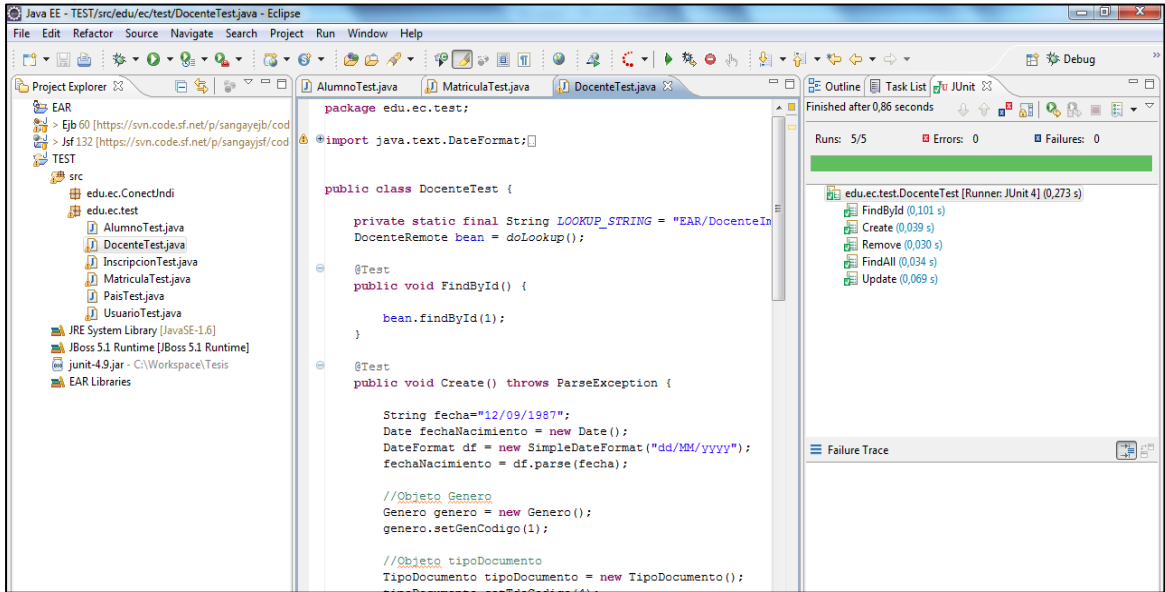


Figura 4.15 Pruebas de la clase docente.

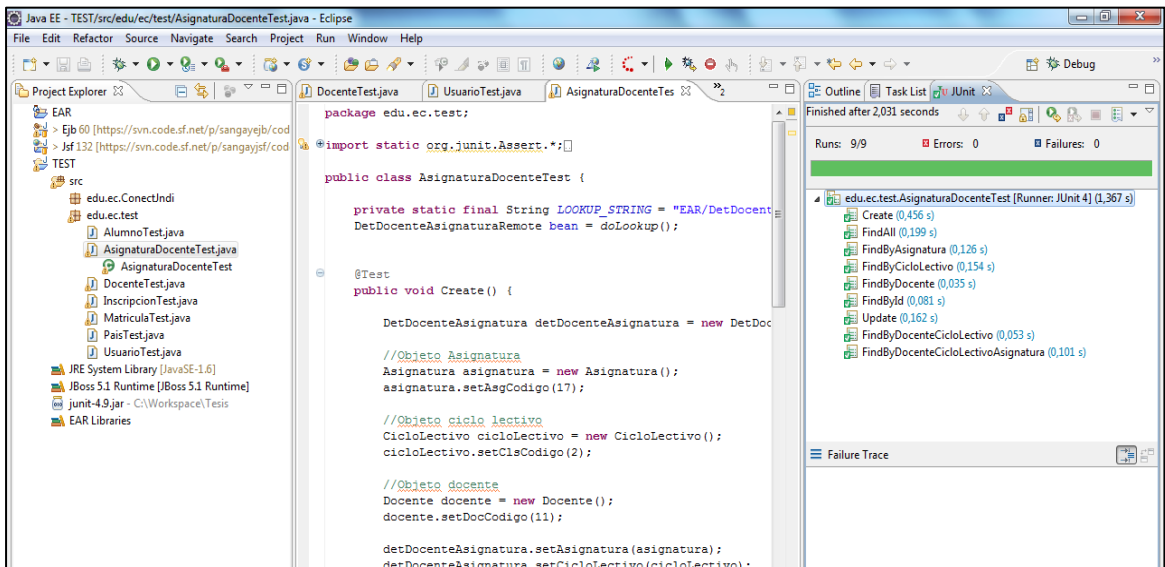


Figura 4.16 Pruebas de la clase asignar asignatura docente.

4.3.4.4 Demo de la Iteración 4

En la Figura 4.17 se puede ver el demo de la Iteración 4; que es la página de administración de docentes del sistema de control escolar.

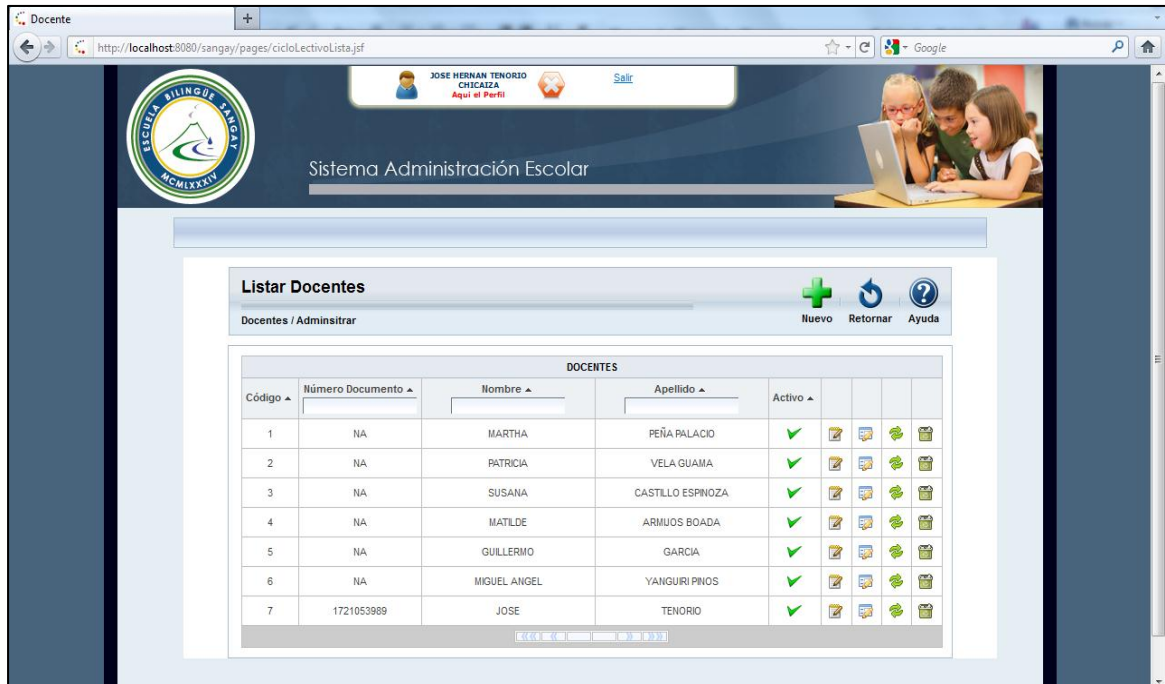


Figura 4.17 Demo de la Iteración 4

4.3.5 Análisis y Desarrollo del requerimiento de la Iteración 5

Para el desarrollo de este requerimiento, se realizará una especificación detallada de cada uno de los requerimientos iniciales que se definieron en la Tabla 4.16.

4.3.5.1 ITERACIÓN (SPRINT) 5

Para el desarrollo de la Iteración 5 se fija una duración de 20 días laborables es decir 4 semanas.

4.3.5.2 Pila de producto de la Iteración 5

La Tabla 4.16 muestra la pila del producto para la Iteración 5.

4.3.5.3 Sprint Backlog

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la Iteración 5 que engloba las historias de usuarios para la gestión escolar. A continuación en la Tabla 4.17 se puede ver la pila de tareas de la Iteración 5.

4.3.5.4 Revisión del Sprint

4.3.5.4.1 Tareas completadas

La Tabla 4.18 muestra las tareas que los desarrolladores ha terminado hasta el momento de la finalización de la Iteración 5.

Tabla 4.16 Pila del producto de la Iteración 5

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
11	Gestión Escolar	5	4	5	Entrar a la aplicación, ir a configuraciones previas ingresar datos de la escuela, verificar la información en el tracking	Esta información puede ser eliminada y modificada por el administrador.
12	Ingreso de asistencia alumnos	4	4	5	Entrar a la aplicación, ir a alumnos, buscar al alumno, ingresar la asistencia del estudiante, verificar la información en el tracking de la pantalla.	Esta información puede ser eliminada y modificada por el docente.
13	Generar horarios escolares	3	3	5	Entrar a la aplicación, ir a horarios y calendarios, ir a horario escolar, ingresar la información que se solicita, ingresar el nuevo horario escolar, verificar la información en el tracking de la pantalla.	Esta información puede ser eliminada y modificada por el personal administrador.
14	Generación de libretas	2	4	5	Entrar a la aplicación, ir a alumnos, ir a generación de libretas, ir a libretas, seleccionar al estudiante que se desea generar la libreta escolar, verificar la libreta de calificaciones en otra pantalla en formato PDF.	Todo reporte se mostrará en formato PDF.

Tabla 4.17 Pila de requerimiento de la Iteración 5

				Días	29 - 02 - Septiembre	05 - 09 - Septiembre	12 - 16 - Septiembre	19 - 23 - Septiembre
Sprint	Inicio	Duración (días)	Elemento del Product Backlog					
5	29-Agosto-2011	20	Gestión Escolar					
Id	Tarea	Delegado	Estado	Horas	20	20	20	20
11	Codificación de la persistencia Entity Beans (JPA) para Ingreso Asistencia por Paralelo	A & J	Completado		X			
11	Elaboración de consultas JPQL para Ingreso Asistencia por Paralelo	A & J	Completado		X			
11	Codificación de Interface y Session Bean (CRUD) para Ingreso Asistencia por Paralelo	A & J	Completado		X			
11	Codificación de la lógica del negocio para Ingreso Asistencia por Paralelo	A & J	Completado		X			
11	Configuración de las seguridades con JAAS para Ingreso Asistencia por Paralelo	A & J	Completado		X			
11	Configuración de los JSF y Baking Bean para Ingreso Asistencia por Paralelo	A & J	Completado		X			
12	Codificación de la persistencia Entity Beans (JPA) para la generación de horarios escolares	A & J	Completado		X			
12	Elaboración de consultas JPQL para la generación de horarios escolares	A & J	Completado		X			

Id	Tarea	Delegado	Estado	Horas	20	20	20	20
12	Codificación de Interface y Session Bean (CRUD) para la generación de horarios escolares	A & J	Completado		X			
12	Codificación de la lógica del negocio para la generación de horarios escolares	A & J	Completado		X			
12	Configuración de las seguridades con JAAS para la generación de horarios escolares	A & J	Completado		X			
12	Configuración de los JSF y Baking Bean para la generación de horarios escolares	A & J	Completado		X			
13	Codificación de la persistencia Entity Beans (JPA) para realizar la Generación de Libretas	A & J	Completado			X		
13	Elaboración de consultas JPQL para realizar la Generación de Libretas	A & J	Completado			X		
13	Codificación de Interface y Session Bean (CRUD) para la Generación de Libretas	A & J	Completado			X		
13	Codificación de la lógica del negocio para realizar la Generación de Libretas	A & J	Completado			X		
13	Configuración de las seguridades con JAAS para realizar la Generación de Libretas	A & J	Completado			X		
13	Configuración de los JSF y Baking Bean para realizar la Generación de Libretas	A & J	Completado			X		

Id	Tarea	Delegado	Estado	Horas	20	20	20	20
14	Codificación de la persistencia Entity Beans (JPA) para realizar el proceso de Agrupación en grados y paralelos	A & J	Completado				X	
14	Elaboración de consultas JPQL para realizar el proceso de Agrupación en grados y paralelos	A & J	Completado				X	
14	Codificación de Interface y Session Bean (CRUD) para realizar el proceso de Agrupación en grados y paralelos	A & J	Completado				X	
14	Codificación de la lógica del negocio para realizar el proceso de Agrupación en grados y paralelos	A & J	Completado					X
14	Configuración de las seguridades con JAAS para realizar el proceso de Agrupación en grados y paralelos	A & J	Completado					X
14	Configuración de los JSF y Baking Bean para realizar el proceso de Agrupación en grados y paralelos	A & J	Completado					X

Tabla 4.18 Tareas completadas de la Iteración 5

Tarea asignada a: Ana Lucia Chávez - José Tenorio Estado Tareas: Completado Numero de Tareas: 24				
Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
11	Codificación de la persistencia Entity Beans (JPA) para Ingreso Asistencia por Paralelo	Sistema de Control Escolar	5	02 – Agosto-2011
11	Elaboración de consultas JPQL para Ingreso Asistencia por Paralelo	Sistema de Control Escolar	5	01 – Agosto-2011
11	Codificación de Interface y Session Bean (CRUD) para Ingreso Asistencia por Paralelo	Sistema de Control Escolar	5	01 – Agosto-2011
11	Codificación de la lógica del negocio para Ingreso Asistencia por Paralelo	Sistema de Control Escolar	5	02 – Agosto-2011
11	Configuración de las seguridades con JAAS para Ingreso Asistencia por Paralelo	Sistema de Control Escolar	5	01 – Agosto-2011
11	Configuración de los JSF y Baking Bean para Ingreso Asistencia por Paralelo	Sistema de Control Escolar	5	31 – Agosto-2011

Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
12	Codificación de la persistencia Entity Beans (JPA) para la generación de horarios escolares	Sistema de Control Escolar	5	02 – Agosto-2011
12	Elaboración de consultas JPQL para la generación de horarios escolares	Sistema de Control Escolar	5	01 – Agosto-2011
12	Codificación de Interface y Session Bean (CRUD) para la generación de horarios escolares	Sistema de Control Escolar	5	01 – Agosto-2011
12	Codificación de la lógica del negocio para la generación de horarios escolares	Sistema de Control Escolar	5	01 – Septiembre - 2011
12	Configuración de las seguridades con JAAS para la generación de horarios escolares	Sistema de Control Escolar	5	02 – Septiembre - 2011
12	Configuración de los JSF y Baking Bean para la generación de horarios escolares	Sistema de Control Escolar	5	02 – Septiembre - 2011
13	Codificación de la persistencia Entity Beans (JPA) para realizar la Generación de Libretas	Sistema de Control Escolar	5	09 – Septiembre - 2011
13	Elaboración de consultas JPQL para realizar la Generación de Libretas	Sistema de Control Escolar	5	09 – Septiembre - 2011

Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
13	Codificación de Interface y Session Bean (CRUD) para la Generación de Libretas	Sistema de Control Escolar	5	08 – Septiembre - 2011
13	Codificación de la lógica del negocio para realizar la Generación de Libretas	Sistema de Control Escolar	5	08 – Septiembre - 2011
13	Configuración de las seguridades con JAAS para realizar la Generación de Libretas	Sistema de Control Escolar	5	07 – Septiembre - 2011
13	Configuración de los JSF y Baking Bean para realizar la Generación de Libretas	Sistema de Control Escolar	5	09 – Septiembre - 2011
14	Codificación de la persistencia Entity Beans (JPA) para realizar el proceso de Agrupación en grados y paralelos	Sistema de Control Escolar	5	16 – Septiembre - 2011
14	Elaboración de consultas JPQL para realizar el proceso de Agrupación en grados y paralelos	Sistema de Control Escolar	5	16 – Septiembre - 2011
14	Codificación de Interface y Session Bean (CRUD) para realizar el proceso de Agrupación en grados y paralelos	Sistema de Control Escolar	5	15 – Septiembre - 2011
14	Codificación de la lógica del negocio para realizar el proceso de Agrupación en grados y paralelos	Sistema de Control Escolar	5	21 – Septiembre - 2011

Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
14	Configuración de las seguridades con JAAS para realizar el proceso de Agrupación en grados y paralelos	Sistema de Control Escolar	5	22 – Septiembre - 2011
14	Configuración de los JSF y Baking Bean para realizar el proceso de Agrupación en grados y paralelos	Sistema de Control Escolar	5	21 – Septiembre - 2011

4.3.5.4.2 Burn Down Chart

La Figura 4.18 presenta el avance del proyecto hasta el momento de la finalización de la Iteración 5.

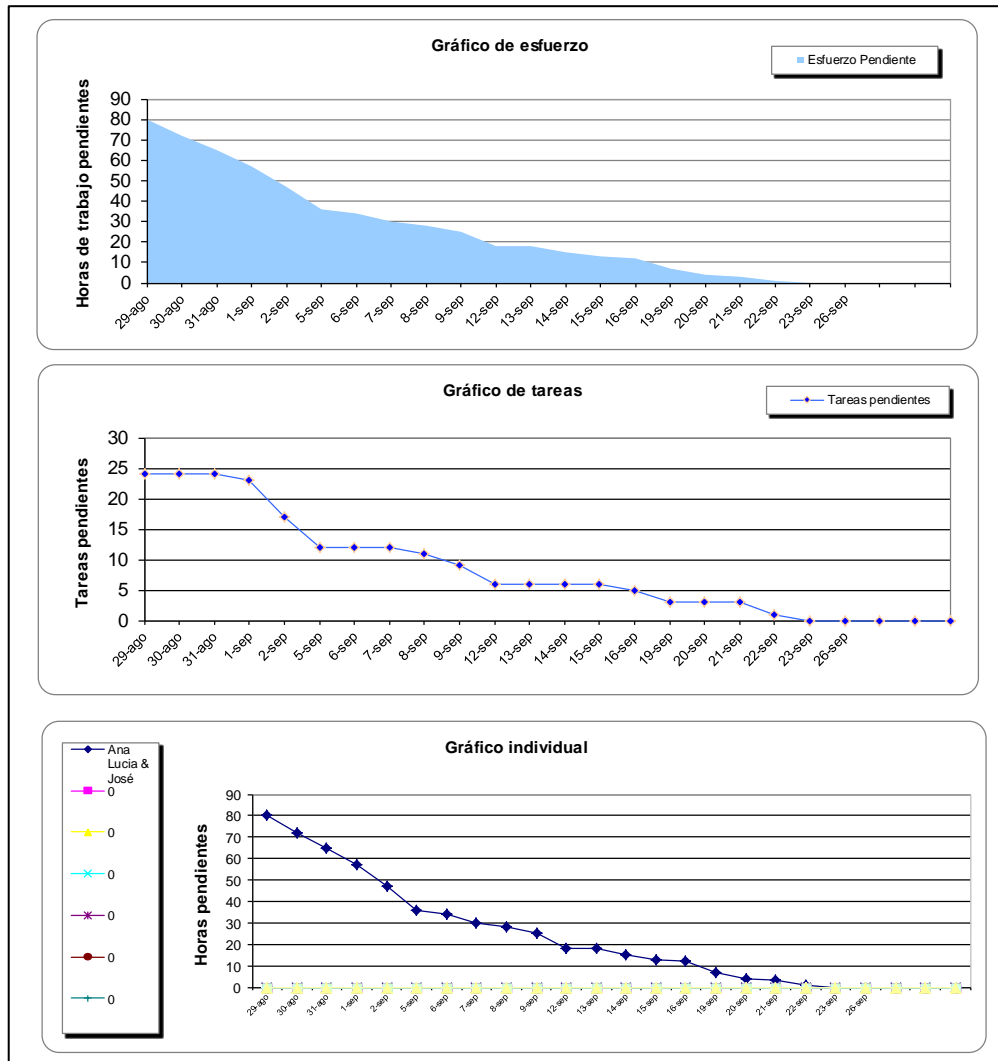


Figura 4.18 Gráfico Burn Down Chart del la Iteración 5

4.3.5.4.3 Pruebas de la Iteración 5

En las Figura 4.19, se puede ver los resultados exitosos de la Iteración 5.

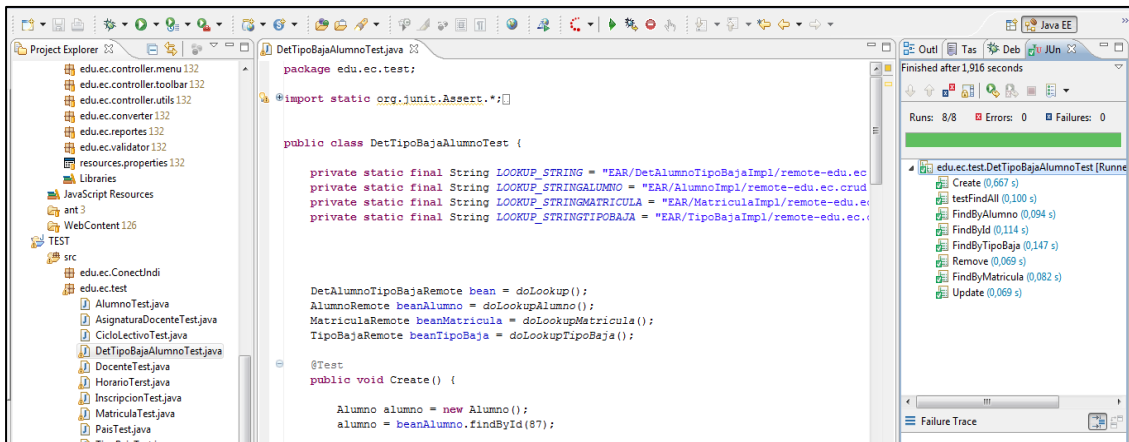


Figura 4.19 Pruebas de la clase horarios

4.3.5.4.4 Demo de la Iteración 5

En la Figura 4.20 se puede ver el demo de la Iteración 5 que es la página de administración de calendarios y horarios del sistema de control escolar.

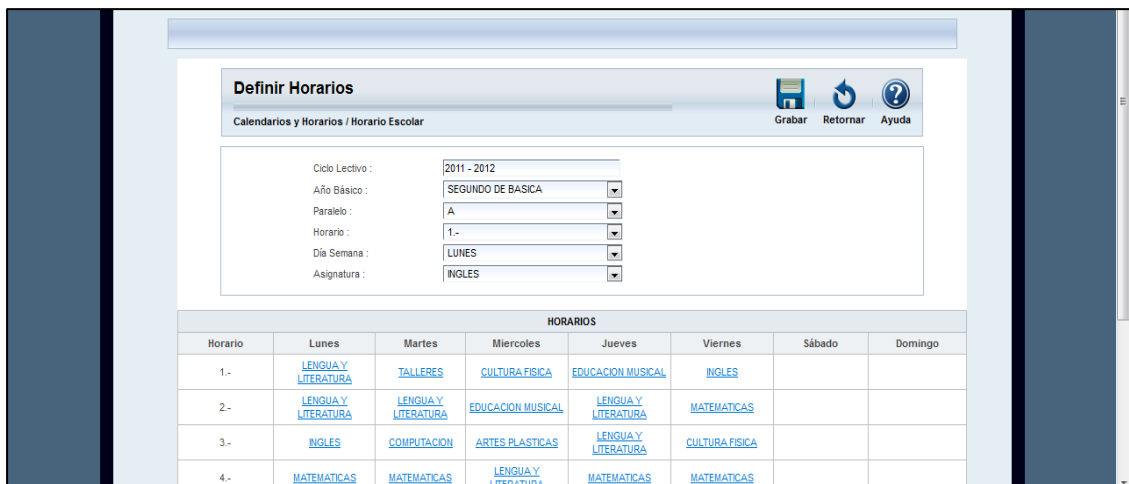


Figura 4.20 Demo del la Iteración 5

4.3.6 Análisis y Desarrollo del requerimiento de la Iteración 6

Para el desarrollo de este requerimiento, se realizará una especificación detallada de cada uno de los requerimientos iniciales que se definieron en la Tabla 4.1.

4.3.6.1 ITERACIÓN (SPRINT) 6

Para el desarrollo del último sprint se fijó una duración de 15 días laborables es decir 3 semanas.

4.3.6.2 Pila de Producto Interacción 6

La Tabla 4.19 muestra la pila de producto para la Iteración 6.

4.3.6.3 Sprint Backlog

En esta fase se detallan las tareas y sub-tareas contenidas dentro de la Iteración 6 que engloba la historia de usuario reportes y consultas. En la Tabla 4.20 la pila de actividades de la Iteración 6.

4.3.6.4 Revisión del Sprint

4.3.6.4.1 Tareas completadas

La Tabla 4.21 muestra las tareas que los desarrolladores ha terminado hasta el momento de la finalización de la Iteración 6.

Tabla 4.19 Pila del producto de la Iteración 6

PILA DE PRODUCTO INICIAL						
ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
15	Reportes y Consultas	1	4	6	Entrar a la aplicación, ir a reportes y consultas, seleccionar el tipo de reporte, verificar el reporte en el archivo PDF.	Todo reporte se mostrará en formato PDF.

Tabla 4.20 Pila de actividades de la Iteración 6

Sprint 6		Inicio 26-Septiembre-2011	Duración (días) 15	Elemento del Product Backlog Reportes y Consultas	Días	26 - 30 - Sept.	03 - 07- Oct.	10 - 14 - Oct.
Id	Tarea	Delegado	Estado	Horas	20	20	20	
15	Realizar los Query necesarios para generar los reportes de inscripción y matricula	A & J	Completado		X			
15	Ingresar los Query en JasperReport para generar los reportes de inscripción y matricula	A & J	Completado		X			
15	Realizar los Query necesarios para generar los reportes de horarios	A & J	Completado		X			
15	Ingresar los Query en JasperReport para generar los reportes de horarios	A & J	Completado		X			
15	Realizar los Query necesarios para generar los reportes de lista docentes	A & J	Completado			X		
15	Ingresar los Query en JasperReport para generar los reportes de lista docentes	A & J	Completado			X		
15	Realizar los Query necesarios para generar los reportes de lista alumnos	A & J	Completado			X		
15	Ingresar los Query en JasperReport para generar los reportes de lista alumnos	A & J	Completado			X		
15	Realizar los Query necesarios para generar los reportes de faltas, atrasos y disciplina	A & J	Completado				X	
15	Ingresar los Query en JasperReport para generar los reportes de faltas, atrasos y disciplina	A & J	Completado				X	
15	Realizar los Query necesarios para generar los reportes de record Estudiantil	A & J	Completado				X	
15	Ingresar los Query en JasperReport para generar los reportes de record Estudiantil	A & J	Completado				X	

Tabla 4.21 Tareas completadas de la Iteración 6

Tarea asignada a: Ana Lucia Chávez - José Tenorio		Estado Tareas: Completado		Numero de Tareas: 12
Id	Nombre Tarea	Nombre Proyecto	Interacción	Fecha Modificación
15	Realizar los Query necesarios para generar los reportes de inscripción y matricula	Sistema de Control Escolar	6	30 Septiembre 2011
15	Ingresar los Query en JasperReport para generar los reportes de inscripción y matricula	Sistema de Control Escolar	6	29 Septiembre 2011
15	Realizar los Query necesarios para generar los reportes de horarios	Sistema de Control Escolar	6	29 Septiembre 2011
15	Ingresar los Query en JasperReport para generar los reportes de horarios	Sistema de Control Escolar	6	30 Septiembre 2011
15	Realizar los Query necesarios para generar los reportes de lista docentes	Sistema de Control Escolar	6	07 Octubre 2011
15	Ingresar los Query en JasperReport para generar los reportes de lista docentes	Sistema de Control Escolar	6	06 Octubre 2011
15	Realizar los Query necesarios para generar los reportes de lista alumnos	Sistema de Control Escolar	6	07 Octubre 2011
15	Ingresar los Query en JasperReport para generar los reportes de lista alumnos	Sistema de Control Escolar	6	07 Octubre 2011
15	Realizar los Query necesarios para generar los reportes de faltas, atrasos y disciplina	Sistema de Control Escolar	6	13 Octubre 2011
15	Ingresar los Query en JasperReport para generar los reportes de faltas, atrasos y disciplina	Sistema de Control Escolar	6	13 Octubre 2011
15	Realizar los Query necesarios para generar los reportes de record Estudiantil	Sistema de Control Escolar	6	12 Octubre 2011
15	Ingresar los Query en JasperReport para generar el record académico	Sistema de Control Escolar	6	13 Octubre 2011

4.3.6.4.2 Burn Down Chart

La Figura 4.21 presenta el avance del proyecto hasta el momento de la finalización de la Iteración 6.

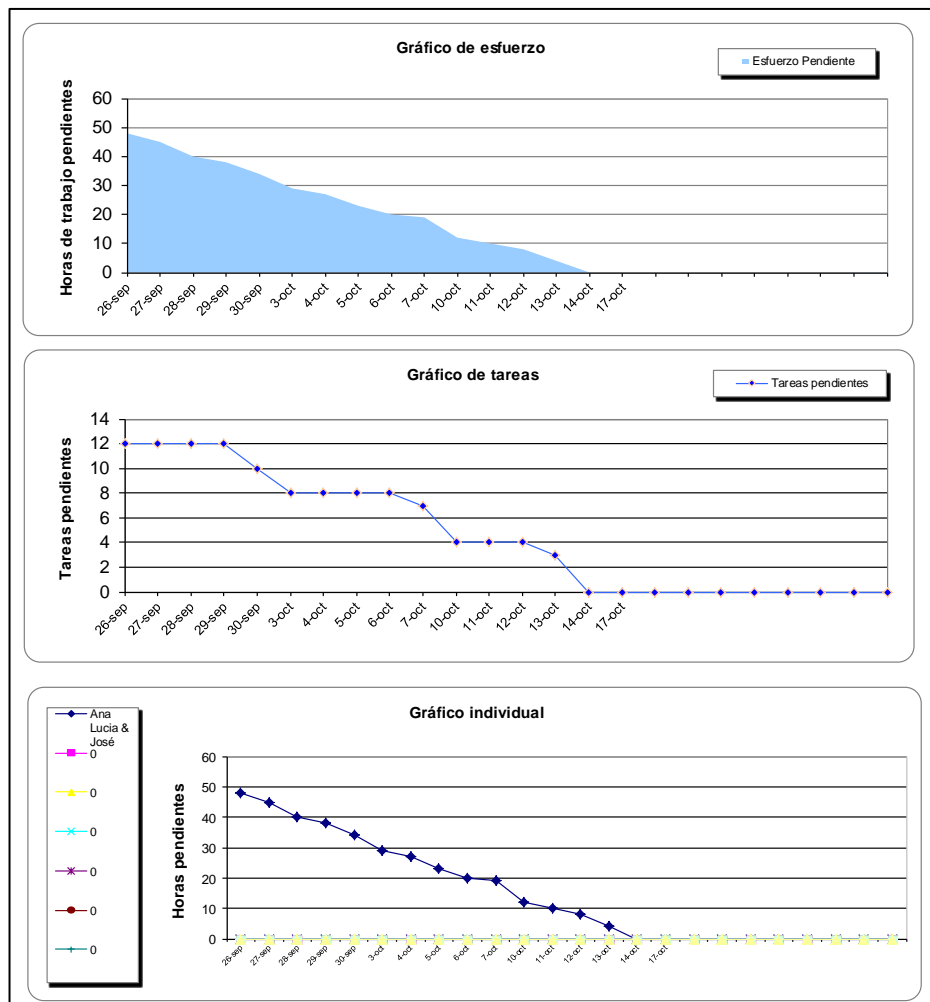


Figura 4.21 Gráfico Burn Down Chart de la Iteración 6

4.3.6.4.3 Demo de la Iteración 6

En la Figura 4.22 se puede ver el demo de la Iteración 6, que es el módulo de generación de reportes del sistema de control escolar.

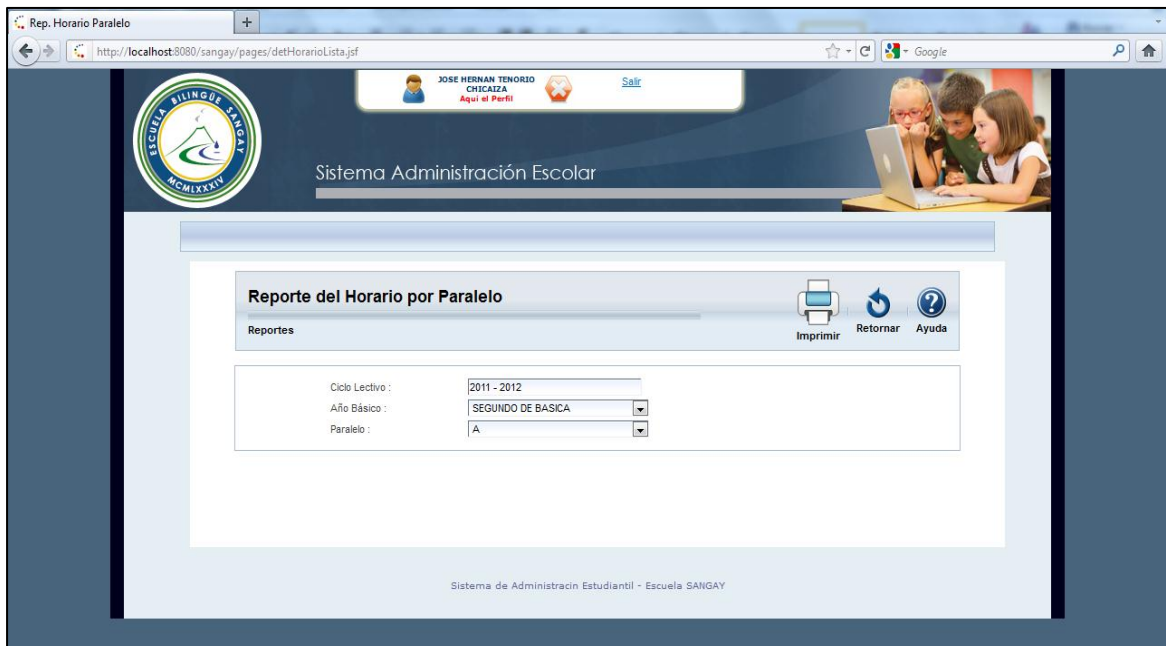


Figura 4.22 Demo de la Iteración 6

Al ser este el último sprint y al haber llevado a cabalidad el desarrollo de la plataforma, no existen tareas sin culminar en el proceso de desarrollo, aunque cabe recalcar que al ser este un prototipo y un sistema en crecimiento constante las tareas seguirán apareciendo de acuerdo con el incremento de las necesidades del product owner a lo largo de la vida útil del sistema, estas deberán ser agregadas al final product backlog para ser lanzadas en una nueva versión.

4.3.7 Product Backlog Final

Al finalizar el proceso de desarrollo se obtiene el listado de tareas basadas en las historias de usuario pero ya delimitadas, asignadas y terminadas como se puede ver en la Tabla 4.22.

Tabla 4.22 Pila del producto final

Pila de Producto Final					
ID	Nombre	Importancia	Estimación	Estado	Sprint
1	Ingreso al sistema	15	4	Completado	1
2	Creación de usuario en el sistema	14	4	Completado	1
3	Ingreso de información del sistema	13	4	Completado	2
4	Ingreso de parámetros generales de alumnos en el sistema	12	4	Completado	2
5	Ingreso de parámetros generales de docentes en el sistema	11	3	Completado	2
6	Ingreso de parámetros generales de calendarios y horarios en el sistema	10	3	Completado	2
7	Administración Alumnos	9	4	Completado	3
8	Inscripción Alumnos	8	4	Completado	3
9	Matriculación Alumnos	7	3	Completado	3
10	Administración Docentes	6	4	Completado	4
11	Gestión Escolar	5	4	Completado	5
12	Ingreso de asistencia alumnos	4	4	Completado	5
13	Generar horarios escolares	3	3	Completado	5
14	Generación de libretas	2	4	Completado	5
15	Reportes y Consultas	1	4	Completado	6

CAPÍTULO 5

IMPLEMENTACION DEL SISTEMA

5.1 Modelo de Datos

Los modelos de datos son usados para escribir una base de datos, ayudando describir la realidad que está reflejada en las historias de usuario que se realizaron en la etapa de planificación del proyecto; el modelo que se va a usar es el entidad – relación.

Al tener varios estándares en el modelo ayuda a que la base de datos sea más entendible; además los estándares permiten que la base de datos mantenida y actualizada con mayor facilidad. La Tabla 5.1 muestra la estandarización para el desarrollo de modelo de datos. Los estándares que se propusieron para el desarrollo del sistema son:

- Los nombres de las tablas de la base de datos deben escribirse con mayúscula, si tiene dos palabras deben estar separadas por un guión bajo, ejemplo:
 - USUARIO
 - TIPO_HORA

- Toda tabla debe tener clave primaria
- La relación entre tablas debe tener claves foráneas con integridad referencial, con eso no se podrán borrar datos relacionados.
- Todos los campos que pertenezcan a las claves primarias deben ubicarse al principio de la tabla. Estos campos son auto numéricos y todos están nombrados con abreviaturas del nombre de las tablas, seguido de un guión bajo; y la palabra CÓDIGO.

Tabla 5.1 Estándar para el desarrollo de modelo de datos

Tabla	Clave Primaria
USUARIO	USU_CODIGO
ASIGNATURA	ASG_CODIGO
TIPO_BAJA	TBJ_CODIGO

5.2 Diseño de la base de datos

Para la construcción de la base de datos se emplea el modelo entidad – relación, lo que se obtiene un modelo completo de la base de datos correctamente normalizados y relacionada constituida tablas, campos y tipos de datos.

La base de datos se construye en base a los requerimientos plasmados en las historias de usuarios con el fin de obtener la visión que tiene los usuarios sobre la manipulación de los datos a través del sistema de control escolar.

El siguiente diagrama representa las entidades requeridas para la satisfacción de las necesidades del cliente y las relaciones entre ellas.

5.3 Modelo entidad relación

En la Figura 5.1 se puede observar el modelo entidad – relación que consta de 57 tablas.

5.4 Diagrama de Clases

En la Figura 5.2 se puede apreciar el diagrama de clases de la aplicación; se puede verificar que consta de 63 clases.

5.5 Diagrama de Arquitectura

En la Figura 5.3 se puede observar el diagrama de arquitectura que contiene las capas del sistema y se explica la manera en que se conectan las diferentes capas del sistema.

5.6 Manual de usuario.

Para la capacitación del sistema se ha elaborado el manual de usuario que se encuentra en la sección anexos para su revisión.

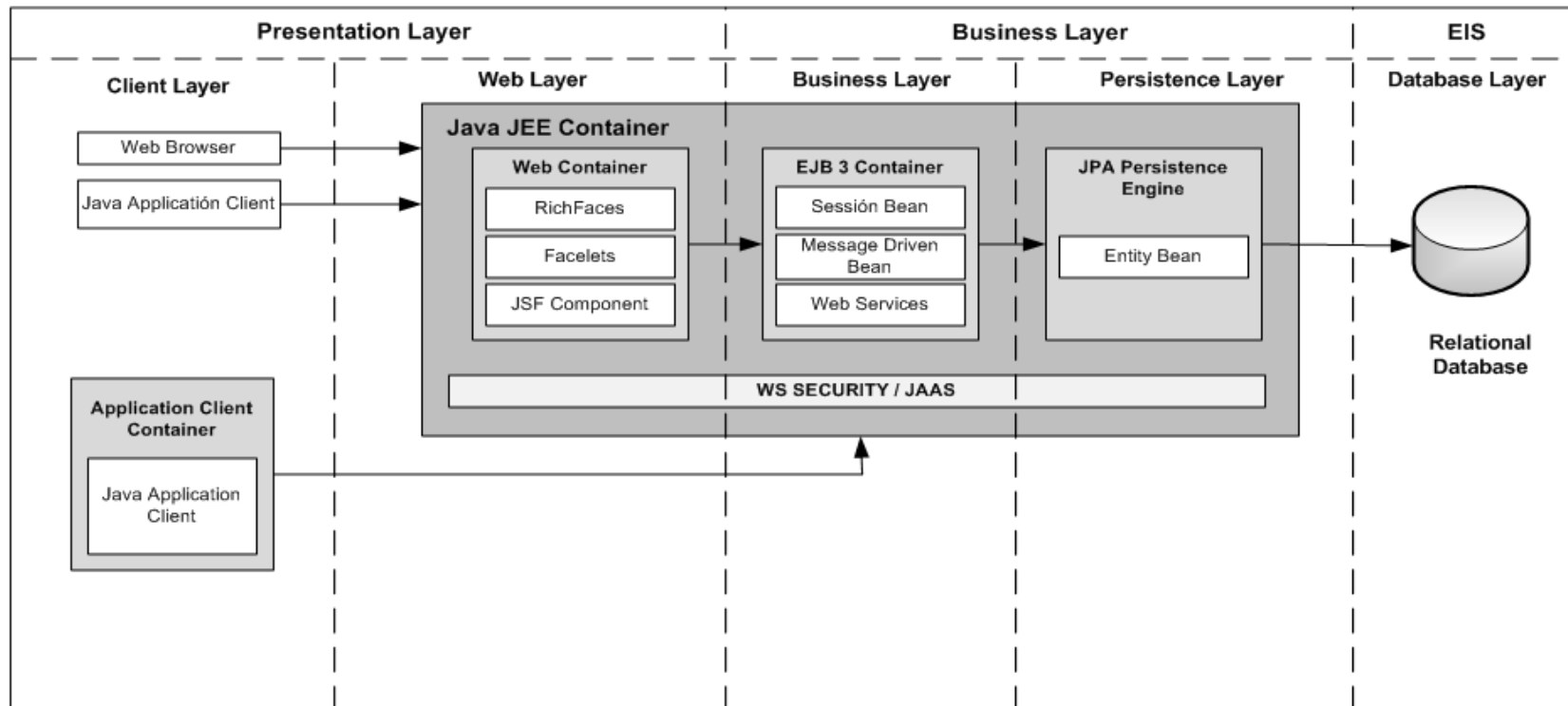


Figura 5.3 Diagrama de arquitectura

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- El uso de la metodología Scrum conjuntamente con Xtreme Programming para el análisis, diseño y desarrollo del sistema de control escolar, permitió obtener un producto de software que automatizó los procesos manuales, dentro de los plazos previstos y cubriendo todas las necesidades que exige la escuela bilingüe Sangay.
- La especificación de requerimientos del sistema desarrollado se lo hizo en forma gradual. A medida que se avanzaba con los diferentes prototipos del sistema, se permitió realizar correcciones y ajustes al producto de software creado.
- La implementación del sistema de control escolar para la escuela bilingüe Sangay, proporciona la reducción de los tiempos a los colaboradores del área académica en la administración y gestión de los procesos académicos, mejorando el desempeño de sus actividades en la realización de tareas de planificación académica, como la de asignar materias a profesores, generar listas del alumnado e ingreso de notas. De esta

manera la institución brinda un mejor servicio a los estudiantes, profesores y docentes. Además de permitir que la información grabada esté disponible al usuario de manera automática, se reducen los ingresos de datos manuales que generan resultados de poca fiabilidad.

- Existe un mayor control de la programación al desarrollar una aplicación distribuida en capas usando patrones de diseño de J2EE, pues facilita el mantenimiento de la aplicación al poder realizar modificaciones de una manera rápida y sencilla.
- Las herramientas utilizadas para el desarrollo del presente proyecto han sido en su mayoría Open Source, economizando de esta manera los costos de construcción y mantenimiento del sistema.

6.2 RECOMENDACIONES

- Para la realización de un sistema que permita automatizar los procesos de una institución educativa, es recomendable conocer y entender los procesos, para recolectar los requerimientos de una manera más fácil y rápida evitando errores en la fase de desarrollo.
- Para el desarrollo de aplicaciones web es recomendable usar herramientas de software libre, para evitar los altos costos de licenciamiento y la poca flexibilidad que presenta el software propietario. Se sugiere el uso de software libre, pues la tendencia actual es la utilización de estas

herramientas que aparte de economizar totalmente la construcción de un sistema, facilitan en gran manera la misma, a más de que permiten reutilizar el código y optimizarlo, dando con esto un buen mantenimiento al sistema que se haya construido.

- Para el desarrollo de proyectos pequeños es aconsejable combinar las metodologías ágiles Scrum con Xtreme Programming ya que proporciona buenos resultados en cortos plazos, y con un equipo de desarrollo no muy extenso; la utilización de estas dos metodologías serán exitosas siempre y cuando exista una buena interacción con el usuario final.
- Al Departamento de Ciencias de Computación se recomienda que se aplique el presente proyecto de tesis como un modelo para quienes se interesen más por conocer sobre la utilización de las metodologías Scrum y Xtreme Programming, así como el uso de herramientas Open Source que pueden ser aprovechadas en proyectos orientados a la obtención rápida de resultados o que tengan una gran cantidad de cambios en el transcurso del desarrollo.
- Se recomienda a la escuela bilingüe Sangay utilizar el actual proyecto de tesis como una guía para la automatización a futuro del proceso de gestión de recursos económicos.

6.3 BIBLIOGRAFÍA

- Henrik, Kniberg. "Scrum y XP desde las trincheras". Estados Unidos de América, Editorial C4Media, 2007. Pág. 122.
- CHROMATIC y APANDI, Díaz Tatiana. "Extreme Programming". Estados Unidos, Primera Edición, Editorial Oreilly & Associates Inc, 2003, Pág. 90.
- BECK, Kent; ANDRES, Cynthia; GAMMA, Erich. "Extreme Programming Explained". Estados Unidos, Segunda Edición, Editorial Addisonwesley, Pág. 189.
- BECK, Kent; GAMMA, Erich. "Una explicación de la Programación Extrema: Aceptar el Cambio". Madrid, Primera Edición, Pearson Educación, 2002, Pág. 216.
- NEWKIRK, Martin. "Programación Extrema En La Práctica". Madrid, Primera Edición, Editorial Pearson Educación, 2002, Pág.200.
- SCHWABER, Ken y BEEDLE, Mike, Agile Software Development With Scrum, Primera Edición, Editorial Prentice Hall, Estados Unidos, 2007, Pág. 158.
- PICHLER, Roman, Agile Product Management With Scrum, Primera Edición, Editorial Addison-wesley, Estados Unidos, 2010, Pág. 168.
- PALACIO, Juan, Flexibilidad con Scrum, principios de diseño e implantación en campos Scrum. SafeCreative. Edición Octubre 2007. Pág. 218
- GRENNING, J. Launching Xp at a Process-Intensive Company. IEEE Software 18:3-9. 2001

- ASTELS David, Test-Driven Development, A practical guide. Prentice Hall PTR 2003.

6.3.1 SITIOS DE REFERENCIA EN INTERNET

- http://www.ecured.cu/index.php/Metodologia_Agil_de_Desarrollo_SXP. Tema: Metodología Ágil de Desarrollo de Software SCRUM-XP. 10-08-2011.
- <http://www.buenastareas.com/ensayos/Sxp-Metodolog%C3%ADa-%C3%81gil-Para-Proyectos-De/1405930.html>. Tema: Metodología Ágil Para Proyectos. 10-11-2011.
- <http://www.chuidiang.com/ood/metodologia/scrum.php>. Tema: SCRUM. 03-12-2011.
- <http://es.wikipedia.org/wiki/Scrum>. Tema: Scrum. 08-12-2011.
- <http://www.xprogramming.com/>. Tema: Xprogramming. 23-12-2011.
- <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-yxp-desde-las-trincheras.pdf>. Tema: Scrum y Xp desde las trincheras. 02-07-2011.
- <http://agilitrix.com/2011/02/5-ways-scrum-creates-safety-vs-xp/>. Tema: 5 Maneras de seguridad de Scrum frente a XP. 10-08-2011.
- <http://www.mountingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming>. Tema: Diferencias entre Scrum and Xtreme Programming. 13-08-2011.
- <http://desarrollodesoftware-mauricio-ntura.blogspot.com/2012/04/scrum-y-xp.html>. Tema: Agilidad: Scrum Vs XP. 20-08-2011.