

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

**“SISTEMA DE GESTIÓN ACADÉMICA DEL PERSONAL QUE
REALIZA ESTUDIOS EN LOS DIFERENTES INSTITUTOS
MILITARES DEL EJÉRCITO”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: ANGEL GUTIERREZ GUTIERREZ

TUTOR METODOLÓGICO: M.Sc. FRANZ CUEVAS QUIROZ
ASESOR: Lic. GERMAN HUANCA TICONA

LA PAZ - BOLIVIA
2017



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mi familia, amig@s y en especial a mi Madre

AGRADECIMIENTOS

Al Tcnl. DAEN. Gonzalo Aldo Leyton Eyzaguirre Jefe de SICyT y al Sr. Tec. I Max Cadena Prado encargado de SICyT por darme la oportunidad, apoyo, colaboración e información para la realización del presente proyecto.

A mi tutor Lic. Franz Cuevas Quiroz y a mi asesor Lic. Germán Huanca Ticona por su seguimiento, constancia, apoyo y consejos.

A los docentes de la carrera de informática.

Muchas Gracias...

RESUMEN

Departamento VI – Educación, es una institución del Ejército encargado de la administración, seguimiento y control del proceso enseñanza-aprendizaje del personal que realiza estudios en los institutos militares del Ejército.

En el presente proyecto de grado se propone el “sistema de gestión académica del personal que realiza estudios en los diferentes institutos militares del Ejército”, el cual está desarrollado por la necesidad de tener un mejor apoyo en la administración de la información de sus integrantes.

En el desarrollo del proyecto se vio conveniente utilizar metodologías ágiles, por el tamaño del proyecto y las entregas del avance del sistema, es así que se realizó la planificación de desarrollo con SCRUM, y se utilizó la metodología TDD en la codificación del sistema. Las herramientas utilizadas son de software libre.

Se realizó la normalización de archivos Excel para luego migrar a una base de datos, la generación de distintos reportes especializados, el acceso restringido al sistema, y lo más importante el kardex de los integrantes que contiene estudios en el país, estudios en el exterior, exámenes de ascenso y el profesorado militar. El acceso a la información es prácticamente instantáneo, los trámites para la obtención de certificaciones, que normalmente duraban en promedio 24 horas ahora se realizan en promedio en una hora.

ABSTRACT

Department VI - Education, is an Army institution in charge of the administration, monitoring and control of the teaching-learning process of the personal that carries out studies in the military institutes of the Army.

In the present project of degree it is proposed the "academic management system of the personnel that carries out studies in the different military institutes of the Army", which is developed by the need to have a better support in the administration of the information of its members.

In the development of the project it was convenient to use agile methodologies, due to the size of the project and the deliveries of the system progress, so that the development planning was done with SCRUM, and the TDD methodology was used in the coding of the system. The tools used are free software.

It was made the normalization of Excel files for later to migrate to a database, the generation of different specialized reports, the restricted access to the system, and most importantly the kardex of the members that contains studies in the country, studies abroad, Promotion exams and military teaching staff. Access to information is practically instantaneous, the procedures for obtaining certifications, which normally last for an average of 24 hours, are now carried out on average in one hour.

CAPÍTULO I

MARCO REFERENCIAL

En las diferentes empresas privadas y públicas, instituciones del estado se ha venido haciendo uso de la tecnología implementando herramientas tecnológicas que le permitan facilitar los procesos y actividades. Aquellas instituciones que han implementado un software informático adecuado para el trabajo que realizan tienen un mayor desarrollo en el funcionamiento de sus tareas, generando ventajas al trabajo manual que se realizaba como también llevando ventajas a otras instituciones que todavía trabajan con métodos manuales.

El Departamento VI del Ejército, encargado de la administración del seguimiento y control del personal que realiza estudios en los diferentes institutos militares del Ejército requiere tener un sistema de información que permita obtener información precisa, detallada y oportuna del historial académico de sus integrantes.

El Departamento VI del Ejército viene implementando las tecnologías de Información y Comunicación. La DTIC (Departamento de Tecnologías de Información y Comunicación) ejecuta actividades coordinadas en los campos de Información y Comunicación para apoyar en la formación de los profesionales de excelencia a través de la administración de la información generada por los Institutos Militares y el empleo de las comunicaciones disponibles, a fin de asesorar y apoyar en el cumplimiento de los objetivos del comando de Institutos Militares.

Es fundamental contar con un software que agilice la información habitual de la institución y lograr así tareas de manera óptima y precisa. El presente proyecto propone desarrollar un sistema de gestión académica, que permitirá el almacenamiento de datos, procesarlos de manera adecuada y contar con información detallada, oportuna y confiable. Esto llevará al acceso rápido y eficaz de la información.

Con el software se registra a los miembros de los institutos adecuadamente. Optimizando las tareas y procedimientos administrativos. En consecuencia el sistema propuesto mejorará el control y el seguimiento de sus integrantes brindando toda la información que se requiera. Además de ofrecer beneficios al personal administrativo e integrantes, también el beneficio llega a la población en general que requieran de alguna certificación militar.

1.1 ANTECEDENTES

1.1.1 De la institución

La institución es la encargada de la administración, seguimiento y control del proceso enseñanza-aprendizaje del personal que realiza estudios en los institutos militares del Ejército, ellos se encargan de llevar un historial de todos sus integrantes desde el momento en que ciudadanos de diferentes partes del país ingresan a los diferentes institutos militares ubicados en distintas partes del país.

Los Institutos Militares dependientes del Departamento VI se listan a continuación:

- Escuela de Comando y Estado Mayor.
- Escuela de Aplicación de Armas.
- Escuela Militar de Sargentos del Ejército.
- Colegio Militar del Ejército.
- Escuela Militar de Inteligencia del Ejército.
- Escuela de Idiomas del Ejército.

- Escuela Militar de Topografía del Ejército.
- Escuela Militar de Música del Ejército.
- Escuela Militar de Equitación del Ejército.
- Instituto Promoción Profesional del Ejército.
- Unidad Educativa del Ejercito “La Paz”.
- Unidad Educativa del Ejercito “Cochabamba”.
- Liceo Militar del Ejército.

1.1.2 Del proyecto

Un sistema de gestión académica es un sistema de información de la administración para establecimientos de educación, para manejar datos de sus estudiantes. Proporciona capacidades para registrar alumnado en cursos, documentación de calificaciones, resultados de las pruebas de los estudiantes y otros resultados de la evaluación, elaboración de horarios de los estudiantes, seguimiento de la asistencia del estudiante y la gestión de muchas otras necesidades de datos relacionados con los estudiantes en un determinado establecimiento de educación. No debe confundirse con un sistema de gestión de aprendizaje o entorno virtual de aprendizaje, donde los materiales del curso, actividades y pruebas de evaluación pueden ser publicados electrónicamente.

Respecto a los proyectos orientados al seguimiento y control académico desarrollados en la carrera de Informática, de la Universidad Mayor de San Andrés a continuación se mencionan algunos.

- SISTEMA WEB DE INFORMACIÓN ACADÉMICO FINANCIERO CENTRO DE CAPACITACIÓN TÉCNICA COMATEC. Año: 2015 Por: Giovanni Joaquin Coaquira Illanes. El sistema automatizó las tareas y actividades más importantes que realiza la institución como ser la inscripción de estudiantes, boleta de inscripción, el seguimiento académico y el control de pagos. Se utilizó la metodología RUP y se empleó las tecnologías Framework Spring Java con PostgreSQL y Apache Tomcat.

- SISTEMA WEB DE INFORMACIÓN Y SEGUIMIENTO ACADÉMICO CASO: PROGRAMA DE ADMISIÓN PREFACULTATIVA CARRERA DE PSICOLOGÍA, Año: 2014 Por: Marizol Roque Caizana. El trabajo hace énfasis en los procesos académicos de: Inscripción de Postulantes, registro de calificaciones y emisión de documentos. Para el análisis, diseño e implementación se aplicó la metodología AUP y para el modelado de aplicaciones web UML basado en Ingeniería Web(UWE). Para el desarrollo del sistema Web se utilizó el framework Spring java, PostgreSQL, Apache TOMCAT y JasperReport.
- SOFTWARE DE GESTIÓN Y SEGUIMIENTO ACADÉMICO CASO: CENTRO DE CAPACITACIÓN TÉCNICA CEINF RM 175/13. Año: 2014 Por: GHILMAR EMIGDIO COPA MAMANI. El Proyecto de Grado titulado software de gestión y seguimiento académico caso: centro de capacitación técnica “CEINF RM 175/13”. El proyecto realizó la implementación del software de gestión y seguimiento académico. El sistema automatiza la inscripción de estudiantes, boleta de inscripción, seguimiento académico, certificado de aprobación y el registro del pago de mensualidades. Este desarrollado utilizando la metodología RUP, las tecnologías implementadas son PHP MySQL y frameworks front-end css javascript.
- SISTEMA DE INFORMACIÓN Y SEGUIMIENTO ACADÉMICO PARA EL INSTITUTO NUEVA ESPERANZA. Año: 2009 Por: Freddy Julio Colque Carvajal. El Proyecto de Grado fue desarrollado en el Centro de Desarrollo Integral “NUEVA ESPERANZA”. El instituto puso en marcha el desarrollo de un Sistema de Información y Seguimiento Académico que coadyuve a un buen manejo de la información generada en el instituto, reduciendo el volumen de papelería, registrándola en una base de datos normalizada y sin redundancia. Esta base de datos se encuentra centralizada en un servidor al cual pueden tener acceso los usuarios dados de alta desde cualquier punto de la institución. La emisión de informes y reportes se realiza de manera automática desde el mismo sistema y de acuerdo a solicitudes realizadas. Para el desarrollo se utilizó la metodología RUP, el Lenguaje de Modelado Unificado UML y las tecnologías Visual Basic.NET y SQLServer.

1.2 PLANTEAMIENTO DEL PROBLEMA

La institución actualmente recibe la información de los diferentes institutos pertenecientes o que dependen de la misma, en archivos Excel, en algunos institutos se tiene sistemas, pero orientados y desarrollados para la administración específica de los institutos. Debido a estas situaciones mencionadas se elabora la siguiente lista de problemas.

- Demora en el acceso a los datos
- Registro en archivos Excel con distintos formatos
- Reportes incompletos
- Restricción inadecuada a los datos
- Registro semiautomatizado de emisión de certificados

Por un deficiente seguimiento y control de los procesos académicos de la institución, esto lleva a una demora en el acceso a los datos, los diversos formatos de archivos Excel no clasifican de buena manera la información y los sistemas independientes no generan servicios para que estos interactúen con otros sistemas.

Planteado el problema, el presente proyecto de grado propone responder lo siguiente:

¿El sistema de gestión académica del personal que realiza estudios en los diferentes Institutos Militares del Ejército permitirá obtener información académica en forma oportuna?

1.3 OBJETIVOS

1.3.1 Objetivo general

Desarrollar un sistema de gestión académica del personal que realiza estudios en los diferentes Institutos Militares del Ejército, permitiendo generar información útil, segura y oportuna para la toma de decisiones.

1.3.2 Objetivos específicos

- Normalizar los formatos de archivos Excel que se tiene actualmente para luego migrar a una Base de datos.
- Diseñar roles para el acceso de usuarios, desarrollando el módulo de usuarios y su validación para el control de la seguridad, el cual permitirá que el sistema pueda ser utilizado y accedido por los usuarios autorizados según las políticas de la institución mediante la asignación de roles a los usuarios.
- Diseñar los reportes necesarios según las políticas de la institución, para su implementación.
- Diseñar el modulo para la emisión de certificados, para tener el historial de certificados.

1.4 JUSTIFICACIÓN

La tecnología es indispensable tanto como software y hardware. La institución cuenta con las condiciones tecnológicas para poder desarrollar e implementar el sistema de información propuesto, cuenta con equipos de computación necesarios que se encuentran en la red de la institución, se tiene acceso a la red de la intranet que es un aspecto fundamental para la implementación del proyecto.

Con la implementación del sistema se beneficiará la institución, en consecuencia el beneficio llegará a todos los integrantes en especial al personal de los Institutos Militares del Ejército de Bolivia. En cuanto a la institución el sistema ayudará en el desempeño eficiente de los procesos de educación y administración de la información del personal de los Institutos Militares, reduciendo la posibilidad de errores que puedan presentarse, minimizando los tiempos de los procesos mecánicos y repetitivos que se realizan, teniendo de ésta forma la información almacenada de manera más ordenada, proporcionando información inmediata y confiable, permitiendo un mejor servicio a los integrantes.

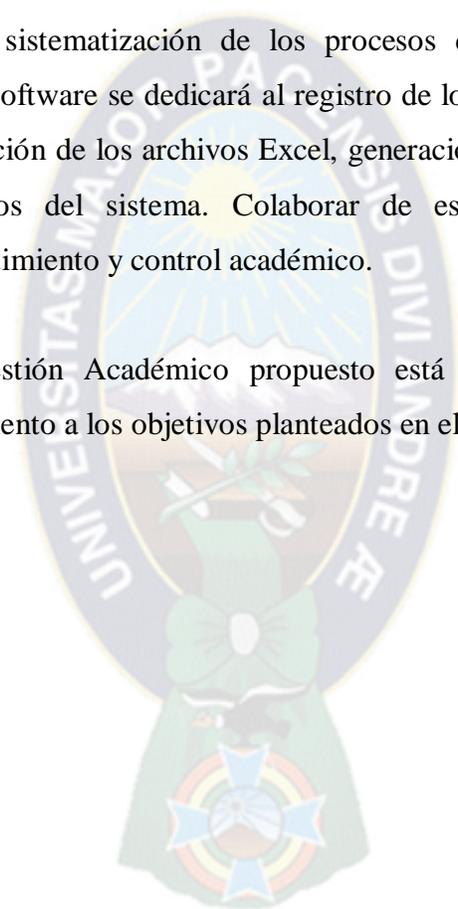
El software reducirá los costos de materiales insumos necesarios utilizados para el procesamiento de la información, lo que permite a la institución minimizar costos, pero el

beneficio también será de manera intangible, se refleja a través de los resultados en tiempo y esfuerzo de trabajo. Por ser una Institución del Estado y debido al Decreto Supremo No. 1793 el sistema será desarrollado en su integridad con software libre reduciendo los costos de desarrollo en un 100%, además la institución será dueño de los derechos de propiedad del software.

1.5 ALCANCES Y LÍMITES

Se desarrollará la sistematización de los procesos de información académica de seguimiento y control. El software se dedicará al registro de los integrantes con su respectivo historial académico, migración de los archivos Excel, generación de Reportes y la creación de distintos roles de usuarios del sistema. Colaborar de esta manera a las autoridades correspondientes en el seguimiento y control académico.

El Sistema de Gestión Académico propuesto está dirigido exclusivamente a la institución dando cumplimiento a los objetivos planteados en el proyecto.



CAPÍTULO II

MARCO TEÓRICO

2.1 INGENIERÍA DEL SOFTWARE

La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza (Sommerville, 2005).

La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad (Pressman, 2010).

La ingeniería de software es una tecnología con varias capas como se aprecia en la figura 2.1. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad.

El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.



Fig. 2.1: Capas de la ingeniería del software

Fuente: Pressman, 2010

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas.

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra.

2.3 ETAPAS DE LA INGENIERÍA DEL SOFTWARE

a) Análisis de requisitos

Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios.

Usualmente el cliente o usuario tiene una visión incompleta e inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El

contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible.

b) Especificación

Es la tarea de describir detalladamente el software a ser escrito, de una forma rigurosa. Se describe el comportamiento esperado del software y su interacción con los usuarios y otros sistemas.

c) Diseño y arquitectura

Determinar cómo funcionará de forma general sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la segunda etapa también conocidas como las entidades de negocio. Generalmente se realiza en base a diagramas que permitan describir las interacciones entre las entidades y su secuenciado.

d) Programación

Se traduce el diseño a código. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados “tangibles”. No necesariamente es la etapa más larga ni la más compleja aunque una especificación o diseño incompletos/ambiguos pueden exigir que, tareas propias de las etapas anteriores se tengan que realizarse en esta.

e) Prueba

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Es una buena praxis realizar pruebas a distintos niveles (por ejemplo primero a nivel unitario y después de forma integrada de cada componente) y por equipos diferenciados del de desarrollo (pruebas cruzadas entre los programadores o realizadas por un área de test independiente).

f) Documentación

Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema. Las tareas de esta etapa se inician ya en la primera fase pero sólo finalizan una vez terminadas las pruebas.

g) Mantenimiento

En esta etapa se realizan un mantenimiento correctivo que consiste en resolver errores y un mantenimiento evolutivo para mejorar las funcionalidades o dar respuesta a nuevos requisitos.

2.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

2.4.1 El agilismo

En 2001, 17 representantes de nuevas metodologías y críticos de los modelos de mejora basados en procesos se reunieron, convocados por Kent Beck, para discutir sobre el desarrollo de software. Fue un grito de ¡basta ya! a las prácticas tradicionales. Estos profesionales, con una dilatada experiencia como aval, llevaban ya alrededor de una década utilizando técnicas que les fueron posicionando como líderes de la industria del desarrollo software. Conocían perfectamente las desventajas del clásico modelo en cascada donde primero se analiza, luego se diseña, después se implementa y, por último (en algunos casos), se escriben algunos tests automáticos y se martiriza a un grupo de personas para que ejecuten manualmente el software, una y otra vez hasta la saciedad. El manifiesto ágil se compone de cuatro principios. Es pequeño pero bien cargado de significado.

Estamos descubriendo mejores maneras de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de esta experiencia hemos aprendido a valorar:

- **Individuos e interacciones** sobre *procesos y herramientas*.

- **Software que funciona** sobre *documentación exhaustiva*.
- **Colaboración con el cliente** sobre *negociación de contratos*.
- **Responder ante el cambio** sobre *seguimiento de un plan* Esto es, aunque los elementos a la derecha tienen valor, nosotros valoramos por encima de ellos los que están a la izquierda.

2.4.2 Scrum

Scrum es una herramienta de gestión de proyectos de desarrollo ágil, para equipos pequeños de hasta 7 personas con roles multifuncionales que les permite organizar mejor el trabajo y dividirlos en iteraciones de alrededor de un mes. Cada iteración debe tener como salida software funcional y de valor para el cliente. Scrum es:

- Ligero
- Fácil de entender
- Extremadamente difícil de llegar a dominar

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. Las reglas de Scrum se describen en el presente documento.

Aunque en equipos de alrededor de 7 personas, se crean vínculos y relaciones muy fuertes y valiosas en todos los sentidos, en ocasiones podemos encontrarnos en solitario y aún así hacer uso efectivo de las ventajas de Scrum. Preparándonos para un posible crecimiento del número de miembros y asumiendo desde el inicio las herramientas y artefactos de Scrum.

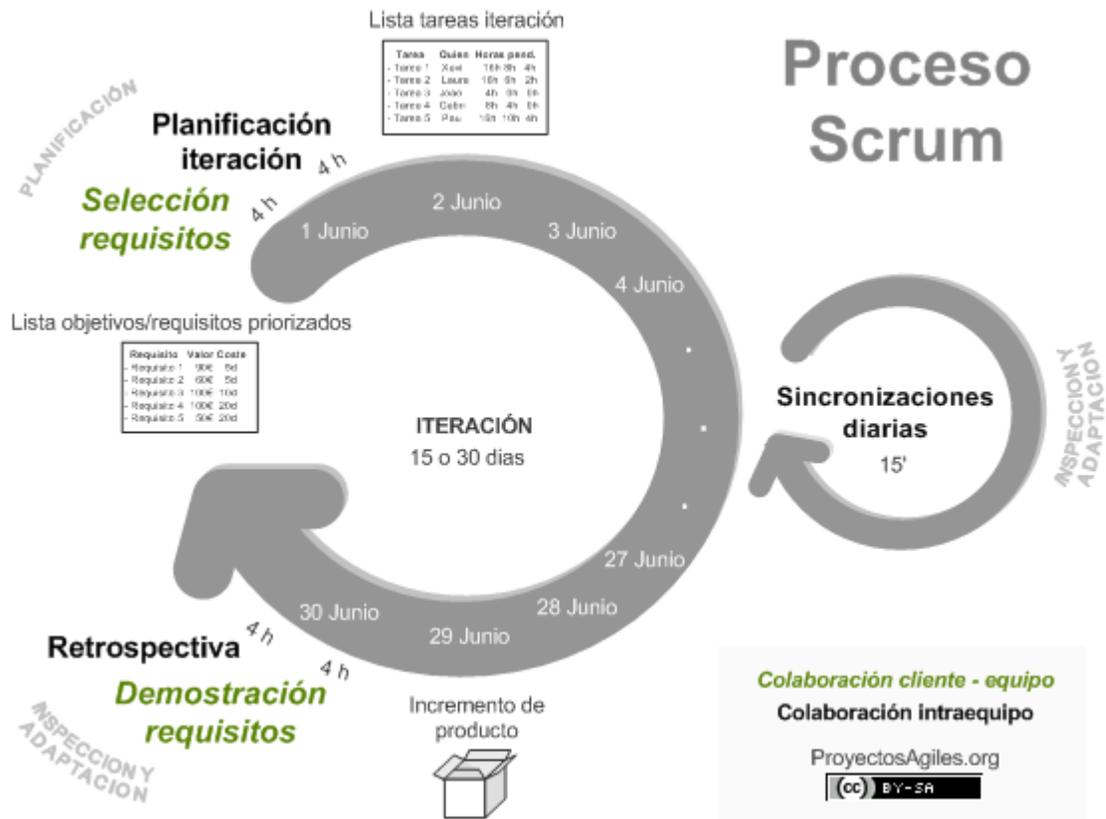


Fig. 2.2: Ciclo del proceso Scrum

Fuente: <http://www.proyectosagiles.org/que-es-scrum>

a) Roles de Scrum

Pues claro, si somos 2 o 3 personas, los roles de SCRUM casi sobran. De todas formas vale la pena mencionarlos:

Dueño del producto (Product Owner). Es el responsable de la visión del producto, de asegurar el retorno de la inversión en el proyecto. Está constantemente re-priorizando los elementos del *Product Backlog*. Es quién toma decisiones concernientes al modelo de negocio.

Decide cuando el producto está listo para una salida o cuando se debe continuar desarrollando en una nueva iteración. Y por supuesto, puede también contribuir como miembro del equipo.

Facilitador (Scrum Máster). Es el encargado de facilitar el proceso de SCRUM, contribuye a crear un ambiente de trabajo favorable y organizado entre los miembros del equipo. Se encarga de escudar al equipo de intromisiones externas para mantener el avance del equipo. Mantiene visible los artefactos de SCRUM (Product Backlog, tareas asignaciones, tiempos de entrega). Promueve mejores prácticas en la ingeniería del software. Y algo muy importante, no tiene autoridad sobre el equipo, su función es más bien como facilitador/coordinador.

Miembros del equipo: Todos los demás miembros que se encargan del desarrollo, programadores, diseñadores, skateholders, experto de negocio y demás.

b) Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo, de tal modo que todos tienen una duración máxima.

El corazón de Scrum es el Sprint, es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

2.4.3 Test Drive Development, TDD

Metodología de Desarrollo Orientado a Pruebas. Se trata de una metodología de desarrollo ágil cuya mayor fortaleza es escribir la mínima cantidad de código posible para obtener el resultado deseado.

TDD es una técnica para diseñar software que se centra en tres pilares fundamentales:

- La implementación de las funciones justas que el cliente necesita y no más.
- La minimización del número de defectos que llegan al software en fase de producción.
- La producción de software modular, altamente reutilizable y preparado para el cambio.

a) Pasos a seguir

Creación de un test, dada una especificación a cumplimentar, creamos un test unitario que debe cumplir la implementación a desarrollar.

Ejecución del test comprobando que el test falla, para cada creación de un nuevo test, ejecutamos el conjunto de tests y verificamos que la implementación que estamos añadiendo no está resuelta por el estado actual del código, dado que los test producen el error correspondiente.

Escribir código para pasar el test, implementamos la funcionalidad, con el mínimo código posible que pase el test. De esta forma, vamos desarrollando paso a paso una funcionalidad completa, pero con la intención de contemplar todos los escenarios posibles.

Ejecutar el test, comprobando que pasa el test, ejecutamos el conjunto de tests para ver que el código que hemos implementado se comporta de la forma esperada.

Refactorizar el código, este paso es importante para, una vez estar seguros de que tenemos una solución para nuestro problema, debemos dejar el código de la forma más legible y eficiente posible.

Repetir, el proceso debe repetirse por cada iteración del desarrollo.

b) Tipos de test

En la siguiente figura se muestra la clasificación de los tests. A la izquierda, se agrupan los tests que pertenecen a desarrolladores y, a la derecha, los que pertenecen al Dueño del Producto. A su vez, algunos tipos de tests contienen a otros.

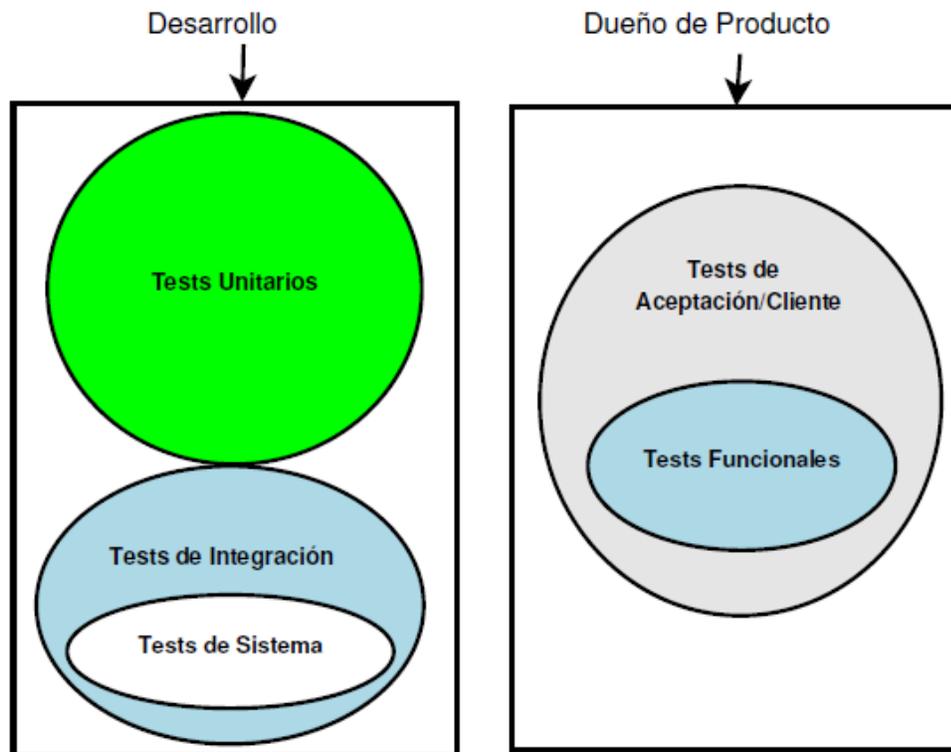


Fig. 2.3: Clasificación de los tests

Fuente: Blé Jurado, 2010

Tests de Aceptación, es un test que permite comprobar que el software cumple con un requisito de negocio, un test de aceptación es un ejemplo escrito con el lenguaje del cliente pero que puede ser ejecutado por la máquina.

Tests Funcionales, Cuando se habla del aspecto funcional, se distingue entre test funcional y test no funcional. Un test funcional es un subconjunto de los tests de aceptación. Es decir, comprueban alguna funcionalidad con valor de negocio. Los tests de aceptación tienen un ámbito mayor porque hay requerimientos de negocio que hablan de tiempos de

respuesta, capacidad de carga de la aplicación, etc; cuestiones que van más allá de la funcionalidad. Un test funcional es un test de aceptación pero, uno de aceptación, no tiene por qué ser funcional.

Tests de Sistema, es el mayor de los tests de integración, ya que integra varias partes del sistema. Se trata de un test que puede ir, incluso, de extremo a extremo de la aplicación o del sistema. Se habla de sistema porque es un término más general que aplicación, pero no se refiere a administración de sistemas.

Tests Unitarios, son los tests más importantes para el practicante TDD, los ineludibles. Cada test unitario o test unidad es un paso que andamos en el camino de la implementación del software. Todo test unitario debe ser:

- Atómico
- Independiente
- Inocuo
- Rápido

Si no cumple estas premisas entonces no es un test unitario, aunque se ejecute con una herramienta tipo xUnit.

Tests de Integración, los tests de integración son la pieza del puzzle que nos faltaba para cubrir el hueco entre los tests unitarios y los de sistema. Los tests de integración se pueden ver como tests de sistema pequeños. Típicamente, también se escriben usando herramientas xUnit y tienen un aspecto parecido a los tests unitarios, sólo que estos pueden romper las reglas.

2.5 SEGURIDAD DEL SISTEMA

2.5.1 Amenazas de seguridad en aplicaciones web

a) Fuerza bruta

Fuerza bruta es un ataque automatizado de prueba y error, utilizado para adivinar los valores (usuarios, contraseña etc.) de los parámetros de la aplicación web. Normalmente las personas usan contraseñas o claves criptográficas débiles que son fáciles de adivinar. Los hackers explotan esta vulnerabilidad de seguridad en páginas web usando un diccionario. Los hackers empiezan un bucle recorriendo el diccionario término a término, en búsqueda de la contraseña válida. El ataque de fuerza bruta es muy popular y pueden llevar a cabo horas, semanas o años en completarse.

b) Autenticación incompleta y débil validación

Autenticación incompleta es un ataque cuando un hacker accede alguna funcionalidad sensible de la aplicación sin tener que autenticarse completamente. En este ataque un hacker podría descubrir URL específica de la funcionalidad sensible través de pruebas de fuerza bruta sobre ubicaciones comunes de ficheros y directorios (/admin), mensajes de error etc. Normalmente muchas aplicaciones, páginas web no tienen seguridad y usan las técnicas de seguridad para aplicaciones convencionales. En un ataque de validación débil el atacante pueda obtener, modificar o recuperar los datos, contraseñas de otros usuarios. Esto ocurre cuando los datos requeridos para validar la identidad de los usuarios, son fácilmente predecible o puedan ser falsificadas. El proceso de validación de datos es una parte importante de las aplicaciones y se deben implementar soluciones de seguridad para aplicaciones.

c) Autorización insuficiente

Autorización insuficiente significa que un usuario tiene acceso a los partes sensibles de la aplicación o sitio web que deberían requerir un aumento de restricciones de control de acceso. Sin algunas medidas de seguridad para aplicaciones, el ataque de autorización

insuficiente podría ser muy dañoso. En el ataque de autorización insuficiente, un usuario autenticado podría controlar toda la aplicación o contenido de la página web. Las recomendaciones dicen que las aplicaciones deben tener políticas de acceso, modificación y prudentes restricciones deben guiar la actividad de los usuarios dentro de la aplicación.

d) Secuestro de sesión

En el ataque de secuestro de sesión un hacker podría deducir o adivinar el valor de sesión id y después puede utilizar ese valor para secuestrar la sesión de otro usuario. Si un hacker es capaz de adivinar la ID de sesión de otro usuario, la actividad fraudulenta es posible. Esto podría permitir a un hacker usar el botón atrás del navegador para acceder las páginas accedidas anteriormente por la víctima. Por esta razón es muy importante tener seguridad para aplicaciones. Otro problema por seguridad para aplicaciones es la expiración de sesión incompleta. Esto resulta cuando una página web permite reutilización de credenciales de sesión antigua. La expiración de sesión incompleta incrementa la exposición de las páginas web para que los hackers roben o se secuestren sesión. La fijación de sesión es otra técnica utilizada por secuestro de sesión. Cuando fuerza un ID de sesión de usuario a un valor explícito, el hacker puede explotar esto para secuestrar la sesión. Posteriormente de que un ID de sesión de usuario ha sido fijado, el hacker esperará para usarlo. Cuando el usuario lo hace, el hacker usa el valor del ID de sesión fijado para secuestro de sesión. Las páginas web que usan las sesiones basadas en cookies sin ningún tipo de seguridad en páginas web, son las más fáciles de atacar. Normalmente la mayoría de ataques de secuestro de sesión tiene la fijación de cookie como motivo.

Sin servicios de seguridad en aplicaciones web contra la fijación de sesión, el hacker puede hacer mucho daño y robar datos confidenciales., la lógica para generar sesión ID, cookie y cada sesión ID deben ser mantenidos confidenciales.

e) Cross-site scripting

Cuando un usuario visita una página web, el usuario espera que haber seguridad en página web y la página web le entregue contenido válido. Cross-site Scripting (XSS) es un

ataque donde la víctima es el usuario. En el ataque de XSS, el hacker fuerza una página web a ejecutar un código suministrado en el navegador del usuario. Con este código el hacker tiene la habilidad de leer, modificar y transmitir datos sensibles accesibles por el navegador. Sin ningún tipo de seguridad en páginas web, un hacker podría robar cookie, secuestrar sesiones, abrir páginas de phishing, bajar malware y mucho más utilizando el ataque de XSS. Hay dos tipos de ataques XSS, persistentes y no persistentes. Ambos ataques pueden causar mucho daño a la reputación de la página web.

f) Desbordamiento de buffer

El desbordamiento de buffer es una vulnerabilidad común en muchos programas, que resulta cuando los datos escritos en la memoria exceden el tamaño reservado en el buffer. Los expertos de empresas de seguridad de aplicaciones, mencionan que durante un ataque de desbordamiento de buffer el atacante explota la vulnerabilidad para alterar el flujo de una aplicación y redirigir el programa para ejecutar código malicioso. Esta vulnerabilidad es muy común a nivel de sistema operativo del servidor de la aplicación.

g) Inyección de código sql

La inyección de código SQL, también conocido como SQL Injection es un ataque muy común y peligroso. Este ataque explota las páginas web que usan SQL como base de datos y construyen sentencias SQL de datos facilitados por el usuario. En el ataque de inyección de código SQL, el hacker puede modificar una sentencia SQL. Con la explotación de la vulnerabilidad, el hacker puede obtener control total sobre la base de datos o también ejecutar comandos en el sistema. Se pueden prevenir la inyección de código SQL con ayuda de saneamiento de los datos facilitados por el usuario.

h) Indexación de directorio

En el ataque de indexación de directorio, un atacante puede acceder todos los ficheros del directorio en el servidor. Sin seguridad en páginas web, esto es equivalente a ejecutar un comando “ls” o “dir”, mostrando los resultados en formato HTML. La información de un

directorio podría contener información que no se espera ser vista de forma pública. Además un hacker puede encontrar la información sensible en comentarios HTML, mensajes de error y en código fuente. La indexación de directorio puede permitir una fuga de datos que proporcione a un hacker los datos para lanzar un ataque avanzado.

i) Path traversal

En el ataque de Path Traversal, un hacker accede a los ficheros, directorios y comandos que residen fuera del directorio “root” de la web. Con el acceso a estos directorios, un atacante puede tener accesos a los ejecutables necesarios para realizar la funcionalidad de la aplicación web e información confidencial de usuarios. En el ataque de path traversal un hacker puede manipular una URL de forma que la página web ejecutará o revelará el contenido de ficheros ubicados en cualquier lugar del servidor web.

j) Denegación de servicio

En un ataque de denegación de servicio (DoS), el motivo es impedir que una página web o aplicación puede funcionar normalmente y sirva la actividad habitual a los usuarios. Los ataques DoS intentan dilapidar todos los recursos disponibles tales como: CPU, memoria, espacio de disco, ancho de banda etc. Cuando estos recursos lleguen un consumo máximo, la aplicación web pasará a estar inaccesible. Hay diferentes tipos de ataques DoS, como a nivel red, nivel dispositivo, nivel aplicación y de diferentes fuentes (DDoS).

Estos son algunos de los ataques cibernéticos sobre aplicaciones web. La metodología de seguridad para páginas web o aplicaciones web debe ser muy diferente de metodología tradicional de seguridad de aplicaciones.

La metodología de seguridad para páginas web o aplicaciones web debe estar basada en un proceso de pruebas manual y automatizadas por medio de scripts propias, revisión de códigos, herramientas propietarias, comerciales y de código abierto que identifica todos los tipos de vulnerabilidades.

2.5.2 Tipos de seguridad para sistemas web

Los tipos de seguridad en los sistemas web se mencionan a continuación.

- Seguridad en el cliente
- Seguridad en el Servidor
- Seguridad en las comunicaciones
- Seguridad en la aplicación

a) Seguridad en el cliente

Uno de los mecanismos de seguridad que se implementan son las validaciones por el lado del cliente. Existen mecanismos de validación provistas por las herramientas que utilizamos para hacer la aplicación, en el caso usamos el framework javascript Extjs que permite la validación de campos, estas validaciones son realizadas antes de que la información introducida llegue al servidor, esto evita que se envíen datos incorrectos al servidor, además se ahorra tiempo ya que si la información es incorrecta simplemente no se le envía al servidor.

b) Seguridad en el servidor

La validación del lado del cliente no es suficiente, también deben realizarse otro tipo de controles por el lado del servidor, ya sea del servidor de aplicaciones o del servidor de base de datos. En el servidor también se filtran los datos que llegan desde el cliente utilizando funciones del lenguaje PHP como ser: addslashes, htmlentities, filter_input, intval, etc.

c) Seguridad en el servidor de aplicaciones

Un servidor de aplicaciones proporciona muchos servicios y no todos son necesarios para el funcionamiento de una aplicación web. Es conveniente deshabilitar lo que no se

necesite en el servidor de aplicaciones, para ello debe configurarse adecuadamente el servidor de aplicaciones.

d) Seguridad en la base de datos

Existen muchos problemas a nivel de base de datos uno de ellos es la Inyección SQL (Lenguaje de Consulta Estructurada), son los ataques realizados contra las bases de datos. En este caso un usuario utiliza debilidades en el diseño de base de datos o de la página web para extraer información o más aun, para manipular información dentro la base de datos. Una forma de evitar este tipo de debilidades es restringiendo los caracteres que los usuarios introducen, por ejemplo las comillas, los puntos y comas y otros. También está el acceso no autorizado a la información. Esto podría solucionarse asignado correctamente los roles que tiene cada usuario en el sistema. El acceso de los usuarios a los formularios web le da acceso solo a los formularios asignados para él.

e) Seguridad en la comunicación

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

f) Seguridad en la aplicación

El control de acceso de los usuarios es una parte fundamental para una aplicación web.

- Autenticación basada en la aplicación, en este caso la aplicación implementa su propio mecanismo de autenticación. Es más costosa pero es más flexible porque permite establecer diferentes permisos y niveles de acceso asignados al usuario.
- Passwords, se recomienda registrar los valores para los nombres de los usuarios, almacenar los password de forma segura protegiendo el acceso a la base de datos, bloquear una cuenta cuando se detecta un número determinado de intentos de acceso incorrectos. Tener una política de recuperación de password en caso de olvido por parte del usuario.
- Sesiones, después de que el usuario se ha autenticado se debe mantener esta autenticación en cada conexión subsiguiente. Para esto se utiliza las variables de sesión, que permiten mantener el estado entre diferentes peticiones HTTP. El procedimiento es el siguiente: Después de autenticarse el usuario recibe un identificador de sesión, este identificador es invisible y lo acompaña en cada petición. Este identificador se almacena en la maquina cliente, mediante una cookie.
- La gestión de sesiones es responsabilidad del programador. Un sistema de gestión de sesiones debe: establecer un tiempo límite de vida para la sesión, pedir una nueva autenticación cuando realice una operación importante, proteger los identificadores de sesión durante su transición y destruir la cookie cuando finalice la sesión para evitar el acceso de otro usuario.

2.6 CALIDAD DEL SOFTWARE

La calidad del software es el *“Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan”*. (Pressman, 2010).

Para medir la calidad del software, se hace uso de un conjunto de métricas de calidad, las mismas poseen la finalidad de evaluar las distintas características que posee el sistema para definir la calidad del mismo. El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad: funcionalidad, confiabilidad, usabilidad, eficiencia y facilidad de mantenimiento y portabilidad. En el presente proyecto se tomara en cuenta los siguientes atributos: funcionalidad, usabilidad y facilidad de mantenimiento.

a) Funcionalidad

Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

La adaptabilidad se refiere a la capacidad del producto software un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

La exactitud consiste la capacidad de entregar resultados correctos.

La interoperabilidad es la habilidad de interactuar con determinados sistemas.

El cumplimiento consiste en la compatibilidad con estándares, convenciones o regulaciones.

La seguridad es la habilidad de prevenir uso no autorizado, tanto intencional como accidental, de programas y datos.

c) Usabilidad

Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.

Entendible, se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.

Aprendible, establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.

Operable, agrupa los conceptos que evalúan la operación y el control del sistema.

Para medir la usabilidad usaremos la siguiente fórmula:

$$U = \frac{\sum_{i=1}^n x_i}{n}$$

Donde:

$\sum_{i=1}^n x_i$ = es la suma de ponderaciones

n = cantidad de preguntas

e) **Facilidad de mantenimiento**

Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.

IEEE Std. 982.1-1988 [IEE93] sugiere un *índice de madurez de software* (IMS) que proporcione un indicio de la estabilidad de un producto de software (con base en cambios que ocurran para cada liberación del producto). Para ello, se determina la siguiente información:

M_T = Numero de módulos de la liberación actual.

F_C = Número de módulos de la liberación actual que cambiaron.

F_A = Número de módulos de la liberación actual que se agregaron.

F_D = Número de módulos de la liberación anterior que se borraron en la liberación actual.

El índice de madurez del software se calcula de la forma siguiente:

$$IMS = \frac{M_T - (F_C + F_A + F_D)}{M_T}$$

Conforme el IMS tiende a 1.0, el producto comienza a estabilizarse. El IMS también puede usarse como una métrica para planificar actividades de mantenimiento de software. El tiempo medio para producir una liberación de un producto de software puede correlacionarse con el IMS, y es posible desarrollar modelos empíricos para esfuerzo de mantenimiento. (Pressman, 2010).

2.7 HERRAMIENTAS DE IMPLEMENTACIÓN

En el presente proyecto se implementará las siguientes herramientas:

- Como Sistema Gestor de Bases de Datos MariaDB (Versión 10.1.19).
- Servidor Web Apache (Versión 2.4).
- Lenguaje de programación PHP (Versión 5.6) junto al ORM redBeanPHP(Versión 3.5).
- Lenguaje de programación javascript junto al Framework ExtJS (Versión 3).
- Lenguaje de marcado HTML5 y Lenguaje de Estilos CSS junto al framework ExtJS (Versión 3).
- Como IDE de desarrollo NETBEANS (Versión 8).
- Para los diagramas de análisis y diseño STARUML.

CAPÍTULO III

MARCO APLICATIVO

Al inicio del proceso se definen todos los elementos que harán parte de la lista del producto (Product Backlog), durante cada Sprint (Iteración) se realizó una reunión en donde se decidió cuáles de esos elementos se iban a desarrollar durante el resto de la iteración (Sprint Backlog). En el proceso de SCRUM se combina con el uso del lenguaje de Modelo Unificado (UML). Se hace un uso aerodinámico del UML: Diagramas de casos de uso, diagramas de clases y diagramas WAE (Web Application Extension).

También se utiliza la metodología de desarrollo de software ágil TDD, Una de las técnicas de pruebas de software que más está alineada con SCRUM y con el desarrollo ágil de software es la de Desarrollo conducido por pruebas, que a su vez es una de las técnicas de programación extrema (Extreme Programming o XP).

TDD es una técnica que cambia el orden establecido en cuanto a primero desarrollar (programar) y luego probar, de manera que primero se define las pruebas (casos de prueba) y a partir de estos se va desarrollando la funcionalidad, repitiendo el ciclo.

3.1 DIAGNOSTICO

La información importante para la institución llega a esta unidad desde todos los institutos dependientes de la misma, estos datos están en diferentes formatos Excel, se los almacena en directorios especiales para posteriormente ser utilizados y consultados. Los trámites, certificaciones y consultas realizadas se requiere desde los últimos 30 años, por lo que se deben registrar todos los datos generados en esos 30 años, ya que se tiene la mayoría de registros en archivos Excel, se propone el aprovechamiento de estos archivos, normalizarlos o estandarizarlos y migrarlos a una base de datos. A continuación se muestra un esquema de la solución.

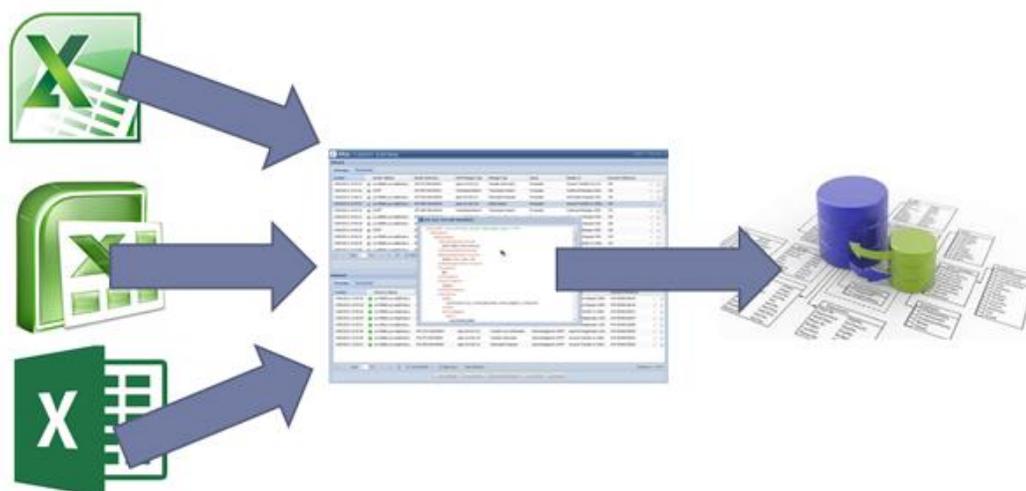


Fig. 3.1: Esquema de migración de archivos Excel a BD

Para el registro de registros actuales, anteriores o futuros se puede hacer uso de los mismos archivos Excel normalizados como también registrarlos uno por uno.

3.2 PLANEACIÓN DEL PROCESO SCRUM

La metodología SCRUM requiere una planeación bien definida, debe establecerse un equipo de trabajo, unos objetivos y estos objetivos serán divididos en ciclos (iteraciones) de máximo 20 días, realizando reuniones cortas de 10 minutos cada día, para verificar el avance, analizar los problemas y determinar rápidamente un plan de acción para solucionarlos.

3.2.1 Definición del Equipo

El equipo está conformado por:

- **Product Owner**, es el dueño del producto, en el desarrollo de este proyecto se considero como dueño del producto a dos personas: Tcnl. DAEN. Gonzalo Leyton Eyzaguirre, Jefe de SICyT(Unidad de Sistemas, Ciencia y Tecnología del Departamento VI), y al Sr. Tec. I Max Cadena Prado, encargado de SICyT.
- **Scrum Master**, es el que asegura de que las etapas de Scrum se lleven a cabo. Este rol será desempeñado por Angel Gutierrez Gutierrez.
- **Development Team**, El equipo de desarrollo se encarga de realizar todo el proceso de creación del producto, este rol será desempeñado por Angel Gutierrez Gutierrez.

3.2.2 Definición de los objetivos del Producto

Tabla 3.1: Product Backlog General

Orden	Título	Descripción
1	Diseño de la Base de Datos	Crear la base de datos, definir las tablas y sus campos, vistas y procedimientos almacenados.
2	Modulo administrar cursos	Desarrollar el módulo para la administración de cursos
6	Modulo administrar integrantes	Desarrollar el módulo para la administración de integrantes
4	Generación de Reportes	Diseñar y desarrollar los reportes (Kardex, Promoción, Certificados Militares y Certificados Civiles, Diplomas, otros)

3.2.3 Descripción de las iteraciones o Sprints

Una vez que se tiene la lista general de productos (Product backlog general) se empezó a descomponer en base a historias de usuario, priorizando las tareas más importantes. A continuación se muestra las iteraciones o sprints realizados durante el desarrollo del proyecto.

Se utilizó como herramienta el software open source Sprintometer.

a) Sprint 0

Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea
✓ <u>s1-h1</u>	Diseño de la base de datos
✓ <u>1</u>	Crear el modelo entidad relacion de la base datos
✓ <u>2</u>	Crear las tablas y sus campos en el gestor de base de datos MariaDB
✓ <u>s1-h2</u>	Definir la estructura de archivos MVC
✓ <u>1</u>	Crear la estructura de archivos MVC que conformaran el sistema
✓ <u>s3-h3</u>	Crear un prototipo para el registro de cursos y sus integrantes
✓ <u>1</u>	crear el CRUD para integrantes
✓ <u>2</u>	crear el CRUD para cursos

Fig. 3.2: Sprint 0

b) Sprint 1

Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea
✓ <u>s1-h1</u>	Registrar integrante
✓ <u>1</u>	crear el frontend para el registro de integrante
✓ <u>2</u>	crear el backend para el registro del integrante
✓ <u>s2-h2</u>	Registrar integrantes importando un archivo excel
✓ <u>1</u>	crear el backend para importar archivos excel
✓ <u>2</u>	crear el frontend para la importacion de archivos excel
✓ <u>3</u>	Añadir las validaciones de cada columna del archivo excel
✓ <u>s2-h3</u>	Registrar curso
✓ <u>1</u>	crear el backend para el registro de curso
✓ <u>2</u>	crear el frontend para el registro de curso

Fig. 3.3: Sprint 1

c) Sprint 2

Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea
✓ <u>s3-h1</u>	Registrar los exámenes de ascenso de cada integrante
✓ <u>1</u>	Crear el backend para el registro de exámenes de ascenso de cada integrante
✓ <u>2</u>	Crear el frontend para el registro de exámenes de ascenso de cada integrante
✓ <u>s3-h2</u>	Registrar los cursos realizados en el exterior de cada integrante
✓ <u>1</u>	Crear el backend para el registro de cursos realizados en el exterior
✓ <u>2</u>	Crear el frontend para el registro de cursos realizados en el exterior
✓ <u>s3-h3</u>	Registrar el profesorado militar de cada integrante
✓ <u>1</u>	Crear el backend para el registro de exámenes de ascenso de cada integrante
✓ <u>2</u>	Crear el frontend para el registro de exámenes de ascenso de cada integrante

Fig. 3.4: Sprint 2

d) Sprint 3

Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea
✓ <u>s4-h1</u>	Generar el reporte por promocion
✓ <u>1</u>	Crear el backend para generar el reporte
✓ <u>2</u>	Crear el frontend para el reporte
✓ <u>s4-h2</u>	Generar el reporte Kardex
✓ <u>1</u>	Crear el backend para generar el reporte
✓ <u>2</u>	Crear el frontend para el reporte
✓ <u>s3-h3</u>	Registrar usuarios del sistema
✓ <u>1</u>	Crear el bakend para el registro de usuarios del sistema
✓ <u>2</u>	crear el frontend para el registro de usuarios del sistema
✓ <u>3</u>	crear el backend para cambiar contraseña
✓ <u>4</u>	crear el frontend para el cambio de contraseña

Fig. 3.5: Sprint 3

e) Sprint 4

Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea
✓ <u>s4-h1</u>	Generar un reporte para el Certificado Militar de Antecedentes
✓ <u>1</u>	Crear el backend para generar el reporte
✓ <u>2</u>	Crear el frontend para el reporte
✓ <u>s4-h2</u>	Generar el reporte Certificado Civil (no haber sido dado de baja de algun institu...
✓ <u>1</u>	Crear el backend para generar el reporte
✓ <u>2</u>	Crear el frontend para el reporte
✓ <u>s5-h3</u>	Registrar la emision de certificados
✓ <u>1</u>	añadir a la base de datos la tabla certificado
✓ <u>2</u>	Crear el backend para la emision de certificados
✓ <u>3</u>	Adicionar el boton para guardar la emision de los certificados

Fig. 3.6: Sprint 4

3.3 ANALISIS DE REQUERIMIENTOS

3.3.1 Actores

El sistema tiene un conjunto de usuarios, se clasifica de la siguiente manera.

Tabla 3.2: Actores

Actor	Descripción
Administrador	Este actor representa <i>al encargado de SICyT(Unidad de Sistemas, Ciencia y Tecnología) del departamento VI</i>
Usuario	Este actor representa <i>a los operadores que se encargaran de registrar a los integrantes, no pueden modificar ni eliminar registros</i>

Invitado	Este actor representa a los miembros de la institución que solo puedan ver o navegar por el sistema sin la posibilidad de realizar ninguna acción.
Integrante	Este actor representa a los integrantes del Ejército ya sean oficiales, suboficiales y/o sargentos que pueden acceder sólo a su historial académico.

3.3.2 Requerimientos Funcionales

Tabla 3.3: Requerimientos funcionales

Código	Requerimiento	Descripción
RF1	Registrar Integrante	El sistema deberá tener una interfaz que permita registrar a cada integrante (Altas, Bajas y Modificaciones).
RF2	Registrar Curso	El sistema deberá tener una interfaz que permita el registro de cursos (Altas, Bajas y Modificaciones).
RF3	Registrar Cursos en el exterior	El sistema deberá tener una interfaz que permita el registro de estudios en el exterior realizado por cada integrante.

RF4	Registrar exámenes de ascenso	El sistema deberá <i>tener una interfaz que permita el registro de los exámenes de ascenso de cada integrante.</i>
RF5	Registrar profesorado militar	El sistema deberá <i>tener una interfaz que permita el registro del profesorado militar de cada integrante</i>
RF6	Generar reporte por promoción	El sistema deberá <i>tener una interfaz que permita la vista previa del reporte por promoción para posteriormente generar el reporte.</i>
RF7	Generar el reporte kardex	El sistema deberá <i>tener una interfaz que permita la vista previa del reporte kardex, para posteriormente generar el reporte.</i>
RF8	Generar el reporte certificado militar	El sistema deberá <i>tener una interfaz que permita la vista previa del reporte certificado militar para posteriormente generar el reporte.</i>
RF9	Registrar usuarios del sistema	El sistema deberá <i>tener una interfaz que permita el registro de usuarios y la asignación de roles.</i>

RF10	Realizar configuraciones de los certificados	El sistema deberá <i>tener una interfaz que permita la configuración de parámetros para la emisión de reportes.</i>
RF11	Registrar emisión de certificados	El sistema deberá <i>tener una interfaz que permita el registro para la emisión de certificados.</i>
RF12	Consultar kardex	El sistema deberá <i>tener una interfaz para el cambio de contraseña de los diferentes usuarios del sistema.</i>

3.3.3 Requerimientos no funcionales

Tabla 3.4: Requerimientos no funcionales

Código	Requerimiento	Descripción
RNF1	Validación de formularios	El sistema deberá <i>validar los campos de todos los formularios existentes en el sistema.</i>
RNF2	Cambiar contraseña	El sistema deberá <i>tener una interfaz para el cambio de contraseña de los diferentes usuarios del sistema.</i>

3.3.4 Diagrama de casos de uso de alto nivel

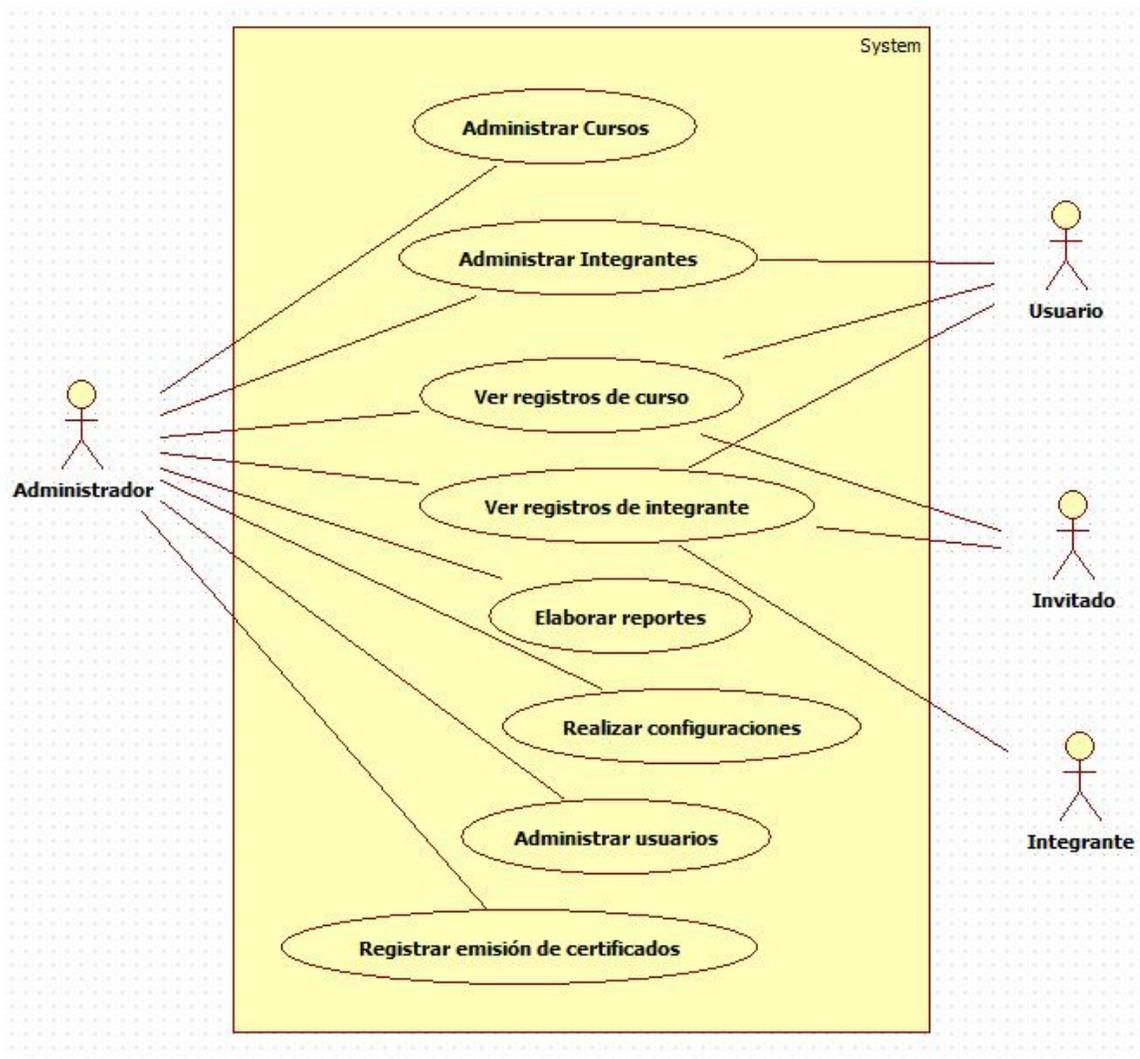


Fig. 3.7: Diagrama de casos de uso de alto nivel

3.3.5 Descripción de los casos de uso

A continuación se realiza la descripción detallada de los casos de uso de alto nivel:

a) Caso de uso: Elaborar reportes

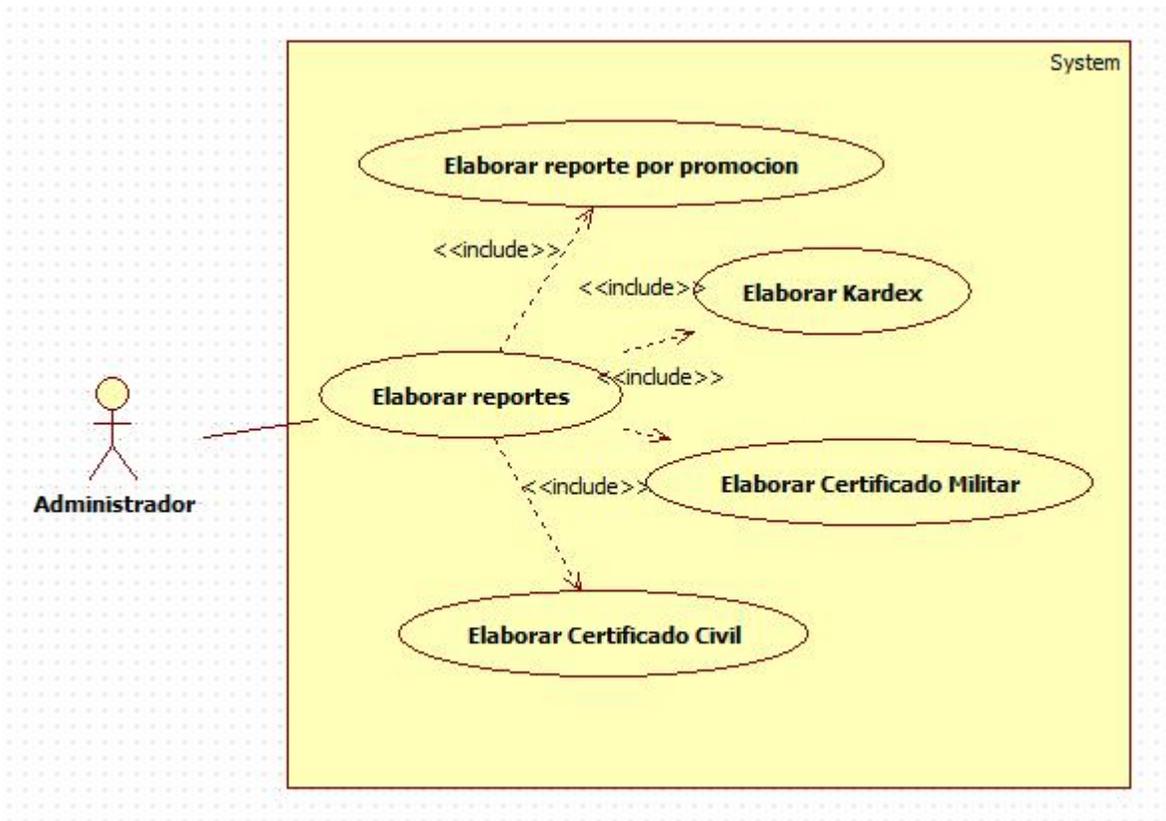


Fig. 3.8: Caso de uso elaborar reportes

Tabla 3.5: Caso de uso Elaborar reportes

Caso de uso: Elaborar reportes	
Nombre	Elaborar reportes
Autor	Angel Gutierrez G.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se elabore los reportes por promoción, kardex, certificado militar y certificado civil, diplomas y otros.
Actores	Administrador
Precondición	Ser usuario autenticado de tipo Administrador
Secuencia normal	Paso Acción
	1 El actor ingresa a las opciones de reportes

	2	El sistema proporciona filtros para generar los diferentes reportes
	3	El sistema proporciona la pre visualización de los reportes
	4	El sistema genera archivos pdf listos para imprimir
Postcondición	El sistema genera los certificados necesarios	
Excepciones	Paso	Acción
	2	El sistema genera mensajes de error si no existe algún filtro o termino de búsqueda

b) Caso de Uso: Realizar configuraciones

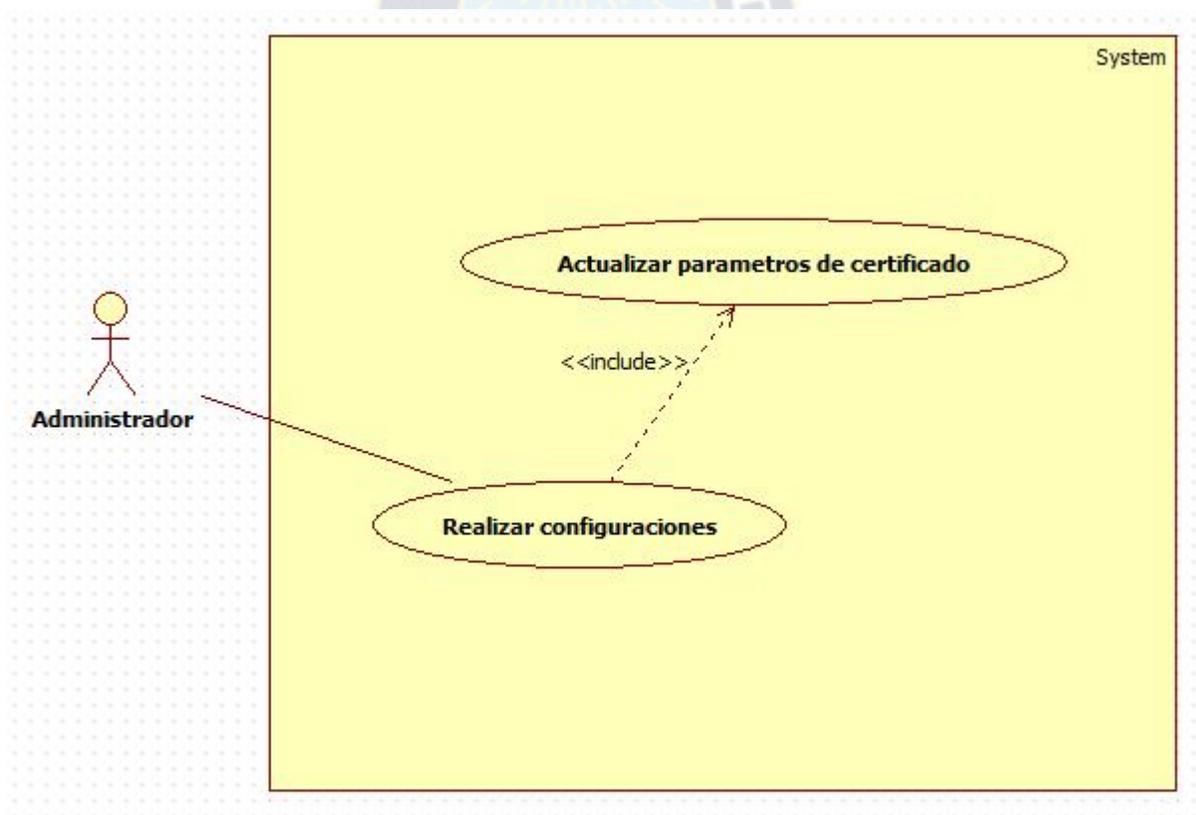


Fig. 3.9: Caso de uso realizar configuraciones

Tabla 3.6: Caso de uso Realizar configuraciones

Caso de uso: Realizar configuraciones	
Nombre	Realizar configuraciones
Autor	Angel Gutierrez G.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se realice las configuración o actualización de los parámetros de los certificados
Actores	Administrador
Precondición	Ser usuario autenticado de tipo Administrador
Secuencia normal	Paso Acción
	1 El actor ingresa a las opciones de parámetros de certificados
	2 El sistema proporciona un listado de todos los parámetros
	3 El actor selecciona el parámetro para ser actualizado.
Postcondición	El sistema actualiza los parámetros de los certificados
Excepciones	Paso Acción
	3 El sistema genera mensajes de error si los parámetros no son válidos

c) Registrar emisión de certificados

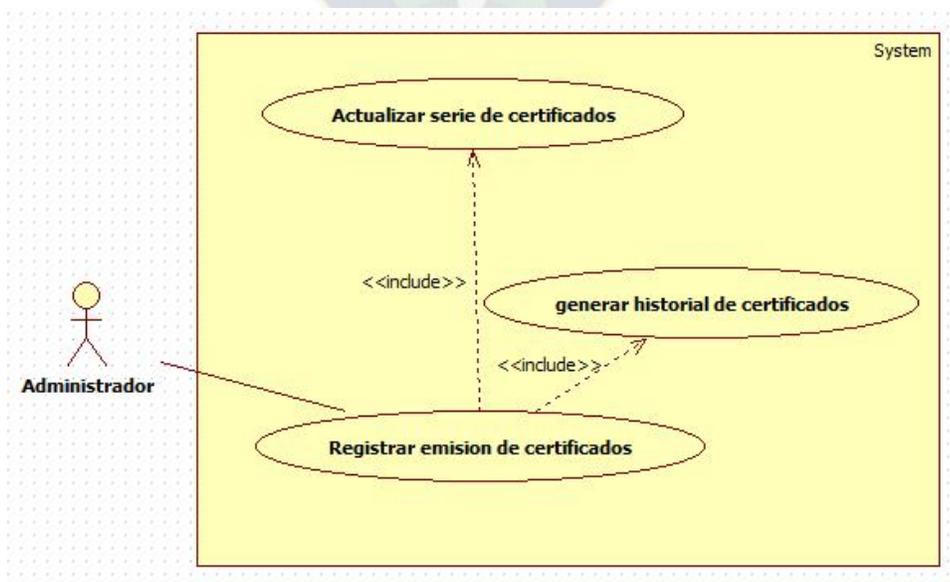
**Fig. 3.10:** Caso de uso registrar emisión de certificados

Tabla 3.7: Caso de uso registrar emisión de certificados

Caso de uso: Registrar emisión de certificados		
Nombre	Registrar emisión de certificados	
Autor	Angel Gutierrez G.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se realice el registro de emisión de certificados	
Actores	Administrador	
Precondición	Ser usuario autenticado de tipo Administrador y debe estar generado algún reporte de certificado	
Secuencia normal	Paso	Acción
	1	El actor presiona el botón que permitirá registrar la emisión de certificados
	2	El sistema actualiza el número de serie del certificado
	3	El sistema almacena el certificado con todos sus datos
	4	El sistema proporciona un listado de los certificados emitidos en un orden descendente de la fecha emitida
Postcondición	El sistema registra la emisión de certificados	
Excepciones	Paso	Acción
	1	El sistema genera un mensaje de confirmación de la emisión de certificado

3.4 MODELO DEL DISEÑO

3.4.1 Diseño de la arquitectura

La arquitectura del sistema es Cliente Servidor con tres capas: Modelo Vista y Controlador

3.4.2 Diagrama de Clases

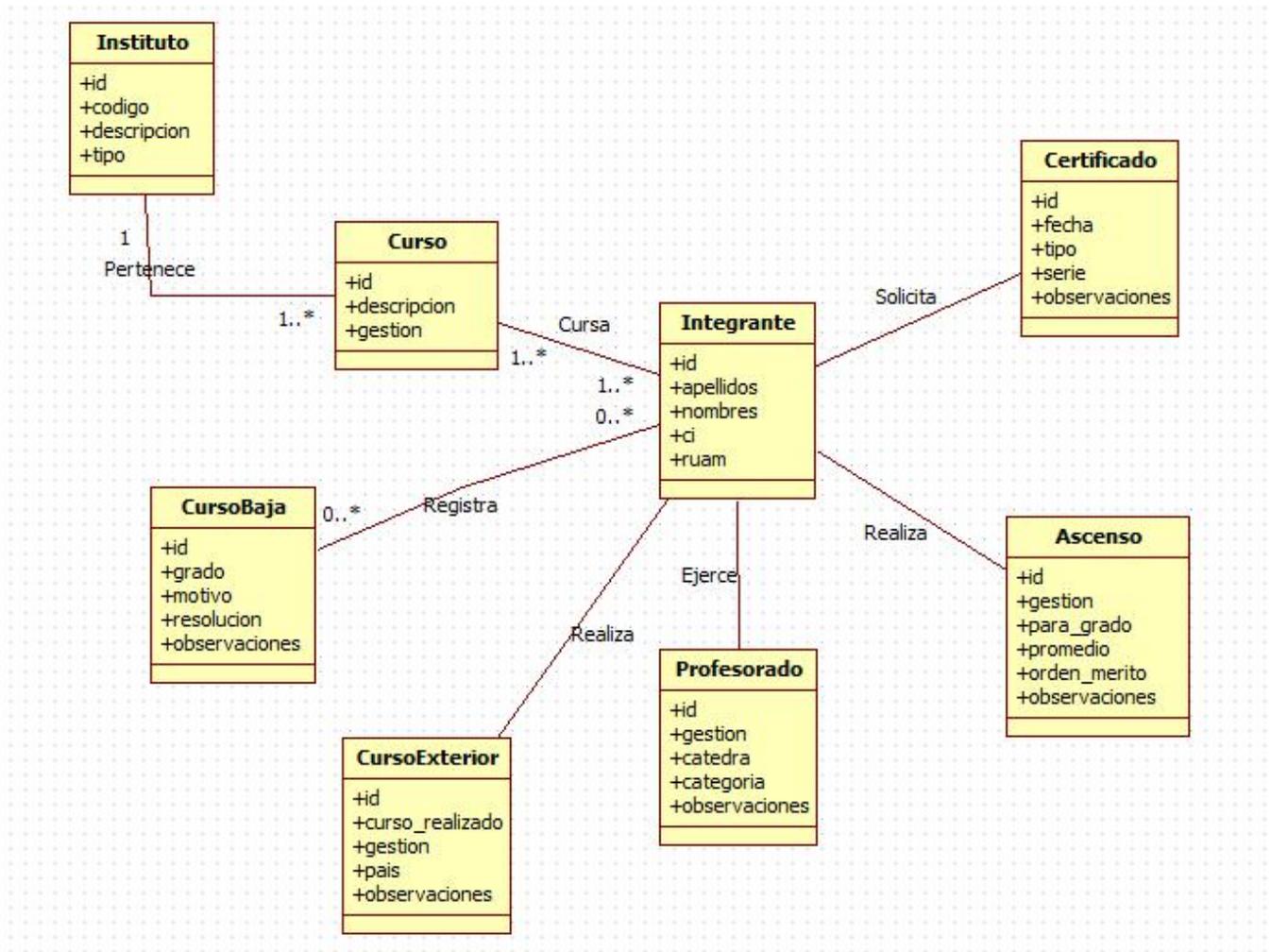


Fig. 3.11: Diagrama de clases

3.4.3 Diagrama entidad relación

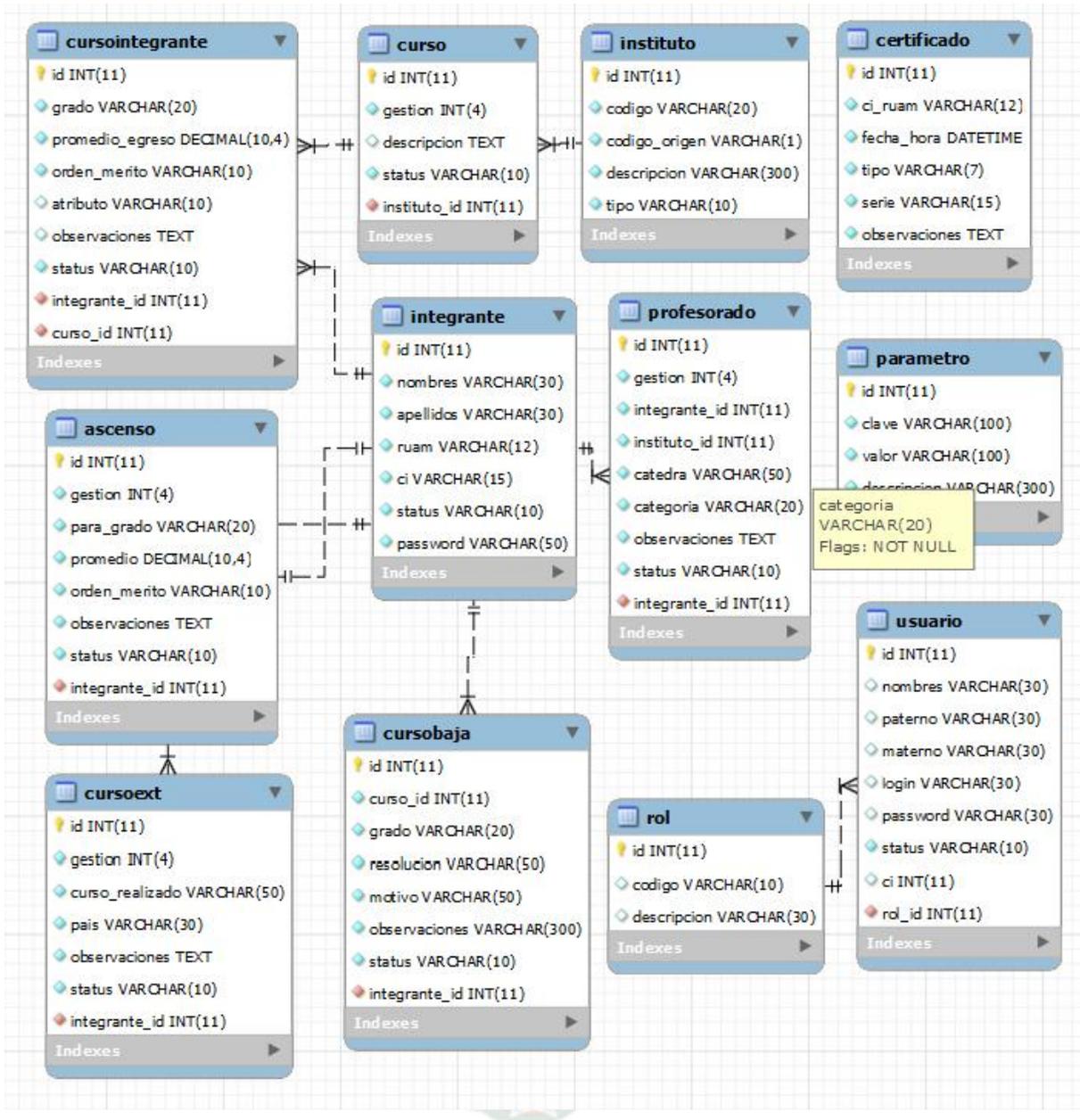


Fig. 3.12: Diagrama entidad relación

3.5 DISEÑO DE LA INTERFAZ PRINCIPAL

La navegación de la aplicación web será de una sola página conocida como un **single-page application (SPA)**, o aplicación de página única es una aplicación web o es un sitio web que cabe en una sola página con el propósito de dar una experiencia más fluida a los usuarios como una aplicación de escritorio. En un SPA todos los códigos de HTML, JavaScript, y CSS se carga de una vez o los recursos necesarios se cargan dinámicamente como lo requiera la página y se van agregando, normalmente como respuesta de las acciones del usuario. El framework javascript ExtJS que se utilizó permite a través de sus componentes realizar peticiones asincrónicas al servidor actualizando las partes necesarias para la representación de datos. Se diseño un menú principal desde donde se realiza todas las operaciones en una misma pantalla.



Fig. 3.13: Diseño de la interfaz principal

Como ejemplo en la siguiente figura mostramos el caso de uso registrar curso, entonces se tiene una estructura de archivos que conformaran las tres capas del modelo vista controlador utilizando Diagramas WAE(Web Application Extension)

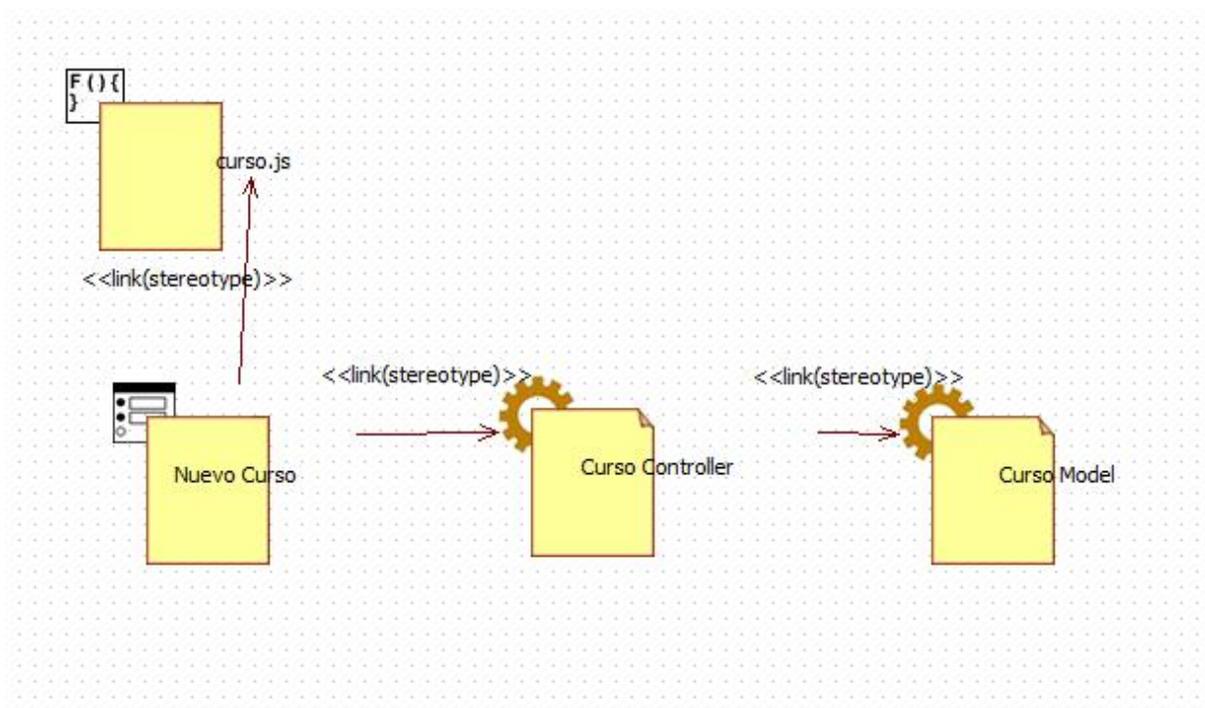


Fig. 3.14: Diagrama WAE Nuevo curso

3.6 DISEÑO DE INTERFACES

Todos los usuarios deberán acceder al sistema introduciendo el nombre de usuario y su respectiva contraseña, como los usuarios están previamente registrados y asignados con un determinado rol, entonces dependiendo del rol el usuario que se tiene se podrá acceder a los módulos que están habilitados en base al rol. En la siguiente figura se muestra el acceso al sistema.



Fig. 3.15: Interfaz principal de acceso al sistema

Si el usuario existe en la base de datos a continuación se muestra la pantalla de bienvenida al sistema, en caso contrario se mostrara el error típico de “Usuario o Contraseña incorrecto”.



Fig. 3.16: Interfaz bienvenida al sistema

En la pantalla anterior se puede observar las opciones o tareas a realizar, en este caso se muestra todas las opciones, ya que se ingreso como usuario administrador. A continuación se muestran algunas capturas en imágenes más importantes del sistema relacionados con los casos de uso.

	GRADO	NOMBRES	APELLIDOS	PROM...	ORDE...
1	CDTE.	JORGE	LOZADA MOSQUEIRA	68.2545	206 / ...
2	CDTE.	CRISTINA ALEZ	FLORES MERCADO	70.4579	205 / ...
3	CDTE.	ANDRÉS	AMESTEGUI ZEBALLOS	71.1469	204 / ...
4	CDTE.	JUÁN FELIPE	VILLA NUÑEZ	71.5610	203 / ...
5	CDTE.	NATALY PABLO	ROCABADO TORRICO	72.1613	202 / ...
6	CDTE.	DIEGO ROCIO	FERRUFINO RODRÍGUEZ	72.2432	201 / ...
7	CDTE.	NINOSCKA ALEJANDRO	SORIA BERDEJA	72.6279	200 / ...

Fig. 3.17: Interfaz administración de cursos

	NOMBRES	APELLIDOS	RUAM	CI
1	KEVIN ROLANDO	CADIMA BERNAL	KCB2013037C	210036 LP
2	RENE	CADIMA ENRIQUEZ	RCE1991058C	1234624 LP
3	WILGE	BUTRON CADIMA	WBC1990047C	2345724 BE
4	JORGE MARCELO	CADIMA PAZ	JCP1989001C	1234567 LP
5	BERNARDO ANTONIO	CADIMA ENRIQUEZ	BCE1987053C	2345730 CB

Fig. 3.18: Interfaz administración de integrantes

	Nombres	Apellidos	COLML	EMEE	EMIE	EAA CB	ECEM	EAA CA
1	JUAN MANUEL	PIEROLA SANDOVAL	9.2850	1 /145	1986			
2	JHONNY	CASTRO MARISCAL	9.2510	2 /145	1986			
3	GARY	GUEVARA IRIATE	9.1100	3 /145	1986			
4	JORGE PASTOR	MENIETA FERRUFINO	9.0890	4 /145	1986		9.5221	1/74
5	JAIME	LORA MOSCOSO	9.0620	5 /145	1986			9.3981
6	GUIDO SIPRIAN	URBINA MACEDA	9.0180	6 /145	1986			11/72
7	CHARLES	MONJE MENESES	8.9650	7 /145	1986		9.1009	18
8	JORGE MARCOS	LOPEZ ASCUI	8.9590	8 /145	1986		2004	9.3428

Fig. 3.19: Interfaz vista previa del reporte por promoción

Inicio SIGAD Cursos Reporte por Promoción

Reporte por promocion

COMANDO GENERAL DEL EJERCITO
DEPARTAMENTO VI - EDUCACION
BOLIVIA

SIG AD

COLMIL PROMOCION 1986

Nº	NOMBRES	APELLIDOS	COLMIL	EMEE	EMIE	EAA CB	ECEM	BAJAS
1	JUAN MANUEL	PEROLA SANDOVAL	9.2880	1/145	1986			
2	JHONNY	CASTRO MARISCAL	9.2510	2/145	1986			
3	GARY	GUEVARA IRIATE	9.1100	3/145	1986			
4	JORGE PASTOR	MENIETA FERRUFINO	9.0890	4/145	1986		9.9221	1/74 2003
5	JAMIE	LORA MOSCOBO	9.0620	5/145	1986			
6	GUIDO SIPRIAN	URBINA MACEDA	9.0190	6/145	1986			
7	CHARLES	MONJE MENESES	9.9990	7/145	1986		9.1009	18 2004
8	JORGE MARCOS	LOPEZ ASCUI	9.9990	8/145	1986			
9	JAMIE	CLAURE BASCON	9.9990	9/145	1986			
10	JUAN OLEGARIO	RAMIREZ GUDIAN	9.9370	10/145	1986			
11	JOHNNY JUAN	MOLLINEDO DEL VILLAR	9.9360	11/145	1986			
12	OVIDIO ENRIQUE	SANTISTEVAN SEGURA	9.9230	12/145	1986			
13	ROBERTO	SANDOVAL FERRER	9.9000	13/145	1986			
14	JUAN BENITO	URIONA AYLON	9.3990	14/145	1986		9.0591	11/94 2006
15	ROBERTO	DAZA VARGAS	9.3730	15/145	1986		9.3148	5/74 2003
16	GROVER REIBERTO	ROJAS USARTE	9.3600	16/145	1986		9.1920	10/74 2003
17	GUSTAVO	AGUILAR ORELLANA	9.3530	17/145	1986			
18	AMILCAR CONSTANTINO	ADUVERI AGUILAR	9.3480	18/145	1986		9.1689	16/88 2006

Fig. 3.20: Reporte por promoción

SIG AD

Kardex de BERNARDO ANTONIO CADIMA ENRIQUEZ (BCE1987053C)

Reporte Kardex de: BERNARDO ANTONIO CADIMA ENRIQUEZ (BCE1987053C)

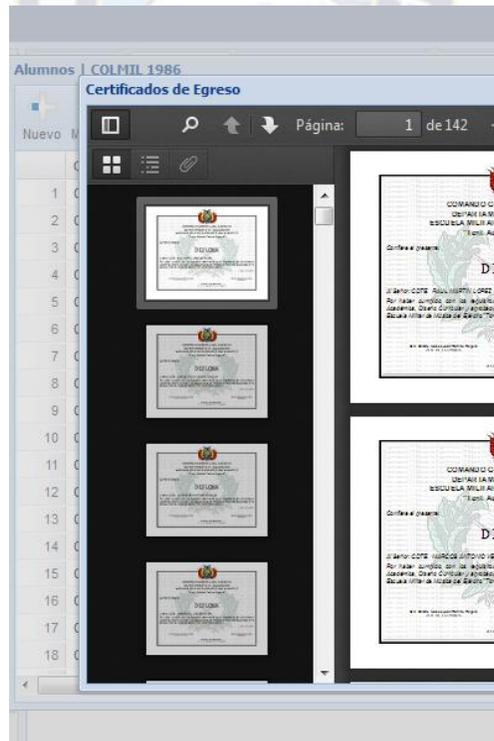
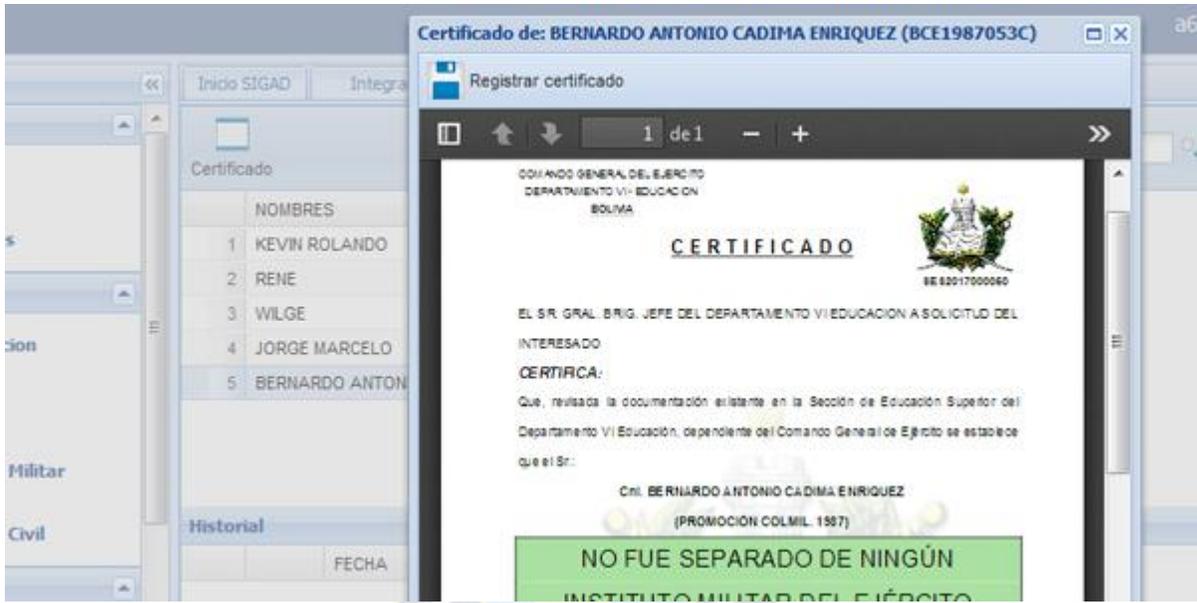
COMANDO GENERAL DEL EJERCITO
DEPARTAMENTO VI - EDUCACION
BOLIVIA

KARDEX ACADEMICO
BERNARDO ANTONIO CADIMA ENRIQUEZ

ESTUDIOS EN EL PAIS

Gestión	Grado	Instituto	Prom
1987	CDTE.	COLMIL	8.4134

Fig. 3.21: Reporte Kardex académico



Los integrantes del Ejército ya sean oficiales, suboficiales o sargentos pueden acceder al historial académico, a un Kardex académico a través de una aplicación android, la siguiente figura muestra la interfaz de ingreso.



Fig. 3.24: Interfaz de ingreso al sistema desde Android

3.7 CREACIÓN DEL CÓDIGO RUAM

RUAM(Registro único académico militar) consiste en un código único que el sistema genera en base a parámetros predefinidos, primeramente se normalizó los archivos Excel, se definió los campos necesarios.

	D	E	F	G	H	I	J	K	L
1	NOMBRES	APELLIDOS	PROMEDIO DE EGRESO	ORDEN DE MERITO	ATRIBUTO	AÑO DE AGRESO	RUAM	OBS	CI
2	JUAN MANUEL	PIEROLA SANDOVAL	9,285	1 /145	MB	1986	JPS1986001C		1515461 LP
3	JHONNY	CASTRO MARISCAL	9,251	2 /145	MB	1986	JCM1986002C		1544124 LP
4	GARY	GUEVARA IRIATE	9,110	3 /145	MB	1986	GG1986003C		1058987 LP
5	JORGE PASTOR	MENDIETA FERRUFINO	9,089	4 /145	MB	1986	JMF1986004C		2445457 LP
6	JAIME	LORA MOSCOSO	9,062	5 /145	MB	1986	JLM1986005C		1515454 LP
7	GUIDO SIPRIAN	URBINA MACEDA	9,018	6 /145	MB	1987	GUM1986006C		2115451 SC

Fig. 3.25: Archivo Excel normalizado

La generación del código RUAM consiste en extraer las iniciales del primer nombre, unirlo con las iniciales de los apellidos, el número de orden de mérito, la gestión y finalmente el instituto de origen como se muestra en la siguiente figura:

NOMBRES: JUAN MANUEL
APELLIDOS: PIEROLA SANDOVAL
ORDEN MERITO: 1/145
AÑO EGRESO: 1986
INSTITUTO MILITAR DE ORIGEN: COLMIL

JPS1986001C

Fig. 3.26: Generación del código RUAM

3.8 APLICACIÓN DE LA METODOLOGÍA TDD

Se hace uso de la herramienta simpletest para PHP. Para ilustrar como sería el procedimiento, vamos a mostrar la implementación en el lado del servidor, específicamente cuando se registran de manera conjunta todo el curso junto a los integrantes, esta parte se lo realiza cargando desde un archivo en formato Excel previamente normalizado como se muestra en la figura 3.25.

Nuestra primera serie de test será la generación del RUAM (Registro único académico militar), mostrado en la figura 3.27.

Creemos el código necesario para la función o el método que generará el RUAM

```
function generateRuam($nombres, $apellidos, $ordenMerito, $gestionEgreso, $codeOrigen) {
    $arrayApellidos = explode(" ", $apellidos);
    $ruam = "";
    $ruam.=substr($nombres, 0, 1);
    if (count($arrayApellidos) == 1) {
        $ruam.=substr($arrayApellidos[0], 0, 1);
    } else {
        $ruam.=substr($arrayApellidos[0], 0, 1) . substr($arrayApellidos[1], 0, 1);
    }
    $ruam.=$gestionEgreso;
    $arrayOrdenMerito = explode("/", $ordenMerito);
    $no = intval(trim($arrayOrdenMerito[0]));
    if ($no > 0 && $no < 10) {
        $ruam.="00" . $no;
    }
    if ($no >= 10 && $no <= 99) {
        $ruam.="0" . $no;
    }
    if ($no > 99) {
        $ruam.=$no;
    }
    $ruam.=$codeOrigen;
    return $ruam;
}
```

Fig. 3.27: Código para la generación RUAM

Creamos el primer test

```
<?php
include './simpletest/autorun.php';
include './functions.php';

class Prueba extends UnitTestCase {

    function test1() {
        $this->assertEqual("JPS1986001C", generateRuam("JUAN MANUEL", "PIEROLA SANDOVAL", "1/145", "1986", "C"));
    }
}
?>
```

Fig. 3.28: Código del test 1

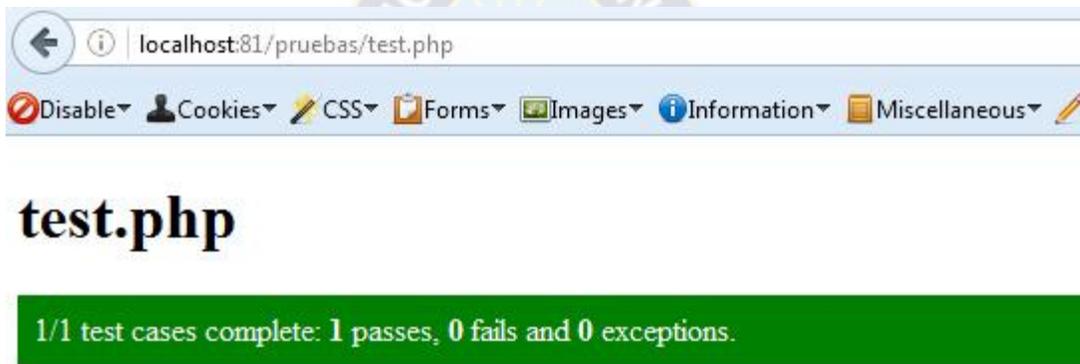


Fig. 3.29: Resultado del test 1, test pasado

```
<?php
include './simpletest/autorun.php';
include './functions.php';

class Prueba extends UnitTestCase {

    function test1() {
        $this->assertEqual("JPS1986001C", generateRuam("JUAN MANUEL", "PIEROLA SANDOVAL", "1/145", "1986", "C"));
    }

    function test2() {
        $this->assertEqual("JPS1986001C", generateRuam("JUAN MANUEL", "PIEROLA SANDOVAL", "1 /145", "1986", "C"));
    }

    function test3() {
        $this->assertEqual("ERROR", generateRuam("JUAN MANUEL", "PIEROLA SANDOVAL", "1145", "1986", "C"));
    }
}
?>
```

Fig. 3.30: Código de los tests

En la figura se puede ver que se añadió dos tests, el segundo test tiene el tercer parámetro un espacio adicional, como en algunos archivos Excel también está registrado de esa forma y debe también de pasar el test, pero en el tercer test, el tercer parámetro viene sin el símbolo de separación “/” por lo que no debería pasar el test como se muestra en la siguiente figura

test.php

Fail: Prueba -> test3 -> Equal expectation fails at character 7 with [JPS1986001C] and [JPS19861145C] at [C:\xampp\htdocs\pruebas\test.php line 28]

1/1 test cases complete: 2 passes, 1 fails and 0 exceptions.

Fig. 3.31: Resultado de los tests, no pasado

Luego implementamos para que el test pase realizando las modificaciones necesarias, adicionamos el código a la función o al método un mensaje de error para después no generar el RUAM y evitar la inserción del registro.

```
$arrayOrdenMerito = explode("/", $ordenMerito);
if (count($arrayOrdenMerito) != 2) {
    return "ERROR";
}
```

Fig. 3.32: Código añadido a la función para generar el error

```
function test3() {
    $this->assertEqual("ERROR", generateRuam("JUAN MANUEL", "PIEROLA SANDOVAL", "1145", "1986", "C"));
}
```

Fig. 3.33: Modificación del test 3 para pasar la prueba

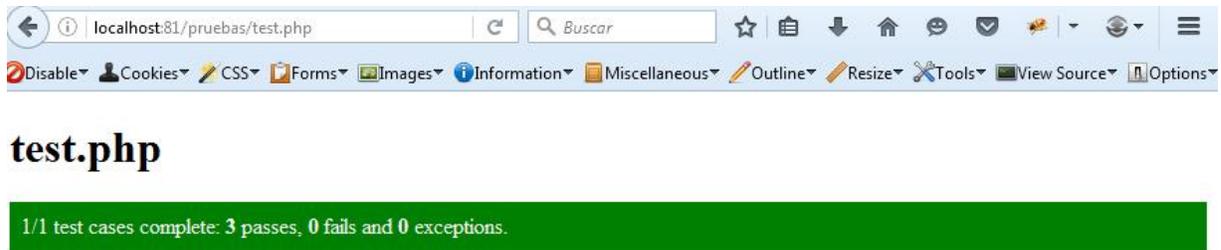


Fig. 3.34: Resultado del test 3, test pasado

Podemos ver que si pasa el test, esta es la forma en la que se implementa el resto del código que permite generar el RUAM. Finalmente podemos refactorizar el código.



CAPITULO IV

SEGURIDAD Y CALIDAD

4.1 SEGURIDAD

a) Verificación de acceso

El sistema tiene desarrollado un módulo de verificaciones y validación de usuarios (login y password) cada usuario adquiere una sesión para la autorización de uso del sistema.

b) Niveles de acceso

Al sistema pueden acceder diferentes tipos de usuarios (Administrador, Usuario, Invitado e Integrante) de la institución, el sistema restringe el acceso a procesos en base a la asignación de un rol.

El sistema permite habilitar nuevos usuarios como también modificar o eliminar usuarios, la eliminación de usuarios como la eliminación de cualquier registro es de manera lógica.

c) Copias de seguridad de la base de datos

Una de las prioridades es contar con copias de seguridad, el proceso consiste en guardar un backup de la base de datos cada cierto periodo de tiempo en un medio de almacenamiento como el disco duro y también un medio almacenamiento extraíble.

La copia de seguridad se guardara como resguardo a cualquier percance no previsto. Por lo cual el administrador o encargado del sistema deberá sacar una copia de la base de datos. Los archivos deberán contener los siguientes datos al momento de grabarlos: Fecha y hora.

4.2 PROPUESTAS DE POLÍTICAS DE SEGURIDAD

En este apartado se dará una clasificación sobre las políticas de seguridad relacionadas a los equipos.

a) Equipamiento

Todo equipo de cómputo que esté conectado a la red debe sujetarse a las normas y procedimientos de instalación que emite la unidad de sistemas.

Todo equipo que sea de propósito específico y tenga una misión crítica asignada, requiere estar ubicado en un área que cumpla con los requerimientos de: Seguridad física, condiciones ambientales y alimentación eléctrica. La protección física de los equipos corresponde a quienes en un principio se les asigna y corresponde notificar los movimientos en caso de que existan.

b) Mantenimiento

La unidad de sistemas, es la encargada del mantenimiento y correctivo de los equipos, la conservación de instalación, la verificación de la seguridad física, y su acondicionamiento específico, para tal fin debe emitir las normas y procedimientos respectivos.

Corresponde a la unidad de sistemas dar a conocer las listas del personal, que tenga acceso a los equipos y brindar los servicios de mantenimiento básico.

Se sugiere como política de seguridad la implantación de:

- Realizar el mantenimiento del sistema de manera periódica como sugerencia anualmente para su buen funcionamiento
- Documentar los cambios que se realicen al sistema.

c) Acceso a los sistemas de información

Tendrán acceso al sistema de información solo aquellos usuarios autorizados por la unidad. La instalación y uso de los sistemas de información se rigen por reglamento de uso de la unidad de sistemas.

Los servidores de bases de datos son delicados, por lo que se prohíben los accesos de cualquier personal, excepto para el administrador de base de datos.

4.3 CALIDAD

Para el presente proyecto se realiza los siguientes atributos de calidad: Funcionalidad, Usabilidad, Mantenimiento y Satisfacción del Usuario.

4.3.1 Funcionalidad

Se analiza los atributos de la funcionalidad.

La adaptabilidad se refiere a la capacidad del producto software un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

El sistema es multiplataforma, la interfaz de usuario se adapta a cualquier navegador web, como también a cualquier dispositivo que se utilice como terminal, entonces el sistema es adaptable

La exactitud consiste la capacidad de entregar resultados correctos.

Después de que el sistema fue probado por un periodo de tiempo por los usuarios, no se encontraron errores en cuanto a los resultados. En consecuencia cumple con la exactitud.

La interoperabilidad es la habilidad de interactuar con determinados sistemas.

El sistema está construido de tal manera que puede crearse módulos para interactuar con otros sistemas como por ejemplo la generación y el consumo de servicios web, entonces es interoperable.

El cumplimiento consiste en la compatibilidad con estándares, convenciones o regulaciones.

El sistema fue realizado todo bajo software libre, por ser institución perteneciente estado bajo el Decreto Supremo No. 1793.

La seguridad es la habilidad de prevenir uso no autorizado, tanto intencional como accidental, de programas y datos.

Al sistema solo se puede acceder a través de un usuario y contraseña, el sistema restringe el acceso a procesos en base a la asignación de un rol, la configuración adecuada del servidor y el acceso a la base de datos aumentan el nivel de seguridad. Entonces el sistema cumple con la seguridad

4.3.2 Usabilidad

Se realiza una evaluación del sistema en base a encuestas de un grupo de usuarios.

Tabla 4.1: Preguntas de usabilidad

Pregunta	PONDERACION
¿La interfaz de usuario es amigable?	85%
¿El manejo del sistema es comprensible y sencillo?	80%
¿El sistema satisface todos los requerimientos?	90%
¿Los datos de salida son confiables?	98%

Para calcular la usabilidad se utilizará la siguiente fórmula:

$$U = \frac{\sum_{i=1}^n x_i}{n}$$

Donde:

$\sum_{i=1}^n x_i$ = es la suma de ponderaciones.

n = cantidad de preguntas

$$\sum_{i=1}^4 x_i = 85 + 80 + 90 + 98 = 353$$

$$n = 4$$

Entonces:

$$U = \frac{\sum_{i=1}^4 x_i}{n} = \frac{353}{4} = 88.25$$

El resultado nos muestra la usabilidad, que es de un 88.25%.

4.3.3 Facilidad de Mantenimiento

Para calcular la facilidad de mantenimiento se tomó los datos de la última iteración o sprint.

M_T = Numero de módulos de la liberación actual.

F_C = Número de módulos de la liberación actual que cambiaron.

F_A = Número de módulos de la liberación actual que se agregaron.

F_D = Número de módulos de la liberación anterior que se borraron en la liberación actual.

$$M_T = 6$$

$$F_C = 1.$$

$$F_A = 2.$$

$$F_D = 0$$

El índice de madurez del software se calcula de la forma siguiente:

$$IMS = \frac{M_T - (F_C + F_A + F_D)}{M_T}$$

$$IMS = \frac{5 - (0 + 1 + 0)}{5}$$

$$IMS = 0.8$$

Por lo tanto, el índice de Madurez del Sistema es de 80%, lo que significa que es mantenible y no requiere de grandes esfuerzos para la adición y el cambio de funcionalidades.

4.3.4 Satisfacción del usuario

Se diseñó un formulario que toma en cuenta la verificación y seguimiento, el cual fue distribuido entre el personal dependiente de la institución donde se obtuvieron los siguientes resultados.

Jefe del Departamento VI	1
Jefes de Unidad	5
Operadores de las diferentes unidades	5

FORMULARIO DE PRUEBA DE SIGAD
Nombre completo:
Fecha/hora
¿El sistema cumple con los requerimientos de la unidad a la cual pertenece? SI NO
Observaciones:
Sugerencias:
FIRMA

Fig. 4.1: Formulario de prueba de SIGAD

Después de realizar la encuesta se obtuvieron los siguientes resultados que se muestran a continuación.

Tabla 4.2: Resultados Formulario de prueba de SIGAD

Dependiente	SI Cumple	NO Cumple
Jefe del Departamento VI	1	0
Jefe de Unidad	5	0
Operadores de las diferentes unidades	5	0
TOTAL	11	0

De los resultados podemos establecer que el sistema si cumple con lo requerido por la institución. Mencionar también que algunos usuarios no siempre estaban de acuerdo al 100%.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Se normalizaron los archivos Excel de distintos formatos a dos formatos. Se generaron los reportes necesarios para el kardex, diplomas y certificados (certificado de antecedentes, certificado de estudios en el exterior, certificado de exámenes de ascenso y profesorado militar) que emite la institución a personal militar como también ciudadanía en general que esté interesado en obtener una certificación, El acceso restringido al sistema permite a los usuarios solo acceder a determinada información. Los trámites que antes tardaban en promedio 24 horas, ahora se reduce a una hora.

El presente proyecto y su implementación llena todas las expectativas de manera satisfactoria en la institución del Departamento VI – Educación del Ejército.

5.2 RECOMENDACIONES

Habiendo concluido el proyecto y mirando hacia un futuro mediato e inmediato se realiza las siguientes recomendaciones:

- Incorporar normas y políticas de uso del sistema.
- Realizar evaluaciones periódicas, con la finalidad de determinar nuevas necesidades.

BIBLIOGRAFÍA

- Balestrini, M. (2001). *Cómo se elabora el proyecto de investigación* Caracas: Consultores Asociados BL, servicio editorial, 5ta edición.
- Goncalves M. (2003). *La relación investigador-comunidad en el trabajo psicosocial comunitario. Su lugar en la producción del conocimiento.* Caracas–Venezuela Facultad de Humanidades y Educación de la Universidad Central de Venezuela.
- Hernández, R. Fernández, C. y Baptista, P. (1999). *Metodología de la investigación.* México: McGraw-Hill.
- Pressman R. (2010). *Ingeniería de Software Un enfoque práctico.* 7ma. Ed.; McGraw-Hill/Interamericana de España: México D.F.
- Sommerville Ian. (2005). *Ingeniería de Software Séptima edición.*; Pearson Educación. S.A. Madrid. 2005
- Schumuller J. (2001). *UML en 24 horas*, 1ra. Ed.; Prentice Hall: Mexico.
- Blé Jurado Carlos (2010). *Diseño Ágil con TDD.* Primera edición. Safe Creative
- Palacio J. (2008). *Flexibilidad con Scrum.* Safe Creative.
- Giovanno Joaquin (2015). *Sistema web de información académico financiero centro de capacitación técnica Comatec.* COMATEC, La Paz.
- Marizol Roque Caizana (2014). *Sistema web de información y seguimiento académico Caso: programa de admisión prefacultativa carrera de psicología,* Carrera de Psicología UMSA, La Paz.

Ghilmar Emigdio Copa Mamani (2014). *Software de gestión y seguimiento académico*
Caso: Centro de capacitación técnica CEINF rm 175/13. Centro de
capacitación técnica CEINF, La Paz.

Freddy Julio Colque Carvajal (2009). *Sistema de información y seguimiento académico*
para el Instituto Nueva Esperanza. Instituto Nueva Esperanza, La Paz.

Franz Limachi Mamani (2008). *Sistema de seguimiento académico Facultad de*
Odontología. Facultad de Odontología UMSA, La Paz.

Victor Hugo Calcina Lopez (2007). *Sistema de seguimiento académico Carrera de*
Arquitectura Universidad Pública de El Alto. Carrera de Arquitectura UPEA,
El Alto La Paz.

(2016) Ejército de Bolivia. Recuperado de <http://www.ejercito.mil.bo/>



DOCUMENTOS

