



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**“DESARROLLO DE UN SISTEMA WEB DE GESTIÓN FINANCIERA
PARA LA FUNDACIÓN “JASPE” UTILIZANDO LA TECNOLOGÍA
DE BASE DE DATOS NO-SQL”**

Trabajo de titulación presentado para optar al grado académico de:
INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES: MOROCHO GUASHPA WILLIAM PATRICIO
CARRANZA LUCAS JERSON KENNY

TUTOR: DR. JULIO SANTILLÁN

Riobamba-Ecuador

2017

©2017, William Patricio Morocho Guashpa, Jerson Kenny Carranza Lucas

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del Trabajo de Titulación certifica que: El Trabajo de investigación: “DESARROLLO DE UN SISTEMA WEB DE GESTIÓN FINANCIERA PARA LA FUNDACIÓN “JASPE” UTILIZANDO LA TECNOLOGÍA DE BASE DE DATOS NO-SQL”, de responsabilidad de los señores: William Patricio Morocho Guashpa, Jerson Kenny Carranza Lucas, ha sido minuciosamente revisado por los miembros del Tribunal, quedando autorizada su presentación.

NOMBRES	FIRMAS	FECHA
Ing. Washington Luna DECANO DE LA F.I.E.	_____	_____
Ing. Patricio Moreno DIRECTOR DE LA E.I.S.	_____	_____
Dr. Julio Santillán DIRECTOR DE TESIS	_____	_____
Ing. Germania Veloz MIEMBRO DEL TRIBUNAL	_____	_____

“Nosotros, William Patricio Morocho Guashpa y Jerson Kenny Carranza Lucas somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.”

William Patricio Morocho Guashpa

Jerson Kenny Carranza Lucas

DEDICATORIA

A Dios, por permitirme llegar a este momento tan especial en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más. A mi padre quien con sus apoyo económico, moral y sus consejos ha sabido guiarme para culminar mi carrera profesional. A mi madre por ser la persona que me ha acompañado durante todo mi trayecto estudiantil y de vida. A mis hijas Valentina y Allison quienes han estado junto a mí durante este arduo camino para convertir me en un profesional. A mi esposa, por su comprensión y apoyo en cada instante.

William.

Este trabajo de titulación va dedicado a en primer lugar a Dios y a mi familia especialmente a mi madre Victoria por brindarme el apoyo incondicional tanto verbal y económico, ya que sin su apoyo no hubiera sido posible este anheloso logro.

A mi hermano Jefferson y Don Carlitos, por darme el aliento y sustento para seguir estudiando día a día, con sus palabras me dio energía para seguir a pesar de los tropiezos que he tenido durante la carrera.

Jerson.

AGRADECIMIENTO

Agradecimiento especial a la Escuela Superior Politécnica de Chimborazo por la educación que nos brindó y nos permitió culminar esta etapa profesional tan importante.

Gracias a nuestro tutor Dr. Julio Santillán quien nos ha guiado en el proceso final de nuestra carrera universitaria; gracias por el tiempo y su predisposición en todo momento.

Y por último pero no menos importante a todas esas personas que nos han respaldado tanto como en lo académico, emocional, dejando en nosotros gratitud y grandes recuerdos.

William

Jerson

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xiii
ÍNDICE DE ANEXOS	xiv
RESUMEN.....	xv
SUMMARY.....	xvi
INTRODUCCIÓN	xvii
CAPÍTULO I	
1 MARCO TEÓRICO REFERENCIAL.....	21
1.1 Sistema Web	21
1.1.1 <i>Concepto</i>	21
1.1.2 <i>Ventajas de un Sistema Web</i>	22
1.1.3 <i>Arquitectura de la aplicación web.</i>	22
1.2 Lenguajes de programación para la web	23
1.2.1 <i>Node.js</i>	23
1.2.2 <i>JavaScript</i>	24
1.2.3 <i>Angularjs</i>	25
1.3 Bases de datos	26
1.3.1 <i>Bases de datos noSQL</i>	26
1.3.1 <i>MongoDB</i>	31
1.3.2 <i>Operaciones MongoDB.</i>	34
1.4 Norma ISO/IEC 9126-3.....	37
1.5 Alcance.....	38
CAPÍTULO II	
2 MARCO METODOLÓGICO.....	39
2.1 Justificación metodología	39
2.2 Fases de la metodología	40
2.3 Tipos y roles de usuarios.	42
2.4 Fase de exploración.....	43
2.4.1 <i>Recopilación de información</i>	43

2.4.2	<i>Diagnosticar situación de la fundación</i>	44
2.4.3	<i>Gestión de riesgos</i>	44
2.4.4	<i>Identificación de riesgos</i>	45
2.5	Fase de planificación	51
2.5.1	<i>Historias de usuarios</i>	51
2.5.2	<i>Prioridad en lo requerimientos</i>	53
2.5.3	<i>Estimación del proyecto</i>	54
2.5.4	<i>Iteración del proyecto</i>	55
2.5.5	<i>Asignar Pareja</i>	55
2.6	Fase de Diseño	56
2.6.1	<i>Arquitectura del sistema</i>	56
2.6.2	<i>Estándar de codificación</i>	59
2.6.3	<i>Diseño de la base de datos</i>	61
2.6.4	<i>Diseño de la interfaz de usuario</i>	62
2.7	Fase de codificación	63
2.7.1	<i>Iteración 1</i>	64
2.7.2	<i>Iteración 2</i>	65
2.7.3	<i>Iteración 3</i>	66
2.7.4	<i>Iteración 4</i>	66
2.7.5	<i>Iteración 5</i>	67
2.7.6	<i>Iteración 6</i>	67
2.8	Fase de pruebas	68
2.9	Fase de finalización	69
2.10	Gestión del proyecto	69
CAPÍTULO III		
3	ANÁLISIS DE RESULTADOS	71
3.1	Generalidades	72
3.2	Determinación de tipos de usuario y tamaños de la muestra.	72
3.3	Definición de los parámetros	74
3.4	Criterios de evaluación	74
3.5	Análisis de Resultados	75
3.5.1	<i>Funcionalidad</i>	77
3.6	Análisis de Rendimiento de la base de datos noSQL frente a una base de datos SQL(PostgreSQL)	84

3.6.1	<i>Plataforma Hardware</i>	84
3.6.2	<i>Plataforma Software</i>	85
3.6.3	<i>Consultas a los gestores de bases de datos</i>	85
3.6.4	<i>Métricas</i>	86
3.6.5	<i>Pruebas de rendimiento</i>	87
3.6.6	<i>Análisis de Resultados</i>	89
	CONCLUSIONES	92
	RECOMENDACIONES	93
	GLOSARIO	
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1: SQL vs noSQL.....	27
Tabla 1 - 2: Tipos y roles de usuarios.....	42
Tabla 2- 2: Tipos de riesgos.....	45
Tabla 3 - 2: Análisis de riesgos.....	46
Tabla 4 - 2: Valoración de Probabilidad.....	47
Tabla 5 - 2: Valoración del impacto.....	47
Tabla 6 - 2: Valoración de la exposición del riesgo.....	48
Tabla 7 - 2: Determinación de la exposición al riesgo.....	48
Tabla 8 - 2: Determinación de la prioridad de riesgo.....	49
Tabla 9 - 2: Hoja de gestión de Riesgo R1.....	50
Tabla 10 - 2: Requisitos del sistema.....	51
Tabla 11 - 2: Prioridad de requerimientos.....	53
Tabla 12-2: Release planning.....	53
Tabla 13 - 2: Asignar Pareja.....	56
Tabla 14 - 2: REST.....	57
Tabla 15 - 2: Metáfora del sistema de la Arquitectura del sistema.....	58
Tabla 16 - 2: Primera tarea ingeniería de la arquitectura del sistema.....	58
Tabla 17 - 2: Segunda tarea ingeniería de la arquitectura del sistema.....	59
Tabla 18 - 2: Metáfora del sistema del estandar de codificación.....	60
Tabla 19 - 2: Primera tarea de ingeniería del estándar de codificación.....	60
Tabla 20 - 2: Metáfora del Diseño de la base de datos.....	61
Tabla 21 - 2: Tarea de ingeniería de la metáfora del Diseño de la base de datos.....	61
Tabla 22 - 2: Tarea de ingeniería de la metáfora del Diseño interfaz de usuario.....	62
Tabla 23 - 2: Tarea de ingeniería de la metáfora del Diseño de la interfaz.....	63
Tabla 24 - 2: Iteración 1.....	64
Tabla 25 - 2: Historia de Usuario Registro de estudiantes.....	64
Tabla 26 - 2: Elaboración de la interfaz de usuario Registro de Estudiantes.....	65
Tabla 27 - 2: Elaboración de la interfaz de usuario Registro de Estudiantes.....	65
Tabla 28 - 2: Iteración 2.....	65
Tabla 29 - 2: Iteración 3.....	66
Tabla 30 - 2: Iteración 4.....	66
Tabla 31 - 2: Iteración 5.....	67
Tabla 32 - 2: Iteración 6.....	67
Tabla 33 - 2: Prueba de Aceptación 1.....	68
Tabla 1 - 3: Definición de los parámetros.....	74
Tabla 2 - 3: Tabla de Likert.....	74
Tabla 3 - 3: Resultados pregunta 1 y 2.....	76
Tabla 4 - 3: Resultados 3 y 4.....	78
Tabla 5 - 3: Resultados pregunta 5 y 6.....	80
Tabla 6 - 3: Resultados pregunta 7 y 8.....	80
Tabla 7 - 3: Cumplimiento funcional.....	82
Tabla 8 - 3: Plataforma Hardware.....	83
Tabla 9 - 3: Plataforma Software.....	84
Tabla 10 - 3: Test N°1.....	84
Tabla 11 - 3: Test N°2.....	84

Tabla 12 - 3: Test N°3.....	85
Tabla 13 - 3: Inserción masiva	87
Tabla 14 - 3: Consultar masiva	88
Tabla 15 - 3: Eliminar masivo.....	88

ÍNDICE DE FIGURAS

Figura 1 - 1: Arquitectura de la aplicación web.....	23
Figura 2 - 1: Estructura de una base datos clave-valor.....	28
Figura 3 - 1: Estructura de una base de datos de grafos.....	29
Figura 4 - 1: Ejemplo de una base de datos de columnas.....	29
Figura 5 - 1: Formato base documental.....	30
Figura 6 - 1: Clasificación de bases de datos noSQL.....	31
Figura 7 - 1: Insertar una colección.....	34
Figura 8 - 1: Insertar varias colecciones.....	34
Figura 9 - 1: Consultar.....	35
Figura 10 - 1: Actualizar varios documentos.....	35
Figura 11 - 1: Eliminar varios documentos.....	35
Figura 1 - 2: Estimación del proyecto en cocomo.....	55
Figura 2 - 2: Diagrama de despliegue.....	58

ÍNDICE DE GRÁFICOS

Gráfico 1 - 2: Burndown chart.....	71
Gráfico 2 - 3: Satisfacción del sistema.....	76
Gráfico 3 - 3: Indicador de Adecuación.	79
Gráfico 4 - 3: Indicador de Exactitud.....	82
Gráfico 5 - 3: Funcionalidad.....	84
Gráfico 5 - 3: Tiempo promedio de inserción de los dos gestores.....	88
Gráfico 6 - 3: Tiempo de respuesta de los select.....	89
Gráfico 7 - 3: Tiempo de respuesta de los " DELETE "	90

ÍNDICE DE ANEXOS

Anexo A: Hojas de gestión de riesgos.

Anexo B: Estándar de Codificación

Anexo C: Diseño de la Base de datos noSQL

Anexo D: Historias de usuario

Anexo E: Planificación del sistema

Anexo F: Pruebas de aceptación

Anexo G: Encuesta

Anexo H: Manual de usuario

Anexo I: Análisis de rendimiento de la base de datos noSQL frente a una SQL(PostgreSQL)

RESUMEN

El presente trabajo de titulación tuvo como objetivo el desarrollo de un sistema web de gestión financiera para la fundación “JASPE” utilizando la tecnología de base de datos noSQL, este sistema fue diseñado para automatizar los procesos de gestión financiera. Para el desarrollo se empleó la metodología ágil de Programación Extrema (XP), esta metodología permite trabajar colaborativamente entre los miembros del equipo de desarrollo y el usuario, para la recopilación de requerimientos, se usó, la fase de exploración que brinda la metodología XP. Para codificación del sistema informático se usa Mongodb, Express, Angularjs y Nodejs (MEAN), para el almacenamiento de los datos se usa la base de datos nosql mongodb que está orientada a documentos. SIWEFI como se ha denominado el Sistema Web Financiero cuenta con los siguientes módulos: registro de estudiante, cobro de matrícula y pensiones, pagos al personal y el módulo contable con sus respectivos reportes. Posterior a la etapa de desarrollo se hicieron capacitaciones al personal de la Fundación. Las pruebas y correcciones fueron realizadas bajo los directivos de Fundación JASPE. Una vez implantado el sistema informático se hicieron encuestas a 70 personas para determinar la funcionalidad del mismo. Los indicadores que se usaron para determinar la funcionalidad fueron tomados del estándar internacional para la evaluación de la calidad del software ISO 9126. El análisis de los resultados reflejó que el 73% estaban totalmente de acuerdo con la funcionalidad del sistema que corresponde a 51 usuarios, concluyendo que el sistema es apto para la utilización de la gestión de los procesos administrativos de la fundación. Para garantizar la funcionalidad del sistema se recomienda seguir el manual de usuario.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <SISTEMA WEB FINANCIERO (SIWEFI)>, <NODEJS (LENGUAJE DE PROGRAMACIÓN)>, <MONGODB (BASE DE DATOS)>, <PROGRAMACIÓN EXTREMA (XP)>, <ANGULARJS (FRAMEWORK)>, <MEAN STACK (FRAMEWORK)>

SUMMARY

The present work of degree had as objective the development of a web system of financial management for the Foundation JASPE, using the technology of non-SQL database. This system was designed to automate the processes of financial management. For the development was used the Agile Methodology Extreme Programming (XP), this methodology allowed to work collaboratively between the members of the development team and the user; for the compilation of requirements used the exploration phase provided by the methodology “XP”; for the coding of the computer system used Mongoddb, Express, Angularjs and Nodejs (MEAN); and, for the storage of data used the Mongoddb noSQL database which is oriented to documents. (FIWESY) as it has been called the Financial Web System has the following modules: Student registration, payment of tuition and pensions, payments to staff and the accounting module with their respective reports. Back to the stage of development were training to the staff of the Foundation. The tests and corrections were carried out under the management of Foundation JASPE. Once in place the computer system were made surveys to 70 people in order to determine its functionality. The indicators which were used to determine its functionality. The indicators which were used to determine the functionality were taken from the international standard for the evaluation of the quality of ISO 9126 software. The analysis of the result reflected that 73% were totally in agreement with the functionality of the Foundation. It is recommended to follow the user manual in order to guarantee the functionality of the system.

KEY WORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <SOFTWARE ENGINEERING>, <FINANCIAL WEB SYSTEM (FIWESY)>, <NODEJS (PROGRAMMING LANGUAGE)>, <MONGODB (DATABASE)>, <EXTREME PROGRAMMING (XP)>, <ANGULARJS (FRAMEWORK)>, <MEAN STACK (FRAMEWORK)>

INTRODUCCIÓN

ANTECEDENTES

Desde décadas atrás la tecnología y los sistemas web se han convertido en un pilar fundamental para diferentes actividades; como por ejemplo las empresariales y en la educación, este motivo ha llevado que múltiples empresas públicas y privadas adquieran sistemas web para administrar de mejor manera la información.

Los sistemas web de gestión financiera permiten administrar cada uno de los recursos que una empresa tiene, de esta manera, afirmar que los egresos que la empresa tiene serán cubiertos para que la misma siga funcionando.

En la actualidad la automatización de procesos se ha convertido en prioridad para las entidades en el sector educativo, ya que permite manipular información de una manera rápida y eficaz. En la fundación “JASPE” ubicada en el cantón Colta provincia de Chimborazo, brinda cursos de capacitación en distintas áreas como belleza, peluquería, música, agricultura, ganadería, elaboración de productos lácteos, panadería y pastelería. Debido a la variedad de cursos existentes, la fundación “JASPE” tiene gran afluencia de estudiantes, los mismos que generan una considerable cantidad de información.

Esta información se manipula por la administración de manera manual y no estructurada (hojas de cálculo, procesadores de texto, correo electrónico) es decir que esta organización carece de la utilización de un sistema informático web; por lo tanto tiene la necesidad de contar con dicho sistema que permita la automatización de los procesos de gestión financiera. El sistema permite tener datos actualizados para la respectiva toma de decisión de la fundación “JASPE”.

El desarrollo del Sistema Web Financiero denominado “SIWEFI”, permite automatizar los procesos de cobranza, los procesos básicos de contabilidad, procesos de costos realizando informes necesarios para la fundación.

La metodología a utilizar para el desarrollo del sistema SIWEFI es la extreme programming conocido mundialmente por sus siglas XP, el mismo que permite la realización del sistema web de forma iterativa e incremental. La metodología permite realizar test mediante las pruebas de aceptación.

La fundación “JASPE” por los distintos procesos genera gran cantidad de datos, los mismos que serán almacenados en el servidor de base de datos MongoDB, el sistema web “SIWEFI” utiliza la tecnología Angular y Javascript.

La arquitectura del sistema SIWEFI es el modelo-vista-controlador, las mismas que están separadas en distintas carpetas. La carpeta *model* permite tener acceso a la base de datos, ingresar datos o realizar las consultas necesarias al servidor de base de datos. La carpeta *public* (vista) tendrá las librerías css, librerías JavaScript, templates e imágenes. La carpeta *controller* permite ejecutar código que recibe del usuario y del servidor, es decir que permite la comunicación entre el modelo y la vista.

El sistema web SIWEFI utiliza el gestor de base de datos MongoDB, el cual es una base de datos noSQL que está orientado a documentos, MongoDB tiene un concepto distinto al de las bases de datos tradicionales. En la base de datos MongoDB no existe tablas, ni registros y no cuenta con lenguaje SQL por esta razón puede procesar gran cantidad de datos con mucha rapidez.

Formulación del problema

La Fundación “JASPE” realiza diferentes procesos manuales, los cuales provocan pérdida de información y la demora en las distintas transacciones, estos procesos han ocasionado malestar en los distintos tipos de usuarios.

Sistematización del problema.

¿Cómo el sistema web de gestión financiera ayudaría a la toma de decisiones en la fundación “JASPE”?

¿Cuáles son los beneficios de aplicar un sistema web de gestión financiera en la fundación “JASPE”?

¿La utilización de tecnologías como la base de datos NO-SQL ayudaría a solucionar el problema de demora en la entrega de la información?

Justificación

Justificación teórica.

En el aspecto teórico se alega por la importancia que se encuentra en las leyes y estatutos que promueven el cambio de la cultura organizacional, a través de modelos enfocados en procesos. Haciendo el uso de herramientas de software como apoyo para la transformación innovadora para la Fundación.

Los sistemas orientados a la web han permitido que se facilite el trabajo colaborativo, proporcionando la distribución de la información sin importar las distancias de los usuarios que accedan a estos sistemas informáticos, también ha permitido que sea compatible con las diferentes plataformas existentes; además se puede acceder a la información en cualquier momento que el usuario lo desee.

Actualmente los sistemas web han evolucionado y han aparecido nuevas tecnologías para almacenar información como es el caso de las bases de datos noSQL, el mismo que tiene un concepto muy diferente al de las bases de datos relacionales. Uno de los beneficios de estas bases es la facilidad que nos brinda para poder escalar; estas bases de datos no SQL están concebidas para obtener una altísima capacidad de volumen de almacenamiento y velocidad de proceso de la información.

Justificación aplicativa.

Debido al crecimiento en las diferentes actividades que se desarrollan dentro de la fundación “JASPE”, existe retardo en las transacciones. Actualmente la fundación “JASPE” no cuenta con ningún sistema web; por lo que se propone crear un sitio web cuya implementación permite a los usuarios realizar transacciones de una manera fácil y eficaz.

El uso de sistemas web se ha convertido en una de las principales herramientas para agilizar procesos y almacenar información de una manera ordenada y oportuna.

Esta aplicación web de gestión financiera cuenta con los módulos de: cobranza, contable, de costos, y de reportes. El mismo que contribuye a llevar una contabilidad más organizada y fiable, ya que la información proporcionada será actualizada y precisa.

Los beneficios que se obtienen con este sistema, están enfocados a mejorar los procesos manuales de la institución cambiándolos por procesos automatizados que disminuyan los tiempos de transacción dentro de la fundación.

Objetivos

Objetivo General

Desarrollar un sistema web de gestión financiera para la fundación “JASPE” utilizando la tecnología de base de datos noSQL.

Objetivos específicos

- Diagnosticar la situación actual de los procesos que lleva a cabo la fundación “JASPE”.
- Desarrollar el sistema web de gestión financiera utilizando la metodología Extreme Program (XP).
- Validar el sistema web utilizando el estándar ISO 9126 para determinar la calidad de software bajo el criterio de funcionalidad.

CAPÍTULO I

1 MARCO TEÓRICO REFERENCIAL

1.1 Sistema Web

El rápido avance de la tecnología genera el desarrollo de nuevas metodologías para la realización de los sistemas web. Cada software orientado a la web tiene un alto porcentaje de interactividad y capacidad de generar estilos a los sitios.

Este avance es posible gracias al desarrollo del internet. Actualmente el internet es una importante fuente de comunicación y los sistemas web surgen como instrumentos imprescindibles para la difusión de la información, los mismos que permiten que los usuarios de la red tengan múltiples servicios (Garrido, 2004, pp. 2-6).

1.1.1 *Concepto*

Se denomina sistema web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web; a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación (Software) que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. (Mora, 2002, pp. 5-10).

Los sistemas web son populares gracias a lo útil que es el navegador web como cliente liviano y también es independiente al sistema operativo siendo de esta manera multiplataforma, facilitando las actualizaciones y manteniendo las aplicaciones web sin distribuir e instalar el software a todos los usuarios potenciales.

1.1.2 *Ventajas de un Sistema Web*

Las ventajas de los sistemas web con respecto a las aplicaciones de escritorio son varias, logrando acoplar de esta manera todos los recursos de la empresa de una manera más práctica.

Uno de los beneficios de utilizar este tipo de sistemas web es el funcionamiento, es decir no importa que sistema operativo esté instalado en el computador del cliente, estas aplicaciones funcionan indistintamente de ello.

Con los sistemas web no se reduda creando clientes para todos los sistemas operativos (Windows, Mac OS X, GNU/ Linux) estos sistemas web se realizan una sola vez y la ejecución es igual en todas partes (Mora, 2002, pp. 47-55).

Los sistemas web no ocupan espacio en el disco duro local y las actualizaciones se realizan de manera inmediata, ya que el software es gestionado por el desarrollador, esto permite que el sistema web siempre sea la última versión.

Otro beneficio de los sistemas que son desarrollados para web es el trabajo a distancia, pudiendo realizar cualquier trabajo con mayor facilidad y desde cualquier parte del mundo lo único que se necesita es un computador con un excelente navegador web y conexión a internet.

1.1.3 *Arquitectura de la aplicación web.*

La arquitectura de una aplicación web está estructurada con el patrón de diseño MVC (Modelo, Vista y Controlador). La capa modelo es el lugar donde se trabaja con los distintos tipos de datos; es decir, esta capa contiene todos los mecanismos necesarios para acceder a la información. Los datos están almacenados en una base de datos, por lo que este modelo tiene los correspondientes selects, updates, inserts, deletes entre otros. La capa vista tiene el código de la interfaz de usuario de la aplicación. La capa controlador contiene el código necesario para responder a las diferentes acciones que se solicita en la aplicación. Cada una de estas capas están estrechamente relacionadas como se puede observar en la figura 1-1(Mora, 2002, pp. 39-45).

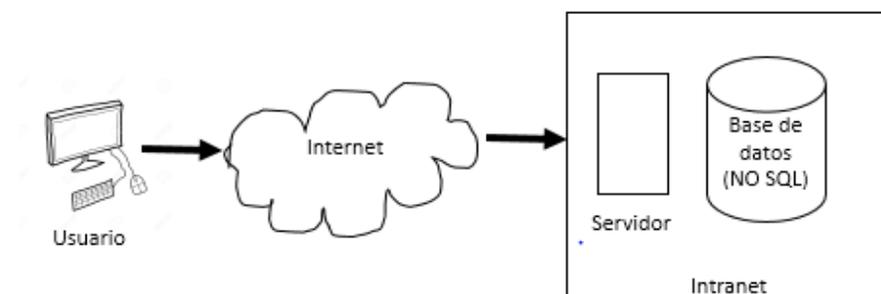


Figura 1 - 1: Arquitectura de la aplicación web.

Realizado por: Carranza Jerson, Morocho William. 2017

1.2 Lenguajes de programación para la web

Mientras el internet evoluciona, las demandas de los usuarios se incrementan, las peticiones cada vez son de sistemas web más sofisticados y personalizados, por esta razón los programadores generan varias soluciones. Esta evolución permite que nuevos lenguajes de programación aparezcan, lo que produce que la realización de las páginas dinámicas sea fáciles de programar, ya que estas páginas permiten la interactividad con el usuario y el almacenamiento de información es a través de un motor de base de datos (Mora, 2002, pp. 39-45).

Entre los lenguajes de programación que se utilizan para el desarrollo del sistema web de gestión tenemos los siguientes:

1.2.1 *Node.js.*

Node.js es simplemente JavaScript del lado del servidor. Las sintaxis y las herramientas son similares con la codificación de JavaScript, por esta razón se ejecuta en el navegador textualmente y funciona correctamente en el servidor ya que node.js es un intérprete de JavaScript. Node.js además tiene herramientas exclusivas las mismas que permiten ser un intérprete de alto nivel de JavaScript (Krol,2015,pp 29-30).

Este entorno de ejecución es multiplataforma y de código abierto. La capa del servidor está basado en el lenguaje de programación ECMAScript que tiene una arquitectura netamente orientada a eventos y basado en el motor v8 de google (Krol,2015,pp 8-9).

Este lenguaje de programación está construida sobre el tiempo de ejecución de JavaScript de google Chrome pensado en el futuro de las aplicaciones escalables de la red. Node.js es una plataforma

que permite a las aplicaciones ejecutarse en tiempo real y distribuirse sin bloqueos lo que admite que la aplicación sea más ligera y muy eficiente (Lightweight Node.js, 2016, pp 127-132).

El motor V8 es el entorno donde JavaScript se ejecuta, el mismo que fue creado para google Chrome, desde el 2008 el código fuente fue abierto pasando a ser software libre, todas las características del motor V8 está escrito en C++ lo cual permite compilar el código fuente como código de máquina en vez de interpretarlo en tiempo real (Pascuali, 2013, pp 15).

Cuando se usa node.js es importante saber a qué se llama NPM, éste es un administrador (gestor) de todos los paquetes existentes de node.js el mismo que nos permite descargar o enlazar librerías de JavaScript ya que cuando se trabaja con node.js en un proyecto nuevo este viene totalmente vacío.

1.2.2 JavaScript

Es el único lenguaje utilizado del lado del cliente lo que en aplicaciones web se refiere, esto permite que se implemente en el navegador web como si fuera parte de ella, teniendo de esta manera innumerables cambios en la interfaz de usuario y las páginas web dinámicas. Cabe resaltar que Javascript tiene formas de ejecutarse del lado del servidor como una de sus ventajas (Pascuali, 2013, pp 9-10).

A continuación se describen las características principales de JavaScript:

- JavaScript no trabaja con ningún concepto de herencia por lo cual está orientada a eventos de manera limitada.
- Es un lenguaje de programación que no necesita ser compilado, el navegador web lo interpreta de forma fácil sin complicaciones acortando tiempos de respuesta.
- JavaScript tiene la característica de ser un lenguaje dinámico, el mismo permite dar respuestas de eventos en tiempo real.
- Este lenguaje es multiplataforma es decir que puede trabajar en diferentes sistemas operativos que actualmente conocemos como es en Windows, Linux y Mac.

En la actualidad existe una variedad de librerías y frameworks, para este lenguaje de programación que es JavaScript como por ejemplo: Backbone.js, Meteor.js, Modernizr, React.js, Angular, etc. Para este proyecto técnico el Framework a usar es Angularjs.

1.2.3 Angularjs

Angularjs es un framework modelo vista controlador (MVC), desarrollada por la empresa multinacional google el mismo que es escrita en Javascript y es de código abierto. Este framework trabaja del lado del cliente permitiendo desarrollar páginas web dinámicas, AngularJS trabaja en grupo con otras tecnologías como HTML, CSS, JavaScript, AngularJS; se ajusta muy bien con librerías de terceros el mismo que ayuda a adaptar y ampliar el HTML tradicional. El cual mejora los contenidos dinámicos de las páginas web de un data binding bidireccional lo que permite la sincronización automática de los modelos y las vistas (Tarasiewicz, 2014, pp 5-10).

El Framework AngularJS, pone poco énfasis en el manejo del modelo en objetos para la representación de documento (DOM) de esta manera mejora testing y el rendimiento.

AngularJS genera algunos objetivos de diseño los mismos que son:

- AngularJS separa el manejo del DOM de la lógica de la aplicación permitiendo de esta manera que existan muchas mejoras para probar el código.
- AngularJS disocia una aplicación del lado del cliente y del lado del servidor esta separación da como resultado que el trabajo de desarrollo avance en paralelo y permite a futuro la reutilización de los dos lados (Cliente y Servidor).
- Este framework permite que los desarrolladores sean guiados en todo el proceso de desarrollo de la aplicación web partiendo del diseño de la interfaz de usuario atravesando el desarrollo de la lógica de negocio y finalmente las pruebas de la aplicación desarrollada (Tarasiewicz, 2014, pp 20-35).

1.3 Bases de datos

Las bases de datos usan sistemas de gestor de base de datos que son un conjunto de programas que facilitan el almacenamiento, modificación y extracción de los datos (Vaish, 2013, pp 7-10).

Este término de bases de datos se utilizó por primera vez en California en el año de 1963. Desde el punto de vista informático las bases de datos son sistemas que están conformados por un conjunto de datos que están almacenados en los discos, los mismos que permite el acceso a distintos programas para que los datos sean manipulados.

Las bases de datos han ido evolucionando con el pasar de los años, apareciendo nuevos conceptos de bases de datos por lo cual en esta sección se hablara de las bases de datos noSQL.

1.3.1 Bases de datos noSQL

No existe un concepto formal de las bases de datos noSQL, pero nos referimos a una amplia clase de sistema de gestión de datos que difieren con el modelo clásico de relaciones entre entidades o llamadas tablas que existen en los sistemas de gestión de las bases de relacionales. (Oracle, 2016).

Actualmente la gran parte de aplicaciones web han sido desarrolladas exclusivamente para que las bases relacionales soporten, pero a medida que los años han ido pasando se ha mejorado con excelencia gracias a una normalización adecuada, esto siempre ha dependido de los requerimientos del entorno y a la utilización de los sistemas de persistencia (Vaish, 2013, pp 7-25).

Con el avance tecnológico, el desarrollo del internet y todos los servicios que actualmente se dispone, los usuarios han empezado a utilizar aplicaciones orientadas a la web, esto ha repercutido en un intercambio de información masivo. Hoy en día se han puesto en auge las redes sociales como el Facebook, Twitter, Orkut, Instagram y los conocidos blogs éstas han hecho que la mayoría de bases de datos relacionales colapsen en gran parte generando problemas de escalabilidad y rendimiento. En este tipo de aplicaciones la información a manipular es no transaccional por este motivo surgen las bases de datos no relaciones (Cattell, 2010, pp 12-27).

Las bases de datos noSQL comienza el surgimiento en las principales empresas de internet como son Google, Amazon, Twitter y Facebook por la cantidad de suscriptores y el intercambio masivo de información, estas empresas tenían que enfrentarse a grandes cambios y desafíos en el tratamiento de la información. A medida que las aplicaciones web en tiempo real han tenido un gran crecimiento; lo cual las necesidades de manejar grandes volúmenes de datos han generado que exista bases de datos noSQL que permitan manejar estructura horizontal, se dice que lo más importante de estas bases son las propiedades de tiempo real y el rendimiento (Pokorny, 2013, pp. 69-82).

Existen diferentes tipos de bases de datos noSQL como son: clave-valor, grafos, de columna y documentales; En las bases de datos noSQL no existe ACID (Atomicity, Consistency, Isolation, Durability) como en las bases de datos relacionales, en las bases de datos noSQL existe la BASE (Basic Availability, Soft State, Eventual Consistency) esto permite que se realice el manejo de datos. (Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases, 2012, pp. 1-4).

En la siguiente Tabla 1-1 se encuentra las principales diferencias con una base datos SQL.

Tabla. 1-1: SQL vs noSQL

Característica	SQL	NoSQL
Tipo	Relacional	No Relacional
Datos	Datos son estructurado	Los datos son almacenados en formato JSON
Escalabilidad	Vertical	Horizontal
Lenguaje	Lenguaje de consulta estructurado	Lenguaje de consulta no estructurado
Flexibilidad	Esquema rígido	No es rígido y flexible
Consultas	Son más fáciles de realizar y los resultados son exactos.	No se tiene un control sobre los datos embebidos y los resultados no son los esperados.

Realizado por: Carranza Jerson, Morocho William. 2017

Las bases de datos noSQL no tienen un mismo modelo de datos ya que al ser sistemas altamente especializados manejan su propia manera de trabajar y por ese motivo se lo agrupo en cuatro categorías que son:

1.3.1.1 Base de datos Clave-Valor.

Este tipo de base de datos es muy sencilla en cuanto a funcionalidad. Esta base de datos identifica a cada elemento con una clave única, lo que permite acceder a la información de una manera rápida. Cada registro está formado por una clave y un valor. El valor puede estar formado por una lista, String o Arrays. En la **figura 2-1** observaremos la estructura de clave-valor.

<u>Key</u>	<u>Value</u>
12345222	Nombre: Reid Garrett Hoffman, Ciudad: California, Edad: 48
345645	Nombre: Lawrence Edward Page, Ciudad: East Lansing, Edad: 43
3242w34	Nombre: Jeff Bezos, Ciudad: Albuquerque, Edad: 52

Figura 2 - 1: Estructura de una base datos clave-valor.
Realizado por: Carranza Jerson, Morocho William. 2017

1.3.1.2 Bases de datos de grafos.

Se utiliza para aquellas relaciones que se pueda representar adecuadamente mediante un grafo, los datos se guardan en estructuras grafos con nodos que son las entidades, con sus respectivas propiedades y las relaciones que tiene como objetivo representar el tipo y la dirección de nuestra relación. El cual facilita almacenar datos bajo muchos niveles. En el **figura 3-1** observamos un ejemplo de su estructura.

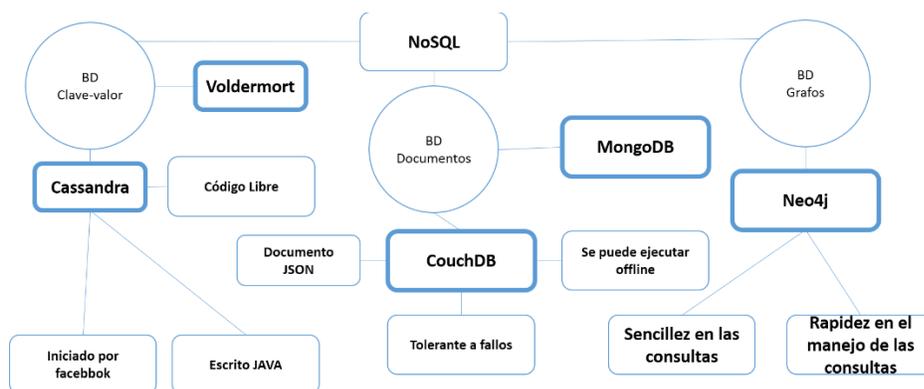


Figura 1.3-1: Estructura de una base de datos de grafos.

Realizado por: Carranza Jerson, Morocho William. 2017

1.3.1.3 Bases de datos Columna

Estas bases de datos orientadas a columnas contienen filas, se adecuan mejor para manejar operaciones analíticas sobre grandes conjuntos de datos. La ventaja es que no es necesario que todas las filas tengan las mismas columnas, lo que permite adicionar una columna o fila en cualquier momento, su estructura se encuentra en la **Figura 4-1**.

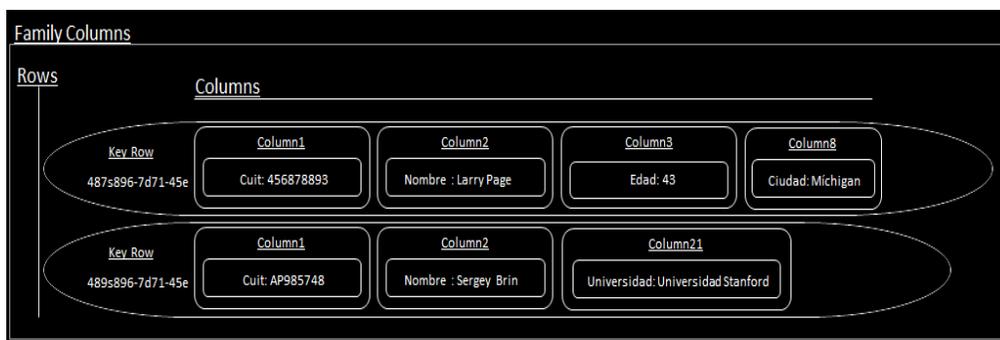


Figura 4 - 1: Ejemplo de una base de datos de columnas.

Realizado por: Carranza Jerson, Morocho William. 2017

1.3.1.4 Bases de datos documentales

Este tipo de base de datos almacena la información en documentos que pueden ser de formato JSON, BSON o XML y donde utiliza un id único para cada registro. En estas bases de datos no es necesario seguir un formato definido para cada documento, lo cual brinda versatilidad y se acoplan a soluciones que se desea añadir. A continuación, en la **figura 5-1** se detalla el formato que se tiene en una base de datos documental.

```
{
  "_id" : ObjectId("5949a90d3791101aa0cea016"),
  "descripcion" : "Ingresos por cobro de matrícula",
  "sugerencia" : "Matricúla",
  "haber" : [
    {
      "codigo_cuenta" : "4.1",
      "detalle" : "cobro de matrícula",
      "cantidad" : 20,
      "_id" : ObjectId("5949a90d3791101aa0cea017")
    }
  ],
  "debe" : [
    {
      "codigo_cuenta" : "1.1.1.1",
      "detalle" : "caja",
      "cantidad" : 20,
      "_id" : ObjectId("5949a90d3791101aa0cea018")
    }
  ],
  "fecha" : ISODate("2017-06-20T23:00:29.763Z"),
  "__v" : 0
}
```

Figura 5 - 1: Formato base documental
Realizado por: Carranza Jerson, Morocho William. 2017

En el siguiente **figura 6-1** se muestra un resumen de las categorías las bases de datos noSQL.

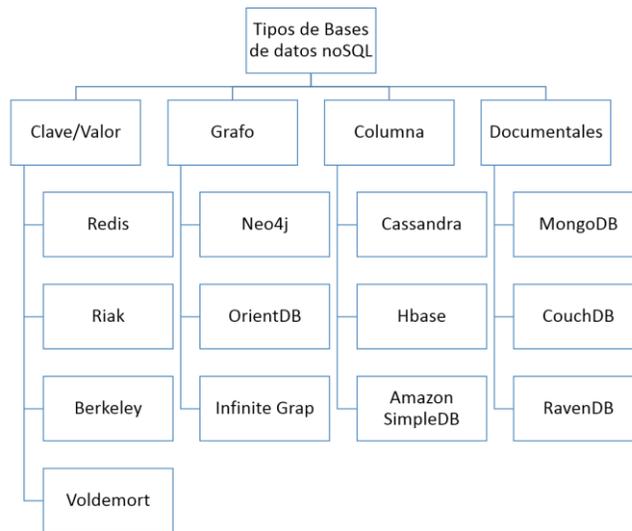


Figura 6 - 1: Clasificación de bases de datos noSQL.
Realizado por: Carranza Jerson, Morocho William. 2017

Las bases de datos noSQL tienen muchas ventajas las cuales son (Anand, 2016, pp 5-10):

- Open source.
- No imponen un esquema a la información que es almacenada.
- Se pueden realizar cambios de esquemas sin tener que parar la base.
- Escalabilidad horizontal.
- Soportan estructuras distribuidas.
- Rendimiento en el procesamiento de datos.

En el nuevo mundo de las bases de datos noSQL no todo es de color rosa, también tiene sus desventajas como:

- No poseen un estándar y su futuro es incierto.
- Las herramientas de administración no son muy usables y no tiene muchas funcionalidades.
- Tienen su propio formato y no son compatibles con la estructura de SQL.

Para este proyecto técnico el enfoque es hacia las bases documentales, estas bases de datos están formado por un conglomerado de programas que permiten almacenar datos en forma de documentos. Algunas de las codificaciones que se usan en la actualidad son XML, BSON, YAML, JSON. Los tipos de bases de datos orientados a documentos tienen un parecido a los registro de las bases relacionales aunque en estas bases las tuplas o filas son menos rígidas ya que no tienen que alinearse a un esquema (Pokorny, 2013, pp. 69-82).

1.3.1 MongoDB

Una de las más conocidas es MongoDB, el mismo que tiene un concepto y estructura distinta al de las bases de datos tradicionales. En esta base de datos no existen tablas, ni registros y también no

posee sentencia SQL, por esta razón es una gran candidata a almacenar los datos de este trabajo de titulación. (Anand, 2016, 2013, pp 1-4).

1.3.1.1 Definición de Mongo DB

MongoDB está orientada a documentos y este sistema de base de datos es multiplataforma, es decir, que no interesa en que sistema operativo está trabajando, MongoDB funciona adecuadamente. Esta base de datos noSQL trabaja con esquemas libres. (Anand, 2016, 2013, pp 1-4).

MongoDB está escrito con el lenguaje C++, el mismo que permite tener relación más cercana con cada uno de los recursos de hardware de la máquina en la que se ejecuta, por lo cual los procesos a realizarse se lo hacen con rapidez en cada tarea (Comparing NoSQL MongoDB to an SQL DB, 2013, pp. 1-6).

También base de dato tiene licencia GNU AGPL lo cual nos indica que se está trabajando con un software que tiene una licencia libre. MongoDB tiene algunas características que son netamente de esta base de datos las mismas que son:

- Es una base de dato de propósito general, los procesos que se ejecutan son rápidos también se puede funcionar como una base de datos relacional o tradicional.
- MongoDB tiene una alta disponibilidad lo que permite que siempre estará para trabajar en el momento que se lo requiera.
- Una de las características más sobresalientes es la escalabilidad permitiendo crear desde un servidor local que está aislado hasta arquitecturas que están distribuidas en gigantes clusters (Anand, 2016, pp 1-4).
- MongoDB tiene una peculiaridad que es el auto balanceado de carga ya que cuenta con varios shards permitiendo de esta manera decidir al balanceador cuando migrar datos.
- Otra característica importante de esta base de dato es la replicación nativa mediante el cual permite la sincronización de datos entre varios servidores.

- Para MongoDB la seguridad es importante por lo cual esta base de dato permite las diferentes autenticaciones y autorizaciones en los distintos niveles de permisos (Anand, 2016, pp 1-4).
- Esta base permite realizar actualizaciones sin dejar de dar servicios es decir siempre estará funcionando haga lo que se haga en la base de datos.
- Mongo actualmente está utilizando objetos JSON para los procesos de guardar y traspasar información, también permite realizar cálculos espaciales y consultas en 2d y 3d (Anand, 2016, pp 1-4).
- Cabe mencionar que las características de MongoDB permiten que no se genere el típico cuello de botella porque tiene un escalamiento horizontal lo que no sucede con las bases de datos relacionales, por lo que procesa grandes cantidades de información sin ningún problema.

1.3.1.2 Terminología básica en MongoDB

En MongoDB cada conjunto de datos se denominará documento, cada uno de estos documentos se podrán agrupar en distintas colecciones, haciendo una comparación se podría decir que cada colección es igual a una tabla de la base de datos relacional, la diferencia se encuentra que en las colecciones se pueden almacenar documentos con formatos variados sin estar sujetos a un esquema fijo (MongoDB Basics, 2014, pp. 1-3).

MongoDB permite implantar o crear los respectivos índices para cada uno de los atributos que se encuentran en los documentos, esto permitirá que MongoDB sea eficiente en la estructura interna, cuando se pretende acceder a la información a través de los atributos de cada contenido.

En la **tabla 2-1** se describen algunos términos que comúnmente se utilizan en la base de datos noSQL, para esto se compara con una de las bases de datos conocida como es PostgreSQL, para esta comparación se escogió términos comunes utilizadas por los desarrolladores.

Tabla. 2-1: Terminología empleada en los sistemas gestores de base de datos SQL y noSQL.

PostgreSQL	MongoDB
Tabla	Colección

Fila	Documentos
Columna	Campo
Joins	Documentos embebidos

Realizado por: Morocho William, Carranza Jerson. 2017

1.3.1.3 Formato de los documentos que trabaja MongoDB.

Los documentos creados en MongoDB tienen el formato binary json la misma que es una versión mejorada de típico json los mismos que permiten realizar las búsquedas de una manera rápida. BSON permite guardar las longitudes de los campos, los índices de cada *array* y toda la información que posteriormente será útil de manera explícita por esta razón los documentos BSON que se crean, ocupan un poco más de espacio a los documentos que son creados en formato json (MongoDB Basics, 2014, pp. 20-33).

Tomando la idea clave que utilizan los sistemas noSQL, los mismos que piensan que el almacenamiento es barato, por lo cual es mejor aprovecharlo al máximo, de esta manera se introduce al incremento de la velocidad para la localización de información que este dentro de un documento.

Lo anteriormente mencionado al momento de la práctica nunca se verá el formato en que realmente se está trabajando porque siempre se trabajará sobre un documento json para realizar cualquier ingreso o consultar la información deseada.

1.3.2 Operaciones MongoDB.

En MongoDB hay diferentes operaciones que son insertar, consultar, actualizar y eliminar documentos.

A continuación se detalla las operaciones que se pueden realizar a los datos en MongoDB.

- **Insertar.**

Para insertar una colección en el documento, se debe utilizar el siguiente formato que se visualiza en la **Figura 7-1**.

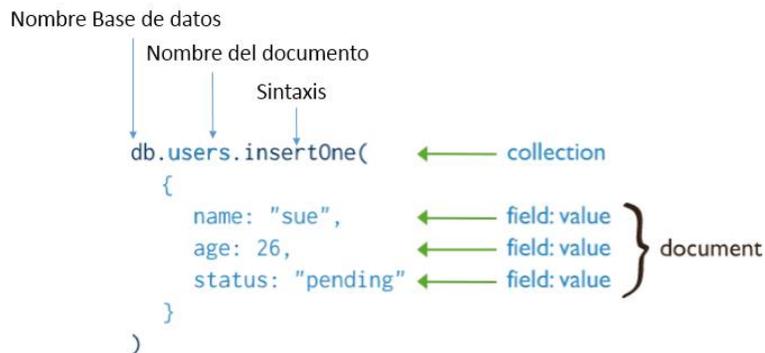


Figura 7 - 1: Insertar una colección
Realizado por: Carranza Jerson, Morocho William. 2017.

Para la inserción de varias colecciones en un documento al mismo tiempo se debe utilizar el formato de la **Figura 8-1**.

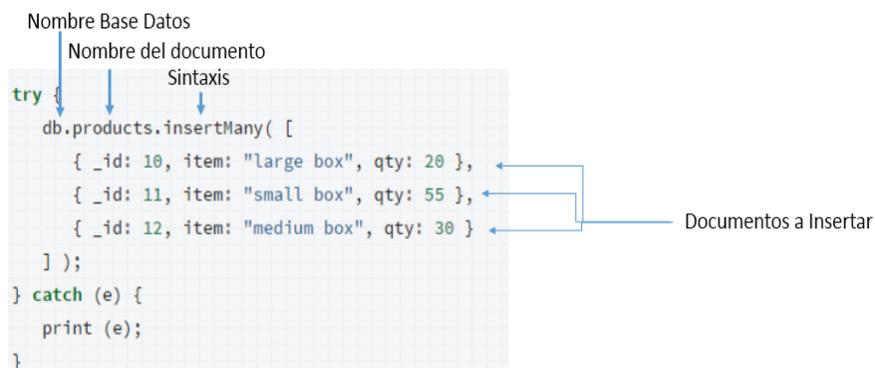


Figura 8 - 1: Insertar varias colecciones
Realizado por: Carranza Jerson, Morocho William. 2017.

Consultar

Para realizar la consulta de la información almacenada en la base de datos se debe seguir el siguiente formato como se observa en la **Figura 9-1**.

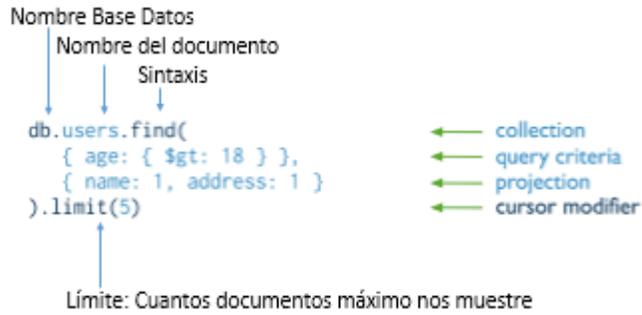


Figura 9 - 1: Consultar

Realizado por: Carranza Jerson, Morocho William. 2017.

Actualizar.

En la modificación de varias colecciones se utiliza el siguiente formato el mismo que se puede observar en la **Figura 10-1**.



Figura 10 - 1: Actualizar varios documentos

Realizado por: Carranza Jerson, Morocho William. 2017.

Eliminar

Borrar varios documentos al mismo tiempo, se utiliza el formato que se observa en la **Figura 11-1** donde se observa toda la sintaxis del código.

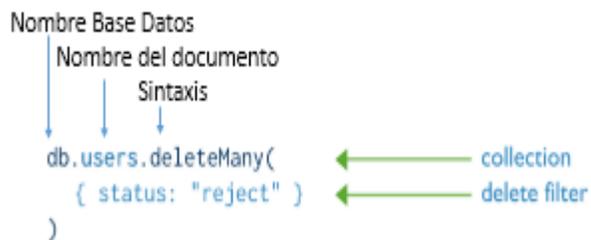


Figura 11 - 1: Eliminar varios documentos

Realizado por: Carranza Jerson, Morocho William. 2017.

1.4 Norma ISO/IEC 9126-3

La norma ISO/IEC 9126 se publica hace muchos años atrás exactamente en el año de 1992, el mismo que es un estándar para determinar la calidad de software bajo distintas métricas. Este estándar está creado, pensado en el análisis de las métricas externas y métricas internas. Es muy importante conocer las características y sub características de este estándar ya que son las bases que se deben aplicar al momento de evaluar la calidad del software. Tenemos 6 características que son la funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad (Definición de Métricas de Calidad para Productos de Software, 2016, pp. 1-6).

En la calidad interna se encuentra las siguientes características: Usabilidad, Funcionalidad y Portabilidad mientras que en la calidad externa se encuentra en las siguientes características que son confiabilidad, eficiencia y mantenibilidad. (Merino, 2017, pp. 19-20).

Para evaluar el proyecto técnico se lo realizará con el parámetro de “funcionalidad”, el mismo que permitirá evaluar el conjunto de funciones y las propiedades específicas del software, se entiende por funciones a todo proceso que realiza el software para satisfacer los requerimientos de la empresa.

El criterio de funcionalidad tiene 5 atributos internos los mismos que pueden ser medidos como son:

- **Adecuación:** tiene que ver con la capacidad que tendrá el software para proveer todo el conjunto de funciones que cumplirán con los requerimientos de usuario.
- **Exactitud:** este atributo hace referencia a los resultados que genera la aplicación web a los procesos que el usuario del sistema realiza, es decir, la precisión en los resultados esperados.
- **Interoperabilidad:** es la capacidad que tendrá la aplicación web en interactuar con varios sistemas específicos.
- **Seguridad:** este indicador permite evaluar la capacidad que el software construido tiene para proteger la información, es decir, que los usuarios o sistemas que no están

respectivamente autorizados no ingresen al sistema para realizar operaciones que pueden dañar la información y la fiabilidad de los datos.

Cumplimiento Funcional: este indicador permite que el software desarrollado cumpla con todo lo anterior mencionado.

1.5 Alcance

El sistema se encargará de llevar el control de los procesos administrativos de la fundación “JASPE” que son el cobro de matrícula y pensiones de los estudiantes y el control de pagos de los profesores, todos esos procesos estarán completamente relacionados con los módulos de contabilidad los cuales son: libro diario, libro mayor, flujo de caja y balance de comprobación.

Los estudiantes realizarán la pre matricula por de la página web y la secretaria aprobará la pre matrícula.

CAPÍTULO II

2 MARCO METODOLÓGICO

En este capítulo se detalla las técnicas y métodos que se utilizaron para el desarrollo del sistema financiero de la fundación JASPE de la ciudad de Colta.

Las metodologías ágiles están ganando terreno a las metodologías tradicionales y acaparando gran interés en la ingeniería de software. El impacto de estas nuevas metodologías es tan grande que ya existen varias conferencias de alto nivel sobre el tema. En el mundo de la ingeniería del software se abrió un gran debate para lidiar que metodología es mejor. (Wood et al, 2013, p.633).

Las metodologías tradicionales son rígidas y dirigidos por la documentación esto hacia que sus procesos ya se encuentren definidos y que los requisitos sean poco cambiantes lo contrario que ofrecen las metodologías ágiles que se ajustan a un amplio rango de proyectos de desarrollo de software.

El trabajo en equipo es considerado vital en la ingeniería de software moderno, por ese motivo para el desarrollo de nuestro sistema se escogió la metodología ágil XP (Extreme Programming) que nos facilita hacer grupo de trabajo. Lo más relevante de esta metodología es la participación del cliente durante todo el proceso es el sello distintivo de XP. (Wood et al, 2013, p.630).

La programación extrema o XP es una metodología ágil y liviana para el desarrollo de sistemas informáticos, sirve para afrontar ambientes cambiantes. Los valores que propone esta metodología es realizar la cosa más sencilla para que funcione que realizar un proceso de codificación complicado y que probablemente no se use.

2.1 Justificación metodología

Las razones que se eligió la metodología XP en este proyecto son:

- Se puede realizar modificaciones de las funcionalidades que ya están realizadas.

- Esta metodología también se caracteriza por potenciar las relaciones interpersonales como punto clave para el desarrollo del software, incentivando el trabajo en equipo, preocupándose por la instrucción de los desarrolladores.
- Otra de las razones de XP es la retroalimentación que existe entre el cliente y el equipo de trabajo.

2.2 Fases de la metodología

En esta etapa se describen las actividades y sus respectivas tareas para conseguir el objetivo principal del proyecto. Para el desarrollo del proyecto se plantea el siguiente plan de trabajo que está dividida en 6 fases.

Las fases que tiene esta metodología son:

- Fase de exploración.
- Fase de planificación del proyecto.
- Fase de diseño.
- Fase de codificación.
- Fase de Pruebas.
- Fase de Finalización

Fase de exploración, en esta fase se tiene la obligación de diagnosticar la situación actual de los procesos que lleva a cabo la fundación “JASPE” con el fin por recopilar información, también en esta fase que el equipo de desarrollo se familiariza con el lenguaje de programación además se gestionarán los *riesgos* que pueden llevar al fracaso el proyecto.

Fase de planificación del proyecto el primer paso a ejecutar son las *historias de usuarios* estas se la definen con el equipo de desarrollo de software y el cliente, tiene la misma funcionalidad que los casos de usos pero la diferencia es que el cliente escribe lo que desea en un lenguaje no técnico.

Estas son técnicas utilizada para detallar los requisitos de software, se trata de tarjetas de papel en las cuales el cliente describe brevemente las características del sistema debe tener, éstos se transformaran en requisitos funcionales y no funcionales, hay que tomar en cuenta que estas

historias de usuarios son flexibles y dinámicos. (Metodologías ágiles en el desarrollo de software, 2003, pp 4).

Hay que resaltar que el tiempo idóneo para realizar una historia es entre una y tres semana.

El siguiente paso de la fase de planificación es crear un plan o llamado *release planning* donde el usuario indica que historia de usuario se realizará y en qué fecha, esta planificación lo realiza el equipo de desarrollo junto al cliente para realizar la planificación se debe tener claro algunos factores que son: El objetivo que se va a cumplir, el tiempo que va a tardar, cuantas personas trabajaran y como se evaluara la calidad del trabajo.

Después de hacer el *release planning* hay que realizar las *iteraciones* que tiene un tiempo aproximado de dos semanas. Al comienzo de la iteración el cliente selecciona las historias de usuarios. El siguiente paso es formar grupo de dos personas para realizar la *programación en pareja* con el fin de mejorar la productividad y calidad del software mientras uno codifica el otro programador analiza si el método está bien diseñado.

Otro factor que se toma en cuenta en la fase de planificación son las *reuniones diarias* con el fin de que los desarrolladores se reúnan y expongan sus problemas, ideas y posibles soluciones, las reuniones debe ser fluidas y todo el equipo de trabajo debe dar su opinión con el fin de resolver cualquier inconveniente.

Además en esta fase se realiza una estimación para calcular las fechas y el costo del proyecto.

Fase de diseño se encuentra el proceso de *diseño simple* se trata de realizar las cosas lo más sencillo posible pero que sea funcional con el fin de desarrollar en el menor tiempo posible. En esta fase hay que realizar las metáforas necesarias que son indispensables para el desarrollo del sistema.

Fase de codificación es donde el equipo de trabajo y el cliente es indispensable para el desarrollo de las historias de usuarios, se debe recalcar que se debe realizar test para verificar que la historia cumple con la funcionalidad especificada.

La codificación debe hacerse bajo un estándar de programación para mantener el código consistente y facilitar su comprensión y escalabilidad. Algunas ventajas que se tiene de la codificación estandarizada son el mejoramiento de la comunicación en los equipos de desarrollo de software, lo cual esto reduce los errores de programación y mejora la calidad del software (Li y Prasad, 2005, p.155).

Por su facilidad de entendimiento y aplicación se escogió el estándar lower Camel Case que nos da una serie de recomendaciones.

- La primera letra debe ser minúscula, ejemplo: listadoEstudianteAprobados.
- Las siguientes palabras empieza con letras mayúsculas, ejemplo: listadoEstudianteAprobados.
- No hay mayúsculas que estén escritas continuamente, un ejemplo incorrecto: listadoESTudiante.
- No existe números en ningún lugar de las palabras, un ejemplo incorrecto: listadoEstudiante1.
- No debe de existir punto “.”, guion bajo “_” o guion medio “-” en las palabras, ejemplos incorrectos: listado_estudiante, listado.estudiante, listado-estudiante.

Fase de pruebas, que son unos de los pilares de esta metodología, esto sirve para examinar el funcionamiento de los códigos que implantados.

Fase de finalización, esta fase sirve para dar capacitaciones a los usuarios y realizar los análisis de los resultados.

2.3 Tipos y roles de usuarios.

Los usuarios para el sistema llamado SIWEFI se describen en la siguiente tabla 1 -2 y son las personas que interactuaran con el sistema.

Tabla 1 - 2: Tipos y roles de usuarios.

Usuario	Funcionalidades / Acciones
Administrador del sistema Secretaria	Gestión procesos administrativos: Matriculas, cursos, profesores, pensiones, roles de pagos. Gráficos: Estudiantes, Cursos, Sueldos.

Estudiantes	Realizar la pre matricula, Verificar en que materias se encuentra matriculado.
-------------	---

Realizado por: Carranza Jerson, Morocho William. 2017

Desarrollo de la metodología

2.4 Fase de exploración.

La fase de exploración de cualquier proyecto de software nos ayuda en que condición se encuentra la entidad a desarrollar el proyecto. Las principales actividades que se tiene en esta metodología son:

- Recopilación de la información.
- Diagnosticar información.
- Gestión de riesgos.

2.4.1 *Recopilación de información*

Para la ejecución de esta tarea se usaron distintas técnicas como la observación y la entrevista.

Con la observación se pudo ver como llevaba los procesos administrativos la secretaria además se percibió cómo los estudiantes tenían que hacer las diferentes transacciones para matricularse o pagar la pensión de un curso.

También se observó otro factor que influía en los procesos de la fundación JASPE los cuales son los profesores, quienes, para recibir su pago, la secretaria debía de ver los pagos de los profesores en sus registros que se encontraban en hojas.

Se entrevistó a dos tipos de personas que fueron la secretaria y los estudiantes. A los estudiantes se les hicieron preguntas para que nos dé su opinión sobre los cursos que ofrece la empresa. Todos los estudiantes estaban satisfechos con el temario que se daban en dichos cursos. También se le hicieron preguntas sobre si estaba satisfecho con la velocidad que la secretaria realiza los distintos

procesos para que se matricule o pague su pensión. Respondieron la mayoría que no, porque existe aglomeración de estudiantes cuando se realiza matriculación.

También se le realizaron preguntas a la secretaria con el objetivo de que deseaba para agilizar los procesos administrativos.

Cabe resaltar que a medida que se ejecutaba esta fase el equipo de trabajo se familiariza con el lenguaje de programación.

2.4.2 Diagnosticar situación de la fundación

Ya dado el primer paso que fue la recopilación de información se diagnosticó la situación de la fundación, con el equipo de trabajo y el cliente se llegó a la conclusión de que la fundación necesita un sistema informático para que agilite los procesos. Para llevar a cabo el sistema una de las herramientas necesarias que debía tener la fundación era un equipo de cómputo. El cliente siempre dispuesto compro un nuevo equipo de cómputo para su posterior ejecución ya que el equipo de cómputo que poseían era obsoleto.

2.4.3 Gestión de riesgos

La gestión del riesgo se define como el proceso de identificar, analizar y cuantificar las probabilidades de pérdidas y efectos secundarios que se desprenden de los desastres, así como de las acciones preventivas, correctivas y reductivas correspondientes que deben emprenderse (Wood et al, 2013, p.637).

Para gestionar los riesgos hay que observarlo y analizarlos con un enfoque estructurado durante el inicio del proyecto y hacerle seguimiento durante todo el proceso, para manejar la incertidumbre relativa a una amenaza, se lo hace con las siguientes actividades las cuales son:

- Identificación de riesgos.
- Estimación de riesgos.
- Plan de riesgos.

Los riesgos implican cambios en las decisiones en cuanto a los procesos, actividades y recursos que son utilizados en el desarrollo del software, existen tres tipos de riesgos en el desarrollo de software, que se detallan en la siguiente tabla 2-2:

Tabla 2- 2: Tipos de riesgos

Tipos de riesgos
Riesgos del proyecto.
Riesgos técnicos.
Riesgos del negocio.

Realizado por: Carranza Jerson, Morocho William 2017

Los riesgos del proyecto son los que amenazan al plan de proyecto, si estos riesgos se hacen realidad es posible que todas las actividades de la planificación se retrasen y esto implica que los costos aumenten.

Los riesgos técnicos amenazan a la calidad del software y a la planificación, si un riesgo técnico llegase a convertirse en realidad, la implementación puede llegar a ser difícil. Estos riesgos identifican problemas de diseño, de interfaz, implementación, verificación y mantenimiento. Las tecnologías de puntas están incluidas en los riesgos técnicos porque el problema a resolver con estas tecnologías tiende a tener una alta probabilidad de ser más difícil a resolver de lo que pensábamos.

Los riesgos de negocios son los que amenazan la viabilidad del software a desarrollar, los principales riesgos de negocio son:

- Desarrollar un sistema que no quiera nadie en realidad.
- Desarrollar un sistema que no se ajusta a la estrategia comercial de la empresa.
- Perder el apoyo de los gestores superiores debido a cambios de enfoque.
- Perder el presupuesto o personal asignado.

Si llegase hacer realidad cualquiera de estos riesgos el proyecto se cancelara.

2.4.4 Identificación de riesgos

La identificación de los riesgos nos ayuda para dar soporte a la planificación del proyecto. Para todos estos riesgos identificados se debe tener un plan de gestión de riesgos para afrontarlos

debidamente, lo primero que se debe hacer es una lista de los posibles riesgos que puede provocar el fracaso de un proyecto.

Cada uno de estos riesgos surge de las consecutivas entrevistas con el director de la fundación JASPE y el análisis del equipo de desarrollo mediante la tormenta de ideas.

En la **tabla 3-2** se detalla los posibles riesgos que pueden aparecer en el desarrollo del proyecto, y se lo categorizará, si es riesgo técnico, riesgo de proyecto o riesgo del negocio.

Tabla 3 - 2: Análisis de riesgos.

Id	Descripción	Categoría	Consecuencias
R1	Muy poca explicación o detalle por parte del usuario en cuanto a sus necesidades	Proyecto	No se cumplirá las tareas de la planificación en la fechas preestablecida.
R2	Diseñar mal la base de datos	Técnico	Demora en el desarrollo del proyecto.
R3	Cambios drásticos y continuos de los requisitos por parte del usuario.	Proyecto	Retraso en el desarrollo del proyecto
R4	Interfaces complejas para que el usuario navegue en el sistema	Técnico	Desinterés de los usuarios por utilizar sistema
R5	Incumplimiento de entregas en fechas establecidas	Técnico	Retraso en la planificación
R6	Cambio de autoridades en la fundación	Negocio	EL proyecto se cancelara.
R7	Falta de apoyo de las autoridades	Negocio	El proyecto quedara inconcluso
R8	No usar estándar en la programación	Técnico	Se tendrá una mala calidad de software.

Realizado por: Carranza Jerson, Morocho William. 2017

2.4.4.1 Valoración de la probabilidad

A continuación, damos la valoración de la prioridad. Cabe recalcar que no existe ningún riesgo con una probabilidad del 0% ya que si existirá no fuera un riesgo y tampoco existe riesgo con probabilidad del 100% ya que si lo hubiera dicho riesgo fuera un problema y se lo debe de resolver antes que comenzara a ejecutarse el proyecto.

La valoración de la prioridad será comprendida en un rango de tres partes entre el 1% y del 99%, se lo realiza con el fin de clasificarlo adecuadamente, los valores para determinar la probabilidad se encuentran en la **tabla 4-2**.

Tabla 4 - 2: Valoración de Probabilidad.

Rango Probabilidad	Descripción	Valor
1% - 33%	Baja	1
33.1% - 67%	Media	2
67.1% - 99%	Alta	3

Realizado por: Carranza Jerson, Morocho William. 2017

2.4.4.2 Valoración del impacto.

En la **tabla 5-2** se describe el detalle del impacto del sistema que se encuentra conformado por el mismo nombre que se le da al riesgo, el retraso que es el tiempo, El impacto técnico donde se describirá las consecuencias ocasionadas, el costo que es el porcentaje que afectaría negativamente al presupuesto del proyecto y el valor.

Tabla 5 - 2: Valoración del impacto.

Impacto	Retraso	Impacto Técnico	Costo	Valor
Bajo	2 días	Ligero retraso en la planificación del proyecto	<3%	1
Moderado	5 Días	Moderado retraso en la planificación del proyecto	<8%	2

Alto	10 Días	Severo retraso en el desarrollo del proyecto	<15%	3
Critico	> 21 Días	El proyecto talvez no pueda ser culminado	>30%	4

Realizado por: Carranza Jerson, Morocho William. 2017

2.4.4.3 Valoración de la exposición del riesgo

En la siguiente **tabla 6-2** se detalla la exposición del riesgo en la cual se describe el riesgo con un rango de valor y con un color correspondiente a cada uno.

Tabla 6 - 2: Valoración de la exposición del riesgo

Exposición al riesgo	Valor	Color
Bajo	1 , 2 y 3	Verde
Media	4, 5 y 6	Amarillo
Alta	> 8	Rojo

Realizado por: Carranza Jerson, Morocho William. 2017

2.4.4.4 Determinación de la exposición del riesgo

En la siguiente **tabla 7-2** se valorará la probabilidad que puede existir en cada riesgo, el impacto y la exposición que se puede presentar cada uno de los riesgos especificados.

Tabla 7 - 2: Determinación de la exposición al riesgo.

Id Riesgo	Probabilidad			Impacto		Exposición	
	Porcentaje	Valor	Descripción	Valor	Impacto	Valor	Exposición
R1	65%	3	Media	3	Alto	9	Alta
R2	12%	1	Media	4	Critico	4	Media

R3	70%	3	Media	2	Moderado	6	Media
R4	25%	1	Baja	3	Alto	3	Bajo
R5	39%	2	Media	2	Moderado	4	Media
R6	20%	1	Baja	4	Critico	4	Media
R7	45%	2	Media	3	Alto	6	Media
R8	23%	1	Media	2	Moderado	2	Bajo

Realizado por: Carranza Jerson, Morocho William. 2017

El siguiente paso es describir la prioridad que tiene cada riesgo con el objetivo de ver cuáles son los que tienen mayor exposición con el fin de darle preferencia la hora de gestionar, lo cual esta detallado en la **tabla 8-2**.

Tabla 8 - 2: Determinación de la prioridad de riesgo.

Id	Prioridad	Descripción	Exposición	
			Valor	Exposición
R1	1	Muy poca explicación o detalle por parte del usuario en cuanto a sus necesidades.	9	Alta
R3	2	Cambios drásticos y continuos de los requisitos por parte del usuario.	6	Media
R7	2	Falta de apoyo de las autoridades	6	Media
R2	3	Diseñar mal la base de datos	4	Media
R6	3	Cambio de autoridades en la fundación.	4	Media
R5	3	Incumplimiento de entregas en fechas establecidas	4	Media
		Interfaces complejas para que el usuario navegue		

R4	5	en el sistema	3	Baja
R8	6	No usar estándar en la programación	2	Baja

Realizado por: Carranza Jerson, Morocho William. 2017

Para cada uno de los riesgos se elabora una hoja de gestión, dando las posibles soluciones y controles que se debe dar en cada uno de los riesgos.

Tabla 9 - 2: Hoja de gestión de Riesgo R1

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R1		FECHA: 03/02/2017	
Probabilidad: Media Valor: 3	Impacto: Alto Valor: 3	Exposición: Alta Valor: 9	Prioridad: 1
DESCRIPCIÓN: Muy poca explicación o detalle por parte del usuario en cuanto a sus necesidades			
REFINAMIENTO			
Causas			
<ul style="list-style-type: none"> • Que el cliente no se explique bien sobre sus necesidades para realizar el sistema o que no se encuentre en las reuniones con el equipo de desarrollo. • No tener un horario establecido para las reuniones 			
Consecuencias			
<ul style="list-style-type: none"> • Tener una mala definición en el diseño del software • El sistema se cumplirá con los requerimientos del usuarios • Retraso en la planificación. 			
REDUCCIÓN			
<ul style="list-style-type: none"> • Reuniones con el cliente. • Hacer estrategias para comprender que es lo que necesita el usuario. 			
SUPERVISIÓN			
<ul style="list-style-type: none"> • Realizar seguimiento de las actividades. 			
GESTIÓN			
<ul style="list-style-type: none"> • Documentar en una tarjeta lo que quiere el usuario. 			

<ul style="list-style-type: none"> • Verificar que cada requerimiento se efectúe al pie de la letra. 	
ESTADO ACTUAL	
Fase de reducción iniciada	X
Fase de Supervisión iniciada	
Gestionando el riesgo	
RESPONSABLES	
William Morocho	
Jerson Carranza	

Realizado por: Carranza Jerson, Morocho William. 2017

Las siguientes hojas de gestión se encuentran en el **ANEXO A**.

2.5 Fase de planificación.

La planificación de un proyecto proporciona un marco de trabajo que permita al director del proyecto hacer estimaciones moderadas de recursos costos y planificación. Entre las principales actividades de esta fase tenemos es la estimación del proyecto, la especificación de requisitos, metáforas y la definición de las iteraciones del proyecto.

2.5.1 Historias de usuarios.

En este ciclo se define los requisitos del usuario y posteriormente se convertirá en historias de usuarios, los requisitos son características que el sistema debe de cumplir, cuyo fin es establecer las funcionalidades necesarias para el sistema informático. Los requisitos son establecidos junto con el usuario, el resultado fue una lista de requerimientos.

A más de los requisitos del sistema también se debe tomar en cuenta las historias técnicas o metáforas del sistema que son necesaria para la implementación del proyecto.

Tomando en cuenta estas acotaciones se definieron 4 historias técnicas o metáforas y veinte seis requisitos o historias de usuarios.

Tabla 10 - 2: Requisitos del sistema

Tipo	Descripción
Técnico	Arquitectura del sistema
Técnico	Estándar de codificación
Técnico	Diseño de la base de datos
Técnico	Diseño de interfaces de usuario
Usuario	Cobro de matrículas
Usuario	Cobro de pensiones
Usuario	Ventas de materiales
Usuario	Pago a profesores
Usuario	Pago Arriendos
Usuario	Compra de materiales
Usuario	Permitir que el estudiante se pueda registrar
Usuario	Autenticación de estudiantes
Usuario	El estudiante pueda observar los cursos
Usuario	Realice la pre matricula (Añadir Curso Carrito)
Usuario	Cancelar la pre matricula (Eliminar curso Carrito)
Usuario	Crear un administrador
Usuario	Autentique administrador
Usuario	Gestión Cursos (Insertar, listar, modificar, eliminar)
Usuario	Gestión Estudiantes (Insertar, listar, modificar, eliminar)
Usuario	Gestión Profesores (Insertar, listar, modificar, eliminar)
Usuario	La misión y visión sea modificable (Insertar, listar, modificar, eliminar)
Usuario	Estado de situación inicial
Usuario	Libro Diario
Usuario	Libro Mayor
Usuario	Balance de comprobación
Usuario	Flujo caja
Usuario	Gráficos Estudiantes
Usuario	Gráficos Cursos
Usuario	Gráficos Sueldos

Realizado por: Carranza Jerson, Morocho William. 2017

2.5.2 Prioridad en lo requerimientos.

Una vez recopilados los requerimientos se define la prioridad junto con el usuario este proceso en la metodología XP se llama release planning, la **tabla 11-2** de prioridad que se encuentra a continuación fueron definidas con el usuario.

Tabla 11 - 2: Prioridad de requerimientos

Prioridad	Criterio
10	Indispensable
9	Esencial
8	Obligatorio
6	Necesario
4	Conveniente

Realizado por: Carranza Jerson, Morocho William. 2017

A continuación se realiza el release planning que está conformado por los requerimientos de usuarios y metáforas y se encuentra priorizados.

Tabla 12-2: Release planning

ID	Descripción	Prioridad
HT-01	Arquitectura del sistema	10
HT-02	Estándar de codificación	10
HT-03	Diseño de la base de datos	10
HT-04	Diseño de interfaces de usuario	10
HU-01	Permitir que el estudiante se pueda registrar	10
HU-02	Autenticación de estudiantes	10
HU-03	El estudiante pueda observar los cursos	10
HU-04	Realice la pre matricula (Añadir Curso Carrito)	10
HU-05	Cancelar la pre matricula (Eliminar curso Carrito)	10
HU-06	Crear un administrador	9
HU-07	Autentique administrador	9
HU-08	Gestión Cursos (Insertar, listar, modificar, eliminar)	9
HU-09	Gestión Estudiantes (Insertar, listar, modificar, eliminar)	9

HU-10	Gestión Profesores (Insertar, listar, modificar, eliminar)	9
HU-11	La misión y visión sea modificable (Insertar, listar, modificar, eliminar)	9
HU-12	Cobro de matrículas	9
HU-13	Cobro de pensiones	9
HU-14	Ventas de materiales	8
HU-15	Estado de situación inicial	8
HU-16	Libro Diario	8
HU-17	Libro Mayor	8
HU-18	Balance de comprobación	8
HU-19	Pago a profesores	8
HU-20	Ver los roles de pagos	8
HU-21	Pago Arriendos	6
HU-22	Compra de materiales	6
HU-23	Flujo de caja	8
HU-24	Gráficos Estudiantes	4
HU-25	Gráficos Cursos	4
HU-26	Gráficos Sueldos	4

Realizado por: Carranza Jerson, Morocho William. 2017

2.5.3 Estimación del proyecto.

Para estimar la duración del proyecto, se utilizó el modelo matemático COCOMO ya que se tiene experiencia en estimar los proyectos, Este modelo permite estimar el tiempo, costo y esfuerzo como se indica en la **figura 1-2**.

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK	
	Requerimiento	8:9000	1200.00	0.54	Non-Specified	32.9	17.8	506.8	21309.70	2.4	1.9	0.0	
Total Lines of Code: 9000						Estimated Effort Sched		PROD	COST	INST	Staff	RISK	
Hours/PM: 152.00						Optimistic	14.2	8.5	633.5	17047.76	1.9	1.7	
						Most Likely	17.8	9.2	506.8	21309.70	2.4	1.9	0.0
						Pessimistic	22.2	9.8	405.4	26637.12	3.0	2.3	

Project Is Saved To File : C:\Users\William\Desktop\tesis 2017\Planificación\d.est

Figura. 1 - 2: Estimación del proyecto en cocomo

Realizado por: Carranza Jerson, Morocho William. 2017.

Teniendo en cuenta que el proyecto tendrá alrededor de 9000 líneas de código, lo ideal es que el proyecto tenga una duración de 9.2 semanas contando con 1.9 personas.

La a usar es metodología XP y solo existe de una pareja programadores entonces para el proyecto hay que realizar la regla de tres para que nos proporcione tiempo exacto. Realizando los cálculos es de 17 semanas y media.

$$\text{Sched} = 9.2 * \frac{1.9}{1}$$

$$\text{Sched} = 17,48$$

2.5.4 Iteración del proyecto

Después de haber realizado la estimación de tiempo y esfuerzo, se inició con el detalle de la organización, el mismo que ayuda a asignar recursos de la mejor manera para que se pueda realizar satisfactoriamente el desarrollo del proyecto.

Los días que se trabajarán en el desarrollo del proyecto será de lunes a viernes, 8 horas diarias, la realización del proyecto iniciará el 1 de febrero del 2017 y culminará el viernes 6 de Junio del 2017 según lo estimado, el plan de proyecto contiene las fechas de las fases, iteraciones e historias las mismas que se detallan en el **ANEXO E** la cual se realizó en Project una herramienta de Microsoft. En el plan del proyecto el desarrollo de la aplicación consta de 89 días laborables, este proyecto tiene un total de 712 horas de trabajo.

2.5.5 Asignar Pareja

La metodología nos propone el rotar parejas en el proyecto, para el trabajo de titulación se cuenta con 2 programadores o una pareja, por lo cual no habrá inconveniente en este proceso, la pareja asignada será.

Tabla 13 - 2: Asignar Pareja

	Nombres	Tipo
Pareja de	William Morocho	Desarrollador
Programadores	Jerson Carranza	Desarrollador

Realizado por: Carranza Jerson, Morocho William. 2017.

2.6 Fase de Diseño.

En esta fase se desarrolla las metáforas del sistema estas permiten la fácil implementación y el desarrollo del proyecto software, las metáforas del proyecto son:

- Arquitectura del sistema
- Estándar de codificación
- Diseño de la base de datos
- Diseño de la interfaz de usuario

A las metáforas se las documenta con fichas técnicas o también llamadas historias técnicas y está formada por las siguientes características.

Historias de técnicas.- Son usadas para administrar las necesidades del programador, las características principales son.

- **ID:** Es el identificador de la historias técnicas, su nomenclatura es **MS-Nº** que significa metáfora del sistema.
- **Nombre de la historia:** Es el calificativo descriptivo que se le da a cada historia técnica.
- **Puntos Estimados:** Son las horas de desarrollo según lo planificado, cada punto estimado equivale a una hora.

Tareas de ingeniería.- Son cada una de las tareas que se realizaron para cumplir con el requerimiento de una historia especificando el esfuerzo que se empleó para cumplir.

2.6.1 Arquitectura del sistema

La arquitectura se diseñó con el objetivo de cumplir que el sistema sea escalable a nivel de interfaz, para que los datos se puedan consumir en dispositivos móviles (Smartphone, tablets).

La arquitectura que se eligió es cliente-servidor, porque a través del protocolo REST permite separar totalmente la interfaz de usuario con el servidor, las API REST permiten que el sistema sea independiente de cualquier lenguaje de programación.

Las operaciones más importantes que se pueden realizar con las REST son cuatro y se detallan en la siguiente tabla:

Tabla 14 - 2: REST

Tipo	Acción
GET	Leer y consultar los datos
PUT	Editar datos
POST	Crear datos
DELETE	Eliminar datos

Realizado por: Carranza Jerson, Morocho William. 2017

Los objetos REST se los manipulan a través de URI, que nos permite acceder a la información.

En el primer servidor está la base de datos NO-SQL documental, MongoDB, la que está encargada de guardar los datos, se configuraron los permisos de usuarios para no tener problemas de vulnerabilidad.

El segundo servidor se encontrará alojada la aplicación tendrá los siguientes paquetes que tendremos son:

- *Models* aquí estará la conexión de la base de datos.
- *Controllers* tendremos las funciones del sistema, se encargará de enviar y recibir peticiones.
- *Public* en este paquete estará las plantillas, archivos CSS, JavaScript, HTML.

Para diagrama de despliegue se usó el estándar UML que se encuentra detallado en la **figura 2 - 2**.

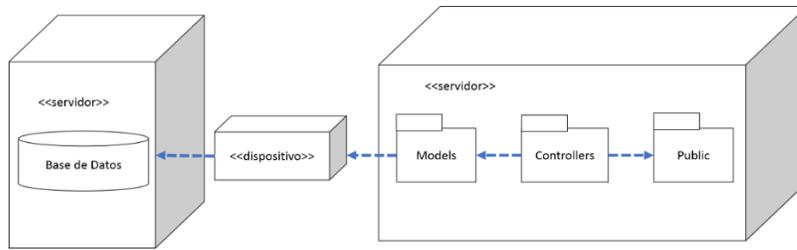


Figura. 2 - 2: Diagrama de despliegue
Realizado por: Carranza Jerson, Morocho William. 2017

La metáfora de la arquitectura del sistema generó una tarea de ingeniería y dos pruebas de aceptación que se detallan a continuación.

Tabla 15 - 2: Metáfora del sistema de la Arquitectura del sistema

Historia Técnica	
Número: MS1	Nombre de la historia: Diseño de la Arquitectura del sistema
Modificación de historia de usuario:	
Usuario: Programador	Iteración Asignada: 1
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como programadores necesitamos implementar el diseño de la Arquitectura para el desarrollo del sistema.	
Observaciones:	
<ul style="list-style-type: none"> La arquitectura debe estar con el diseño MVC 	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Verificar en el diagrama UML el modelo MVC. Existencia de los servidores para la aplicación web. Existencia de las carpetas para el modelo, vista (public), controlador.

Realizado por: Carranza Jerson, Morocho William. 2017

Tareas de ingenierías de la arquitectura del sistema.

Tabla 16 - 2: Primera tarea ingeniería de la arquitectura del sistema

Tarea de ingeniería	
Historia de Usuario: Diseño de la Arquitectura del sistema	
Número Tarea: T1MS1	Nombre de la Tarea: Verificar en el diagrama UML el modelo MVC.

Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 15/02/17	Fecha de Fin: 15/02/2017
Programador Responsable: William Morocho	
Descripción: Verificar en el diagrama de paquetes para establecer las relaciones entre el modelo, vista que se llama public, controlador de la aplicación web.	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> Existencia de las carpetas para el modelo, vista (public), controlador. 	

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 17 - 2: Segunda tarea ingeniería de la arquitectura del sistema

Tarea de ingeniería	
Historia de Usuario: Diseño de la Arquitectura del sistema	
Número Tarea: T2MS1	Nombre de la Tarea: Diseño del despliegue de la arquitectura.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 16/02/17	Fecha de Fin: 16/02/17
Programador Responsable: Jerson Carranza	
Descripción: Realizar el diseño del despliegue de la arquitectura para posteriormente implementarla en dos servidores, el primero tendrá la base de datos y el segundo la aplicación	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> Existencia de los servidores para la aplicación web. 	

Realizado por: Carranza Jerson, Morocho William. 2017

2.6.2 *Estándar de codificación*

Por su facilidad de entendimiento y aplicación se escogió el estándar lower Camel Case la cual nos da una serie de recomendaciones.

- La primera letra debe ser minúscula, ejemplo: listadoEstudianteAprobados.
- Las siguientes palabras empieza con letras mayúsculas, ejemplo: listadoEstudianteAprobados.
- No hay mayúsculas que estén escritas continuamente, un ejemplo incorrecto: listadoESTudiante.
- No debe de existir punto “.”, guion bajo “_” o guion medio “-” en las palabras, ejemplos incorrectos: listado_estudiante, listado.estudiante, listado-estudiante.

En el **ANEXO B** se describe todas consideraciones que se debe tomar en cuenta en el desarrollo del sistema “SIWEFI”.

Para la ejecución de la metáfora del estándar de codificación de acuerdo con la metodología que se eligió se describe como historia de usuario con sus respectivas tareas de ingeniería y pruebas de aceptación como resultado de esta metáfora o historia se generó una tarea de ingeniería y una prueba de aceptación, que se encuentra detallado en la tabla.

Tabla 18 - 2: Metáfora del sistema del estandar de codificación

Historia Técnica	
Número: MS2	Nombre de la historia: Estándar de codificación.
Modificación de historia de usuario:	
Usuario: Programadores	Iteración Asignada: 1
Prioridad en el Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 8
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 8
Descripción: Como programadores necesitamos utilizar un estándar de codificación para poder llevar orden en el código.	
Observaciones: Que el estándar tenga correlación del lenguaje de programación.	

Historia de Usuario - Pruebas de Aceptación

- Existencia del estándar de codificación.

Realizado por: Carranza Jerson, Morocho William. 2017

Tarea de ingeniería del estándar de codificación.

Tabla 19 - 2: Primera tarea de ingeniería del estándar de codificación

Tarea de ingeniería	
Historia de Usuario: Estándar de codificación.	
Número Tarea: T1 MS2	Nombre de la Tarea: Investigación del estándar de codificación a utilizarse.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 17/02/17	Fecha de Fin: 17/02/17
Programador Responsable: Jerson Carranza	
Descripción: Recopilar información del estándar de codificación a utilizarse en el desarrollo.	
Pruebas de Aceptación: Existencia del estándar de codificación.	
Realizado por: Carranza Jerson, Morocho William. 2017	

2.6.3 Diseño de la base de datos

La base de datos es una pieza fundamental en el desarrollo de la aplicación web, ya que es el lugar donde se almacenará todos los datos que se generen en los distintos procesos que realice el sistema web, también la base de datos permite acceder a toda la información actualizada. Cabe resaltar que no existe ningún estándar para realizar un esquema de la base de datos noSQL, para realizar el esquema nos basamos en una propuesta de Cecilio Alvarez para lo cual se ha realizado el diseño de base de datos noSQL ANEXO C.

La metáfora de diseño de base de datos con relación a lo establecido por la metodología ágil XP se describe como una historia de usuario, la misma que generan tareas de ingenierías y las respectivas pruebas de aceptación en el desarrollo e implementación de la base, la misma que se detalla en la tabla.

Tabla 20 - 2: Metáfora del Diseño de la base de datos

Historia Técnica	
Número: MS3	Nombre de la historia: Diseño de la Base de Datos
Modificación de historia de usuario:	
Usuario: programadores	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 40
Riesgo en el Desarrollo: Alta	Puntos Reales: 40
Descripción: Como programadores se necesita realizar el diseño de la base de datos para no tener complicaciones en el desarrollo del proyecto.	
Observaciones:	
<ul style="list-style-type: none"> • La Base de Datos debe estar desarrollada en mongoBD. 	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • La base de datos fue aceptada por los desarrolladores del sistema. • La base de datos existe.

Realizado por: Morocho William, Carranza Jerson. 2017

Tabla 21 - 2: Tarea de ingeniería de la metáfora del Diseño de la base de datos

Tarea de Ingeniería	
Historia de Usuario: Diseño de la base de datos	
Número Tarea: T1	Nombre de la Tarea: Diseño de la base de datos noSQL.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 40

Fecha de Inicio: 20/02/17	Fecha de Fin: 24/02/17
Programador Responsable: Jerson Carranza	
Descripción: Creación del diseño de la base de datos noSQL la misma que permitirá verificar cuantos documento se necesite para el desarrollo de SIWEFI	
Pruebas de Aceptación: Que la base de datos sea embebida.	

Realizado por: Jerson Carranza, Morocho William. 2017

2.6.4 *Diseño de la interfaz de usuario*

En esta metáfora se realiza el diseño de las interfaces de usuario, es decir, se representa de manera más intuitiva cada diseño de interfaz ya que este diseño será aplicado en todo el proyecto de desarrollo software, cada elemento que compone la interfaz de usuario fue minuciosamente pensado en el bienestar del usuario.

Esta metáfora permite que el usuario y el equipo de desarrollo establezcan mutuos acuerdos sobre la estructura visible del proyecto conociendo a esto como la vista de la aplicación web, este diseño tiene que ver con los colores que se van a utilizar, el tipo de letra para los distintos encabezados, como estará organizado cada uno de los datos dentro de la aplicación web es decir esta metáfora permite que el sistema web este funcional y al agrado de la vista del usuario final quien es la o las personas quienes manipularan el aplicativo. El resultado de realizar esta metáfora del sistema se generó una tarea de ingeniería y 2 pruebas de aceptación las mismas que están detalladas en la tabla.

Metáfora del sistema del diseño de las interfaces.

Tabla 22 - 2: Tarea de ingeniería de la metáfora del Diseño interfaz de usuario

Historia Técnica	
Número: MS4	Nombre de la historia: Diseño de las interfaz de usuario
Modificación de historia de usuario:	
Usuario: Programadores	Iteración Asignada: 1
Prioridad en el Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como programadores se necesita detallar todo los elementos que se introducirían en el diseño de la interfaz de usuario a utilizarse, para facilitar el manejo de la aplicación web.	

Observaciones:

- Los botones en la interfaz de usuario deben estar en la parte inferior derecha.
- Los colores utilizados son acordes a la imagen corporativa de la fundación.

Historia de Usuario - Pruebas de Aceptación

- No se podrán modificar las claves primarias.
- Para la utilización del sistema debe ser obligatorio autenticarse.

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 23 - 2: Tarea de ingeniería de la metáfora del Diseño de la interfaz

Tarea de Ingeniería	
Historia de Usuario: Diseño de las interfaces de usuario	
Número Tarea: T1	Nombre de la Tarea: Diseño de las interfaces de usuario del sistema SIWEFI
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 27/02/17	Fecha de Fin: 28/02/17
Programador Responsable: William Morocho	
Descripción: Creación de plantilla del proyecto SIWEFI con los estándares que se han quedado de acuerdo.	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> - Verificar que el diseño de la interfaz de usuario este acorde con el estándar establecido. - Aceptación de la interfaz por los directivos de la fundación. 	

Realizado por: Carranza Jerson, Morocho William. 2017

2.7 Fase de codificación

En el desarrollo de este proyecto se lo ha dividido en iteraciones y cada iteración debe proporcionar un resultado completo.

En esta fase da a conocer a detalle el desarrollo de las historias de usuarios por medio de tarjetas.

Para el desarrollo del sistema financiero se lo dividió en 6 iteraciones, cada una de estas iteraciones cuenta con una determinada cantidad de historias de usuarios y esta a su vez tienen una fecha de inicio y fecha fin también se necesita un esfuerzo requerido para su desarrollo.

Para el adecuado manejo de la metodología, se realizaron las siguientes fichas técnicas que fueron utilizadas para la documentación de las iteraciones del proyecto.

Historias de usuario.- Son usadas para administrar los requerimientos de usuario, las características principales son.

- **ID:** Es el identificador de la historias de usuarios, nomenclatura es **HU-Nº**.
- **Nombre de la historia:** Es el calificativo descriptivo que se le da a cada historia.
- **Puntos Estimados:** Son las horas de desarrollo según lo planificado, cada punto estimado equivale a una hora.

2.7.1 Iteración 1

En la primera iteración cuenta con 5 historias de usuarios y tiene una duración de 10 días con 80 puntos estimados y horas de trabajo. Dicha iteración está dedicada al módulo de los estudiantes.

A continuación se detalla las historias de usuarios de la primera iteración.

Tabla 24 - 2: Iteración 1

ID	Nombre de la historia	Puntos estimados
HU_01	Registro de estudiantes	24
HU_02	Autenticación de estudiantes	16
HU_03	Vistas Cursos	16
HU_04	Añadir Curso (Carrito)	16
HU_05	Eliminar Curso (Carrito)	8
Total		80

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 25 - 2: Historia de Usuario Registro de estudiantes.

Historia de Usuario	
Número: HU_01	Nombre de la historia: Registro de estudiantes.
Modificación de historia de usuario:	
Usuario: estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta	Puntos Reales: 24
Descripción: Como estudiante deseó registrarme para poder acceder al sistema.	
Observaciones: Los campos que se requieren para registrar a un estudiante son los siguientes:	

Cédula, contraseña, nombres, apellidos paterno, apellido materno, sexo.

Historia de Usuario - Pruebas de Aceptación
--

- Si todos los campos no están llenos el botón registro se bloquea.
- Si el número de cédula introducido es incorrecto emitirá el respectivo mensaje de error.

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 26 - 2: Elaboración de la interfaz de usuario Registro de Estudiantes

Tarea de ingeniería	
Historia de Usuario: Registro de estudiantes.	
Número Tarea: TI_01HU01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 01/03/17	Fecha de Fin: 01/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el registro de los estudiantes.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 27 - 2: Elaboración de la interfaz de usuario Registro de Estudiantes

Tarea de ingeniería	
Historia de Usuario: Registro de estudiantes.	
Número Tarea: TI_02 HU01	Nombre de la Tarea: Elaboración del controlador para el registro de los estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 02/03/17	Fecha de Fin: 03/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el registro de los estudiantes.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Carranza Jerson, Morocho William. 2017

Las siguientes historias de usuarios y tareas de ingenierías se encuentran en el **ANEXO D**.

2.7.2 Iteración 2

La iteración numero dos está dedicado al módulo del administrador y está compuesta por 6 historias de usuarios.

Tabla 28 - 2: Iteración 2

ID	Nombre de la historia	Puntos estimados

HU_06	Crear administrador	16
HU_07	Autenticar (Adminstrador)	8
HU_08	CRUD Curso	16
HU_09	CRUD Estudiante	16
HU_10	CRUD Profesores	16
HU_11	Eliminar Curso (Carrito)	8
Total		80

Realizado por: Carranza Jerson, Morocho William. 2017

2.7.3 Iteración 3

Las historias de usuarios de esta iteración corresponden al módulo cobranzas se encuentran detallado en la siguiente tabla.

Tabla 29 - 2: Iteración 3

ID	Nombre de la historia	Puntos estimados
HU_12	Cobro de matrícula	32
HU_13	Cobro de pensiones	24
HU_14	Venta de materiales de trabajo	24
Total		80

Realizado por: Carranza Jerson, Morocho William. 2017

2.7.4 Iteración 4

La cuarta iteración está dedicada al módulo contable y consta de cuatro historias de usuarios que se detallan en la siguiente tabla.

Tabla 30 - 2: Iteración 4

ID	Nombre de la historia	Puntos estimados
----	-----------------------	------------------

HU_15	Estado situación inicial	32
HU_16	Libro Diario	24
HU_17	Libro Mayor	32
HU_18	Balance de comprobación	32
Total		120

Realizado por: Carranza Jerson, Morocho William. 2017

2.7.5 Iteración 5

El desarrollo está dedicado a los egresos de la fundación, las historias de usuarios se detallan en la siguiente tabla.

Tabla 31 - 2: Iteración 5

ID	Nombre de la historia	Puntos estimados
HU_19	Pago a profesores	32
HU_20	Ver roles de pagos	16
HU_21	Pago de arriendo	16
HU_22	Comprar de material	16
Total		80

Realizado por: Carranza Jerson, Morocho William. 2017

2.7.6 Iteración 6

La última iteración está dedicada a reportes y gráficos consta de 4 historias de usuarios que se detalla a continuación.

Tabla 32 - 2: Iteración 6

ID	Nombre de la historia	Puntos estimados
HU_23	Flujo de caja	24

HU_24	Gráficos Estudiantes	16
HU_25	Gráficos Cursos	16
HU_26	Gráficos Sueldos	48
Total		120

Realizado por: Carranza Jerson, Morocho William. 2017

2.8 Fase de pruebas

Utilizando la metodología XP (extreme programming) se generaron metáforas del sistema e historias de usuario, que fueron definidas en la planificación del desarrollo del sistema web, la misma que se detalla en el **ANEXO E** del calendario de actividades, para aumentar la calidad del sistema y evitar efectos colaterales al momento de realizar alguna modificación se realizó 115 pruebas de aceptación las cuales tuvieron resultados exitosos. Cada una de estas pruebas son aplicadas al finalizar las tareas de ingeniería definidas en cada una de las historias de usuarios.

Prueba de aceptación del diseño de la arquitectura del sistema.

Tabla 33 - 2: Prueba de Aceptación 1

Prueba de Aceptación	
Código: P1MS1	Historia de Usuario: Diseño de la Arquitectura del sistema.
Nombre: Existencia de las carpetas para el modelo, vista (public), controlador.	
Responsable: William MOROCHO	Fecha: 15/02/17
Descripción: En el diagrama UML verificar el diseño Modelo Vista Controlador.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de las carpetas. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Crear de las carpetas Modelo vista (public), controlador. • Crear las subcarpetas. 	
Resultado esperado: Funcione correctamente la aplicación	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Carranza Jerson, Morocho William. 2017

Las demás pruebas de aceptación que están orientados al desarrollo se encuentran el **ANEXO F**.

2.9 Fase de finalización

En esta fase las tareas que se realizaron fueron.

- Capacitación del usuario.

Los programadores del sistema se involucran en un proceso educacional con los usuarios que es conocido como capacitación. A lo largo del ciclo de vida de desarrollo de sistemas los usuarios siempre estaban involucrados en el desarrollo de la aplicación.

- Presentación Documentación.

En la documentación se encuentra el manual de usuario que es un documento de comunicación técnica que busca brindar asistencia a los sujetos que usan un sistema. Más allá de su especificidad, los autores de los manuales intentan apelar a un lenguaje ameno y simple para llegar a la mayor cantidad posible de receptores.

2.10 Gestión del proyecto

Es de vital importancia gestionar el proyecto, se lo realiza con el fin de dar seguimiento a los entregables, según lo planificado en cada iteración.

La duración del proyecto fue de 712 horas.

En la fase de exploración se hicieron todas las tareas las cuáles fueron la recopilación de información y realizar la gestión de los posible riesgos del proyecto, se los realizo sin ningún contratiempo y tuvo una duración de 40 horas.

En la fase de planificación se establecieron las historias de usuarios y en qué número de iteración se la desarrollará. En esta fase no hubo ningún problema y se cumplió con las fechas según lo planificado, esta fase tuvo una duración de 40 horas.

En la fase de diseño se realizó las historia técnicas o metáforas del sistema, aunque el usuario no las conoce, son de vital importancia para el funcionamiento del software, dichas historias

técnicas se definieron con los miembros del proyecto: Arquitectura del sistema, estándar de codificación, diseño de la base de datos y definir la interfaz de usuario. En esta fase se cumplió todas las tareas según lo planificado.

La siguiente fase es la de codificación, en la que se dividió en seis módulos.

En el primer módulo llamado módulo de estudiantes, no se tuvo inconveniente y se entregó en el tiempo establecido.

El segundo módulo dedicado al administrador se los desarrollo según lo planificado y la entrega se le realizó a tiempo.

El tercer y cuarto módulo se los desarrollo según lo planificado y la entrega se le realizó a tiempo.

En el quinto módulo hubo tardanza debido al difícil manejo de extraer información en los datos embebidos de la base de datos MongoDB. Los datos embebidos son rápidos ya que no accede a otra tabla como en las bases de datos relaciones por que la información se encuentra en el mismo documento sin embargo una de su gran desventaja es no poder realizar consultas con el resultado esperado. Para resolver este inconveniente se programaron métodos adicionales con el fin de encontrar los datos correctos, una de la solución más factible fue la creación de arreglos para poder colocar las consultas de los datos embebidos con el objetivo de manipularlos. Debido a este contratiempo el retraso de este módulo fue de 20 horas.

En el sexto módulo se avanzó rápido y se recuperó el tiempo que se perdió en el módulo anterior, esto se debe a la experiencia que se obtuvo durante el desarrollo y también al elegir librerías que permiten el fácil manejo de los gráficos, lo cual facilito el desarrollo en este módulo, en la siguiente tabla nos muestra cómo se desarrolló el proyecto.

BURNDOWN CHART

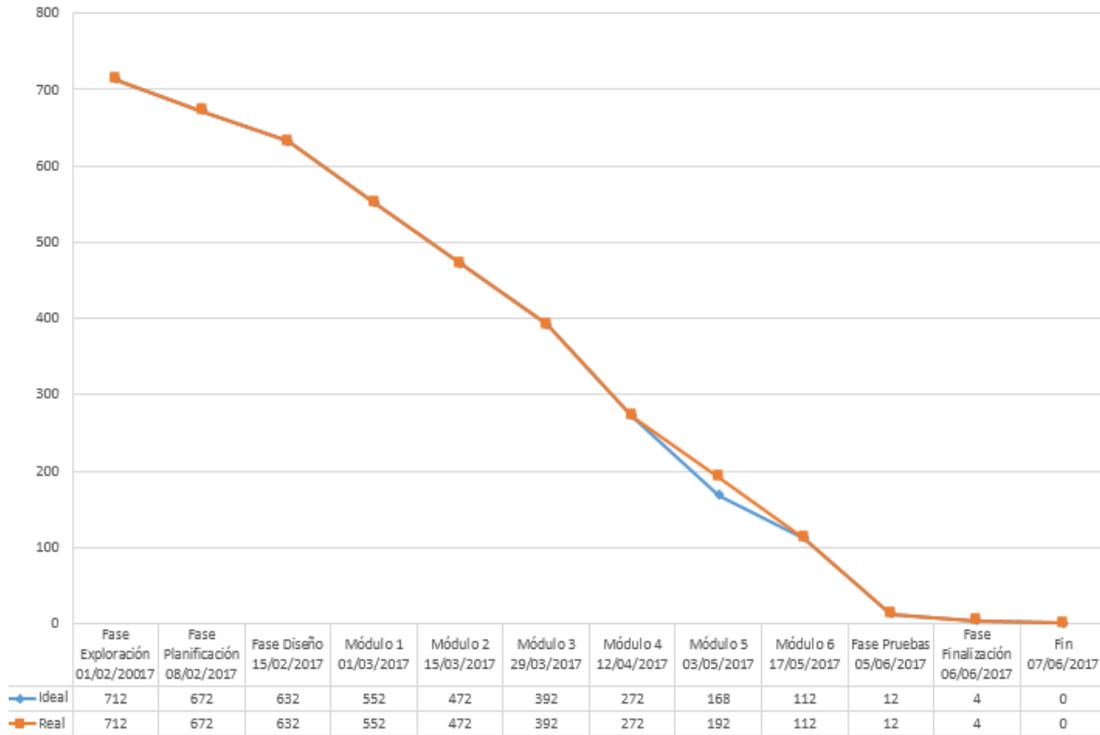


Gráfico 1 - 2: Burndown chart

Realizado por: Carranza Jerson, Morocho William. 2017.

CAPÍTULO III

3 ANÁLISIS DE RESULTADOS

En este capítulo se detalla el análisis de todos los resultados obtenidos, las mismas que se obtuvieron utilizando diversas técnicas como son las encuestas y entrevistas, que se realizaron a los estudiantes y al personal administrativo de la fundación “JASPE”, con el objetivo de conseguir información necesaria para medir la funcionalidad del sistema web “SIWEFI”.

Actualmente la fundación JASPE cuenta con 313 estudiantes y 10 personas que manejan la parte administrativa, estos datos se los usarán para determinar el tamaño de la muestra.

3.1 Generalidades

Con el objetivo de determinar si el sistema desarrollado sirve de apoyo para el personal administrativo y los estudiantes de la fundación “JASPE”, se ha realizado un estudio donde se utiliza la evaluación del estándar ISO/IEC 9126 con respecto al criterio de funcionalidad, donde se han tomado en cuenta los siguientes indicadores: Adecuación, Exactitud, Interoperabilidad, Seguridad y cumplimiento funcional.

3.2 Determinación de tipos de usuario y tamaños de la muestra.

En esta sección se determinará cuantos tipos de usuario y la cantidad a quienes se aplicaran el test de funcionalidad.

Ecuación N° 1: Fórmula para el cálculo de la muestra

$$n = \frac{N\sigma^2 Z^2}{e^2(N - 1) + Z^2\sigma^2}$$

Donde:

n=el tamaño de la muestra de la población.

N=tamaño de la población.

σ=desviación estándar.

Z= nivel de confianza

e=error de la muestra.

➤ Ecuación tamaño muestra de estudiantes

Con un nivel de confianza del 95%, con un margen de error del 10% y con una desviación estándar del 0,5.

$$n = \frac{313 * 0,5^2 * 1,96^2}{(0,1^2(313 - 1)) + (1,96^2 * 0,5^2)}$$

$$n=74$$

Si $\frac{n}{N} > 10\%$ la muestra debe ser corregida.

$$\frac{74}{313} > 10\%$$

$$0,23 > 10\%$$

$$23\%$$

En este caso el valor de la muestra supera al 10% de la población total por lo cual debe ser protegida con esta fórmula.

$$nc = \frac{N \cdot n}{N + (n - 1)}$$

$$nc = \frac{313 * 74}{(313) + (74 - 1)}$$

$$nc = 60$$

Luego de aplicar la fórmula el nuevo valor de nuestra muestra es 60.

➤ **Determinar el tamaño de la muestra del personal administrativo.**

El número de personas en la planta administrativo son de 10 por lo tanto no es necesario sacar muestra.

Después de sacar las muestras, se obtuvo el número de persona de 70, lo que equivale al número de usuarios que examinarán el sistema "SIWEFI".

3.3 Definición de los parámetros

Los indicadores que están dentro de la iso/iec 9126 bajo el criterio de funcionalidad se detallan en la **tabla 1-3**.

Tabla 1 - 3: Definición de los parámetros

Indicador	Descripción
Adecuación	Capacidad del software para proveer el conjunto de funciones que cumplirá los requerimientos de usuario.
Exactitud	Precisión en los resultados esperados.
Interoperabilidad	Interactuar con otros sistemas.
Seguridad	Es para evaluar la protección de la información.
Cumplimiento Funcional	Que cumpla con todos los indicadores antes mencionado.

Fuente: ISO 9126.

3.4 Criterios de evaluación

Para medir si el sistema web “SIWEFI” es funcional, se utilizó como instrumento las encuestas y entrevista a los estudiantes y personal administrativo de la fundación “JASPE”, seleccionando a los encuestados a partir de la muestra que anteriormente fue detallada. Previamente se realizó capacitaciones al personal administrativo y a los estudiantes seleccionados. Para realizar la medición de satisfacción a todos los usuarios quienes utilizaran el sistema se utilizó la escala de Likert, que se encuentra detallada en la **tabla 2-3**.

Tabla 2 - 3: Tabla de Likert

Valor Cualitativo	Valor
Totalmente de acuerdo	5
De acuerdo	4
Indiferente	3
En desacuerdo	2
Totalmente en desacuerdo	1

Fuente: Likert Escala.

3.5 Análisis de Resultados

Los resultados que se describen fueron obtenidos a través de las encuestas, que fueron bajo los indicadores del criterio de funcionalidad. La encuesta se detalla en el **ANEXO G**.

Las preguntas 1 y 2 corresponden si el sistema informático le está ayudando en la gestión de la fundación y están dirigidos al personal administrativo. Existen 10 Personas que son equivalentes al 100%.

La pregunta número 1 se formuló de la siguiente manera: *¿El sistema te permite encontrar rápidamente la información?*

En esta pregunta respondieron 7 personas que estaban totalmente de acuerdo y 3 personas que están de acuerdo.

$$X5 = \frac{7 * 100}{10} = 70\%$$

$$X4 = \frac{3 * 100}{10} = 30\%$$

La pregunta número 2 se formuló de la siguiente manera: *¿El sistema ayuda a mejorar tus tiempos en los procesos de la empresa?*

En esta pregunta número respondieron 8 personas que estaban totalmente de acuerdo y 2 personas de acuerdo.

$$X5 = \frac{8 * 100}{10} = 80\%$$

$$X4 = \frac{2 * 100}{10} = 20\%$$

De esta manera obtenemos los porcentajes de la pregunta 1 y 2, que se encuentra detallado en la **tabla 3-3**.

Tabla 3 - 3: Resultados pregunta 1 y 2

Valor	5	4	3	2	1
Pregunta 1	70%	30%	0%	0%	0%
Pregunta 2	80%	20%	0%	0%	0%
Promedio	75%	25%	0%	0%	0%

Realizado por: Carranza Jerson, Morocho William. 2017

En el siguiente gráfico, presentamos los resultados en el diagrama pastel para tener mejor percepción de los resultados de la pregunta 1 y 2.

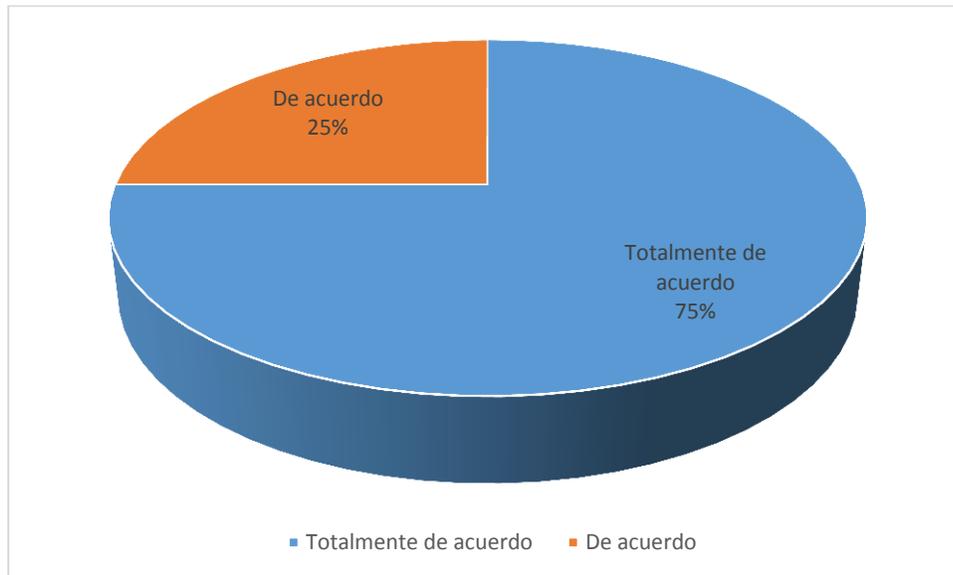


Gráfico 1 - 3: Satisfacción del sistema

Realizado por: Carranza Jerson, Morocho William. 2017.

En el gráfico podemos observar que el 75 % del personal administrativo están totalmente de acuerdo de que el sistema le ayuda a gestionar los procesos administrativos de la fundación.

Este análisis permite determinar como el sistema SIWEFI está ayudando a la fundación “JASPE”. Antes, de que el sistema se desarrolle, esta empresa se demoraba en buscar la información requerida por que se manejaba de forma manual utilizando diferentes software administrativos

como el Word, Excel entre otros, actualmente el sistema permite tener búsquedas avanzadas de los datos requeridos por el usuario del sistema.

El sistema SIWEFI además ha permitido mejorar los tiempos en las diferentes transacciones que se realiza en la fundación “Jaspe”, los usuario ya no tienen que esperar en secretaria para ser atendidos, ahora con SIWEFI todo el tramite empezando desde la pre matricula se los realiza de manera automática atreves de la web.

3.5.1 Funcionalidad.

Esta sección tiene relación directa con el objetivo específico, que es validar el sistema con el estándar ISO/IEC 9127 bajo el criterio de funcionalidad.

Para comprobar si el sistema es funcional, se realizó 7 preguntas donde abarca cuatros indicadores que son; adecuación, exactitud, interoperabilidad y seguridad.

3.5.1.1 Adecuación

Las preguntas 3 y 4 hacen referencia al indicador de adecuación, en la que nos indica si el software proporciona un conjunto apropiado de funciones para las tareas y objetivos del usuario.

Para tener resultados del indicador adecuación, se realizaron a las 70 personas, 60 son estudiantes y 10 son personal administrativo de la fundación. Por lo tanto, 70 personas equivalen al 100 %.

La pregunta número 3 se encuentra formulada de la siguiente forma: *¿Los módulos del sistema son suficientes para cumplir con sus procesos administrativos?*

En dicha pregunta número 3, respondieron 26 personas que estaban totalmente de acuerdo.

$$X5 = \frac{26 * 100}{70} = 37.14\%$$

43 personas estaban de acuerdo.

$$X4 = \frac{43 * 100}{70} = 61.46\%$$

Y una persona contesto que le era indiferente.

$$X3 = \frac{1 * 100}{70} = 1.40\%$$

En la pregunta número 4 que se la formuló de la siguiente forma: *¿Estás satisfecho con las funcionalidades del sistema?*

Respondieron 20 personas que estaban totalmente de acuerdo.

$$X5 = \frac{20 * 100}{70} = 28.58\%$$

43 personas estaban de acuerdo.

$$X4 = \frac{46 * 100}{70} = 65.71\%$$

Y una persona contesto que le era indiferente.

$$X3 = \frac{4 * 100}{70} = 5.71\%$$

De esta manera obtenemos los porcentajes de la pregunta 3 y 4 referente al indicador de adecuación, que se encuentra detallado en la **tabla 4-3**.

Tabla 4 - 3: Resultados 3 y 4

Valor	5	4	3	2	1
Pregunta 3	37.14%	61.46%	1.40%	0%	0%
Pregunta 4	28.58%	65.71%	5.71%	0%	0%
Promedio	32.86%	63.585%	3.555%	0%	0%

Realizado por: Carranza Jerson, Morocho William. 2017

En el siguiente gráfico, presentamos los resultados en el diagrama pastel para tener mejor percepción de los resultados de la pregunta 3 y 4 referente al indicador adecuación.

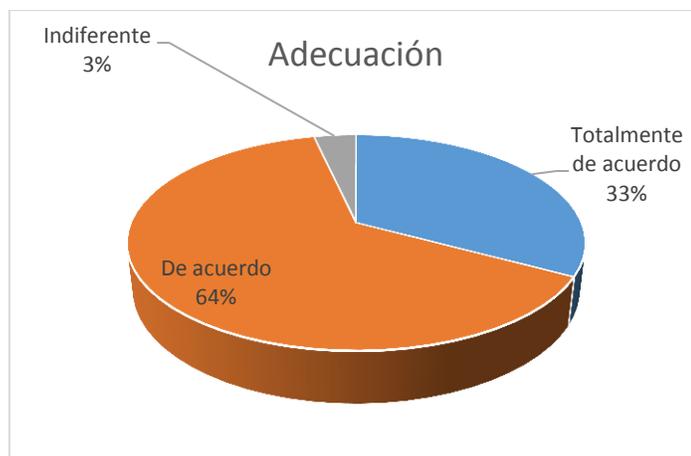


Gráfico 2 - 3: Indicador de Adecuación.
Realizado por: Carranza Jerson, Morocho William. 2017.

En este gráfico podemos observar que el 33 % de las personas encuestadas están de acuerdo, el 64% está totalmente de acuerdo, y al 3% le parece indiferente en la adecuación del sistema.

Los módulos con que SIWEFI cuenta han sido minuciosamente extraídos de la necesidad que la empresa tenía, esto permite crear módulos para que el sistema pueda funcionar adecuadamente en los distintos procesos administrativos que a diario se lleva a cabo. Estos resultados que se obtienen de la encuesta permiten observar que los usuarios se sientan satisfechos con el funcionamiento del sistema web de gestión financiera SIWEFI.

3.5.1.2 Exactitud.

Las preguntas 5 y 6 hacen referencia al indicador de exactitud, en la que nos indica si el software proporciona resultados correctos.

Para tener resultados del indicador de exactitud, se realizaron a las 70 personas, 60 son estudiantes y 10 son personal administrativo de la fundación. Por lo tanto, 70 personas equivalen al 100 %.

En la pregunta número 5, quedó de la siguiente manera formulada *¿Los resultados que proporciona el sistema son los esperados?* Respondieron 55 personas que estaban totalmente de acuerdo, 11 personas están de acuerdo y a 4 persona le parece indiferente.

$$X5 = \frac{55 * 100}{70} = 78.57\%$$

$$X4 = \frac{11 * 100}{70} = 15.71\%$$

$$X3 = \frac{4 * 100}{70} = 5.72\%$$

En la pregunta número 6 quedo de la siguiente manera formulada: *¿El lenguaje utilizado en el sistema es claro y conciso?* Respondieron 63 personas que estaban totalmente de acuerdo, 7 personas están de acuerdo que el lenguaje utilizado en el sistema es claro y conciso.

$$X5 = \frac{63 * 100}{70} = 90\%$$

$$X4 = \frac{7 * 100}{70} = 10\%$$

La mayoría está totalmente de acuerdo que el lenguaje utilizado en el sistema es claro y conciso.

Tabla 5 - 3: Resultados pregunta 5 y 6.

Valor	5	4	3	2	1
Pregunta 5	78.57%	15.71%	5.72%	0%	0%
Pregunta 6	90%	10%	0%	0%	0%
Promedio	84.29 %	12.86%	2.86%	0%	0%

Realizado por: Carranza Jerson, Morocho William. 2017

Gracias a la encuesta realizada en la fundación “JASPE”, permite determinar que los resultados que se obtiene al utilizar el sistema SIWEFI son los adecuados, permitiendo de esta manera tener

exactitud en cada reporte, en cada gráfico estadístico y en los procesos del área de contabilidad. Por esta razón se tiene una aceptación por parte del usuario en 84,29%.

3.5.1.3 Seguridad.

Las preguntas 7 y 8 hacen referencia al indicador de seguridad, en la que nos indica si el software previene acceso no autorizado.

Para poseer los resultados del indicador de seguridad, se realizaron a las 70 personas, 60 son estudiantes y 10 son personal administrativo de la fundación. Por lo tanto, 70 personas equivalen al 100 %.

En la pregunta número 7, está relacionado de quienes pueden acceder a la información del administrador. Y se formuló de la siguiente manera: *¿La información solo puede acceder el administrador?*

Lo cual 68 respondieron personas que estaban totalmente de acuerdo y 2 personas están de acuerdo.

$$X5 = \frac{68 * 100}{70} = 97.14\%$$

$$X4 = \frac{2 * 100}{70} = 2.86\%$$

Para la pregunta número 8, se les pregunto a los 10 administrativos de la fundación y está relacionado con la asignación de permisos y se encuentra formulada de la siguiente forma: *¿Puede asignar permisos a los usuarios el sistema?*

6 respondieron personas que estaban totalmente de acuerdo con la funcionalidad del establecimiento de permisos y 4 personas están de acuerdo.

$$X5 = \frac{5 * 10}{10} = 60\%$$

$$X4 = \frac{5 * 10}{10} = 40\%$$

A continuación, en la **tabla 6 -3** se muestran los resultados válidos a la pregunta 7 y 8 que se refiere a la seguridad del sistema.

Tabla 6 - 3: Resultados pregunta 7 y 8

Valor	5	4	3	2	1
Pregunta 7	97.14%	2.86%	0%	0%	0%
Pregunta 8	60%	40%	0%	0%	0%
Promedio	78.57 %	21.43%	0%	0%	0%

Realizado por: Carranza Jerson, Morocho William. 2017

En el siguiente gráfico, presentamos los resultados en el diagrama pastel para tener mejor percepción de los resultados de la pregunta 7 y 8 referente al indicador exactitud.



Gráfico 3 - 3: Indicador de Seguridad.

Realizado por: Carranza Jerson, Morocho William. 2017.

En el gráfico se observa que el 84 % de las personas encuestadas están totalmente de acuerdo, el 13 % está totalmente de acuerdo, y al 3 % le parece indiferente en la adecuación del sistema.

Para la fundación “JASPE” la seguridad de la información es importante, por esta razón en una entrevista que se sostuvo con el director de la fundación se llegó a un acuerdo, el mismo que permitió crear distintos tipos de usuario. Cada uno de estos tenía permisos, privilegios y limitaciones dentro del sistema de acuerdo al rol que ellos manejaban. Por esta razón el sistema SIWEFI tiene su aceptación dentro de la empresa.

3.5.1.4 Interoperabilidad.

Este indicador consiste si el sistema puede interactuar con otros sistemas. Se realizó una pregunta la cual se encuentra formulada de la siguiente forma.

¿El sistema “SIWEFI” puede interactuar con otros sistemas?

Para ello se preguntó a los administrativos de la empresa. Absolutamente todos estaban de totalmente de acuerdo.

$$X5 = \frac{10 * 100}{10} \rightarrow 100\%$$

3.5.1.5 Cumplimiento funcional.

Estos resultados están separados por los indicadores anteriores, también es conocido como el indicador del cumplimiento funcional.

Tabla 7 - 3: Cumplimiento funcional

Valor	5	4	3	2	1
Adecuación	32.86%	63.59%	3.55%	0%	0%
Exactitud	84.29%	12.86%	2.86%	0%	0%
Interoperabilidad	78.57%	21.43%	0%	0%	0%
Seguridad	100%	0%	0%	0%	0%
Promedio	73.93 %	24.47%	1.6%	0%	0%

Realizado por: Carranza Jerson, Morocho William. 2017

Dando como resultado, que el 73.93 % de las personas encuestadas están totalmente de acuerdo de que el sistema es totalmente funcional. Un 24.47 % están de acuerdo con la funcionalidad del sistema y a una menoría le parecía indiferente la funcionalidad del sistema. En el grafico observamos los resultados de la funcionalidad del sistema.

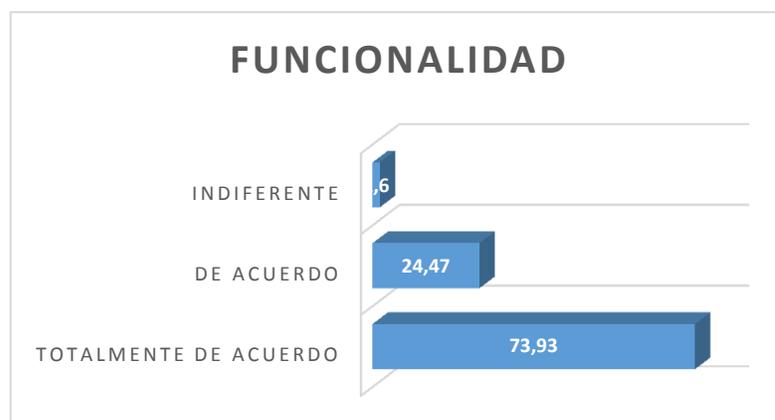


Gráfico 4 - 3: Funcionalidad
Realizado por: Carranza Jerson, Morocho William. 2017

Análisis Final: con la evaluación de la adecuación, exactitud, interoperabilidad, seguridad y cumplimiento funcional realizada por los estudiantes y personal administrativos se logró evaluar la funcionalidad del sistema, se establece como aceptada obteniendo un resultado del 73.93%, considerado un producto funcional.

3.6 Análisis de Rendimiento de la base de datos noSQL frente a una base de datos SQL(PostgreSQL)

Analizar el rendimiento de la base de datos MongoDB es importante, ya que nos permite obtener conclusiones en cuanto a su capacidad de procesamiento de datos. Para realizar este análisis se desarrolló una comparación entre la base de datos PostgreSQL y MongoDB.

3.6.1 Plataforma Hardware

El equipo que se utilizó en las pruebas de rendimiento, tiene las siguientes características.

Tabla 8-3: Plataforma hardware

Tipo:	HP Pavilion Entertainment
Procesador:	Intel® Core™ i7-M480
Velocidad:	1,67 GHz
RAM	6,00 GB
Disco Duro:	500 GB

Realizado por: Carranza Jerson, Morocho William. 2017

3.6.2 Plataforma Software

Tabla 9-3: Plataforma Software

Sistema Operativo:	Windows 10 Pro 64bits
Gestor de Base de datos:	Postgresql 9.3 Community Server 64bits
Gestor de Base de datos:	MongoDB 2.6.10 64bits
Lenguaje de programación:	NODE.js

Realizado por: Carranza Jerson, Morocho William. 2017

3.6.3 Consultas a los gestores de bases de datos

Tabla 10-3: Test N°1

Test N°1	
Operación	INSERT
Objetivo	Obtener el tiempo de la inserción de 1.000, 10.000, 100.000 y 1.000.000 de registros en los dos sistemas gestores de base de datos.
Proceso	Se procede a utilizar un script, que realiza la inserción de miles de registros en una tabla/colección que poseen un modelo de datos similar.
Procedimiento	
<ol style="list-style-type: none"> 1. Crear el script en JavaScript para insertar los miles de registros en la tabla y colección de los gestores de base de datos. 2. Iniciar el servicio PostgreSQL 3. Ejecutar el script 4. Obtener el tiempo utilizado para la tarea. 5. Detener el servicio PostgreSQL 6. Iniciar el servicio MongoDB 7. Ejecutar el Script 8. Obtener el tiempo utilizado para la tarea 	

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 11-3: Test N°2

Test N°2	
Operación	Select
Objetivo	Obtener el tiempo de las diferentes consultas que se realizan a los registros.
Proceso	Se mide el tiempo de respuesta de las transacciones select.
Procedimiento	
<ol style="list-style-type: none"> 1. Crear el script para seleccionar los registros en la tabla. 2. Iniciar el servicio PostgreSQL 3. Ejecutar el script 4. Obtener el tiempo utilizado para la tarea. 5. Detener el servicio PostgreSQL 6. Iniciar el servicio MongoDB 7. Ejecutar el Script 8. Obtener el tiempo utilizado para la tarea 	

Realizado por: Carranza Jerson, Morocho William. 2017

Tabla 12-3: Test N°3

Test N°3	
Operación	DELETE
Objetivo	Obtener el tiempo al borrar un número determinado de filas en los gestores de bases de datos.
Proceso	Luego de realizar la inserción masiva de registros (actualmente la tabla y colección “Estudiante” cuenta con 1.111.000 registros), se procederá a medir el tiempo que los dos gestores tardan en la eliminación de los registros.
Procedimiento	
<ol style="list-style-type: none"> 1. Crear el script para eliminar todos los miles de registros en la tabla y colección de los gestores de base de datos. 2. Iniciar el servicio PostgreSQL 3. Ejecutar el script 4. Obtener el tiempo utilizado para la tarea. 5. Detener el servicio PostgreSQL 6. Iniciar el servicio MongoDB 7. Ejecutar el Script 8. Obtener el tiempo utilizado para la tarea 	

Realizado por: Carranza Jerson, Morocho William. 2017

3.6.4 Métricas

Para medir el rendimiento de las bases de datos en lo que se refiere a tareas de consulta y manipulación de datos, es necesario una métrica común. El factor más importante para una aplicación es el tiempo requerido para completar una tarea, y en el caso de las bases de datos es el tiempo requerido para completar una transacción.

Los scripts creados miden el tiempo requerido para completar un número determinado de transacciones, ya que cada transacción en sí misma es insignificante. También, se obtiene el promedio del tiempo de ejecución de cada benchmark (técnica utilizada para medir el rendimiento de un sistema) realizado, con el fin de tener una idea global de cómo se comportan las bases de datos frente a factores externos como: un proceso del sistema operativo utilizando temporalmente el CPU o una operación de I/O.

3.6.5 Pruebas de rendimiento

A continuación se muestra los resultados simplificados de las pruebas de rendimiento que se hizo en los gestores de base de datos PostgreSQL y MongoDB. Para cada uno de los resultados de los script se ha realizado capturas de pantalla los mismos que se pueden observar en el **ANEXO I**. En estas capturas se pueden observar el tiempo que se demora cada uno de los gestores de a base de datos en realizar una transacción sea esta un *INSERT*, *SELECT* O *DELETE*.

3.6.5.1 Operación: Inserción masiva de registro

La inserción masiva de registros en los dos gestores de base de datos se realiza desde el mismo gestor de base de datos. El objetivo de esta prueba es observar el comportamiento de la base de datos, según el número de registros insertados.

La siguiente tabla, muestra el tiempo de inserción para diferentes números de registros en MongoDB y PostgreSQL.

Tabla 13-3: Inserción masiva

Número de registros insertados	Tiempo de inserción(milisegundos)	
	PostgreSQL	MongoDB
1.000	137	21
10.000	1567	150
100.000	18454	1468
1.000.000	1148778	19211
Promedio de tiempo	292234	5212,5

Realizado por: Carranza Jerson, Morocho William. 2017

Durante la primera fase de pruebas, que es la fase de carga de registros, se han insertado en total 1.111.000 registros. En la tabla anterior se puede visualizar que MongoDB realiza las inserciones rápidamente en comparación de PostgreSQL.

3.6.5.2 Operación: Consulta masiva

Se realiza la consulta masiva, el tiempo de respuesta se medirá en milisegundos, cabe mencionar que para el select de 1.111.000 millones de registros; se utilizó las mismas características hardware antes mencionadas en la tabla de plataforma hardware.

La siguiente tabla muestra el tiempo de ejecución de las consultas en los dos gestores de bases de datos con los diferentes números de registro:

Tabla 14-3: Consulta Masiva

Número de registros insertados	Tiempo de consulta(milisegundos)	
	PostgreSQL	MongoDB
1.000	137	2
1.0000	1567	26
100.000	5561	127
1.111.000	62840	1126
Promedio de tiempo	17526,25	320,25

Realizado por: Carranza Jerson, Morocho William. 2017

3.6.5.3 Operación: Eliminación Masiva

Se realizarán la eliminación de todos los registros empezando por los 1.111.000 millones de registros; luego se procederá a eliminar 100.000 registros, seguido de estos se eliminara 10.000 y 1000 registros en los dos gestores de base de datos.

En la siguiente tabla se muestra los tiempos de cada una de las respectivos DELETE que se realizaron en las respectivas bases de datos.

Tabla 15-3: Eliminar Masivo.

Número de registros insertados	Tiempo de eliminados(miliseundos)	
	PostgreSQL	MongoDB
1000	62	22
10000	342	162
100000	3805	1198
1111000	1148778	16880
Promedio de tiempo	288246,75	4565,5

Realizado por: Carranza Jerson, Morocho William. 2017

3.6.6 *Análisis de Resultados*

Durante la fase de carga, se comenzó con la inserción de 1.000, 10.000, 100.000 y 1.000.000 de registros. Luego de ejecutar cada uno de los scripts arrojaron resultados significativos en favor de MongoDB, en el proceso de ingresar registros se vio una notable ventaja con respecto a la base de dato noSQL MongoDB, se tiene que el tiempo de respuesta de PostgreSQL al ingresar 1.000 registros es de 137 milisegundos y de MongoDB es 21 milisegundos, en la segunda fase de inserción para PostgreSQL al ingresar 10.000 registros es de 1567 milisegundos y para MongoDB es de 150 milisegundos, en la tercera fase de inserción para PostgreSQL al ingresar 100.000 registros es de 18454 milisegundos y para MongoDB es de 1468 milisegundos, cave recalcar que mientras más aumentaba los números de registros; más notoria era la ventaja de la base de datos MongoDB por lo cual se procedió a la cuarta fase de inserción masiva con 1.000.000 millón de registros para PostgreSQL se obtuvo 1148778 milisegundos en tiempo de respuesta y para MongoDB 192211.

De todo este proceso de inserción se sacó un promedio en el cual se afirma que para el gestor de base de datos PostgreSQL el tiempo de respuesta es de 292234 mili segundos y para MongoDB es de 5212,5 mili segundos esto permite mencionar que MongoDB es 56 veces más rápido en la inserción de registros que PostgreSQL.

A continuación, se muestra un gráfico del tiempo de inserción de los dos gestores:



Gráfico 2 - 3: Tiempo promedio de inserción de los dos gestores.

Realizado por: Carranza Jerson, Morocho William. 2017.

Para la segunda fase de pruebas, el rendimiento de los dos gestores se medirá en milisegundos. En esta fase de consulta de registros, se puede observar que MongoDB mantiene la ventaja sobre PostgreSQL a medida que el número de registros recuperados se incrementa. Realizando el promedio de los tiempos de transacción en las consultas de los datos se tiene que PostgreSQL se demora 292234 milisegundos en comparación de MongoDB que se demora 5212, este análisis permite llegar a la conclusión que MongoDB es 54 veces más veloz que PostgreSQL como se puede observar en el Gráfico 6-3.

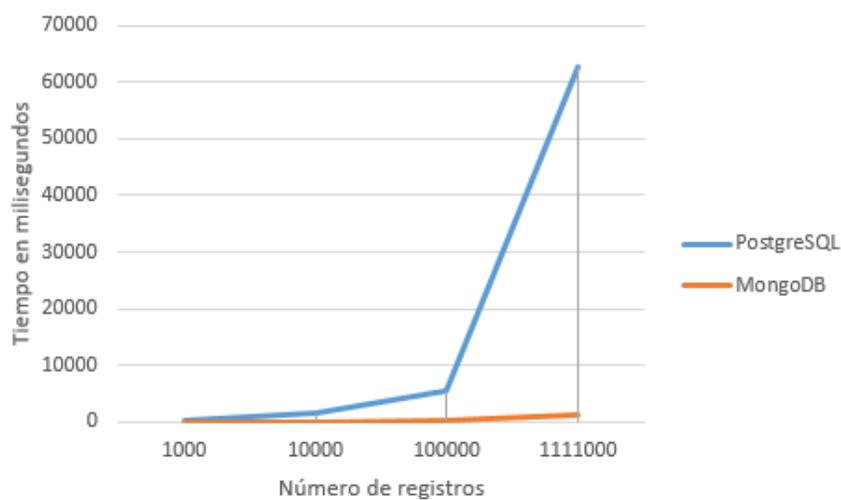


Gráfico 6 - 3: Tiempo de respuesta de los select.

Realizado por: Carranza Jerson, Morocho William. 2017.

El mismo comportamiento que anteriormente se observó en las gráficas, se tiene en la eliminación de los registros, sacando el promedio de las mismas se tiene que el gestor de base de datos PostgreSQL se demora en realizar esta transacción 1152987 milisegundos y MongoDB se demora 18262 milisegundos en realizar esta transacción, con estos datos se puede decir que MongoDB al realizar este proceso es 63 veces más rápido que PostgreSQL.

En el siguiente gráfico se muestra el tiempo de eliminación de los dos gestores de base de datos.

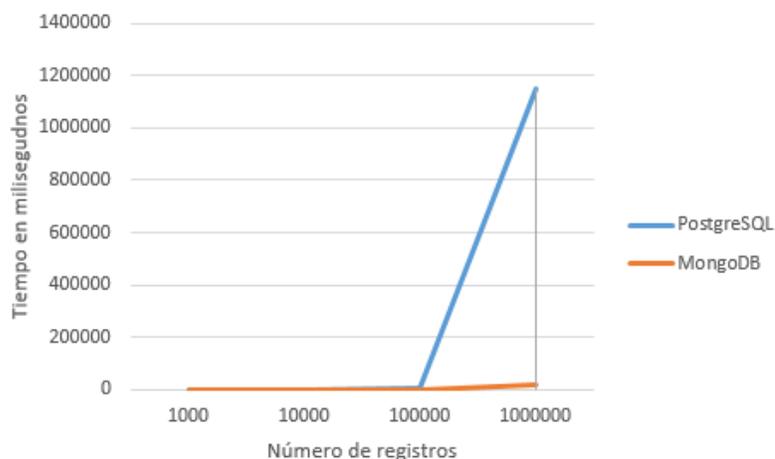


Gráfico 7 - 3: Tiempo de respuesta de los “DELETE”.
Realizado por: Carranza Jerson, Morocho William. 2017.

CONCLUSIONES.

- A través de las técnicas que se aplicaron en la fundación “JASPE”, como fue la entrevista y la observación, se logró determinar las necesidades existentes en la empresa, esto permitió obtener los requerimientos del usuario para el desarrollo del sistema SIWEFI.
- Para el desarrollo del sistema SIWEFI se aplicó la metodología XP, el mismo que permitió realizar una programación organizada e interactiva con el usuario, logrando de esta manera tener una menor tasa de errores y realizar versiones nuevas del sistema SIWEFI.
- Tras realizar la validación y los análisis de los indicadores de funcionalidad bajo los criterios de: adecuación, exactitud, interoperabilidad y seguridad; a través de un test se pudo determinar que el nivel de aceptación del sistema de gestión financiera es del 73.04 % por lo que se considera al sistema SIWEFI es funcional.
- El uso de la base de datos MongoDB, permitió realizar cambios en el proyecto, como añadir campos o arreglos que se habían obviado al principio. Sin embargo, en los query, para realizar consultas se tuvieron problemas al acceder a los campos embebidos, porque los resultados no eran los esperados.
- Al realizar las pruebas de inserción, se ha demostrado que en términos de tiempos de ejecución, MongoDB supera a PostgreSQL, esto se debe a que MongoDB no impone un esquema a los documentos que son almacenados en la colección. Debido a que, cada documento puede tener su propio conjunto definido de campos, sin tener la necesidad de alterar la estructura o crear otra colección.

RECOMENDACIONES.

- Es aconsejable aplicar diversas técnicas para obtener información adecuada de los procesos que lleva a cabo la empresa, esto permitirá desarrollar un correcto análisis de las necesidades de la empresa sea cual sea la actividad que esta desempeñe.
- Al momento de utilizar la metodología XP, se debe tener en claro cuáles son los respectivos roles dentro del desarrollo de software para que no exista contratiempos en la entrega del proyecto.
- Para verificar la calidad del producto software, es aconsejable aplicar algunas de las métricas internas importantes como son usabilidad, funcionalidad y eficiencia; las mismas que permiten verificar la calidad del producto software. Estas métricas están en la norma ISO 9126.
- Tener en cuenta que cuando se usa bases de datos documentales, no se debe embeber datos a más de dos niveles, ya que se complica al momento de realizar consultas, porque los resultados no son los esperados. Por eso es recomendable crear otro documento.
- Para el modelamiento de las bases de datos noSQL, no existe un estándar definido por lo cual, se recomienda futuras investigaciones en este campo, esto facilitaría a los desarrolladores crear diagramas entendibles y universales.
- Es aconsejable utilizar bases de datos SQL y noSQL en ambientes de desarrollo grandes para sacar provecho de los dos mundos ya que no existe una base de datos ideal para un sistema específico.

GLOSARIO

AngularJS: angular es un framework que tiene el código abierto originario de JavaScript.

NodeJS: es un entorno de código abierto para javascript, el mismo que está preparado para la capa del servidor. Este entorno está basado en el lenguaje ECMAScript el mismo que está basado en el motor V8 de google.

Sistema Web: es aquel sistema que la mayoría de sus procesos se encuentran a través de la red Word Wide Web.

Extreme Programming(XP): es un término que está escrito en inglés que significa programación extrema la misma que es una metodología ágil para el desarrollo de software.

noSQL: son sistemas de gestión de bases de datos que no utilizan el lenguaje SQL como principal, es decir que difieren de las bases de datos tradicionales.

ISO: es la Organización Internacional de Normalización donde se crean estándares internacionales.

Json: es un formato de texto ligero que permite realizar el intercambio de datos.

Metáforas del sistema: una metáfora es una historia que describe cómo funciona el programa.

SIWEFI: sistema web de gestión financiera para la fundación JASPE.

Historia de usuario: es una forma para especificar los requisitos del usuario.

Prueba de aceptación: permite verificar que la historia de usuario se cumplió adecuadamente.

MongoDB: es una base de datos noSQL que brinda escalabilidad, rendimiento y gran disponibilidad permitiendo a las empresas ser ágiles.

Tarea de ingeniería: en las tareas de ingeniería se almacena información relevante para identificar y detallar la historia de usuario.

Api: es un conjunto de funciones y procedimientos que se ofrece como biblioteca para que pueda ser utilizado por otro software.

Rest: es una forma de arquitectura de software para sistemas hipermedia distribuidos en la red .

JavaScript: es un lenguaje de programación que tiene el estándar ECMAScript, es un lenguaje orientado a objetos.

BIBLIOGRAFÍA

1. **ANAND, V; RAO, C.** "MongoDB and Oracle NoSQL: A technical critique for design decisions", *1st International Conference on Emerging Trends in Engineering, Technology and Science*[en línea], 2016 (India), volumen (2), pp. 1-4. [Consulta: 22 de marzo de 2017]. ISBN 978-146736725-7. Disponible en: <http://ieeexplore.ieee.org/abstract/document/7602984/>
2. **BESADA, E; BERMÚDEZ, J; DE LA TORRE, L; LOPEZ, J; DE LA CRUZ,J.**” Lightweight Node.js & EJS-based Web Server for Remote Control Laboratories”. *IFAC-PapersOnLine*[en línea], 2016,(España) volumen (49),pp. 127-132.[Consulta:22 de marzo 2017].ISSN 2341-7564. Disponible en: <https://doi.org/10.1016/j.ifacol.2016.07.165>
3. **CANÓS, H; LETELIER, P; PENADÉS, M.** “Metodologías ágiles en el desarrollo de software”. *Metodologías Ágiles en el Desarrollo de Software* [en línea], 2003, (España) 1(10), pp. 1-8. [Consulta: 23 agosto 2009]. ISSN 46022. Disponible en: http://www.academia.edu/download/33257239/METODOLOGIAS_AGLES.pdf#page=9
4. **CATTELL, R.** “*Scalable SQL and NoSQL data stores*”. ACM SIGMOD[en línea], 2010, (USA)volumen(39), pp. 12-27.[Consulta: 22 de marzo 2017]. ISSN 0163-5808 .Disponible en: <https://pdfs.semanticscholar.org/51d6/588ff7c1994f035a5a3be8d2e8ca62b78f22.pdf>
5. **GARRIDO, J.** “Arquitectura y diseño de sistemas Web modernos”. *Revista de Ingeniería Informática del CIIRM*,2004, (España) volumen(1), pp. 1-6. [Consulta: 30 de marzo de 2017]. ISSN: 1698-8841. Disponible en: http://pegaso.ls.fi.upm.es/~sortega/html_css/files/Arquitectura_y_diseno_de_sistemas_web_modernos.pdf

6. **HOWS, D; MEMBREY, P; PLUGG** *MongoDB Basics*[en línea]. New York-USA: Apress,2014. [Consulta: 23 de marzo de 2017]. Disponible en: <http://site.ebrary.com/lib/espoch/reader.action?docID=10682482&p00=history+NoSQL&ppg=47>

7. **KROL, J.** *Web Development with MongoDB and NodeJS*[en línea].Olton-USA: Packt Publishing, 2014. [21 March 2017]. Disponible en: <http://site.ebrary.com/lib/espoch/detail.action?docID=10944922>

8. **LI, Xiaosong; PRASAD, Christine.** “Effectively teaching coding standards in programming”. *Proceedings of the sixth conference on Information technology education*, volumen. 15, nº 5 (2005), (New Zealand). pp. 239-244.

9. **LÓPEZ, A; SÁNCHEZ, A; MONTEJANO, G.** “Definición de Métricas de Calidad para Productos de Software”. *En XVIII Workshop de Investigadores en Ciencias de la Computación* [en línea], 2016(Argentina) volumen(1), pp. 1-6. [Consulta: 23 de marzo de 2017]. ISBN: 978-950-698-377-2. Disponible en: <http://hdl.handle.net/10915/53444>

10. **MERINO, Cristian.** “Desarrollo Del Sistema Académico Del Sindicato De Choferes Profesionales 4 De Octubre Aplicando El Framework Js” [En línea]. (tesis pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela Ingeniería en Sistemas. Riobamba, Ecuador. 2017. pp. 19- 20 [Consulta: 22 de marzo 2017]. Disponible en: <http://hdl.handle.net/10915/5541>

11. **MORA, S.** *Programación de aplicaciones web: historia, principios básicos y clientes web* [en línea]. Alicante-España: Editorial Club Universitario, 2002. [Consulta: 30 de marzo de 2017]. Disponible en: https://books.google.es/books?hl=es&lr=&id=r9CqDYh2-loC&oi=fnd&pg=PR3&dq=aplicaciones+web&ots=MIFOYm8QH1&sig=vh4tMzvscQAkW25PgoYKNJ_5uxw#v=onepage&q=aplicaciones%20web&f=false

12. **ORACLE.** ¿Qué es una Base de Datos NoSQL? [blog]. [Consulta: 30 de marzo de 2017]. Disponible en: https://blogs.oracle.com/uncafeconoracle/entry/base_de_datos_nosql1
13. **PARKER, Z; POE, S; VRBSKY, S.** “Comparing nosql MongoDB to and sql db”. In *Proceedings of the 51st ACM Southeast Conference*[en línea], 2013,(USA) volumen(1), pp.1-6.[Consulta: 23 de marzo de 2017]. ISBN: 978-1-4503-1901-0. Disponible en: https://www.researchgate.net/profile/SV_Vrbsky/publication/261848669_Comparing_nosql_MongoDB_to_an_sql_db/links/565c618608aefe619b25294b.pdf
14. **PASQUALI, S.** *Mastering Node.js*[en línea]. Olton-USA: Packt Publishing, 2013. [Consulta: 22 March 2017]. Disponible en: <http://site.ebrary.com/lib/epoch/reader.action?ppg=32&docID=10813424&tm=1490199473914>
15. **POKORNY, J.** “NoSQL databases: a step to database scalability in web environment”. *International Journal of Web Information Systems* [en línea], 2013,(Czech Republic),volumen(9), pp. 69-82. [Consulta: 22 de marzo de 2017]. ISSN 1744-0084. Disponible en: <http://www.emeraldinsight.com/doi/pdfplus/10.1108/17440081311316398>
16. **TARASIEWICZ, P.** *AngularJS*[en línea]. Vancouver-Canada: Brainy Software,2014.[Consulta: 22 Marzo 2017]. Disponible en: <http://site.ebrary.com/lib/epoch/detail.action?docID=10993739&p00=angularjs>
17. **TAURO, C; ARAVINDH, S; SHREEHARSHA, A.** “Comparative study of the new generation, agile, scalable, high performance NOSQL databases”. *International Journal of Computer Applications*[en línea], 2012,(India), volumen (48), p. 1-4.[Consulta: 22 de marzo de 2017]. ISSN 0975 – 888. Disponible en: <http://www.ijcaonline.org/archives/volume48/number20/7461-0336>
18. **VAISH, G.** *Getting Started with Nosql*[en línea]. Birmingham-Reino Unido: Packt Publishing, 2013. [Consulta: 22 de marzo 2017]. Disponible en:

<https://books.google.es/books?id=oPiT-V2eYTsc&lpg=PT9&ots=sGvBj0IHju&dq=history%20NoSQL%20%20vaish&lr&hl=es&pg=PT9#v=onepage&q&f=false>

19. **WOOD, S; MICHAELIDES, G; THOMSON, C.** “Successful Extreme Programming: Fidelity To The Methodology Or Good Teamworking?”. *Information and Software Technology* [en línea], 2013, (United Kingdom), 55(4), pp.660-672. [Consulta: 21 marzo 2017]. ISSN 1090-7807. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0950584912002091>

ANEXOS

ANEXO A: Hojas de gestión de riesgos

Hoja de gestión de Riesgo R2

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R2		FECHA: 03/02/2017	
Probabilidad: Media Valor: 1	Impacto: Crítico Valor: 4	Exposición: Alta Valor: 4	Prioridad: 3
DESCRIPCIÓN: Diseñar mal la base de datos			
REFINAMIENTO			
Causas <ul style="list-style-type: none">• Mala definición de los requerimientos funcionales.• Poca experiencia por parte del programador en realizar diseño de base de datos.			
Consecuencias <ul style="list-style-type: none">• Retraso en la planificación• Redundancia de datos.			
REDUCCIÓN <ul style="list-style-type: none">• Definir correctamente los requerimientos con el cliente.			
SUPERVISIÓN <ul style="list-style-type: none">• Realizar reuniones con el equipo de desarrollo y el cliente.			
GESTIÓN <ul style="list-style-type: none">• Respaldo en la documentación en el desarrollo del proyecto• Capacitación a los desarrolladores con cursos de bases de datos			
ESTADO ACTUAL			
Fase de reducción iniciada		X	
Fase de Supervisión iniciada			
Gestionando el riesgo			
Responsables William Morocho Jerson Carranza			

Realizado por: Carranza Jerson, Morocho William. 2017

Hoja de gestión de Riesgo R3

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R3		FECHA: 07/02/2017	
Probabilidad: Media Valor: 2	Impacto: Moderado Valor: 2	Exposición: Media Valor: 4	Prioridad: 4
DESCRIPCIÓN: Cambios drásticos y continuos de los requisitos por parte del usuario.			
REFINAMIENTO			
<p>Causas</p> <ul style="list-style-type: none"> • Los usuarios finales no tienen en claro lo que quiere que haga el sistema web. • Utilización de un modelo de desarrollo de software no adecuado. • Falta de comunicación entre el usuario y el equipo de desarrollo. <p>Consecuencias</p> <ul style="list-style-type: none"> • Pérdida de tiempo. • Mal diseño del proyecto de desarrollo. • El tiempo de entrega se vuelve extenso. 			
REDUCCIÓN			
<ul style="list-style-type: none"> • Reunirse continuamente entre el usuario y el equipo de desarrollo del proyecto. • Realizar prototipos del proyecto software para que los usuarios tengan el sistema palpable. 			
SUPERVISIÓN			
<ul style="list-style-type: none"> • Los usuarios verificarán que lo que el equipo de desarrollo está realizando está cumpliendo con sus necesidades. • El ambiente de trabajo que se cree debe ser el adecuado para el usuario y el desarrollador. 			
GESTIÓN			
<ul style="list-style-type: none"> • Supervisar continuamente los requerimientos de usuario. 			
ESTADO ACTUAL			
Fase de reducción iniciada			
Fase de Supervisión iniciada X			
Gestionando el riesgo			
RESPONSABLES			
William Morocho			
Jerson Carranza			

Realizado por: Carranza Jerson, Morocho William. 2017

Hoja de gestión de Riesgo R4

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R4		FECHA: 07/02/2017	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Bajo Valor: 3	Prioridad: 5
DESCRIPCIÓN: Interfaces complejas para que el usuario navegue en el sistema.			
REFINAMIENTO			
<p>Causas</p> <ul style="list-style-type: none"> • Escoger interfaces complejas para el diseño del proyecto software. • No tener mucho conocimiento en el diseño de interfaz. • Dificultades que el usuario tenga para acceder al sistema. <p>Consecuencias</p> <ul style="list-style-type: none"> • Mal diseño de la interfaz del sistema. • Pérdida de tiempo. 			
REDUCCIÓN			
<ul style="list-style-type: none"> • La interfaz del usuario debe ser lo más sencillo posible. • Realizar capacitaciones de la manipulación del sistema para que los usuarios no tengan problemas. 			
SUPERVISIÓN			
<ul style="list-style-type: none"> • Verificar que el usuario este complacido con la interfaz y no tenga complicaciones. 			
GESTIÓN			
<ul style="list-style-type: none"> • Volver a diseñar la interfaz de proyecto de desarrollo. • Re planificar en el diseño de la interfaz de usuario. 			
ESTADO ACTUAL			
Fase de reducción iniciada			
Fase de Supervisión iniciada X			
Gestionando el riesgo			
RESPONSABLES			
William Morocho			
Jerson Carranza			

Realizado por: Carranza Jerson, Morocho William. 2017

Hoja de gestión de Riesgo R5.

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R5		FECHA: 07/02/2017	
Probabilidad: Valor:	Impacto: Valor:	Exposición: Valor:	Prioridad:
DESCRIPCIÓN: Incumplimiento de entregas en fechas establecidas			
REFINAMIENTO Causas: <ul style="list-style-type: none">• Irresponsabilidad en el equipo de trabajo.• Incumplir con las tareas asignadas. Consecuencias <ul style="list-style-type: none">• Retraso en el desarrollo del proyecto.• No cumplir con la planificación.			
REDUCCIÓN <ul style="list-style-type: none">• Realizar supervisiones continuas.• Revisiones con las tareas en un periodo cortó. (2 días).			
SUPERVISIÓN <ul style="list-style-type: none">• Definir un líder.			
GESTIÓN <ul style="list-style-type: none">• Definir un jefe que se encargue de hacer supervisiones continuas de las tareas asignadas de cada integrante del equipo.			
ESTADO ACTUAL			
Fase de reducción iniciada		X	
Fase de Supervisión iniciada		X	
Gestionando el riesgo			
RESPONSABLES William Morocho Jerson Carranza			

Realizado por: Carranza Jerson, Morocho William. 2017

Hoja de gestión de Riesgo R6

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R6		FECHA: 06/02/2017	
Probabilidad: Baja Valor: 1	Impacto: Critico Valor: 4	Exposición: Media Valor: 4	Prioridad: 4
DESCRIPCIÓN: Cambio de autoridades en la fundación.			
REFINAMIENTO			
Causas			
<ul style="list-style-type: none"> • Nueva administración de la fundación “JASPE”. • Personal suplente que se asigna a la fundación. • Los directivos de la fundación toman nuevas decisiones. 			
Consecuencias			
<ul style="list-style-type: none"> • Pausa del proyecto software que está en desarrollo. • Se generan nuevos requerimientos por el nuevo personal. • Cambio en la planificación del proyecto de desarrollo. 			
REDUCCIÓN			
<ul style="list-style-type: none"> • Reuniones con las personas encargadas en el área de contabilidad. • Reuniones con el equipo asignado para el proyecto de desarrollo de software. 			
SUPERVISIÓN			
<ul style="list-style-type: none"> • Cerciorarse de que los requerimientos dados por el usuario se cumplan de la mejor forma. 			
GESTIÓN			
<ul style="list-style-type: none"> • Documentar las reuniones que se tienen con los directivos de la fundación cuando se tome decisiones. 			
ESTADO ACTUAL			
Fase de reducción iniciada X			
Fase de Supervisión iniciada			
Gestionando el riesgo			
RESPONSABLES			

William Morocho
Jerson Carranza

Realizado por: Carranza Jerson, Morocho William. 2017

Hoja de gestión de Riesgo R7.

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R7		FECHA: 06/02/2017	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Media Valor: 6	Prioridad: 3
DESCRIPCIÓN: Falta de apoyo de las autoridades para implementar el sistema.			
REFINAMIENTO			
Causa: <ul style="list-style-type: none">• No tiene el suficiente dinero para ejecutar el sistema. Consecuencia: <ul style="list-style-type: none">• El sistema no se podría implementado.			
REDUCCIÓN			
<ul style="list-style-type: none">• Antes de la ejecución del proyecto, verificar los costos.			
SUPERVISIÓN			
<ul style="list-style-type: none">• Verificar si existen hosting gratuitos.			
GESTIÓN			
<ul style="list-style-type: none">• Subir el software en un hosting gratuito para la implementación del sistema.			
ESTADO ACTUAL			
Fase de reducción iniciada X			
Fase de Supervisión iniciada X			
Gestionando el riesgo			
RESPONSABLES			
William Morocho			

Hoja de gestión de Riesgo R8.

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R8		FECHA: 07/02/2017	
Probabilidad: 1 Valor: Media	Impacto: 2 Valor: Moderado	Exposición: 2 Valor: Moderado	Prioridad: 2
DESCRIPCIÓN: No usar estándar en la programación			
REFINAMIENTO			
Causas: <ul style="list-style-type: none"> • Desconocimiento de estándar por parte del desarrollador. • No establecer un estándar cuando comienza el proyecto. Consecuencias <ul style="list-style-type: none"> • Cuando otro programador quiera modificar el código perderá tiempo. • No podrá pasar la calidad de los estándares. 			
REDUCCIÓN			
<ul style="list-style-type: none"> • Establecer un plan de estándar de codificación al inicio. 			
SUPERVISIÓN			
<ul style="list-style-type: none"> • Establecer un jefe para revisión de código. 			
GESTIÓN			
<ul style="list-style-type: none"> • Revisar en cada tarea que se esté usando ese estándar en la pareja de programadores 			
ESTADO ACTUAL			
Fase de reducción iniciada		X	
Fase de Supervisión iniciada		X	
Gestionando el riesgo			

RESPONSABLES

William Morocho

Jerson Carranza

Realizado por: Carranza Jerson, Morocho William. 2017

ANEXO A: Estándar de Codificación

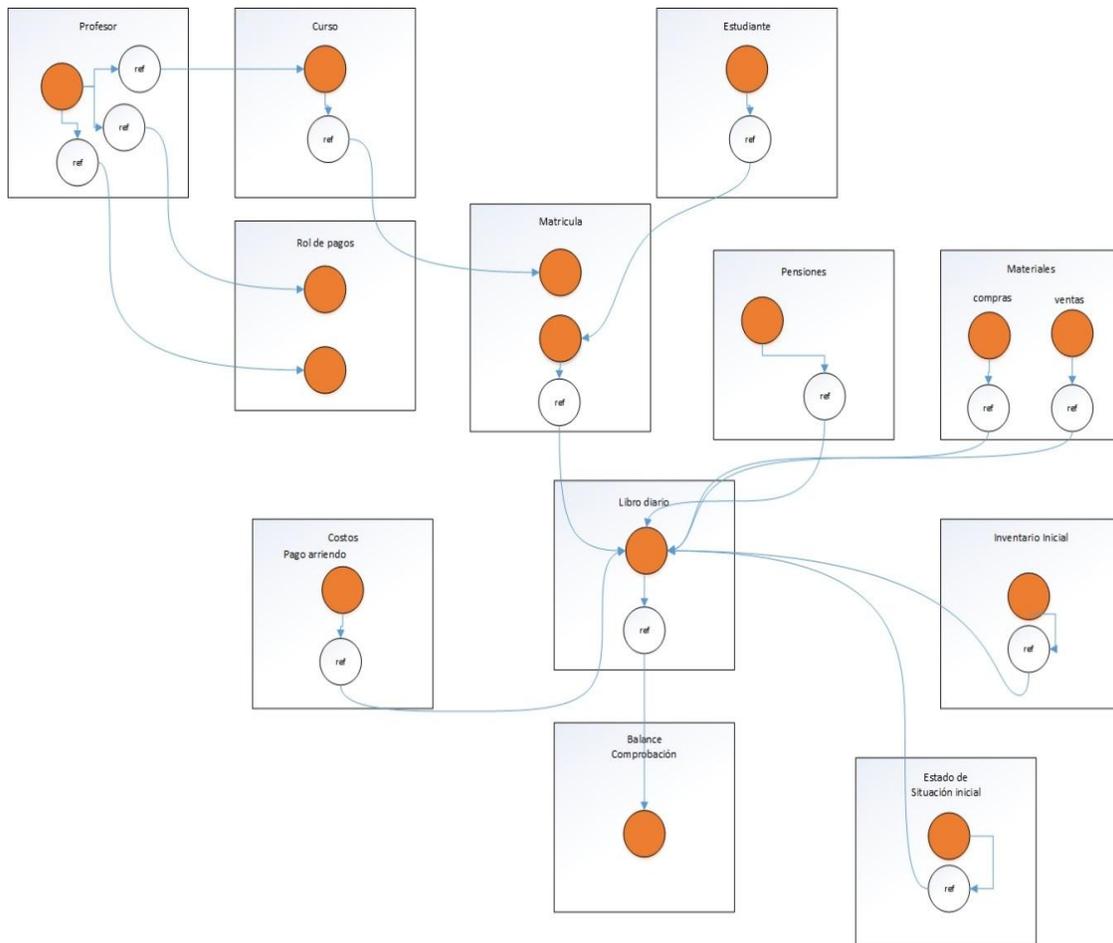
Documentación javaScript	<p>A continuación se especifican algunas etiquetas para funciones y clases:</p> <p>///"Encabezado", añade un cabecera descriptiva.</p> <p>Autor, añade información sobre los desarrolladores del código.</p> <p>Fecha y Versión, permite incluir información sobre la fecha de código y versión del código.</p> <p>Descripción, se le añadirá un breve detalle sobre lo que hará la clase.</p> <p>Callback, Retorna información después que el usuario realicé una petición.</p> <p>Err, Si existiera un error, lo asignaría a la variable <i>err</i>, para después usarlo correctamente.</p>
Sentencias de importación	<p>Para importar clases o librerías lo realizaremos en el siguiente orden.</p> <ol style="list-style-type: none">1. Librerías.2. Clases de la aplicación. <p>Con el siguiente formato</p> <pre>var nombreLibreria = require("nombreLibreria"); var nombreClase = require("nombreClase");</pre>
Sangría	<p>En las clases se establecerá 4 caracteres como unidad de sangría.</p>
Longitud de	<p>La longitud de línea no debe superar los 100 caracteres por motivos de</p>

línea	visualización e impresión.
Espacios en blanco	<p>Se utilizarán espacios en blanco en los siguientes casos:</p> <ol style="list-style-type: none"> 1. En cada coma en un listado de argumentos. Por ejemplo: funcion(a, b, c); 2. Cuando se exporta una función. Por ejemplo: exports.funcion = función() 3. Para llamar una función o librería. var clase = require('user.js'); var libreriaExpress = require('user.js'); 4. Cuando necesitamos separar las expresiones incluidas en la sentencia "for". Por ejemplo: for (expresion1; expresion2; expresion3). 5. Para asignar una valor a una variable. Ejemplo: Var \$scope.datos = datos. 6. Cuando enviamos una variable por la uri, ejemplo: \$http.get('/estudiante/listcedula/'+ id);
Carpetas o paquetes	Se escribirán siempre con letras minúsculas para no entrar en conflicto con los nombres de clases o interfaces.
Clases e interfaces	<p>Los nombres de clases deben ser sustantivos seguida de su nombre designado con la primera letra en minúscula y si existe una clase conformada por dos palabras el primer carácter de la primera palabra será con minúscula y el primer carácter de la segunda palabra será mayúscula.</p> <ul style="list-style-type: none"> • No usar acrónimos o abreviaturas. • Los nombres deben ser simples y descriptivos. • Ejemplo: estudianteController.
Métodos	Cuando el método esté compuesto por varias palabras cada una de ellas tendrá

	<p>la primera letra en mayúsculas, excepto la primera palabra.</p> <ul style="list-style-type: none"> funcionInsertarEstudiante
Variables	<p>Se definirán usando solo minúsculas que corresponde al nombre asignado.</p> <p>Las variables nunca podrán comenzar con caracteres especiales, números. Los nombres de variables deben ser cortos y sus significados tienen que expresar con la función que desempeñan en el código. Ejemplo</p> <ul style="list-style-type: none"> variablecontador

Realizado por: Carranza Jerson, Morocho William. 2017

ANEXO B: DISEÑO DE LA BASE DE DATOS NOSQL



Realizado por: Carranza Jerson, Morocho William. 2017

DICCIONARIO DE DATOS

DOCUMENTO DE ARRIENDO

Nombre	Campo	Tipo	Descripción
Código Arriendo	_id	ObjectId	Almacena el id curso (genera automáticamente)
Mes en letras	meses	String	Almacena el nombre del mes que canceló.
Mes en número	mesnumero	Number	Almacena el mes en número.
Cantidad	cantidad	Number	Almacena la cantidad que canceló.
Año	Anio	Number	Almacena el año que canceló
Fecha	Fecha	Date	Almacena la fecha

Realizado por: Carranza Jerson, Morocho William. 2017

DOCUMENTO CUENTA.

Nombre	Campo	Tipo	Descripción
Código cuenta	_id	ObjectId	Almacena el id de la cuenta (se genera automáticamente)
Código de la cuenta	codigocuenta	String	Almacena el código de acuerdo un plan de cuentas.
Nombre de la cuenta	nombrecuenta	String	Almacena el nombre la cuenta.

Realizado por: Carranza Jerson, Morocho William. 2017

DOCUMENTO DE CURSO.

Nombre	Campo	Tipo	Descripción
Código de cursos	_id	ObjectId	Almacena el id curso (genera automáticamente). También es el nombre que se le coloca al imagen.
Nombre del curso	Nombre	String	Almacena el nombre del curso.

Descripción de curso	Descripción	String	Almacena la descripción del curso.
Extensión de la imagen	Extensión	String	Almacena la extensión de la imagen del curso.
Matrícula del curso	Precio	Number	Almacena el precio de la matrícula del curso.
Pensión del curso	Pension	Number	Almacena la pensión del curso.
Duración del curso	Meses	Number	Almacena cuantos meses dura el curso.
Fecha de inicio	fecha_inicio	Date	Almacena la fecha de inicialización del curso.
Fecha de creación del curso	Date	Date	Almacena en fecha se creó el curso.

Realizado por: Carranza Jerson, Morocho William. 2017

DOCUMENTO ESTUDIANTE.

Nombre	Campo	Tipo	Descripción
Código de estudiantes	_id	ObjectId	Almacena el id estudiante (genera automáticamente).
Cédula	Cedula	String	Almacena el número de cédula.
Nombres del estudiante	nombres	String	Almacena los nombres del estudiante
Apellido Paterno	apellido1	String	Almacena el apellido paterno del estudiantes
Apellido Materno	apellido2	String	Almacena el apellido materno.
Sexo del estudiante	Sexo	String	Almacena el sexo del estudiante en palabras.
Auxiliar	auxiliar	String	Almacena un estado, y sirve para identificar si el estudiante se encuentra matriculado en alguna materia.

Cursos	Cursos	Array	Contienen los cursos de los estudiantes embebidos.
Código del curso	cursos.id_cursos	ObjectId	Almacena el código del curso
Nombre del curso	cursos.nombre_curso	String	Almacena el nombre del curso
Precio del curso	cursos.precio_cursos	String	Almacena el precio del curso
Estado del curso	cursos.estado	String	Nos permite ver si el estudiante se encuentra pre matriculado o matriculado.
Pagos	cursos.pagos	Array	Contiene los pagos de los estudiantes de cada curso
Mes de pago	cursos.pagos.mes	Number	Almacena el mes de pago del estudiante.
Mes de cantidad	cursos.pagos.cantidad	Number	Almacena la cantidad del pago, en dicho curso.
Tipo de pago	cursos.pagos.tipo	Number	Almacena el tipo de pago con que se canceló la pensión.
Fecha canceló	cursos.pagos.fecha	Date	Fecha que canceló el estudiante la pensión del curso.

Realizado por: Carranza Jerson, Morocho William. 2017

DOCUMENTO LIBRO DIARIO.

Nombre	Campo	Tipo	Descripción
Código del libro diario	_id	ObjectId	Almacena el id del asiento contable (genera automáticamente).
Descripción	Descripción	String	Almacena la descripción del asiento.
Fecha del asiento	Fecha	Date	Almacena cuando se generó el asiento
Haber	Haber	array	Es un campo embebido y guarda los datos del haber.
Código de	haber.codigo_cuenta	string	Almacena el código de la cuenta del

cuenta			haber.
Detalle	haber.detalle	string	Almacena el detalle del haber.
Identificador del haber	haber_id	ObjectId	Se genera automáticamente el id
Debe	Debe	Array	Es un campo embebido y guarda los datos del debe.
Código de cuenta	debe.codigo_cuenta	string	Almacena el código de la cuenta del debe.
Detalle	debe.detalle	string	Almacena el detalle del debe.
Identificador del debe	debe_id	ObjectId	Se genera automáticamente el id

Realizado por: Carranza Jerson, Morocho William. 2017

Documento Matrícula

Nombre	Campo	Tipo	Descripción
Código matrícula	_id	ObjectId	Almacena el id matrícula (genera automáticamente)
Fecha de la matrícula	fecha_matricula	Date	Almacena la fecha de la matrícula.
Estudiantes	estudiantes	Array	Almacena los datos de los estudiantes embebidos.
Nombre del estudiante	estudiantes.nombres	String	Almacena el nombre del estudiante.
Apellido Paterno	estudiantes.apellido1	String	Almacena el apellido paterno del estudiante matriculado.
Apellido Materno	estudiantes.apellido2	String	Almacena el apellido materno del estudiante matriculado.
Cédula	estudiantes .cedula	String	Almacena la cedula del estudiante.
Sexo	estudiantes.sexo	String	Almacena el sexo del estudiante matriculado.
Cursos	Cursos	Array	Almacena los datos de los curso elegidos por los estudiantes embebidos.

Nombre del curso	cursos.nombre	String	Almacena el nombre del curso que eligio el estudiante
------------------	---------------	--------	---

Realizado por: Carranza Jerson, Morocho William. 2017

Documento rol de pagos.

Nombre	Campo	Tipo	Descripción
Código del rol de pagos	_id	ObjectId	Almacena el id del rol de pagos (genera automáticamente)
Cédula	cedula	String	Almacena la cédula del profesor..
Nombre	nombre	String	Almacena el nombre completo del profesor.
Mes	Mes	Number	Almacena el mes que se le canceló.
Sueldo	comisión	Number	Almacena cuanto se canceló al trabajador
IESS trabajador	iesstrabajador	Number	Le descuenta un porcentaje al sueldo del trabajador.
IESS Patrono	iesspatrono	Number	Almacena el aporte por parte del patrono.
Fecha	Date	Date	Fecha que se canceló el rol de pagos.

Realizado por: Carranza Jerson, Morocho William. 2017

Documento del estado de situación inicial

Nombre	Campo	Tipo	Descripción
Identificador	Id	ObjectId	Almacena el id del Estado de Situación Inicial
Patrimonio	patrimonio	Array	Contiene las cuentas y cantidad que se encuentra en patrimonio.
Cuenta	patrimonio.cuenta	String	Almacena las cuentas que se encuentra en el patrimonio
Cantidad	patrimonio.cantidad	Number	Almacena la cantidad las cantidades

			de cada cuenta del administrador.
Código de cuenta	codigo_cuenta	String	Almacena el código de cuenta de cada asiento
Pasivo	pasivo	Array	Contiene las cuentas y cantidad que se encuentra en el pasivo.
Cuenta	pasivo.cuenta	String	Almacena las cuentas que se encuentra en el pasivo
cantidad	pasivo.cantidad	Number	Almacena la cantidad las cantidades de cada cuenta del administrador.
Código de cuenta	codigo_cuenta	String	Almacena el código de cuenta de cada asiento
Activo	activo	Array	Contiene las cuentas y cantidad que se encuentra en el activo.
Cuenta	activo.cuenta	String	Almacena las cuentas que se encuentra en el activo
cantidad	pasivo.cantidad	Number	Almacena la cantidad las cantidades de cada cuenta del administrador.
Código de cuenta	codigo_cuenta	String	Almacena el código de cuenta de cada asiento

Realizado por: Carranza Jerson, Morocho William. 2017

Documento profesores.

Nombre	Campo	Tipo	Descripción
Identificador	_id	ObjectId	Almacena el id del profesor
Cédula	Cedula	String	Almacena la cédula del profesor.
Nombre	nombre	String	Nombre del profesor.
Apellido	apellido	String	Almacena el apellido del profesor.
Sueldo	sueldo	Number	Almacena el sueldo del trabajador.

Realizado por: Carranza Jerson, Morocho William. 2017

Documento de usuarios.

Nombre	Campo	Tipo	Descripción
Identificador	_id	ObjectId	Almacena el id del usuario
Nombre	name	String	Almacena el nombre del usuario.
email	email	email	Almacena el correo del usuario.
Constraseña	password	Base64	Almacena la clave del usuario.

Realizado por: Carranza Jerson, Morocho William. 2017

Documento de Fundación.

Nombre	Campo	Tipo	Descripción
Identificador	_id	ObjectId	Almacena de la fundación.
Somos	somos	String	Almacena la información de la fundación.
Visión	vision	String	Almacena la visión de la fundación.
Misión	mision	String	Almacena la misión de la fundación.

Realizado por: Carranza Jerson, Morocho William. 2017

ANEXO D: Historia de usuarios.

Iteración 1

HU_01 Registro de estudiantes: en esta historia se desarrolló el registro de los estudiantes permitiendo de esta manera que el estudiante ingrese al sistema de acuerdo con el número de cédula y la contraseña, de esta manera el estudiante podrá seleccionar el curso a seguir. Como resultado de la elaboración de esta historia se generó 2 tareas de ingeniería y 4 pruebas de aceptación, las mismas que se detallaran a continuación en la tabla.

Historia de Usuario Registro de estudiantes.

Historia de Usuario	
Número: HU_01	Nombre de la historia: Registro de estudiantes.
Modificación de historia de usuario:	
Usuario: estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta	Puntos Reales: 24
Descripción: Como estudiante deseó registrarme para poder acceder al sistema.	
Observaciones: Los campos que se requieren para registrar a un estudiante son los siguientes: Cédula, contraseña, nombres, apellidos paterno, apellido materno, sexo.	

Historia de Usuario - Pruebas de Aceptación

- Si todos los campos no están llenos el botón registro se bloquea.
- Si el número de cédula introducido es incorrecto emitirá el respectivo mensaje de error.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario Registro de Estudiantes

Tarea de ingeniería	
Historia de Usuario: Registro de estudiantes.	
Número Tarea: TI_01HU01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 01/03/17	Fecha de Fin: 01/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el registro de los estudiantes.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario Registro de Estudiantes

Tarea de ingeniería	
Historia de Usuario: Registro de estudiantes.	
Número Tarea: TI_02 HU01	Nombre de la Tarea: Elaboración del controlador para el registro de los estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 02/03/17	Fecha de Fin: 03/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el registro de los estudiantes.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_02 Autenticación: en esta historia se elaboró la autenticación de usuario y se implementó sesiones los mismos que permitirán al usuario ingresar al sistema, esta autenticación consta de varias restricciones dependiendo del tipo de usuario que se haya autenticado. Como resultado de esta autenticación se obtuvo 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se muestran en la siguiente tabla.

Historia de usuario Autenticación

Historia de Usuario	
Número: HU_02	Nombre de la historia: Autenticación
Modificación de historia de usuario:	
Usuario: estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como estudiante deseé autenticarme para poder acceder al sistema dependiendo del tipo de usuario.	
Observaciones: los usuarios de tipo estudiante solo podrán registrarse en un curso. El usuario de tipo administrador tendrá acceso a todas las funciones.	

Historia de Usuario - Pruebas de Aceptación

- Si la autenticación es correcta se re direcciona a la página de opciones.

Realizado por: Jerson Carranza, William Morocho. 2017

Las siguientes tablas se encuentran detalladas las tareas de ingenierías.

Elaboración de la interfaz de usuario de la autenticación.

Tarea de ingeniería	
Historia de Usuario: Autenticación	
Número Tarea: TI_01HU02	Nombre de la Tarea: Elaboración de la interfaz de usuario de la autenticación.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 06/03/17	Fecha de Fin: 06/03/17
Programador Responsable: Jerson Carranza	

Descripción: Elaboración de la interfaz necesaria para la autenticación de los estudiantes.
Pruebas de Aceptación: La interfaz de la autenticación este correctamente diseñada bajo los estándares establecidos.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la autenticación de los usuarios.

Tarea de ingeniería	
Historia de Usuario: Autenticación.	
Número Tarea: TI_01HU02	Nombre de la Tarea: Elaboración del controlador para la autenticación de los usuarios.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 07/03/17	Fecha de Fin: 07/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la autenticación de los usuarios.	
Pruebas de Aceptación: Comprobar que la autenticación se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_03 Vistas cursos: se desarrolló esta historia de usuario la misma que permite visualizar los cursos existentes dentro de la fundación “Jaspe”, estas vistan deben tener el estándar que anteriormente se estableció con el usuario. En el desarrollo de esta historia de usuario se obtuvo como resultado 2 metáforas de sistema y 3 pruebas de aceptación.

Historia de usuario Vistas cursos.

Historia de Usuario	
Número: HU_03	Nombre de la historia: Vistas cursos.

Modificación de historia de usuario:	
Usuario: estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como estudiante deseó obtener las vistas de todos los cursos disponibles.	
Observaciones: Las vistas de los cursos deben estar disponibles para los estudiantes.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Las vistas de los cursos estén con la información adecuada.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario.

Tarea de ingeniería	
Historia de Usuario: Vistas cursos.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 08/03/17	Fecha de Fin: 08/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para las vistas de curso.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para las vistas de los cursos.

Tarea de ingeniería	
Historia de Usuario: Vistas cursos.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para las vistas de los cursos.

Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 09/03/17	Fecha de Fin: 09/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para generar las vistas de los cursos.	
Pruebas de Aceptación: Comprobar que las vistas se generen el orden adecuado.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_04 Añadir curso: esta historia de usuario se desarrolló con el fin de que el estudiante pueda seleccionar los diferentes cursos que ofrece la fundación, mediante esta función se realiza la pre matrícula, cabe resaltar que cuando un estudiante añade un curso al carrito está realizando la pre matrícula, de esta historia de usuario generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la tabla.

Historia de usuario Añadir curso

Historia de Usuario	
Número: HU_04	Nombre de la historia: Añadir curso (carrito).
Modificación de historia de usuario:	
Usuario: estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como estudiante deseó añadir los cursos al carrito de adquisición para iniciar el curso.	
Observaciones: Los cursos seleccionados se mostraran en el perfil del estudiante.	

Historia de Usuario - Pruebas de Aceptación

- Mostrar los cursos seleccionados en el perfil del estudiante.

Realizado por: Jerson Carranza, William Morocho. 2017

Las siguientes tablas detallan las tareas de ingenierías que se generó en esta historia de usuario.

Elaboración de la interfaz de usuario de añadir los cursos.

Tarea de ingeniería	
Historia de Usuario: Añadir curso (carrito).	
Número Tarea: TI_01HU04	Nombre de la Tarea: Elaboración de la interfaz de usuario de añadir los cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 10/03/17	Fecha de Fin: 13/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para verificar el funcionamiento del carrito.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador Añadir curso

Tarea de ingeniería	
Historia de Usuario: Añadir curso (carrito).	
Número Tarea: TI_02HU04	Nombre de la Tarea: Elaboración del controlador para el funcionamiento del carrito.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 13/03/17	Fecha de Fin: 13/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para generar listado de cursos en el carrito.	
Pruebas de Aceptación: Comprobar que las vistas se generen el orden adecuado.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_05 Eliminar curso (Carrito): en esta historia de usuario permite al estudiante eliminar los cursos que ha seleccionado, para eliminar dichos cursos se debe verificar que aún no este

matriculado en los mismos; caso contrario se acercara directamente a la fundación. Como resultado de esta historia de usuario se generó 2 tareas de ingeniería y 3 pruebas de aceptación los mismos que se pueden observar en la tabla.

Eliminar Curso (carrito)

Historia de Usuario	
Número: HU_05	Nombre de la historia: Eliminar Curso (carrito).
Modificación de historia de usuario:	
Usuario: estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 8
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 8
Descripción: Como estudiante deseó eliminar los cursos del carrito para escoger solamente los cursos que deseo.	
Observaciones: Existirá el nombre del curso y el costo, el mismo donde se confirmara al momento de dar clic en el botón eliminar.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Mostrar los cursos seleccionados en el perfil del estudiante y que aparezca el botón eliminar.

Realizado por: Jerson Carranza, William Morocho. 2017

Tareas de ingenierías que se generó en la historia de usuario.

Primera Tarea de ingeniería de eliminar curso (carrito).

Tarea de ingeniería	
Historia de Usuario: Eliminar curso (carrito).	
Número Tarea: TI_01 HU05	Nombre de la Tarea: Elaboración de la interfaz de usuario de eliminar los cursos seleccionados.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4
Fecha de Inicio: 14/03/17	Fecha de Fin: 14/03/17

Programador Responsable: Jerson Carranza
Descripción: Elaboración de la interfaz necesaria para verificar la eliminación de los cursos.
Pruebas de Aceptación: La interfaz de eliminar cursos está correctamente diseñada bajo los estándares establecidos.

Realizado por: Jerson Carranza, William Morocho. 2017

Segunda Tarea de ingeniería de Eliminar curso (carrito).

Tarea de ingeniería	
Historia de Usuario: Eliminar curso (carrito).	
Número Tarea: TI_02 HU05	Nombre de la Tarea: Elaboración del controlador para eliminar los cursos del carrito.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4
Fecha de Inicio: 14/03/17	Fecha de Fin: 14/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para generar listado y eliminar los cursos no deseados.	
Pruebas de Aceptación: Comprobar que el curso se elimina.	

Realizado por: Jerson Carranza, William Morocho. 2017

Iteración 2

HU_06 Crear un administrador: en esta historia de usuario se desarrolló esta función que permite crear varios administradores, el administrador creado accederá a todas las funciones del sistema SIWEFI, para crear administradores se necesita llenar varios campos como el nombre, correo, contraseña. Como resultado de esta historia se generó 2 tareas de ingeniería y 3 pruebas de aceptación los mismos que se pueden observar en la tabla.

Historia de usuario crear un administrador.

Historia de Usuario	
Número: HU_06	Nombre de la historia: Crear un administrador.
Modificación de historia de usuario:	
Usuario: administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16

Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como administrador deseó crear un perfil de administrador para acceder a todas las fundiciones de la aplicación y delegar funciones a futuros administradores.	
Observaciones: Existirá los campos siguientes: el nombre, correo, contraseña, repita contraseña.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Al momento de crear un administrador genere el mensaje de confirmación.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario para crear un administrador.

Tarea de ingeniería	
Historia de Usuario: Crear un administrador.	
Número Tarea: TI_01HU06	Nombre de la Tarea: Elaboración de la interfaz de usuario para crear un administrador.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 15/03/17	Fecha de Fin: 15/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el registro del administrador.	
Pruebas de Aceptación: La interfaz de registro del administrador este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para crear cuenta al administrador.

Tarea de ingeniería	
Historia de Usuario: Crear un administrador.	
Número Tarea: TI_02 HU06	Nombre de la Tarea: Elaboración del controlador para crear cuenta al administrador.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 16/03/17	Fecha de Fin: 16/03/17

Programador Responsable: William Morocho
Descripción: Elaboración del controlador para crear la cuenta del administrador.
Pruebas de Aceptación: Comprobar que el administrador se ha creado correctamente.

Realizado por: Jerson Carranza, William Morocho. 2017

HU_07 Autenticarse: en esta historia de usuario se desarrolló el autenticar para que los administradores se autenticuen y puedan ingresar al sistema como súper usuario, por ende puedan acceder libremente a todas las funciones. Para autenticarse se necesita estar registrado e ingresar el correo y la contraseña. En este desarrollo se generó 2 tareas de ingeniería y 3 pruebas de aceptación los mismos que se pueden observar en la tabla.

Historia de usuario Autenticarse Administrador

Historia de Usuario	
Número: HU_07	Nombre de la historia: Autenticarse.
Modificación de historia de usuario:	
Usuario: administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 8
Riesgo en el Desarrollo: Alta	Puntos Reales: 8
Descripción: Como administrador deseó autenticarme para poder acceder a todas las funciones del sistema.	
Observaciones: El usuario de tipo administrador tendrá acceso a todas las funciones del sistema.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Si la autenticación es correcta se re direcciona a la página administrador.

Realizado por: Jerson Carranza, William Morocho. 2017

Tareas de ingenierías de la historia de autenticar estudiante, se detallan en las siguientes tablas.

Elaboración de la interfaz de usuario de la autenticación Administrador

Tarea de ingeniería
Historia de Usuario: Autenticarse.

Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario de la autenticación.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4
Fecha de Inicio: 17/03/17	Fecha de Fin: 17/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la autenticación de los administradores.	
Pruebas de Aceptación: La interfaz de la autenticación este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la autenticación del administrador.

Tarea de ingeniería	
Historia de Usuario: Autenticarse.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para la autenticación de los administradores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4
Fecha de Inicio: 17/03/17	Fecha de Fin: 17/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la autenticación de los administradores.	
Pruebas de Aceptación: Comprobar que la autenticación se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_08 CRUD Curso: esta historia de usuario se desarrolló para que el administrador pueda crear los diferentes cursos que se dictan en fundación Jaspe, también en esta etapa se creó funciones para eliminar los cursos que ya se dictaron, otra de las funciones añadidas fue el modificar la misma que permitirá editar luego de haber creado un curso con cualquier error gramatical u ortográfico y finalmente la función que se agregó fue el visualizar la misma que permitirá listar todos los cursos que se han creado por el administrador del sistema. En esta historia de usuario se generaron 8 tareas de ingeniería y 12 pruebas de aceptación las mismas que se detallan en tabla.

Historia de usuario CRUD Curso.

Historia de Usuario	
Número: HU_08	Nombre de la historia: CRUD Curso.
Modificación de historia de usuario:	
Usuario: administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como administrador deseó realizar el CRUD de los cursos para crear, modificar, eliminar y visualizar los cursos.	
Observaciones: El administrador podrá realizar todas las acciones del CRUD en la aplicación.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Crear curso se muestre el mensaje de confirmación. • Modificar curso muestre el mensaje de confirmación. • Eliminar curso muestre el respectivo mensaje. • Los cursos creados se visualicen en el orden respectivo.

Realizado por: Jerson Carranza, William Morocho. 2017

Tareas de ingeniería que se generaron para la creación de cursos, fueron dos la primera que está dedicada a la elaboración de la interfaz y la segunda que se encarga de la elaboración del controlador.

Elaboración de la interfaz de usuario de crear cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_01HU08	Nombre de la Tarea: Elaboración de la interfaz de usuario de crear cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 20/03/17	Fecha de Fin: 20/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la creación de cursos.	
Pruebas de Aceptación: La interfaz de la creación de cursos este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la creación de los cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_02 HU08	Nombre de la Tarea: Elaboración del controlador para la creación de los cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 20/03/17	Fecha de Fin: 20/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la creación de cursos.	
Pruebas de Aceptación: Comprobar que la creación de cursos se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Tareas de ingeniería que se generaron para la modificación de los cursos.

Elaboración de la interfaz de usuario de modificar cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_03	Nombre de la Tarea: Elaboración de la interfaz de usuario de modificar cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 20/03/17	Fecha de Fin: 20/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la modificación de cursos.	
Pruebas de Aceptación: La interfaz de la modificación de cursos este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la modificación de los cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_04	Nombre de la Tarea: Elaboración del controlador para la modificación de los cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 20/03/17	Fecha de Fin: 20/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la modificación de cursos.	
Pruebas de Aceptación: Comprobar que la modificación de cursos se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Se generaron dos tareas ingenierías para la funcionalidad de eliminar cursos

Elaboración de la interfaz de usuario de eliminar cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_05	Nombre de la Tarea: Elaboración de la interfaz de usuario de eliminar cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 21/03/17	Fecha de Fin: 21/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la eliminación de cursos.	
Pruebas de Aceptación: La interfaz de la eliminación de cursos este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la eliminación de los cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_06	Nombre de la Tarea: Elaboración del controlador para la eliminación de los cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 21/03/17	Fecha de Fin: 21/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la eliminación de cursos.	
Pruebas de Aceptación: Comprobar que la eliminación de cursos se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Se generaron dos tareas ingenierías para la funcionalidad de visualizar cursos

Elaboración de la interfaz de usuario para visualizar los cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_07	Nombre de la Tarea: Elaboración de la interfaz de usuario para visualizar los cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 21/03/17	Fecha de Fin: 21/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para visualizar los cursos.	
Pruebas de Aceptación: La interfaz para visualizar los cursos está correctamente diseñada bajo los	

estándares establecidos.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para listar cursos.

Tarea de ingeniería	
Historia de Usuario: CRUD Curso.	
Número Tarea: TI_08	Nombre de la Tarea: Elaboración del controlador para listar cursos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 21/03/17	Fecha de Fin: 21/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para listar cursos.	
Pruebas de Aceptación: Comprobar que listar cursos se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_09 CRUD Estudiante: En esta historia de usuario se desarrolló para que el administrador pueda ingresar al sistema a los diferentes estudiantes que todavía no han hecho la pre matrícula, cuando el administrador ingresa al sistema a un estudiante la contraseña es la misma cédula del estudiante, también en esta etapa se creó funciones para eliminar a los estudiantes que ya han cumplido su ciclo académico, otra de las funciones añadidas fue el modificar la misma que permite editar luego de haber creado a un estudiante cuando exista algún error en la información de registro y finalmente la función que se agregó fue el visualizar la misma que permite listar todos los estudiantes que se han creado por el administrador del sistema. En esta historia de usuario se generaron 8 tareas de ingeniería y 12 pruebas de aceptación las mismas que se detallan en la siguiente tabla.

Historia de usuario CRUD Estudiante.

Historia de Usuario	
Número: HU_09	Nombre de la historia: CRUD Estudiante.

Modificación de historia de usuario:	
Usuario: administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 16
Descripción: Como administrador deseó realizar el CRUD de los estudiantes para crear, modificar, eliminar y visualizar a todos los estudiantes.	
Observaciones: El administrador podrá realizar todas las acciones del CRUD en la aplicación.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Crear estudiante se muestre el mensaje de confirmación. • Modificar estudiante muestre el mensaje de confirmación. • Eliminar estudiante muestre el respectivo mensaje. • Los estudiantes creados se visualicen en el orden respectivo.

Realizado por: Jerson Carranza, William Morocho. 2017

Las tareas de ingeniería que se generaron para la creación de estudiantes son las siguientes.

Elaboración de la interfaz de usuario de crear estudiantes Administrador

Tarea de ingeniería	
Historia de Usuario: CRUD Estudiante.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario de crear estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 22/03/17	Fecha de Fin: 22/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la creación de estudiantes.	
Pruebas de Aceptación: La interfaz de la creación de estudiantes este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la creación de los estudiantes Administrador

Tarea de ingeniería	
Historia de Usuario: CRUD Estudiante.	

Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para la creación de los estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 22/03/17	Fecha de Fin: 22/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la creación de estudiantes.	
Pruebas de Aceptación: Comprobar que la creación de estudiantes se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Tareas de ingeniería que se generaron para la modificación de estudiantes

Elaboración de la interfaz de usuario de modificar estudiante.

Tarea de ingeniería	
Historia de Usuario: CRUD Estudiante.	
Número Tarea: TI_03	Nombre de la Tarea: Elaboración de la interfaz de usuario de modificar estudiante.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 22/03/17	Fecha de Fin: 22/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la modificación de estudiante.	
Pruebas de Aceptación: La interfaz de la modificación de estudiante este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la modificación de estudiantes.

Tarea de ingeniería

Historia de Usuario: CRUD Estudiante.	
Número Tarea: TI_04	Nombre de la Tarea: Elaboración del controlador para la modificación de estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 22/03/17	Fecha de Fin: 22/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la modificación de estudiantes.	
Pruebas de Aceptación: Comprobar que la modificación de estudiantes se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario de eliminar estudiantes.

Tarea de ingeniería	
Historia de Usuario: CRUD Estudiante.	
Número Tarea: TI_05	Nombre de la Tarea: Elaboración de la interfaz de usuario de eliminar estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 23/03/17	Fecha de Fin: 23/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la eliminación de estudiantes.	
Pruebas de Aceptación: La interfaz de la eliminación de estudiantes este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la eliminación de los estudiantes.

Tarea de ingeniería	
Historia de Usuario: CRUD estudiantes.	
Número Tarea: TI_06	Nombre de la Tarea: Elaboración del controlador para la eliminación de los estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 23/03/17	Fecha de Fin: 23/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la eliminación de estudiantes.	
Pruebas de Aceptación: Comprobar que la eliminación de estudiantes se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario para visualizar los estudiantes.

Tarea de ingeniería	
Historia de Usuario: CRUD Estudiante.	
Número Tarea: TI_07	Nombre de la Tarea: Elaboración de la interfaz de usuario para visualizar los estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 23/03/17	Fecha de Fin: 23/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para visualizar los estudiantes.	
Pruebas de Aceptación: La interfaz para visualizar estudiantes está correctamente diseñada bajo	

los estándares establecidos.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para listar estudiantes.

Tarea de ingeniería	
Historia de Usuario: CRUD Estudiante	
Número Tarea: TI_08	Nombre de la Tarea: Elaboración del controlador para listar estudiantes.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 23/03/17	Fecha de Fin: 23/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para listar estudiantes.	
Pruebas de Aceptación: Comprobar que liste los estudiantes que realice correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_10 CRUD Profesores: esta historia de usuario se desarrolló para que el administrador pueda ingresar al sistema a los diferentes profesores que serán parte de la planta de docentes de la fundación Jaspe, también en esta etapa se creó funciones para eliminar a los profesores que ya no están dentro de institución, otra de las funciones añadidas fue el modificar la misma que permitirá editar luego de haber creado a un profesor cuando exista algún error en la información de registro y finalmente la función que se agregó fue el visualizar la misma que permite listar todos los profesores que se han creado por el administrador del sistema. En esta historia de usuario se generaron 8 tareas de ingeniería y 12 pruebas de aceptación las mismas que se detallan en la siguiente tabla.

CRUD Profesores.

Historia de Usuario	
Número: HU_10	Nombre de la historia: CRUD Profesores.
Modificación de historia de usuario:	
Usuario: administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta	Puntos Reales: 16

Descripción: Como administrador deseó realizar el CRUD de los profesores para crear, modificar, eliminar y visualizar a todos los profesor.

Observaciones: El administrador podrá realizar todas las acciones del CRUD en la aplicación.

Historia de Usuario - Pruebas de Aceptación

- Crear profesor se muestre el mensaje de confirmación.
- Modificar profesor muestre el mensaje de confirmación.
- Eliminar profesor muestre el respectivo mensaje.
- Los profesores creados se visualicen en el orden respectivo.

Realizado por: Jerson Carranza, William Morocho. 2017

A continuación las tareas de ingenierías que se generó en esta historia de usuario

Elaboración de la interfaz de usuario de crear profesores.

Tarea de ingeniería	
Historia de Usuario: CRUD Profesor.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario de crear profesores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 24/03/17	Fecha de Fin: 24/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la creación de profesores.	
Pruebas de Aceptación: La interfaz de la creación de profesores este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la creación de los profesores.

Tarea de ingeniería	
Historia de Usuario: CRUD Profesor.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para la creación de los profesores.

Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 24/03/17	Fecha de Fin: 24/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la creación de profesores.	
Pruebas de Aceptación: Comprobar que la creación de profesores se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario de modificar profesor.

Tarea de ingeniería	
Historia de Usuario: CRUD Profesor.	
Número Tarea: TI_03	Nombre de la Tarea: Elaboración de la interfaz de usuario de modificar profesor.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 24/03/17	Fecha de Fin: 24/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la modificación de profesor.	
Pruebas de Aceptación: La interfaz de la modificación de profesor este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la modificación de profesores.

Tarea de ingeniería

Historia de Usuario: CRUD Profesor.	
Número Tarea: TI_04	Nombre de la Tarea: Elaboración del controlador para la modificación de profesores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 24/03/17	Fecha de Fin: 24/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la modificación de profesores.	
Pruebas de Aceptación: Comprobar que la modificación de profesores se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario de eliminar profesores.

Tarea de ingeniería	
Historia de Usuario: CRUD Profesor.	
Número Tarea: TI_05	Nombre de la Tarea: Elaboración de la interfaz de usuario de eliminar profesores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 27/03/17	Fecha de Fin: 27/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la eliminación de profesores.	
Pruebas de Aceptación: La interfaz de la eliminación de profesores este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la eliminación de los profesores.

Tarea de ingeniería	
Historia de Usuario: CRUD profesores.	
Número Tarea: TI_06	Nombre de la Tarea: Elaboración del controlador para la eliminación de los profesores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 27/03/17	Fecha de Fin: 27/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la eliminación de profesores.	
Pruebas de Aceptación: Comprobar que la eliminación de profesores se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario para visualizar los profesores.

Tarea de ingeniería	
Historia de Usuario: CRUD Profesor.	
Número Tarea: TI_07	Nombre de la Tarea: Elaboración de la interfaz de usuario para visualizar los profesores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 27/03/17	Fecha de Fin: 27/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para visualizar los profesores.	
Pruebas de Aceptación: La interfaz para visualizar profesores está correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para listar profesores.

Tarea de ingeniería	
Historia de Usuario: CRUD Profesor	
Número Tarea: TI_08	Nombre de la Tarea: Elaboración del controlador para listar profesores.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 27/03/17	Fecha de Fin: 27/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para listar profesores.	
Pruebas de Aceptación: Comprobar que listar profesores se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_11 CRUD Misión y Visión: esta historia de usuario se desarrolló para que el administrador pueda ingresar al sistema la misión y visión de la fundación Jaspe, los mismos que serán visualizados en la página principal del sistema, también en esta etapa se creó funciones para eliminar la visión y misión del sistema, otra de las funciones añadidas fue el modificar la misma que permitirá editar luego de haber creado la misión y visión cuando exista algún error ortográfico o gramatical y finalmente la función que se agrego fue el visualizar la misma que permitirá visualizar la misión y visión de la fundación en la página principal. En esta historia de usuario se generaron 8 tareas de ingeniería y 12 pruebas de aceptación las mismas que se detallan en la siguiente tabla.

CRUD Misión y Visión.

Historia de Usuario	
Número: HU_11	Nombre de la historia: CRUD Misión y Visión.
Modificación de historia de usuario:	
Usuario: administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta	Puntos Reales: 16
Descripción: Como administrador deseó realizar el CRUD de la Misión y Visión para crear, modificar, eliminar y visualizar a todos los profesores.	
Observaciones: El administrador podrá realizar todas las acciones del CRUD en la aplicación.	

Historia de Usuario - Pruebas de Aceptación

- Crear misión y visión se muestre el mensaje de confirmación.
- Modificar misión y visión muestre el mensaje de confirmación.
- Eliminar misión y visión muestre el respectivo mensaje.
- Los Misión y Visión creados se visualicen en el orden respectivo.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario de crear Misión y Visión.

Tarea de ingeniería	
Historia de Usuario: CRUD Misión y visión.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario de crear Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la creación de Misión y Visión.	
Pruebas de Aceptación: La interfaz de la creación de Misión y Visión este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la creación de los Misión y Visión.

Tarea de ingeniería	
Historia de Usuario: CRUD Misión y visión.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para la creación de los Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 1
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la creación de Misión y Visión.	

Pruebas de Aceptación: Comprobar que la creación de Misión y Visión se realiza correctamente con la base de datos.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario de modificar misión y visión.

Tarea de ingeniería	
Historia de Usuario: CRUD Misión y visión.	
Número Tarea: TI_03	Nombre de la Tarea: Elaboración de la interfaz de usuario de modificar misión y visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 1
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: Jerson CARRANZA	
Descripción: Elaboración de la interfaz necesaria para la modificación de misión y visión.	
Pruebas de Aceptación: La interfaz de la modificación de misión y visión este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la modificación de Misión y Visión.

Tarea de ingeniería	
Historia de Usuario: CRUD Misión y visión.	
Número Tarea: TI_04	Nombre de la Tarea: Elaboración del controlador para la modificación de Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 1

Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la modificación de Misión y Visión .	
Pruebas de Aceptación: Comprobar que la modificación de Misión y Visión se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario de eliminar Misión y Visión.

Tarea de ingeniería	
Historia de Usuario: CRUD Misión y visión.	
Número Tarea: TI_05	Nombre de la Tarea: Elaboración de la interfaz de usuario de eliminar Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 1
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la eliminación de Misión y Visión.	
Pruebas de Aceptación: La interfaz de la eliminación de Misión y Visión este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para la eliminación de los Misión y Visión.

Tarea de ingeniería

Historia de Usuario: CRUD Misión y Visión.	
Número Tarea: TI_06	Nombre de la Tarea: Elaboración del controlador para la eliminación de los Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 1
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la eliminación de Misión y Visión.	
Pruebas de Aceptación: Comprobar que la eliminación de Misión y Visión se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario para visualizar los Misión y Visión.

Tarea de ingeniería	
Historia de Usuario: CRUD Misión y visión.	
Número Tarea: TI_07	Nombre de la Tarea: Elaboración de la interfaz de usuario para visualizar los Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para visualizar los Misión y Visión.	
Pruebas de Aceptación: La interfaz para visualizar Misión y Visión está correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para listar Misión y Visión.

Tarea de ingeniería

Historia de Usuario: CRUD Misión y visión	
Número Tarea: TI_08	Nombre de la Tarea: Elaboración del controlador para listar Misión y Visión.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 2
Fecha de Inicio: 28/03/17	Fecha de Fin: 28/03/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para listar Misión y Visión.	
Pruebas de Aceptación: Comprobar que listar Misión y Visión se realiza correctamente con la base de datos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Iteración 3

HU_12 Cobro de matrículas: esta historia de usuario permite a la secretaria realizar el respectivo cobro de matrículas a todos los estudiantes que hicieron la pre matrícula, también la secretaria después de realizar esta acción cambiará los estados de los estudiantes a estudiantes legalmente matriculados. Esta historia de usuario generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la siguiente tabla.

Historia de usuario cobro de matrículas.

Historia de Usuario	
Número: HU_12	Nombre de la historia: Cobro de matrículas.
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 3
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 32
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 32
Descripción: Como secretaria deseó realizar el cobro de matrículas para poder registrar en el sistema.	
Observaciones: el estudiante debe haber hecho la pre matricula.	

Historia de Usuario - Pruebas de Aceptación

- Al finalizar el cobro de la matricula genere el mensaje de confirmación.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario.

Tarea de ingeniería	
Historia de Usuario: Cobro de matrículas.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 29/03/17	Fecha de Fin: 30/03/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el cobro de matrículas.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el cobro de matrícula

Tarea de ingeniería	
Historia de Usuario: Cobro de matrículas.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el cobro de matrículas.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 31/03/17	Fecha de Fin: 03/04/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el cobro de matrículas.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_13 Cobro de pensiones: esta historia de usuario permite a la secretaria realizar el respectivo cobro de pensiones cada mes a todos los estudiantes legalmente matriculados, también la secretaria podrá emitir el comprobante de pagos que corresponde a cada estudiante. Esta historia de usuario generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la siguiente tabla.

Historia de usuario Cobro de pensiones.

Historia de Usuario	
Número: HU_13	Nombre de la historia: Cobro de pensiones.
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 3
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 24
Descripción: Como secretaria deseó realizar el cobro de pensiones para poder registrar en el sistema.	
Observaciones: el estudiante debe estar matriculado.	

Historia de Usuario - Pruebas de Aceptación

- Al finalizar el cobro pensiones genere el mensaje de confirmación.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario para el cobro de pensiones.

Tarea de ingeniería	
Historia de Usuario: Cobro de pensiones.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario para el cobro de pensiones.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 04/04/17	Fecha de Fin: 04/04/17
Programador Responsable: Jerson Carranza	

Descripción: Elaboración de la interfaz necesaria para el cobro de pensiones.
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el cobro de pensiones.

Tarea de ingeniería	
Historia de Usuario: Cobro de pensiones.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el cobro de pensiones.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 05/04/17	Fecha de Fin: 06/04/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el cobro de pensiones.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_14 Venta de materiales de trabajo: en esta historia de usuario permite a la secretaria vender los productos que se comercializan en la fundación Jaspe para la utilización en los distintos cursos que esta institución imparte diariamente. Esta historia de usuario generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden visualizar en la tabla.

Ventas de materiales de trabajo.

Historia de Usuario	
Número: HU_14	Nombre de la historia: Ventas de materiales de trabajo.
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 3
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 24
Descripción: Como secretaria deseó realizar la venta de materiales de trabajo para poder registrar en el sistema.	

Observaciones: se venderá el material a todo tipo de persona.

Historia de Usuario - Pruebas de Aceptación

- Al finalizar el cobro por los materiales de trabajo genere el mensaje de confirmación.

Realizado por: Jerson Carranza, William Morocho. 2017

Iteración 4

HU_15 Estado Situación Inicial: en esta historia de usuario permite a la contadora realizar todo el estado situación inicial de la fundación es decir permitirá realizar la acción de contar, pesar, medir y valorar todos los bienes existentes en la fundación, también se puede ingresar todo lo que la empresa adeuda. Es un registro detallado de todos los bienes, valore o servicios tanto que posee o adeuda la empresa a una fecha dada. Esta historia de usuario generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden ver en la tabla.

Historia de usuario Estado situación inicial.

Historia de Usuario	
Número: HU_15	Nombre de la historia: Estado situación inicial
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 4
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 32
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 32
Descripción: Como contadora deseó realizar el estado de situación inicial para poder registrar el proceso contable de la fundación.	
Observaciones: las cuentas a utilizar deben estar registradas en la base de datos.	

Historia de Usuario - Pruebas de Aceptación

- Al finalizar el inventario inicial genere el mensaje de confirmación.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario Estado de situación inicial.

Tarea de ingeniería	
Historia de Usuario: Estado de situación inicial.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario

	estado de situación inicial.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 12/04/17	Fecha de Fin: 12/04/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el estado de situación inicial.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el Estado de situación inicial

Tarea de ingeniería	
Historia de Usuario: Estado de situación inicial.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el estado de situación inicial.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 24
Fecha de Inicio: 13/04/17	Fecha de Fin: 17/04/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el estado de situación inicial.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_16 Libro diario: en esta historia de usuario se desarrolló el libro diario, el mismo que permitirá que la contadora registre de forma metódica y cronológica todas las afectaciones económicas de la fundación “Jaspe” en forma de asientos contables. También permite visualizar el libro diario a la contadora y generar su respectivo informe. En esta historia de usuario se generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden visualizar en la siguiente tabla.

Historia de usuario libro diario

Historia de Usuario

Número: HU_16	Nombre de la historia: Libro diario.
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 4
Prioridad en el Negocio: Alta	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta	Puntos Reales: 24
Descripción: Como contadora deseó realizar el libro diario para poder registrar el proceso contable de la fundación.	
Observaciones: las cuentas a utilizar deben estar registradas en la base de datos.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Visualizar el libro diario con todas las cuentas respectivas

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario del libro diario.

Tarea de ingeniería	
Historia de Usuario: Libro diario.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 18/04/17	Fecha de Fin: 18/04/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el libro diario.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el libro diario

Tarea de ingeniería	
Historia de Usuario: Libro diario.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el libro diario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 19/04/17	Fecha de Fin: 20/04/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el libro diario.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_17 Libro Mayor: en esta historia de usuario se desarrolló el libro mayor, el que se encarga de recoger todas las operaciones económicas registradas en las distintas cuentas contables de la fundación de manera cronológica. Tenemos un libro mayor por cada cuenta contable que hemos utilizado en la contabilidad. En esta historia de usuario se generó 2 tareas de ingeniería.

Historia de Usuario: Libro mayor.

Historia de Usuario	
Número: HU_17	Nombre de la historia: Libro Mayor.
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 4
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 32
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 32
Descripción: Como contadora deseó ver las cuentas que están en el libro mayor que se encuentran implicada en la contabilidad.	
Observaciones: las cuentas a utilizar deben estar registradas en la base de datos.	

Historia de Usuario - Pruebas de Aceptación

- Visualizar el libro mayor y que me indique si el saldo es deudor o acreedor.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario del libro mayor.

Tarea de ingeniería	
Historia de Usuario: Libro mayor.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 21/04/17	Fecha de Fin: 21/04/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el libro mayor.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el libro mayor.

Tarea de ingeniería	
Historia de Usuario: Libro mayor	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el libro mayor.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 24
Fecha de Inicio: 22/04/17	Fecha de Fin: 26/04/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el libro mayor.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_18 Balance de comprobación: en esta historia de usuario se desarrolló el balance de comprobación el mismo que permitirá a la contadora realizar el fin del periodo contable, con esta acción se determinara si la fundación Jaspe ha generado utilidad o perdida, esta información se genera gracias al libro mayor general. En este desarrollo se generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la tabla.

Historia de usuario Balance de comprobación

Historia de Usuario	
Número: HU_18	Nombre de la historia: Balance de comprobación
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 4
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 32
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 32
Descripción: Como contadora deseó realizar el Balance de comprobación para poder registrar el proceso contable de la fundación.	
Observaciones: las cuentas a utilizar deben estar registradas en la base de datos.	

Historia de Usuario - Pruebas de Aceptación

- Visualizar el Balance de comprobación con todas las cuentas respectivas

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario balance de comprobación

Tarea de ingeniería	
Historia de Usuario: Balance de comprobación.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 27/04/17	Fecha de Fin: 27/04/17

Programador Responsable: Jerson Carranza
Descripción: Elaboración de la interfaz necesaria para el Balance de comprobación.
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.
Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para él Balance de comprobación.

Tarea de ingeniería	
Historia de Usuario: Balance de comprobación.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para él Balance de comprobación.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 24
Fecha de Inicio: 28/04/17	Fecha de Fin: 02/05/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el balance de comprobación.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

Iteración 5

HU_19 Pago a profesores: en esta historia de usuario se desarrolló el pago a profesores el mismo que permite generar un comprobante de pago y un registro automático en el libro diario de todos los pagos que se realicen a los profesores, también permite generar informes con su respectivo encabezado y pie de página de la fundación. El desarrollo de esta historia de usuario generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la tabla.

Historia de usuario Pago a profesores.

Historia de Usuario	
Número: HU_19	Nombre de la historia: Pago a profesores.
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 5
Prioridad en el Negocio: Alta	Puntos Estimados: 32

(Alta / Media / Baja)	
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 40
Descripción: Como contadora deseó realizar el pago a profesores para poder registrar en el sistema para tener los registros de las actividades de la fundación.	
Observaciones: El profesor debe estar registrado en el sistema.	

Historia de Usuario - Pruebas de Aceptación

- Al realizar el pago a profesores generar el comprobante de pago.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario del pago de profesores

Tarea de ingeniería	
Historia de Usuario: Pago a profesores.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos Reales: 8
Fecha de Inicio: 03/05/17	Fecha de Fin: 03/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el pago de profesores.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el pago a profesores.

Tarea de ingeniería	
Historia de Usuario: Pago a profesores.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el pago a profesores.
Tipo de Tarea: Desarrollo	Puntos Reales: 32
Fecha de Inicio: 04/05/17	Fecha de Fin: 09/05/17
Programador Responsable: William Morocho	

Descripción: Elaboración del controlador para el pago a profesores.
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.

Realizado por: Jerson Carranza, William Morocho. 2017

HU_20 Ver roles de Pagos: en esta historia de usuario se puede observar el pago a los profesores que trabaja dentro de la fundación Jaspe, En el desarrollo de esta historia de usuario se generó 2 tareas de ingeniería.

Historia de usuario Ver roles de Pagos

Historia de Usuario	
Número: HU_20	Nombre de la historia: Ver roles de Pagos
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 5
Prioridad en el Negocio: Alta	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta	Puntos Reales: 32
Descripción: Como contadora deseó observar los roles de pagos detallado por cada mes de cada uno de los profesores de la fundación	
Observaciones: El personal administrativo debe estar registrado en el sistema.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Al observar los roles de pagos, que se despliegue en una ventana modal

Realizado por: Jerson Carranza, William Morocho. 2017

Interfaz de usuario Ver roles de Pagos

Tarea de ingeniería

Historia de Usuario: Ver roles de Pagos	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos Reales: 8
Fecha de Inicio: 10/05/17	Fecha de Fin: 11/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para ver roles de Pagos	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para Ver los roles de Pagos

Tarea de ingeniería	
Historia de Usuario: Ver roles de Pagos	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para ver roles de Pagos
Tipo de Tarea: Desarrollo	Puntos Reales: 16
Fecha de Inicio: 11/05/17	Fecha de Fin: 12/05/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para observar roles de Pagos	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_21 Pago de arriendo: en esta historia de usuario se desarrolló el registro del pago de arriendo de las aulas que utiliza la institución para impartir clases, esta funcionalidad permite llevar los pagos. En este desarrollo se generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se observan en tabla.

Historia de usuario Pago de arriendo.

Historia de Usuario	
Número: HU_21	Nombre de la historia: Pago de arriendo.
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 5
Prioridad en el Negocio: Alta	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta	Puntos Reales: 20
Descripción: Como contadora deseó realizar el Pago de arriendo para poder registrar en el sistema para tener los registros de las actividades de la fundación.	
Observaciones: El personal administrativo debe estar registrado en el sistema.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Al realizar el Pago de arriendo generar el comprobante de pago.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz de usuario Pago de arriendo

Tarea de ingeniería	
Historia de Usuario: Pago de arriendo.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 13/05/17	Fecha de Fin: 14/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el Pago de arriendo.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración del controlador para el Pago de arriendo.

Tarea de ingeniería	
Historia de Usuario: Pago de arriendo.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el Pago de arriendo.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 15/05/17	Fecha de Fin: 16/05/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el Pago de arriendo.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_22 Compra de materiales: en esta historia de usuario se desarrolló la compra de materiales para determinar cuánto se está gastando en la compra de los mismos y llevar el registro respectivo de cada compra. En el desarrollo de esta historia se generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la tabla.

Historia de usuario compra de materiales.

Historia de Usuario	
Número: HU_22	Nombre de la historia: Compra de materiales.
Modificación de historia de usuario:	
Usuario: contadora	Iteración Asignada: 5
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 16
Riesgo en el Desarrollo: Alta	Puntos Reales: 20
Descripción: Como contadora deseó realizar la compra de materiales para poder registrar en el sistema.	
Observaciones: después de la compra los materiales deberán registrarse en el sistema.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> Al registrar la compra de materiales generar mensaje de confirmación.

Realizado por: Jerson Carranza, William Morocho. 2017

Elaboración de la interfaz para la compra de materiales

Tarea de ingeniería	
Historia de Usuario: Compra de materiales.	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 17/05/17	Fecha de Fin: 17/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para la Compra de materiales.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para la compra de materiales.

Tarea de ingeniería	
Historia de Usuario: Compra de materiales.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para la Compra de materiales.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 18/05/17	Fecha de Fin: 19/05/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para la Compra de materiales.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

Iteración 6

HU_23 Flujo de caja: en esta historia de usuario se desarrolló el flujo de caja el mismo que permite observar las variaciones de entrada y salida de caja en un periodo determinado, también se lo conoce como flujo neto, es importante porque nos indica la liquidez de la empresa. En el desarrollo de esta historia se generó 2 tareas de ingeniería y 3 pruebas de aceptación las mismas que se pueden observar en la tabla.

Historia de usuario flujo de caja

Historia de Usuario	
Número: HU_23	Nombre de la historia: Flujo de caja.
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 6
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta (Alto / Medio / Bajo)	Puntos Reales: 24
Descripción: Como secretaria deseó realizar el flujo de caja para observar cuanto tengo en caja.	
Observaciones: el flujo de caja permite visualizar los ingresos y egresos de cada mes.	

Historia de Usuario - Pruebas de Aceptación

- Que se visualice correctamente los datos del flujo de caja.

Realizado por: Jerson Carranza, William Morocho. 2017

Interfaz de usuario flujo de caja

Tarea de ingeniería	
Historia de Usuario: Flujo de caja	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 8
Fecha de Inicio: 22/05/17	Fecha de Fin: 23/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el flujo de caja.	
Pruebas de Aceptación: La interfaz este correctamente diseñada bajo los estándares establecidos.	

Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para el flujo de caja

Tarea de ingeniería	
Historia de Usuario: Flujo de caja.	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para el flujo de caja
Tipo de Tarea: Desarrollo	Puntos de Estimados: 16
Fecha de Inicio: 24/05/17	Fecha de Fin: 25/05/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para el flujo de caja.	
Pruebas de Aceptación: Comprobar que los datos estén almacenados correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_24 Gráficos Estudiantes: En esta historia de usuario se desarrolló gráficos que permite observar cuantos estudiantes existe en cada curso. En el desarrollo de esta historia se generó 2 tareas de ingeniería.

Historia de usuario gráfico estudiantes

Historia de Usuario	
Número: HU_24	Nombre de la historia: Gráficos de estudiantes
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 6
Prioridad en el Negocio: Alta	Puntos Estimados: 32
Riesgo en el Desarrollo: Alta	Puntos Reales: 16
Descripción: Como secretaria deseó observar el número estudiantes existe en los cursos	
Observaciones: Se debe tener matriculados en distintos cursos	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Que se visualice correctamente cada uno de los gráficos generados.

Realizado por: Jerson Carranza, William Morocho. 2017

Tarea de ingeniería

Tarea de ingeniería	
Historia de Usuario: Gráficos de estudiantes	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4
Fecha de Inicio: 26/05/17	Fecha de Fin: 26/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el reporte de los gráficos.	
Pruebas de Aceptación: Observar gráficos en pastel y barra.	

Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para los gráficos respectivos.

Tarea de ingeniería	
Historia de Usuario: Gráficos de estudiantes	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para los gráficos respectivos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 12
Fecha de Inicio: 26/05/17	Fecha de Fin: 29/05/17

Programador Responsable: William Morocho
Descripción: Elaboración del controlador para los gráficos.
Pruebas de Aceptación: Comprobar que los datos estén correctamente.

Realizado por: Jerson Carranza, William Morocho. 2017

HU_25 Gráficos Cursos: En esta historia de usuario se desarrolló gráficos que permite observar cuánto dinero generó cada curso. En el desarrollo de esta historia se generó 2 tareas de ingeniería.

Historia de usuario gráfico estudiantes

Historia de Usuario	
Número: HU_25	Nombre de la historia: Gráficos de cursos
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 6
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta	Puntos Reales: 16
Descripción: Como secretaria deseó observar que curso genero más ingresos económicos	
Observaciones: Se debe tener matriculados y además que exista pago de pensiones.	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Que se visualice correctamente cada uno de los gráficos generados.

Realizado por: Jerson Carranza, William Morocho. 2017

Tarea de ingeniería

Tarea de ingeniería	
Historia de Usuario: Gráficos de cursos	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4

Fecha de Inicio: 30/05/17	Fecha de Fin: 30/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el reporte de los gráficos.	
Pruebas de Aceptación: Observar gráficos en pastel y barra.	

Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para los gráficos respectivos.

Tarea de ingeniería	
Historia de Usuario: Gráficos de cursos	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para los gráficos respectivos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 12
Fecha de Inicio: 30/05/17	Fecha de Fin: 31/05/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para los gráficos.	
Pruebas de Aceptación: Comprobar que los datos estén correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

HU_26 Gráficos Sueldos: En esta historia de usuario se desarrolló un gráfico que permite observar cuánto dinero se gastó en cada por el concepto de pagos a profesores. En el desarrollo de esta historia se generó 2 tareas de ingeniería.

Historia de usuario gráfico estudiantes.

Historia de Usuario	
Número: HU_26	Nombre de la historia: Gráficos de sueldos
Modificación de historia de usuario:	
Usuario: secretaria	Iteración Asignada: 6
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta	Puntos Reales: 16
Descripción: Como secretaria deseó observar cuanto se gasto en sueldos por concepto a profesores.	
Observaciones: Se debe tener pagado los sueldos a los profesores para que se genere correctamente los gráficos	

Historia de Usuario - Pruebas de Aceptación
<ul style="list-style-type: none"> • Que se visualice correctamente cada uno de los gráficos generados.

Realizado por: Jerson Carranza, William Morocho. 2017

Tarea de ingeniería

Tarea de ingeniería	
Historia de Usuario: Gráficos de sueldo	
Número Tarea: TI_01	Nombre de la Tarea: Elaboración de la interfaz de usuario.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 4
Fecha de Inicio: 01/05/17	Fecha de Fin: 01/05/17
Programador Responsable: Jerson Carranza	
Descripción: Elaboración de la interfaz necesaria para el reporte de los gráficos.	
Pruebas de Aceptación: Observar gráficos en pastel y barra.	

Realizado por: Jerson Carranza, William Morocho. 2017

Controlador para los gráficos respectivos.

Tarea de ingeniería	
Historia de Usuario: Gráficos de sueldo	
Número Tarea: TI_02	Nombre de la Tarea: Elaboración del controlador para los gráficos respectivos.
Tipo de Tarea: Desarrollo	Puntos de Estimados: 12
Fecha de Inicio: 02/05/17	Fecha de Fin: 05/06/17
Programador Responsable: William Morocho	
Descripción: Elaboración del controlador para los gráficos.	
Pruebas de Aceptación: Comprobar que los datos estén correctamente.	

Realizado por: Jerson Carranza, William Morocho. 2017

ANEXO E: Planificación del sistema.

Nombre de tarea	Duración	Comienzo	Fin	Predecesora
▸ Fase exploración	5 días	mié 01/02/17	mié 08/02/17	
Recopilación Información	1 día	mié 01/02/17	mié 01/02/17	
Diagnosticar situación fundación	1 día	jue 02/02/17	jue 02/02/17	2
Gestión de riesgos	3 días	vie 03/02/17	mar 07/02/17	3
Reuniones	0 días	mié 08/02/17	mié 08/02/17	4
▸ Fase de planificación	5 días	mié 08/02/17	mié 15/02/17	
Historias de usuarios	2 días	mié 08/02/17	jue 09/02/17	5
release planning	1 día	vie 10/02/17	vie 10/02/17	7
Estimación	1 día	lun 13/02/17	lun 13/02/17	8
Iteraciones	1 día	mar 14/02/17	mar 14/02/17	9
Asignar pareja	0 días	mié 15/02/17	mié 15/02/17	10
▸ Fase de Diseño	10 días	mié 15/02/17	mar 28/02/17	
Arquitectura del sistema	2 días	mié 15/02/17	jue 16/02/17	11
Estándar de codificación	1 día	vie 17/02/17	vie 17/02/17	13
Diseño de la base de datos	5 días	lun 20/02/17	vie 24/02/17	14
Diseño de interfaces de usuario	2 días	lun 27/02/17	mar 28/02/17	15
▸ Fase de codificación	68 días?	mié 01/03/17	vie 02/06/17	
▸ Iteración 1	10 días	mié 01/03/17	mar 14/03/17	
▸ Módulo Estudiante	10 días	mié 01/03/17	mar 14/03/17	
H1 Registro	3 días	mié 01/03/17	vie 03/03/17	16
H2 Autenticación	2 días	lun 06/03/17	mar 07/03/17	20
H3 Vistas cursos	2 días	mié 08/03/17	jue 09/03/17	21
H4 Añadir Curso (Carrito)	2 días	vie 10/03/17	lun 13/03/17	22
H5 Eliminar Curso(Carrito)	1 día	mar 14/03/17	mar 14/03/17	23
▸ Iteración 2	10 días	mié 15/03/17	mar 28/03/17	
▸ Módulo Administrador	10 días	mié 15/03/17	mar 28/03/17	
H6 Crear un administrador	2 días	mié 15/03/17	jue 16/03/17	24
H7 Autenticarse	1 día	vie 17/03/17	vie 17/03/17	27
H8 CRUD Curso	2 días	lun 20/03/17	mar 21/03/17	28
H9 CRUD Estudiante	2 días	mié 22/03/17	jue 23/03/17	29
H10 CRUD Profesores	2 días	vie 24/03/17	lun 27/03/17	30
H11 CRUD Misión y Visión	1 día	mar 28/03/17	mar 28/03/17	31

Nombre de tarea	Duración	Comienzo	Fin	Predecesora
Iteración 3	10 días	mié 29/03/17	mar 11/04/17	
Módulo de cobranza	10 días	mié 29/03/17	mar 11/04/17	
H12 Cobro de matrículas	4 días	mié 29/03/17	lun 03/04/17	32
H13 Cobro de pensiones	3 días	mar 04/04/17	jue 06/04/17	35
H14 Ventas de materiales de trabajo	3 días	vie 07/04/17	mar 11/04/17	36
Iteración 4				
Módulo contable				
H15 Estado de situación inicial	4 días	mié 12/04/17	lun 17/04/17	37
H16 Libro Diario	3 días	mar 18/04/17	jue 20/04/17	40
H17 Libro Mayor	4 días	vie 21/04/17	mié 26/04/17	41
H18 Balance de comprobación	4 días	jue 27/04/17	mar 02/05/17	42
Iteración 5				
Módulo de costos				
H19 Pago a profesores	4 días	mié 03/05/17	lun 08/05/17	43
H20 Ver Roles Pagos	2 días	mar 09/05/17	mié 10/05/17	46
H21 Pago de Arriendo	2 días	jue 11/05/17	vie 12/05/17	47
H22 Compra de materiales	2 días	lun 15/05/17	mar 16/05/17	48
Iteración 6				
Módulo Reportes				
H23 Flujo de caja	3 días	mié 17/05/17	vie 19/05/17	49
H24 Gráficos Estudiantes	4 días	lun 22/05/17	jue 25/05/17	52
H25 Gráficos Cursos	3 días	vie 26/05/17	mar 30/05/17	53
H26 Gráficos Sueldos	3 días	mié 31/05/17	vie 02/06/17	54
Fase de Pruebas	1 día	lun 05/06/17	lun 05/06/17	
Realización de pruebas	1 día	lun 05/06/17	lun 05/06/17	55
Fase de Finalización	1 día	mar 06/06/17	mié 07/06/17	
Capacitación del usuario	1 día	mar 06/06/17	mar 06/06/17	57
Presentación Documentación	0 días	mié 07/06/17	mié 07/06/17	59

ANEXO F: PRUEBAS DE ACEPTACIÓN

Prueba de Aceptación	
Código: P2MS1	Historia de Usuario: Diseño de la Arquitectura del sistema.
Nombre: Existencia de los servidores para la aplicación web.	
Responsable: Jerson Carranza	Fecha: 16/02/17
Descripción: Verificar la existencia de los servidores a utilizar.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Verificar el funcionamiento del servidor.	
Pasos de ejecución: <ul style="list-style-type: none">• Elegir los servidores que estén acorde a las necesidades de la aplicación.• Verificar que tenga los requisitos necesarios como memoria RAM, espacio.	
Resultado esperado: Funcione correctamente la aplicación	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Pruebas de aceptación del estándar de codificación

Prueba de Aceptación	
Código: P1MS2	Historia de Usuario: Estándar de codificación.
Nombre: Existencia del estándar de codificación	
Responsable: Jerson Carranza	Fecha: 17/02/17
Descripción: Existencia de información verídica del estándar de codificación.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Existencia del estándar de codificación.	
Pasos de ejecución: <ul style="list-style-type: none">• Investigación del estándar de codificación.• Recopilación de información acerca del estándar de codificación.	
Resultado esperado: Que la información de la investigación sea entendible para el equipo	

del trabajo.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de aceptación del diseño de la base de datos.

Prueba de Aceptación	
Código: P1MS2	Historia de Usuario: Diseño de la base de datos.
Nombre: Que la base de datos sea embebida.	
Responsable: Jerson Carranza	Fecha: 24/02/17
Descripción: la base de datos noSQL debe estar desarrollado en mongoDB.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Tener instalado mongoDB. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Diseño físico de la base de datos. Creacion de los documento. 	
Resultado esperado: que la base de datos este creada en mongoDB.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: P1MS4	Historia de Usuario: Diseño de las interfaces de usuario.
Nombre: Verificar que el diseño de la interfaz de usuario este acorde con el estándar establecido.	
Responsable: William MOROCHO	Fecha: 27/02/17
Descripción: los usuarios del sistema web ingresan a cada una de las pantallas.	
Condiciones de Ejecución:	
Tener conexión a la red.	
Pasos de ejecución:	
Verificar la información del modelo de la interfaz gráfica.	
Verificar si el diseño está acorde al modelo de interfaz gráfica.	
Resultado esperado: Que los modelos sean acorde a lo requerido por la fundación.	

Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de aceptación 2 diseño interfaces

Prueba de Aceptación	
Código: P2MS4	Historia de Usuario: Diseño de interfaces de usuario.
Nombre: Aceptación de la interfaz por los directivos de la fundación.	
Responsable: Jerson Carranza	Fecha: 28/02/17
Descripción: Aceptación del diseño de interfaz por parte de la fundación de acuerdo a los requerimientos presentados.	
Condiciones de Ejecución: Que la interfaces de usuario estén creadas.	
Pasos de ejecución: Visualización de las interfaces de usuario. Verificación por parte de los usuarios de la fundación cada una de las interfaces.	
Resultado esperado: Aceptación del diseño de las interfaces por parte de los usuarios de la fundación.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Pruebas de aceptación del registro de estudiantes.

Prueba de Aceptación	
Código: PA_01H01	Historia de Usuario: Registro de estudiantes.
Nombre: Si todos los campos no están llenos el botón registro se bloquea.	
Responsable: William Morocho	Fecha: 03/03/17
Descripción: Se requiere verificar que no haya registro, si todos los campos no están llenos.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> Existencia de la interfaz del registro de los estudiantes.
Pasos de ejecución: <ul style="list-style-type: none"> El usuario no llena los datos en los campos establecidos.
Resultado esperado: el botón registro se bloquea y no permite registrarse.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H01	Historia de Usuario: Registro de estudiantes.
Nombre: Si el número de cédula introducido es incorrecto emitirá el respectivo mensaje de error.	
Responsable: William MOROCHO	Fecha: 03/03/17
Descripción: Se requiere verificar que al ingresar una cédula incorrecta emita el mensaje de error.	
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz del registro de los estudiantes. Existencia del campo donde ingresara la cédula. 	
Pasos de ejecución: <ul style="list-style-type: none"> Ingresa una cédula incorrecta. Click en registrarse. 	
Resultado esperado: mensaje mencionando que la cédula ingresada es incorrecta.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H01	Historia de Usuario: Registro de estudiantes.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William MOROCHO	Fecha: 01/03/17
Descripción: Que las preguntas estén estandarizadas de acuerdo a las interfaz.	
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz del registro de asistencia. 	
Pasos de ejecución: <ul style="list-style-type: none"> El usuario ingresa a la interfaz de registro Verificar la interfaz de registro 	
Resultado esperado: Presentación de la interfaz de registro de los estudiantes.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación

Código: PA_04H01	Historia de Usuario: Registro de estudiantes.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 03/03/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página registro de estudiantes. • Llenar todos los campos que se requieran. • Dar clic en el botón registrarse • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Pruebas de aceptación de la autenticación de estudiantes

Prueba de Aceptación	
Código: PA_01H02	Historia de Usuario: Autenticación.
Nombre: Si la autenticación es correcta se re direcciona a la página de opciones.	
Responsable: William Morocho	Fecha: 07/03/17
Descripción: Se requiere verificar que la autenticación de estudiantes se re direcciona adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de la autenticación. • Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos establecidos. • Clic en ingresar. 	
Resultado esperado: mensaje de autenticación satisfactorio y redirección a la página de opciones.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H02	Historia de Usuario: Autenticación.
Nombre: La interfaz de la autenticación este correctamente diseñada bajo los estándares	

establecidos.	
Responsable: William Morocho	Fecha: 06/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la página de autenticación de estudiante. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de autenticación. Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de autenticación de los estudiantes.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H02	Historia de Usuario: Autenticación.
Nombre: Comprobar que la autenticación se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 07/03/17
Descripción: verificar si la autenticación se está realizando correctamente y re direccionando a las paginas adecuadas.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la página autenticación. Llenar todos los campos que se requieran. Dar clic en el botón autenticar. Verificar si está comparando con la base de datos. 	
Resultado esperado: mensaje de autenticación correcta.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H03	Historia de Usuario: Vistas cursos.
Nombre: Las vistas de los cursos estén con la información adecuada.	
Responsable: William Morocho	Fecha: 09/03/17
Descripción: Se requiere verificar las vistas de los cursos estén con la información necesaria.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de la vista de los cursos. 	

Pasos de ejecución:
<ul style="list-style-type: none"> El estudiante observa en la página de las vistas y verifica si existe información.
Resultado esperado: Las vistas contendrán información.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H03	Historia de Usuario: Vistas cursos.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 08/03/17
Descripción: Las imágenes y el texto estén estandarizadas de acuerdo con la interfaz.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de vista de los cursos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de vista de los cursos Verificar la interfaz de vistas del curso. 	
Resultado esperado: Presentación de la interfaz de vistas de los cursos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H03	Historia de Usuario: Vistas de cursos.
Nombre: Comprobar que las vistas se generen el orden adecuado.	
Responsable: Jerson Carranza	Fecha: 09/03/17
Descripción: los cursos que en la fundación se imparten se mostrarán al estudiante.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la página vistas de cursos. Verificar la información y las imágenes generadas. 	
Resultado esperado: Las imágenes y la información de muestran en el orden correcto.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H04	Historia de Usuario: Añadir curso (carrito)..
Nombre: Mostrar los cursos seleccionados en el perfil del estudiante.	
Responsable: William Morocho	Fecha: 13/03/17

Descripción: Se requiere verificar los cursos seleccionados.
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz del carrito de los estudiantes.
Pasos de ejecución: <ul style="list-style-type: none"> El estudiante observa en la página del perfil del estudiante.
Resultado esperado: El listado de cursos inscritos.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H04	Historia de Usuario: Añadir curso (carrito).
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 10/03/17
Descripción: Las imágenes y el texto estén estandarizadas de acuerdo con la interfaz.	
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz de vista de los cursos. 	
Pasos de ejecución: <ul style="list-style-type: none"> El usuario ingresa a la interfaz del carrito Verificar la interfaz de vistas del curso. 	
Resultado esperado: Presentación de la interfaz del carrito	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H04	Historia de Usuario: Añadir curso (carrito).
Nombre: Comprobar que las vistas se generen el orden adecuado.	
Responsable: Jerson CARRANZA	Fecha: 09/03/17
Descripción: los cursos que en la fundación se imparten se mostrarán al estudiante.	
Condiciones de Ejecución: <ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución: <ul style="list-style-type: none"> Ingresar al perfil del estudiante. Verificar el listado de cursos registrados. 	
Resultado esperado: Listado de los cursos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación

Código: PA_01H05	Historia de Usuario: Eliminar curso (carrito).
Nombre: Mostrar los cursos seleccionados en el perfil del estudiante y que aparezca el botón eliminar.	
Responsable: William MOROCHO	Fecha: 14/03/17
Descripción: Se requiere eliminar los cursos que se seleccionó y no deseo.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz del carrito de los estudiantes. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El estudiante observa en la página del perfil del estudiante. El estudiante clic en el botón eliminar. 	
Resultado esperado: Eliminar el curso seleccionado.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H05	Historia de Usuario: Eliminar curso (carrito).
Nombre: La interfaz de eliminar cursos está correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 14/03/17
Descripción: Los cursos que se listaron se podrán eliminar.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz donde se encuentra la lista de los cursos y el botón eliminar. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El estudiante ingresara al perfil del mismo. Verificara la lista generada. Verifica que exista el botón eliminar. 	
Resultado esperado: Aparece la interfaz eliminar curso con su respetivo botón en cada curso.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H05	Historia de Usuario: Eliminar curso (carrito).
Nombre: Comprobar que el curso se elimina.	
Responsable: Jerson Carranza	Fecha: 14/03/17
Descripción: los cursos que en la fundación se imparten se mostrarán al estudiante y se podrán eliminar lo que no se desea.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar al perfil del estudiante. Verificar el listado de cursos registrados. Clic en el botón eliminar. 	
Resultado esperado: El curso se elimina.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H06	Historia de Usuario: Crear un administrador.
Nombre: Al momento de crear un administrador genere el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 16/03/17
Descripción: Se requiere crear un administrador.	
Condiciones de Ejecución: <ul style="list-style-type: none">Existencia de la interfaz para crear administradores.	
Pasos de ejecución: <ul style="list-style-type: none">El administrador ingresa a la interfaz de creación.Llena todos los campos necesarios para la creación.Clic en el botón registrarse.	
Resultado esperado: mensaje de confirmación que el administrador se creó correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H06	Historia de Usuario: Crear un administrador.
Nombre: La interfaz de registro del administrador este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 15/03/17
Descripción: La interfaz de usuario que el administrador utilizara tiene que estar bajo los estándares establecidos.	
Condiciones de Ejecución: <ul style="list-style-type: none">Existencia de la interfaz de registro del administrador.	
Pasos de ejecución: <ul style="list-style-type: none">El administrador ingresara a la interfaz de registro.Verificara la interfaz de registro.	
Resultado esperado: La interfaz está bajo los estándares establecidos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H06	Historia de Usuario: Crear un administrador.
Nombre: Comprobar que el administrador se ha creado correctamente.	
Responsable: Jerson Carranza	Fecha: 16/03/17
Descripción: La cuenta del administrador se cree de forma correcta.	
Condiciones de Ejecución: <ul style="list-style-type: none">Debe existir el controller creado.Existencia del modelo.	

Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz de registro de administrador. • Llenar todos los campos necesarios. • Clic en el botón registrar. • Verificar registro en la base de datos. 	
Resultado esperado: datos guardados correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H07	Historia de Usuario: Autenticarse.
Nombre: Si la autenticación es correcta se re direcciona a la página administrador.	
Responsable: William Morocho	Fecha: 17/03/17
Descripción: Se requiere verificar que la autenticación del administrador se re dirccione adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de autenticación. • Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos establecidos. • Click en iniciar sesión. 	
Resultado esperado: mensaje de autenticación satisfactorio y redirección a la página de administración.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_02H07	Historia de Usuario: Autenticarse.
Nombre: La interfaz de la autenticación este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 17/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página autenticarse para el administrador. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de autenticación. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de autenticación para el administrador.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H07	Historia de Usuario: Autenticarse.
Nombre: Comprobar que la autenticación se realiza correctamente con la base de datos.	

Responsable: Jerson Carranza	Fecha: 17/03/17
Descripción: verificar si la autenticación se está realizando correctamente y re direccionando a las paginas adecuadas.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página autenticación. • Llenar todos los campos que se requieran. • Dar clic en el botón autenticar. • Verificar si está comparando con la base de datos. 	
Resultado esperado: mensaje de autenticación correcta.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H08	Historia de Usuario: CRUD Curso.
Nombre: Crear curso se muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se requiere verificar que al momento de crear un curso se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de creación de curso. • Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos establecidos. • Clic en guardar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_02H08	Historia de Usuario: CRUD Curso.
Nombre: Modificar curso muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se requiere verificar que al momento de modificar un curso se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de modificación de curso. • Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos establecidos. • Clic en guardar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H08	Historia de Usuario: CRUD Curso.
Nombre: Eliminar curso muestre el respectivo mensaje.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se requiere verificar que al momento de eliminar un curso se muestre el mensaje de confirmación.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Existencia de la interfaz de eliminación de curso. • Existencia del botón eliminar 	
Pasos de ejecución: <ul style="list-style-type: none"> • Clic en botón eliminar. • Sale el mensaje de confirmación. • Clic en el botón ok. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_04H08	Historia de Usuario: CRUD Curso.
Nombre: Los cursos creados se visualicen en el orden respectivo.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se requiere verificar que visualice todos los cursos que se han creado.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Existencia de la interfaz de visualización de cursos. • Existencia de cursos. 	
Pasos de ejecución: <ul style="list-style-type: none"> • El usuario ingresa a la interfaz de visualización de cursos. • Verificar la interfaz de registro 	
Resultado esperado: Presentación de la interfaz de visualización de cursos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_05H08	Historia de Usuario: CRUD Curso.
Nombre: La interfaz de la creación de cursos este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 20/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Existencia de la página crear cursos. 	

Pasos de ejecución:
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de creación de cursos. • Verificar los colores y la posición de los botones.
Resultado esperado: Presentación de la interfaz de creación de los cursos.
Evaluación de la prueba: Satisfactorio.
Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_06H08	Historia de Usuario: CRUD Curso.
Nombre: Comprobar que la creación de cursos se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 20/03/17
Descripción: verificar si la creación de cursos se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz creación cursos. • Llenar todos los campos que se requieran. • Clic en guardar. • Verificar si está guardando en la base de datos. 	
Resultado esperado: mensaje se guardó correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_07H08	Historia de Usuario: CRUD Curso.
Nombre: La interfaz de modificación de cursos este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 20/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página modificar cursos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de modificación de cursos. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de modificación de los cursos.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_08H08	Historia de Usuario: CRUD Curso.
Nombre: Comprobar que la modificación de cursos se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 20/03/17
Descripción: verificar si la modificación de cursos se está realizando correctamente.	

Condiciones de Ejecución:
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo.
Pasos de ejecución:
<ul style="list-style-type: none"> • Ingresar a la interfaz modificación de cursos. • Llenar todos los campos que se requieran. • Clic en guardar. • Verificar si está modificando en la base de datos.
Resultado esperado: mensaje se actualizo correctamente.
Evaluación de la prueba: Satisfactorio.
Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_09H08	Historia de Usuario: CRUD Curso.
Nombre: La interfaz de la eliminación de cursos este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página eliminar cursos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de eliminación de cursos. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de eliminación de los cursos.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_10H08	Historia de Usuario: CRUD Curso.
Nombre: Comprobar que la eliminación de cursos se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 21/03/17
Descripción: verificar si la eliminación de cursos se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz eliminación de cursos. • Clic en eliminar. • Verificar si está eliminando de la base de datos. 	
Resultado esperado: mensaje se eliminó correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_11H08	Historia de Usuario: CRUD Curso.
Nombre: La interfaz para visualizar los cursos está correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página listar cursos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de listar cursos. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz listar cursos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_12H08	Historia de Usuario: CRUD Curso.
Nombre: Comprobar que listar cursos se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 21/03/17
Descripción: verificar si listar cursos se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz listar cursos. • Verificar si está visualizando todos los cursos existentes en la base de datos. 	
Resultado esperado: listado de cursos existentes en la base de datos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H09	Historia de Usuario: CRUD Estudiante.
Nombre: Crear estudiante se muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 23/03/17
Descripción: Se requiere verificar que al momento de crear un estudiante se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de creación de estudiante. • Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos establecidos. • Clic en guardar. 	

Resultado esperado: mensaje de confirmación.
Evaluación de la prueba: Satisfactorio.
Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H09	Historia de Usuario: CRUD Estudiante.
Nombre: Modificar estudiante muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 23/03/17
Descripción: Se requiere verificar que al momento de modificar un estudiante se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de modificación de estudiante. Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Llenar los campos establecidos. Clic en guardar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H09	Historia de Usuario: CRUD Estudiante.
Nombre: Eliminar estudiante muestre el respectivo mensaje.	
Responsable: William Morocho	Fecha: 23/03/17
Descripción: Se requiere verificar que al momento de eliminar un estudiante se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de eliminación de estudiantes. Existencia del botón eliminar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Clic en botón eliminar. Sale el mensaje de confirmación. Clic en el botón ok. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_04H09	Historia de Usuario: CRUD Estudiante.
Nombre: Los estudiantes creados se visualicen en el orden respectivo.	
Responsable: William Morocho	Fecha: 23/03/17
Descripción: Se requiere verificar que visualice todos los estudiantes que se han creado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de visualización de estudiantes. Existencia de estudiantes. 	

Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de visualización de estudiantes. • Verificar la lista de los estudiantes 	
Resultado esperado: Presentación de la interfaz de visualización de estudiantes.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_05H09	Historia de Usuario: CRUD Estudiante.
Nombre: La interfaz de la creación de estudiantes este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 22/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página crear estudiantes. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de creación de estudiante. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de creación de estudiantes.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_06H09	Historia de Usuario: CRUD Estudiante.
Nombre: Comprobar que la creación de estudiantes se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 22/03/17
Descripción: verificar si la creación de estudiantes se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz creación estudiantes. • Llenar todos los campos que se requieran. • Clic en guardar. • Verificar si está guardando en la base de datos. 	
Resultado esperado: mensaje se guardó correctamente.	

Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_07H09	Historia de Usuario: CRUD Estudiante.
Nombre: La interfaz de modificación de estudiante este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 22/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Existencia de la página modificar estudiante.	
Pasos de ejecución: <ul style="list-style-type: none">• El usuario ingresa a la interfaz de modificación de estudiante.• Verificar los colores y la posición de los botones.	
Resultado esperado: Presentación de la interfaz de modificación de estudiante.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_08H09	Historia de Usuario: CRUD Estudiante.
Nombre: Comprobar que la modificación de estudiante se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 22/03/17
Descripción: verificar si la modificación de estudiante se está realizando correctamente.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Debe existir el controller creado.• Existencia del modelo.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar a la interfaz modificación de estudiante.• Llenar todos los campos que se requieran.	

<ul style="list-style-type: none"> • Clic en guardar. • Verificar si está modificando en la base de datos.
Resultado esperado: mensaje se actualizo correctamente.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_09H09	Historia de Usuario: CRUD Estudiante.
Nombre: La interfaz de la eliminación de estudiante este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 21/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página eliminar estudiantes. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de eliminación de estudiantes. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de eliminación de los estudiantes.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_10H09	Historia de Usuario: CRUD Estudiante.
Nombre: Comprobar que la eliminación de estudiantes se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 21/03/17
Descripción: verificar si la eliminación de estudiantes se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz eliminación de estudiantes. • Clic en eliminar. • Verificar si está eliminando de la base de datos. 	

Resultado esperado: mensaje se eliminó correctamente.
Evaluación de la prueba: Satisfactorio.
Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_11H09	Historia de Usuario: CRUD Estudiante.
Nombre: La interfaz para visualizar los estudiantes está correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 23/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la página listar estudiantes. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de listar estudiantes. Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz listar estudiantes.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_12H09	Historia de Usuario: CRUD Estudiante.
Nombre: Comprobar que listar estudiante se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 23/03/17
Descripción: verificar si listar estudiante se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la interfaz listar estudiante. Verificar si está visualizando todos los estudiantes existentes en la base de datos. 	
Resultado esperado: listado de estudiantes existentes en la base de datos.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H10	Historia de Usuario: CRUD Profesor.
Nombre: Crear profesor se muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 27/03/17
Descripción: Se requiere verificar que al momento de crear un profesor se muestre el	

mensaje de confirmación.
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz de creación de profesor. Existencia de campos.
Pasos de ejecución: <ul style="list-style-type: none"> Llenar los campos establecidos. Clic en guardar.
Resultado esperado: mensaje de confirmación.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H10	Historia de Usuario: CRUD Profesor.
Nombre: Modificar profesor muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 27/03/17
Descripción: Se requiere verificar que al momento de modificar un profesor se muestre el mensaje de confirmación.	
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz de modificación de profesor. Existencia de campos. 	
Pasos de ejecución: <ul style="list-style-type: none"> Llenar los campos establecidos. Clic en guardar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H10	Historia de Usuario: CRUD Profesor.

Nombre: Eliminar profesor muestre el respectivo mensaje.	
Responsable: William Morocho	Fecha: 27/03/17
Descripción: Se requiere verificar que al momento de eliminar un profesor se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de eliminación de profesores. • Existencia del botón eliminar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Clic en botón eliminar. • Sale el mensaje de confirmación. • Clic en el botón ok. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_04H10	Historia de Usuario: CRUD Profesor.
Nombre: Los profesores creados se visualicen en el orden respectivo.	
Responsable: William Morocho	Fecha: 27/03/17
Descripción: Se requiere verificar que visualice todos los profesores que se han creado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de visualización de profesores. • Existencia de profesores. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de visualización de profesores. • Verificar la lista de los profesores 	
Resultado esperado: Presentación de la interfaz de visualización de profesores.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_05H10	Historia de Usuario: CRUD Profesores.
Nombre: La interfaz de la creación de profesores este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 24/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página crear profesores. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de creación de profesor. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de creación de profesores.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_06H10	Historia de Usuario: CRUD Profesor.
Nombre: Comprobar que la creación de profesores se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 24/03/17
Descripción: verificar si la creación de profesores se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz creación profesores. • Llenar todos los campos que se requieran. • Clic en guardar. • Verificar si está guardando en la base de datos. 	
Resultado esperado: mensaje se guardó correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_07H10	Historia de Usuario: CRUD Profesor.
Nombre: La interfaz de modificación de profesor este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 24/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página modificar profesor. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de modificación de profesor. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de modificación de profesor.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_08H10	Historia de Usuario: CRUD Profesor.
Nombre: Comprobar que la modificación de profesor se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 24/03/17
Descripción: verificar si la modificación de profesor se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz modificación de profesor. • Llenar todos los campos que se requieran. • Clic en guardar. • Verificar si está modificando en la base de datos. 	

Resultado esperado: mensaje se actualizo correctamente.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_09H10	Historia de Usuario: CRUD Profesor.
Nombre: La interfaz de la eliminación de profesor este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 27/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página eliminar profesores. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de eliminación de profesores. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de eliminación de los profesores.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_10H10	Historia de Usuario: CRUD Profesor.
Nombre: Comprobar que la eliminación de profesores se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 27/03/17
Descripción: verificar si la eliminación de profesores se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz eliminación de profesores. • Clic en eliminar. • Verificar si está eliminando de la base de datos. 	
Resultado esperado: mensaje se eliminó correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_11H10	Historia de Usuario: CRUD Profesor.
Nombre: La interfaz para visualizar los profesores está correctamente diseñada bajo los estándares establecidos.	

Responsable: William Morocho	Fecha: 27/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la página listar profesores. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de listar profesores. Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz listar profesores.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_12H10	Historia de Usuario: CRUD Profesores.
Nombre: Comprobar que listar profesor se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 27/03/17
Descripción: verificar si listar profesor se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la interfaz listar profesor. Verificar si está visualizando todos los profesores existentes en la base de datos. 	
Resultado esperado: listado de profesores existentes en la base de datos.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Crear misión y visión se muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se requiere verificar que al momento de crear misión y visión se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de creación de misión y visión. Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Llenar los campos establecidos. Clic en guardar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación

Código: PA_02H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Modificar misión y visión muestre el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se requiere verificar que al momento de modificar un misión y visión se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de modificación de misión y visión. • Existencia de campos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos establecidos. • Clic en guardar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Eliminar misión y visión muestre el respectivo mensaje.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se requiere verificar que al momento de eliminar misión y visión se muestre el mensaje de confirmación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de eliminación de Misión y Visión. • Existencia del botón eliminar 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Clic en botón eliminar. • Sale el mensaje de confirmación. • Clic en el botón ok. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_04H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Los Misión y Visión creados se visualicen en el orden respectivo.	
Responsable: William MOROCHO	Fecha: 28/03/17
Descripción: Se requiere verificar que visualice todos los Misión y Visión que se han creado.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de visualización de Misión y Visión. • Existencia de Misión y Visión. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de visualización de Misión y Visión. • Verificar la lista de los Misión y Visión 	
Resultado esperado: Presentación de la interfaz de visualización de Misión y Visión.	

Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_05H11	Historia de Usuario: CRUD Misión y visión.
Nombre: La interfaz de la creación de Misión y Visión este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Existencia de la página crear Misión y Visión.	
Pasos de ejecución: <ul style="list-style-type: none">• El usuario ingresa a la interfaz de creación de misión y visión.• Verificar los colores y la posición de los botones.	
Resultado esperado: Presentación de la interfaz de creación de Misión y Visión.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_06H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Comprobar que la creación de Misión y Visión se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 28/03/17
Descripción: verificar si la creación de Misión y Visión se está realizando correctamente.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Debe existir el controller creado.• Existencia del modelo.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar a la interfaz creación Misión y Visión.• Llenar todos los campos que se requieran.• Clic en guardar.• Verificar si está guardando en la base de datos.	
Resultado esperado: mensaje se guardó correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_07H11	Historia de Usuario: CRUD Misión y visión.
Nombre: La interfaz de modificación de misión y visión este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la página modificar misión y visión. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de modificación de misión y visión. • Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz de modificación de misión y visión.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_08H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Comprobar que la modificación de misión y visión se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 28/03/17
Descripción: verificar si la modificación de misión y visión se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la interfaz modificación de misión y visión. • Llenar todos los campos que se requieran. • Clic en guardar. • Verificar si está modificando en la base de datos. 	
Resultado esperado: mensaje se actualizo correctamente.	

Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_09H11	Historia de Usuario: CRUD Misión y visión.
Nombre: La interfaz de la eliminación de misión y visión este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Existencia de la página eliminar misión y visionés.	
Pasos de ejecución: <ul style="list-style-type: none">• El usuario ingresa a la interfaz de eliminación de Misión y Visión.• Verificar los colores y la posición de los botones.	
Resultado esperado: Presentación de la interfaz de eliminación de los Misión y Visión.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_10H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Comprobar que la eliminación de Misión y Visión se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 28/03/17
Descripción: verificar si la eliminación de Misión y Visión se está realizando correctamente.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Debe existir el controller creado.• Existencia del modelo.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar a la interfaz eliminación de Misión y Visión.• Clic en eliminar.• Verificar si está eliminando de la base de datos.	
Resultado esperado: mensaje se eliminó correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_11H11	Historia de Usuario: CRUD Misión y visión.
Nombre: La interfaz para visualizar los Misión y Visión está correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 28/03/17
Descripción: Se debe comprobar que la interfaz de usuario este de acuerdo al estándar establecido.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la página listar Misión y Visión. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de listar Misión y Visión. Verificar los colores y la posición de los botones. 	
Resultado esperado: Presentación de la interfaz listar Misión y Visión.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_12H11	Historia de Usuario: CRUD Misión y visión.
Nombre: Comprobar que listar misión y visión se realiza correctamente con la base de datos.	
Responsable: Jerson Carranza	Fecha: 28/03/17
Descripción: verificar si listar misión y visión se está realizando correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la interfaz listar misión y visión. Verificar si está visualizando todos los Misión y Visión existentes en la base de datos. 	
Resultado esperado: listado de Misión y Visión existentes en la base de datos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H12	Historia de Usuario: Cobro de matrículas.
Nombre: Al finalizar el cobro de la matricula genere el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 03/04/17
Descripción: Se requiere verificar que exista registro, cuando se haga la matricula.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de cobro de matrículas. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Primero los estudiantes deberán hacer la pre matricula. Se dirige a generar el listado de los prematriculados. 	

<ul style="list-style-type: none"> • Se matricula a todos los estudiantes. • Clic en el botón matricula.
Resultado esperado: mensaje de confirmación.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H12	Historia de Usuario: Cobro de matrículas.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 30/03/17
Descripción: el listado de estudiantes estén estandarizadas de acuerdo a las interfaz.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de cobro de matrículas. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de cobro de matrículas. • Verificar la interfaz de cobro de matrículas. 	
Resultado esperado: Presentación de la interfaz de cobro de matrículas.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H12	Historia de Usuario: Cobro de matrículas.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson CARRANZA	Fecha: 03/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página cobro de matrículas. • Dar clic en el botón matricular. • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H13	Historia de Usuario: Cobro de matrículas.
Nombre: Al finalizar el cobro pensiones genere el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 06/04/17
Descripción: Se requiere verificar que exista registro, cuando se cobre las pensiones.	

Condiciones de Ejecución:
<ul style="list-style-type: none"> Existencia de la interfaz de cobro de pensiones.
Pasos de ejecución:
<ul style="list-style-type: none"> Los estudiantes deben estar matriculados. Se dirige a generar el listado de los matriculados. Clic en el botón cobro de pensión.
Resultado esperado: mensaje de confirmación.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H13	Historia de Usuario: Cobro de pensiones.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 04/04/17
Descripción: el listado de estudiantes esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de cobro de pensiones. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de cobro de pensiones. Verificar la interfaz de cobro de pensiones. 	
Resultado esperado: Presentación de la interfaz de cobro de pensiones.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H13	Historia de Usuario: Cobro de pensiones.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 06/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresa a la página cobro de pensiones. Dar clic en el botón cobrar. Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H14	Historia de Usuario: Ventas de materiales de trabajo.
Nombre: Al finalizar el cobro por los materiales de trabajo genere el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 11/04/17
Descripción: Se requiere verificar que exista registro, cuando se venda los materiales de trabajo.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de venta de materiales de trabajo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Se dirige a generar el listado de los materiales. Clic en el botón venta. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H14	Historia de Usuario: Ventas de materiales de trabajo.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 07/04/17
Descripción: el listado de materiales esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de venta de materiales de trabajo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de venta de materiales de trabajo. Verificar la interfaz de venta de materiales de trabajo. 	
Resultado esperado: Presentación de la interfaz de venta de materiales de trabajo.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H14	Historia de Usuario: Ventas de materiales de trabajo.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 11/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la página de venta de materiales de trabajo. Dar clic en el botón cobrar. Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	

Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H15	Historia de Usuario: Inventario Inicial.
Nombre: Al finalizar el inventario inicial genere el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 17/04/17
Descripción: Se requiere verificar que exista registro, cuando se realice el inventario inicial.	
Condiciones de Ejecución: <ul style="list-style-type: none">Existencia de la interfaz de inventario inicial.	
Pasos de ejecución: <ul style="list-style-type: none">Llenar los campos respectivos con las cuentas contablesClic en el botón registrar.	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H15	Historia de Usuario: Inventario Inicial.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 12/04/17
Descripción: el inventario inicial esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución: <ul style="list-style-type: none">Existencia de la interfaz de inventario inicial.	
Pasos de ejecución: <ul style="list-style-type: none">El usuario ingresa a la interfaz de inventario inicial.Verificar la interfaz de venta de materiales de trabajo.	
Resultado esperado: Presentación de la interfaz de inventario inicial.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H15	Historia de Usuario: Inventario Inicial.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 17/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución: <ul style="list-style-type: none">Debe existir el controller creado.Existencia del modelo.	

Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página de inventario inicial. • Dar clic en el botón registrar. • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_02H16	Historia de Usuario: Estado de situación inicial.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 118/04/17
Descripción: el estado de situación inicial esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz del estado de situación inicial. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz del estado de situación inicial. • Verificar la interfaz del estado de situación inicial. 	
Resultado esperado: Presentación de la interfaz del estado de situación inicial.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H16	Historia de Usuario: Estado de situación inicial.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 21/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página del estado de situación inicial. • Dar clic en el botón registrar. • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H16	Historia de Usuario: Estado de situación inicial.
Nombre: Al finalizar el estado de situación inicial genere el mensaje de confirmación.	
Responsable: William Morocho	Fecha: 21/04/17
Descripción: Se requiere verificar que exista registro, cuando se realice el estado de situación inicial.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de estado de situación inicial. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Llenar los campos respectivos con las cuentas contables Clic en el botón registrar. 	
Resultado esperado: mensaje de confirmación.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H17	Historia de Usuario: Libro diario.
Nombre: Visualizar el libro diario con todas las cuentas respectivas.	
Responsable: William Morocho	Fecha: 26/04/17
Descripción: Se requiere verificar que las cuentas generadas en el libro diario sean las correctas.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de libro diario. Realizar alguna transacción contable 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Generar libro diario. 	
Resultado esperado: se visualiza en libro diario con sus cuantas respectivas.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_02H17	Historia de Usuario: libro diario.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 24/04/17
Descripción: el libro diario esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz del libro diario. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario ingresa a la interfaz del libro diario. Verificar la interfaz del libro diario. 	
Resultado esperado: Presentación de la interfaz del libro diario.	

Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H17	Historia de Usuario: Libro diario
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 26/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Debe existir el controller creado.• Existencia del modelo.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar a la página del libro diario.• Verificar en la base de datos que la información fue almacenada.	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H18	Historia de Usuario: Balance de comprobación.
Nombre: Visualizar el Balance de comprobación con todas las cuentas respectivas.	
Responsable: William Morocho	Fecha: 02/05/17
Descripción: Se requiere verificar que las cuentas generadas en el balance de comprobación sean las correctas.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Existencia de la interfaz del Balance de comprobación.• Realizar alguna transacción contable	
Pasos de ejecución: <ul style="list-style-type: none">• Generar libro balance de comprobación.	
Resultado esperado: se visualiza en balance de comprobación con sus cuentas respectivas.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H18	Historia de Usuario: Balance de comprobación.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 27/04/17
Descripción: Balance de comprobación esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Existencia de la interfaz del Balance de comprobación.	

Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz del Balance de comprobación. • Verificar la interfaz del Balance de comprobación. 	
Resultado esperado: Presentación de la interfaz del Balance de comprobación	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H18	Historia de Usuario: Balance de comprobación.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 02/05/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página del Balance de comprobación. • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H19	Historia de Usuario: Pago a profesores.
Nombre: Al realizar el pago a profesores generar el comprobante de pago.	
Responsable: William Morocho	Fecha: 08/05/17
Descripción: Se requiere verificar que exista registro, cuando se realice el pago a profesores.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de pago a profesores. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos respectivos en la interfaz. • Clic en el botón pagar. 	
Resultado esperado: genera el comprobante de pago.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_02H19	Historia de Usuario: Pago a profesores.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 03/05/17
Descripción: el pago a profesores esté estandarizadas de acuerdo a la interfaz establecida.	

Condiciones de Ejecución:
<ul style="list-style-type: none"> Existencia de la interfaz de pago a profesores.
Pasos de ejecución:
<ul style="list-style-type: none"> El usuario ingresa a la interfaz de pago a profesores. Verificar la interfaz de pago a profesores.
Resultado esperado: Presentación de la interfaz de pago a profesores.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H19	Historia de Usuario: Pago a profesores.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 21/04/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Debe existir el controller creado. Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Ingresar a la página de pago a profesores. Dar clic en el botón pagar Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H20	Historia de Usuario: Pago a personal administrativo.
Nombre: Al realizar el Pago a personal administrativo generar el comprobante de pago.	
Responsable: William Morocho	Fecha: 10/05/17
Descripción: Se requiere verificar que exista registro, cuando se realice el Pago a personal administrativo.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> Existencia de la interfaz de Pago a personal administrativo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> Llenar los campos respectivos en la interfaz. Clic en el botón pagar. 	
Resultado esperado: genera el comprobante de pago.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H20	Historia de Usuario: Pago a personal administrativo.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 09/05/17
Descripción: el Pago a personal administrativo esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de Pago a personal administrativo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de Pago a personal administrativo. • Verificar la interfaz de Pago a personal administrativo. 	
Resultado esperado: Presentación de la interfaz de Pago a personal administrativo.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_03H20	Historia de Usuario: Pago a personal administrativo.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 10/05/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página de Pago a personal administrativo. • Dar clic en el botón pagar • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	
Realizado por: Jerson Carranza, William Morocho. 2017	

Prueba de Aceptación	
Código: PA_01H21	Historia de Usuario: Pago de arriendo.
Nombre: Al realizar el Pago de arriendo generar el comprobante de pago.	
Responsable: William Morocho	Fecha: 12/05/17
Descripción: Se requiere verificar que exista registro, cuando se realice el Pago de arriendo.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> Existencia de la interfaz de Pago de arriendo.
Pasos de ejecución: <ul style="list-style-type: none"> Llenar los campos respectivos en la interfaz. Clic en el botón pagar.
Resultado esperado: genera el comprobante de pago.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H21	Historia de Usuario: Pago de arriendo.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 11/05/17
Descripción: el Pago de arriendo esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución: <ul style="list-style-type: none"> Existencia de la interfaz de Pago de arriendo. 	
Pasos de ejecución: <ul style="list-style-type: none"> El usuario ingresa a la interfaz de Pago de arriendo. Verificar la interfaz de Pago de arriendo. 	
Resultado esperado: Presentación de la interfaz de Pago de arriendo.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H20	Historia de Usuario: Pago de arriendo.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 12/05/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	

Condiciones de Ejecución:
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo.
Pasos de ejecución:
<ul style="list-style-type: none"> • Ingresar a la página de Pago de arriendo. • Dar clic en el botón pagar • Verificar en la base de datos que la información fue almacenada.
Resultado esperado: datos almacenados correctamente.
Evaluación de la prueba: Satisfactorio.

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H22	Historia de Usuario: Compra de materiales.
Nombre: Al realizar la Compra de materiales generar mensaje de comprobación.	
Responsable: William MOROCHO	Fecha: 12/05/17
Descripción: Se requiere verificar que exista registro, cuando se realice la compra de materiales.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de Compra de materiales.. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Llenar los campos respectivos en la interfaz. • Clic en el botón registrar. 	
Resultado esperado: genera el mensaje de confirmación	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H22	Historia de Usuario: Compra de materiales.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 15/05/17
Descripción: la Compra de materiales esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de Compra de materiales. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de Compra de materiales. • Verificar la interfaz de Compra de materiales. 	
Resultado esperado: Presentación de la interfaz de Compra de materiales.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación

Código: PA_03H22	Historia de Usuario: Compra de materiales.
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 16/05/17
Descripción: con los datos que se llenaron en todos los campos verificar si los mismos se guardaron adecuadamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página de Compra de materiales. • Dar clic en el botón comprar • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos almacenados correctamente.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H23	Historia de Usuario: Flujo de caja
Nombre: Que se visualice correctamente los datos del flujo de caja.	
Responsable: William MOROCHO	Fecha: 19/05/17
Descripción: Se requiere verificar que los datos que se visualicen en el flujo de caja sean los correctos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de flujo de caja. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Clic en el botón generar flujo de caja. 	
Resultado esperado: se visualiza el flujo de caja con los datos correctos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H23	Historia de Usuario: Flujo de caja.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 17/05/17
Descripción: el flujo de caja esté estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de flujo de caja. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de flujo de caja. • Verificar la interfaz de flujo de caja. 	
Resultado esperado: Presentación de la interfaz de flujo de caja.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H23	Historia de Usuario: flujo de caja
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 19/05/17
Descripción: con los datos generados debes ser los adecuados.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página de flujo de caja. • Dar clic en el botón generar • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos correctamente visualizados.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H24	Historia de Usuario: Ingresos mensuales
Nombre: Que se visualice correctamente los Ingresos mensuales.	
Responsable: William Morocho	Fecha: 23/05/17
Descripción: Se requiere verificar que los datos que se visualicen en los ingresos mensuales sean los correctos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de ingresos mensuales. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Clic en el botón generar ingresos mensuales. 	
Resultado esperado: se visualiza los ingresos mensuales con los datos correctos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H24	Historia de Usuario: Ingresos mensuales.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 17/05/17
Descripción: los ingresos mensuales estén estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de Ingresos mensuales 	

<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario ingresa a la interfaz de Ingresos mensuales. • Verificar la interfaz de Ingresos mensuales.
<p>Resultado esperado: Presentación de la interfaz de Ingresos mensuales.</p>
<p>Evaluación de la prueba: Satisfactorio.</p>

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H24	Historia de Usuario: Ingresos mensuales
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 23/05/17
Descripción: con los datos generados debes ser los adecuados.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la página de Ingresos mensuales • Dar clic en el botón generar • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos correctamente visualizados.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H25	Historia de Usuario: Egresos mensuales
Nombre: Que se visualice correctamente los Egresos mensuales.	
Responsable: William MOROCHO	Fecha: 25/05/17
Descripción: Se requiere verificar que los datos que se visualicen en los Egresos mensuales sean los correctos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de Egresos mensuales. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Clic en el botón generar Egresos mensuales. 	
Resultado esperado: se visualiza los Egresos mensuales con los datos correctos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H25	Historia de Usuario: Egresos mensuales.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William MOROCHO	Fecha: 24/05/17
Descripción: los Egresos mensuales estén estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz de Egresos mensuales 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario ingresa a la interfaz de Egresos mensuales. • Verificar la interfaz de Egresos mensuales. 	
Resultado esperado: Presentación de la interfaz de Egresos mensuales.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H25	Historia de Usuario: Egresos mensuales
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 23/05/17
Descripción: con los datos generados debes ser los adecuados.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Debe existir el controller creado. • Existencia del modelo. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Egresar a la página de Egresos mensuales • Dar clic en el botón generar • Verificar en la base de datos que la información fue almacenada. 	
Resultado esperado: datos correctamente visualizados.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_01H26	Historia de Usuario: Gráficos.
Nombre: Que se visualice correctamente cada uno de los gráficos generados.	
Responsable: William Morocho	Fecha: 02/06/17
Descripción: Se requiere verificar que los datos que se visualicen en cada uno de los gráficos sean los correctos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Existencia de la interfaz para los gráficos. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Clic en el botón generar gráficos. 	
Resultado esperado: se visualiza los gráficos con los datos correctos.	

Evaluación de la prueba: Satisfactorio.
--

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_02H26	Historia de Usuario: Gráficos.
Nombre: La interfaz este correctamente diseñada bajo los estándares establecidos.	
Responsable: William Morocho	Fecha: 26/05/17
Descripción: los gráficos estén estandarizadas de acuerdo a la interfaz establecida.	
Condiciones de Ejecución: <ul style="list-style-type: none">Existencia de la interfaz de gráficos	
Pasos de ejecución: <ul style="list-style-type: none">El usuario ingresa a la interfaz de gráficos.Verificar la interfaz de gráficos.	
Resultado esperado: Presentación de la interfaz de los gráficos.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

Prueba de Aceptación	
Código: PA_03H26	Historia de Usuario: Gráficos
Nombre: Comprobar que los datos estén almacenados correctamente.	
Responsable: Jerson Carranza	Fecha: 02/06/17
Descripción: con los datos generados debes ser los adecuados.	
Condiciones de Ejecución: <ul style="list-style-type: none">Debe existir el controller creado.Existencia del modelo.	
Pasos de ejecución: <ul style="list-style-type: none">Egresar a la página de gráficosDar clic en el botón generarVerificar en la base de datos que la información fue almacenada.	
Resultado esperado: datos correctamente visualizados.	
Evaluación de la prueba: Satisfactorio.	

Realizado por: Jerson Carranza, William Morocho. 2017

ANEXO G: Encuesta



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA ESCUELA DE INGENIERÍA EN SISTEMAS

Objetivo: Obtener información sobre la funcionalidad del sistema; Encuesta aplicada personal administrativos.

1. ¿El sistema te permite encontrar rápidamente la información?
 - Totalmente de acuerdo
 - De acuerdo
 - Indiferente
 - En desacuerdo
 - Totalmente en desacuerdo

2. ¿El sistema ayuda a mejorar tus tiempos en los procesos de la empresa?
 - Totalmente de acuerdo

- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

3. ¿Los módulos del sistema son suficientes para cumplir con sus procesos administrativos? Adecuación

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

4. ¿Estás satisfecho con las funcionalidades del sistema? Adecuación

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

5. ¿Los resultados que proporciona el sistema son los esperados? Indicador Exactitud

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

6. ¿El lenguaje utilizado en el sistema es claro y conciso? Indicador Exactitud

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

7. ¿La información solo puede acceder solo el administrador? Seguridad

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

8. ¿Puede asignar permisos a los usuarios el sistema? Seguridad

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

9. ¿El sistema “SIWEFI” puede interactuar con otros sistemas? Indicador Interoperabilidad.

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

Encuesta dirigida a los estudiantes



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

Objetivo: Obtener información sobre la funcionalidad del sistema; Encuesta aplicada a los estudiantes de la fundación “JASPE”.

1. ¿Los módulos del sistema son suficientes para cumplir con sus procesos administrativos?
Adecuación

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

2. ¿Estás satisfecho con las funcionalidades del sistema? Adecuación

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

3. ¿Los resultados que proporciona el sistema son los esperados? Indicador Exactitud

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

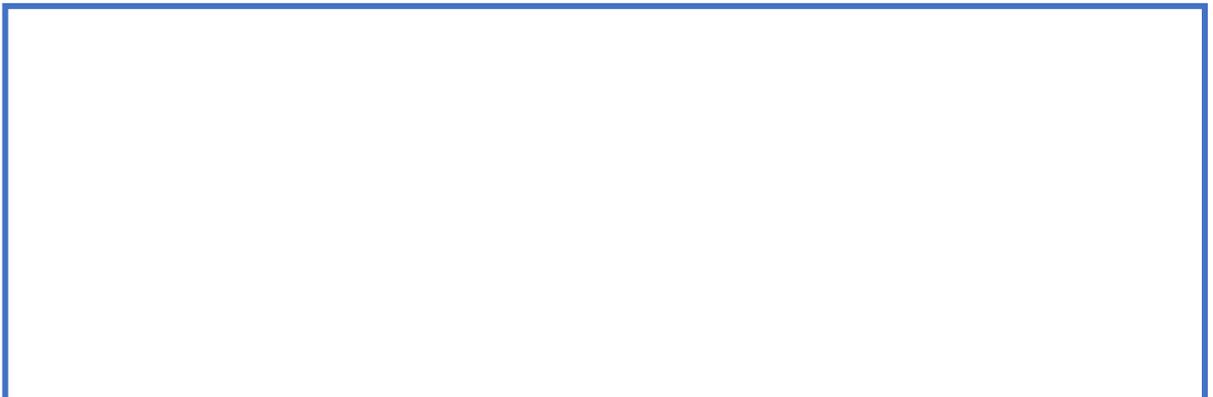
4. ¿El lenguaje utilizado en el sistema es claro y conciso? Indicador Exactitud

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

5. ¿La información solo puede acceder el administrador? Seguridad

- Totalmente de acuerdo
- De acuerdo
- Indiferente
- En desacuerdo
- Totalmente en desacuerdo

Anexo H: Manual de usuario



SIWEFI.

Es una herramienta de software online, este sistema web financiera ayuda a colaborar los procesos administrativos de la fundación “JASPE”

Introducción

Este manual tiene como objetivo aprender utilizar las funcionalidades básicas de SIWEFI.

¿Cómo acceder a SIWEFI?

Para acceder a la parte administrativa del sistema se requiere un correo electrónico y contraseña.



A login form with three input fields and one button. The first field is labeled 'Correo electrónico', the second 'Contraseña', and the button is labeled 'Iniciar sesión'.

Correo electrónico	
Contraseña	Iniciar sesión

¿No tiene cuenta?

En caso que no posee cuenta se puede registrar. Para ello deberá ingresar el nombre, correo electrónico y su contraseña.

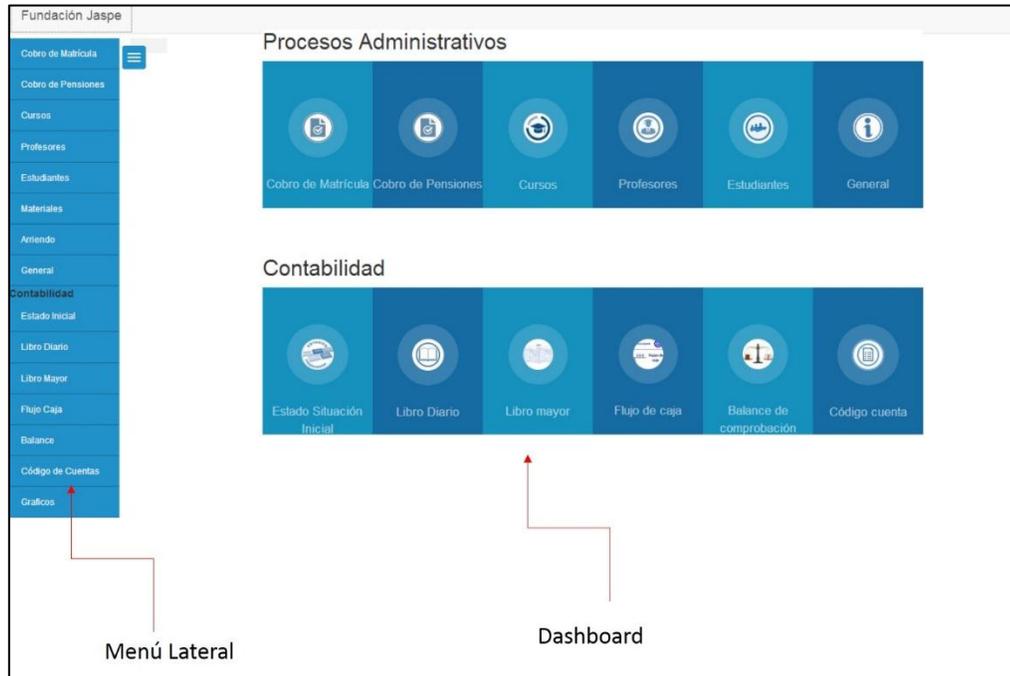


A registration form titled 'Regístrate' with four input fields and one button. The fields are labeled 'Nombre', 'Correo electrónico', 'contraseña', and 'Repita la contraseña'. The button is labeled 'Registrarse'.

Regístrate	
Nombre	
Correo electrónico	
contraseña	
Repita la contraseña	
Registrarse	

Dashboard

Cuando se ingresa como administrador se mostrará todas las funcionalidades que posee el administrador en un menú.



El dashboard está dividida en dos secciones: La primera es para gestionar los procesos administrativos de la fundación “JASPE”, entre los módulos más importantes se encuentra el cobro de matrícula, cobro de pensiones, cursos, profesores y estudiantes.

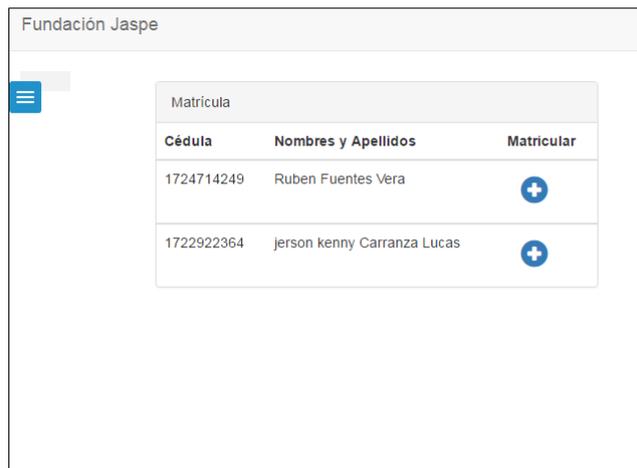
La segunda parte es para la sección de contabilidad donde tenemos los módulos de estado de situación inicial, libro diario, libro mayor, Flujo de caja y balance de comprobación.

El menú lateral nos sirve para acceder a funciones del sistema rápido cuando estemos realizando un proceso y deseamos cambiar a otro.

Módulo cobro de matrícula

La secretaria podrá aprobar la pre matrícula de los estudiantes que desea inscribirse en los cursos que ofrece la fundación.

Saldrá una lista de estudiante que han hecho la pre matrícula.



The screenshot shows a web interface for 'Fundación Jaspe'. On the left, there is a blue sidebar menu icon. The main content area displays a table titled 'Matrícula' with the following data:

Cédula	Nombres y Apellidos	Matricular
1724714249	Ruben Fuentes Vera	+
1722922364	jerson kenny Carranza Lucas	+

Cuando hacemos click en el más nos abrirá una ventana modal donde nos mostrará los datos de los estudiantes con las materias que escogió.

Fundación "Jaspe" 

Cobro de Matrícula

Cédula 1724714249

Nombres FUENTES VERA RUBEN

Fecha emisión: 06/01/2017

Atendido por: Carolina Alban

Detalle	Precio
Piano	\$20 -

[Matricular](#)

Para eliminar una materia que no desee el estudiante damos click en el signo menos, si estamos de acuerdo damos click en el signo más.

Módulo cobro de pensiones

La secretaria podrá cobrar las pensiones de los estudiantes. Se desplegará la lista de estudiante con los cursos que eligió.

Buscamos al estudiante que deseamos cobrar la pensión y le damos click en el curso a cancelar.

Fundación Jaspe

☰

Cobro de pensiones

Cédula	Nombres y Apellidos	Cursos
1724714249	Ruben Fuentes Vera	Guitarra

Al realizar click en el curso se nos mostrará una ventana modal y nos observaremos la información del estudiante en ese curso.

Fundación "Jaspe" 

Cobro de Pensión

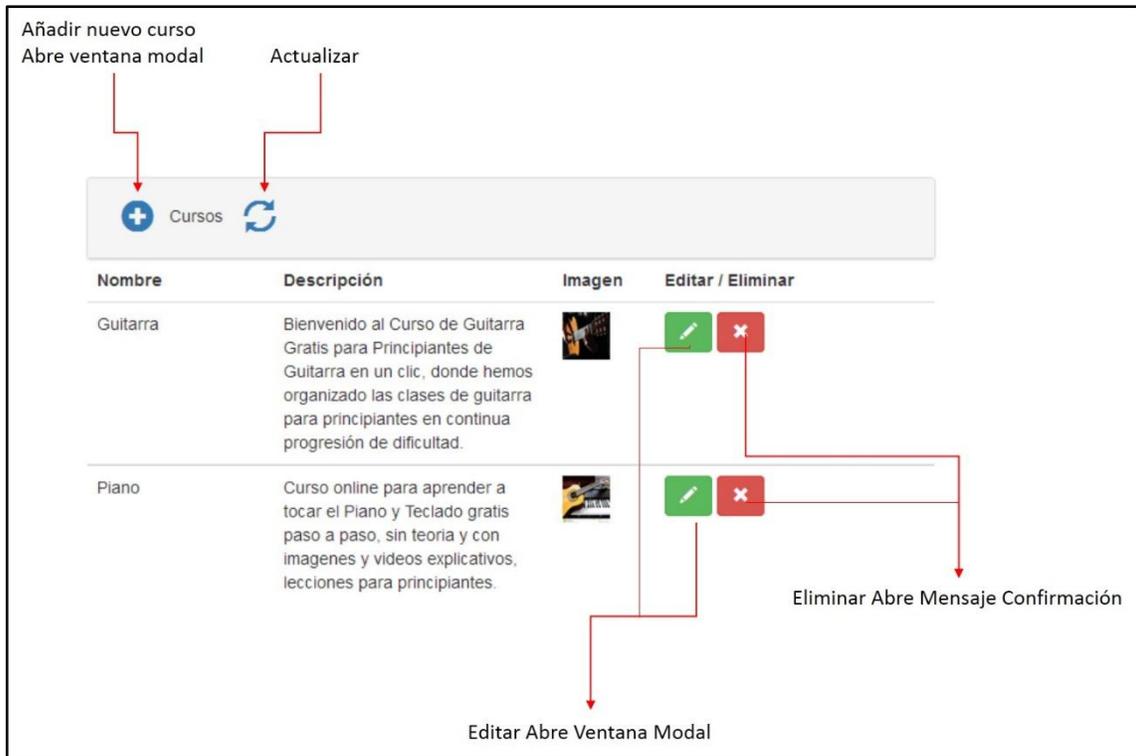
Estudiante: Ruben Fuentes Vera Cédula: 1724714249	Fecha: 1-6-2017 Atendido: Carolina Alban
Cancelado todos los meses	
Meses cancelados: 2 Mayo Junio	
<input type="button" value="Cobrar"/> <input type="button" value="Cancelar"/>	

Datos del estudiante **Meses cancelados** **Fecha y Persona que lo atendió**

Módulo curso

En esta sección podremos añadir, editar, eliminar y observar los cursos activos que posee la fundación.

En esta imagen detallamos el nombre, descripción e imagen del curso además posee los botones para realizar las acciones nombradas anteriormente.



Para añadir un curso es necesario es necesario escribir todos los datos que son nombre del curso, costo de la matrícula y de la pensión, que duración tiene el curso, en qué fecha inicia y seleccionar una imagen para la portada.

Añadir nuevo curso

Nombre del nuevo curso <input type="text" value="Curso"/>	Costo de la matrícula <input type="text" value="Precio"/> \$
Duración del curso en meses <input type="text" value="ej: 3"/>	Costo de la pensión <input type="text" value="ej. 30\$"/> \$
Descripción <input type="text" value="Descripción"/>	
Fecha de inicio del curso <input type="text" value="01-06-2017"/>	Seleccione una imagen <input type="button" value="Examinar..."/> No se ha seleccionado ningún archivo.

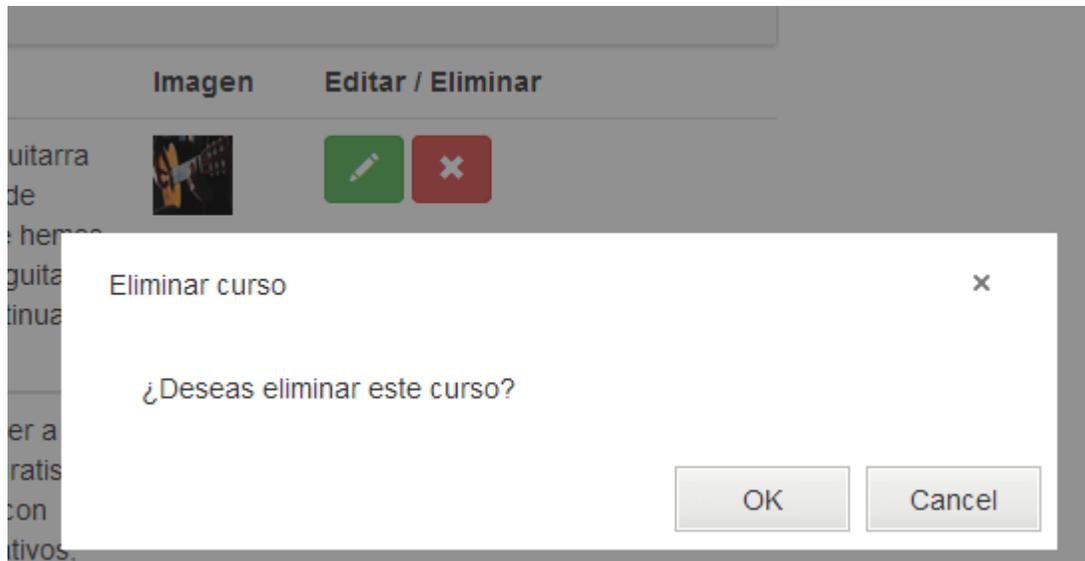
Fecha que comienza el curso Imagen para la portada del curso

Costo de la matrícula
Costo de la pensión



Cuando damos click en el botón de eliminar nos mostrará un mensaje para confirmar la eliminación del curso. Cuando ya eliminemos no aparecerá en cursos activos de la fundación.

Nota: Cuando se elimina un curso no afecta a los estudiantes que están tomando dicho curso.



Módulo estudiante

En esta sección podremos añadir, editar, eliminar y observar los estudiantes que posee la función.

Podemos listar los estudiantes según los cursos que se encuentren matriculados los estudiantes.

Filtrar estudiantes por curso

The screenshot shows a web interface for managing students. At the top, there is a title 'Filtrar estudiantes por curso'. Below it is a header bar with a blue plus icon, the text 'Nómina Estudiantes', a dropdown menu labeled 'curso' with options 'curso', 'Guitarra', and 'Piano', and a refresh icon. Below the header is a table with columns: 'Cédula', 'Nomb', 'Apellidos', and 'Modificar / Eliminar'. The table contains three rows of student data. Below the table, there is a PDF icon labeled 'PDF' and two buttons labeled 'Editar' and 'Eliminar'. Red arrows point from the labels 'Ver en formato PDF', 'Editar', and 'Eliminar' to their respective UI elements.

Cédula	Nomb	Apellidos	Modificar / Eliminar
1724714223	Jerson	Carranza Lucas	 
1724714249	Ruben	Fuentes Vera	 
1722922364	jerson kenny	Carranza Lucas	 

Ver en formato PDF

Editar

Eliminar

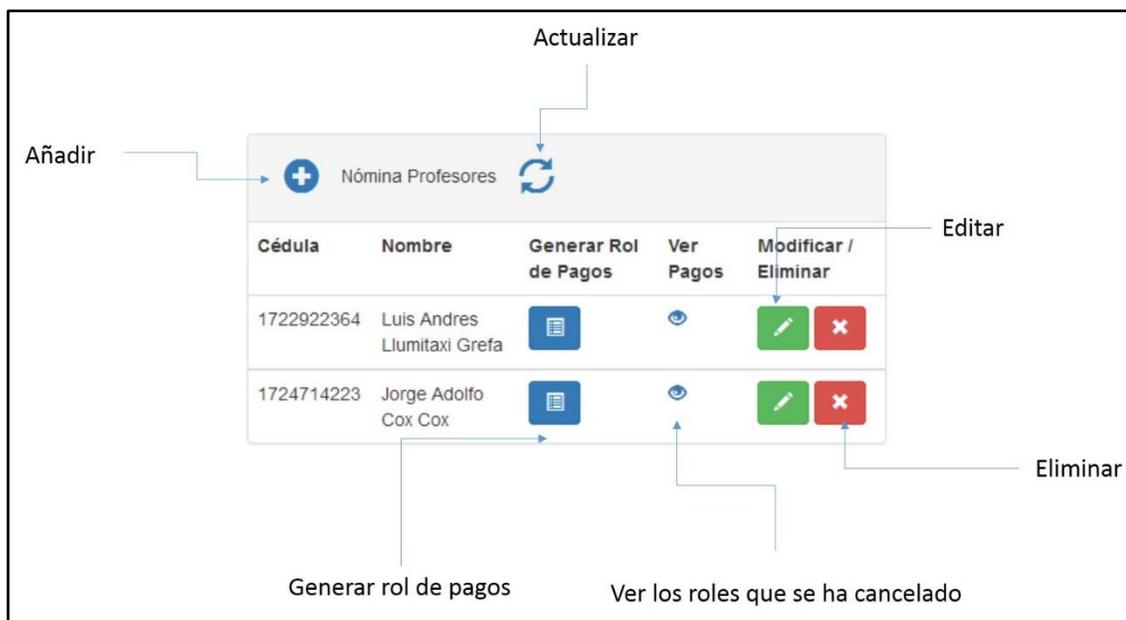
Cuando demos click en el botón editar se nos mostrará una pantalla modal en que podremos editar los datos del estudiante.

También contamos con el botón eliminar para borrar a un estudiante, para proceder ejecutar esta acción es necesario, escribir la cédula de identificación del estudiante.

Módulo Profesores

En esta sección podremos añadir, editar, eliminar y observar los profesores que posee la función.

Otra funcionalidad es emitir rol de pagos a los profesores, y observar los meses de pagos que se les cancelo a los profesores.



Cuando damos click en añadir un profesor nos mostrará una ventana modal para ingresar los datos, donde debemos ingresar la cédula, nombres completos, apellidos.

Registro de profesor

Cédula

Nombres Completos

Apellidos

Sueldo

La siguiente opción de ver los roles nos mostrará con detalle cuales meses se le pagaron a los profesores.



The screenshot displays a modal window titled "Fundación 'JASPE'" with the subtitle "Información de roles de pagos". It shows details for employee "Jorge Adolfo Cox Cox" with ID "1724714223". A table lists payment data for April 2017, including salary, commission, IESS contribution, loan, and net amount. Below the table is a card with the employee's name and ID, and icons for list, eye, edit, and delete.

Mes	Fecha de cobro	Sueldo	Comisión	Aportación IESS	Prestamo	Neto
Abril	2017-05-24	\$720	\$0	\$68.04	\$0	\$651.96

1724714223 Jorge Adolfo Cox Cox

Módulo General

En esta sección se encuentra datos sobre la fundación, misión y visión cuyos datos saldrá en la página principal de los estudiantes.

Fundación Jaspe jerson

- Cobro de Matrícula
- Cobro de Pensiones
- Cursos
- Profesores
- Estudiantes
- Materiales
- Arriendo
- General
- Contabilidad**
- Estado Inicial

Quienes somos

Por más de 26 años hemos acompañado a los chimboracences , ofreciéndoles completos cursos practicos.

Misión de la fundación

En la fundación Jaspe, estamos comprometidos con alcanzar excelentes resultados financieros a largo plazo integrando la sostenibilidad en nuestra estrategia de negocios, a la vez que dejamos una

Visión de la fundación

En la fundación Jaspe, estamos comprometidos con alcanzar excelentes resultados financieros a largo plazo integrando la sostenibilidad en nuestra estrategia de negocios, a la vez que dejamos una

En la siguiente sección se encuentra lo que es la parte contable.

Como primer punto tenemos el **Estado De Situación Inicial** donde podemos insertar, editar y listar las cuentas.

Estado de situación inicial

Fundación "JASPE"

Estado de situación inicial

Al 2 De Enero del 2017

ACTIVO		Pasivo	
caja	\$8000	documentos por pagar	\$500
		Total Pasivos	\$500
		Patrimonio	
		capital	\$7500
		Total Patrimonio	\$7500
Total de activos	\$8000	T Patrimonio + T Pasivo	\$8000

PDF

Insertar un nuevo estado de situación inicial

Editar

Descargar en formato PDF

Nota: Cuando se inserta un nuevo estado de situación inicial se borra el anterior.

Podemos descargar en formato pdf el estado de situación inicial.

Libro Diario.

Todas las transacciones que se hicieron en el proceso administrativo, están ligados a la parte contable.

Columna del Debe Columna del Haber

Libro Diario		Debe	Haber
Fecha	Detalle		
2017-05-24	Nº 1		
	caja	\$ 8000	
	documentos por pagar		\$ 500
	capital		\$ 7500
	Por :Estado de situación inicial		
2017-05-24	Nº 2		
	mercadería	\$ 80	
	caja		\$ 80
	Por :adquisición de Guantes		
2017-06-01	Nº 14		
	caja	\$ 50	
	mercadería		\$ 50
	Por :venta de Guantes		
2017-06-01	Nº 15		
	caja	\$ 25	
	Cobro de pensión		\$ 25
	Por :Cobro de pensión		
	Total	\$10274.48	\$10274.48

Número de asiento

Descargar en PDF

Total Debe y Haber

En libro diario podemos observar el número de asiento, la fecha que se realizó la transacción la cantidad en debe y en haber.

Además podemos descargar la información en formato PDF.

Libro mayor

En esta funcionalidad del sistema podemos seleccionar la cuenta para que nos devuelva el saldo de la cuenta elegida.

Cuenta

Libro Mayor cuenta: caja

Cuenta : CAJA
Código : 1.1.1.1

Debe		Haber	
Fecha	Cant	Fecha	Cant
2017-05-24	\$ 800		
2017-05-24	\$ 50	2017-05-24	\$ 60
2017-05-24	\$ 50	2017-05-24	\$ 720
2017-05-24	\$ 20	2017-05-24	\$ 80.28
2017-05-24	\$ 25	2017-05-24	\$ 800
2017-06-01	\$ 25	2017-05-24	\$ 89.2
2017-06-01	\$ 50	2017-06-01	\$ 200
2017-06-01	\$ 25		
Total	\$ 8245	Total	\$ 2029.48

Total del Debe ←

Deudor 6215.52

Saldo Deudor o Acreedor

→ Total del Haber

Balance de comprobación

EL balance sirve para asegurar que todos los asientos contables correspondientes que se han registrados en los libros, de acuerdo con el concepto del sistema de partida doble en la contabilidad. Si los valores del balance de comprobación no concuerdan, estas diferencias pueden ser indagadas y resueltas antes de que se preparen los estados financieros.

Código	Nombre	Debe	Haber	Deudor	Acreedor
1.1.1.1	caja	\$8245	\$2029.48	\$6215.52	\$0
1.1.2.6	mercadería	\$140	\$175	\$0	\$35
2.1.4.01	servicios cobrados por anticipado	\$0	\$20	\$0	\$20
2.1.4.1	Cobro de pensión	\$0	\$50	\$0	\$50
2.1.1	Documentos por Pagar	\$0	\$500	\$0	\$500
2.3.2.2	pago arriendo	\$200	\$0	\$200	\$0
3.1.1	Capital	\$0	\$7500	\$0	\$7500
4.1.1	sueños y salarios	\$1376.36	\$0	\$1376.36	\$0
4.1.2	aporte less	\$143.64	\$0	\$143.64	\$0
4.1.3	aporte less patrono	\$169.48	\$0	\$169.48	\$0
		\$10274.48	\$10274.48	\$8105	\$8105

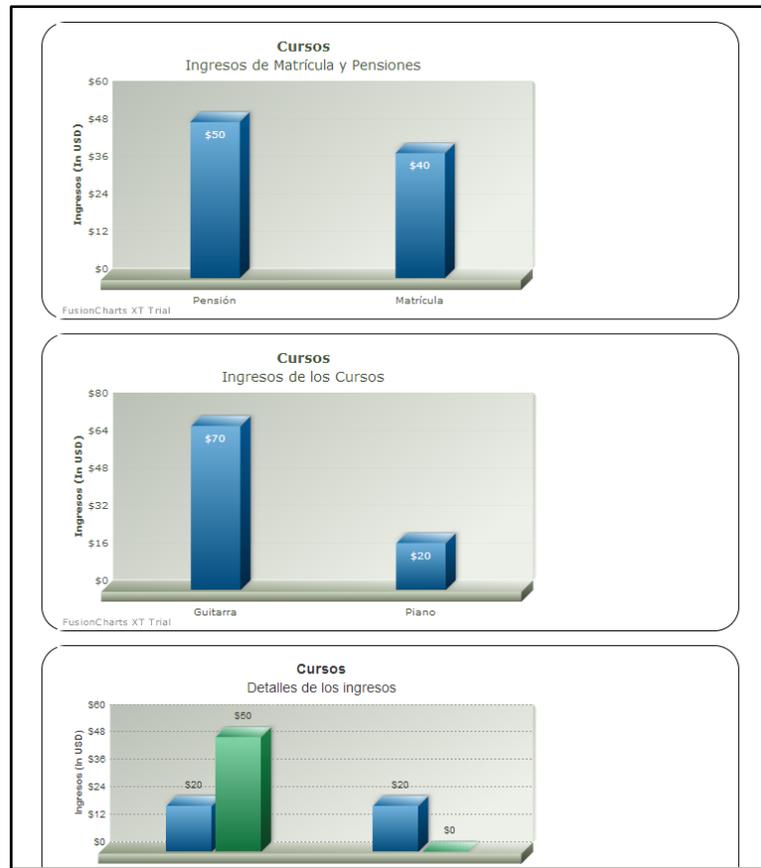
Saldo

Gráficos

Nos servirá para tener una mejor percepción sobre los datos de la fundación. Para acceder debemos de ir al menú lateral y dar click en *gráficos* el siguiente paso es elegir una de las opciones para que se nos muestre los gráficos.



En la opción de cursos nos mostrará los ingresos que generaron cada curso, con sus respectivos detalles.



En la opción estudiantes podremos observar cuantos estudiantes están matriculados en los cursos de manera gráfica.

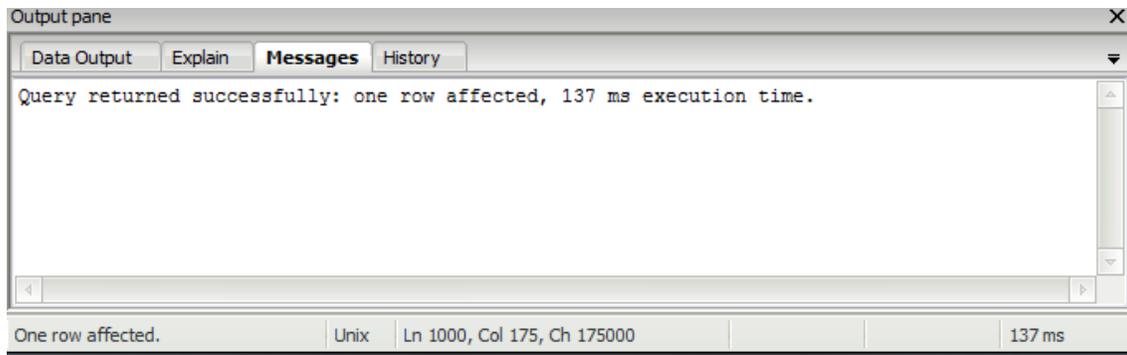


También tenemos el botón para poder descargar los gráficos.

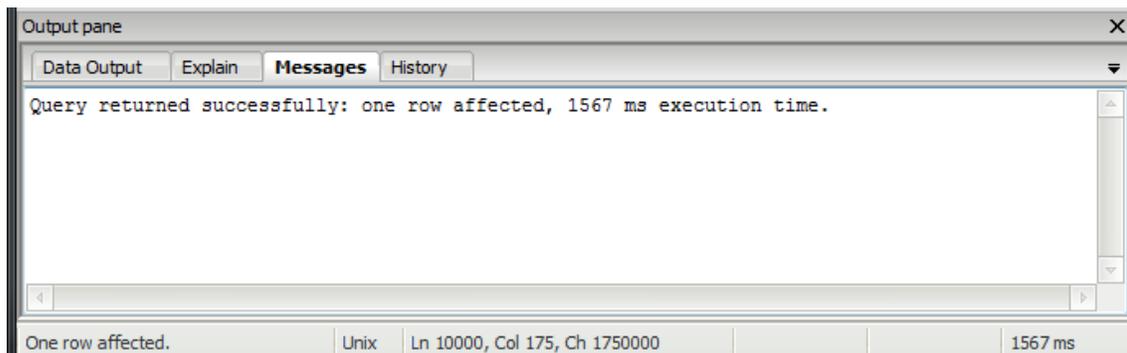
Anexo I: Análisis de Rendimiento de la base de datos noSQL frente a una SQL(PostgreSQL)

PostgreSQL

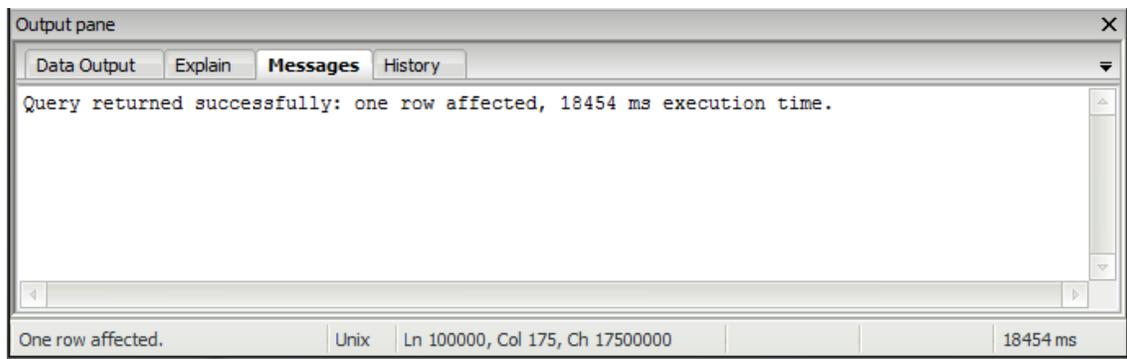
Ingresar 1000 registros en un tiempo de 137 ms



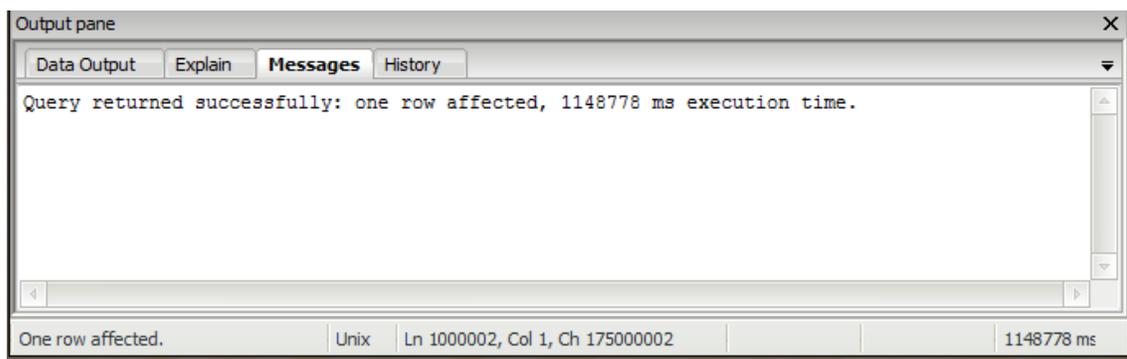
Ingresar 10000 registros en un tiempo de 1,225ms



Ingresar 100,000 registros en un tiempo en 18454ms

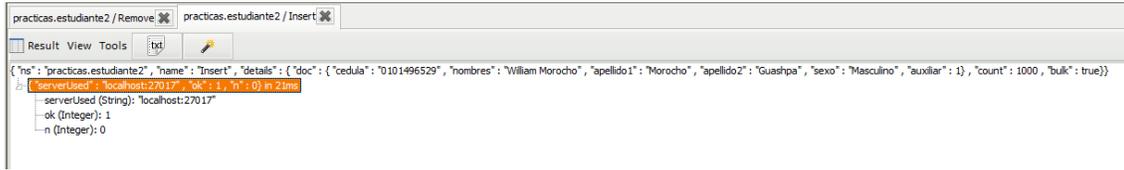


1,000,000 de registros en un tiempo de 1148778ms



Resultados de inserción en mongoDB

1,000 registros en 21 milisegundos



```
practicass.estudiante2 / Remove  practicass.estudiante2 / Insert
Result View Tools txt
{"ns": "practicass.estudiante2", "name": "Insert", "details": {"doc": {"cedula": "0101496529", "nombres": "William Morocho", "apellido1": "Morocho", "apellido2": "Guashpa", "sexo": "Masculino", "auxiliar": 1}, "count": 1000, "bulk": true}}
  serverUsed (String): localhost:27017, "ok": 1, "n": 0 in 21ms
  serverUsed (String): localhost:27017
  ok (Integer): 1
  n (Integer): 0
```

10,000 registros en 150 milisegundos



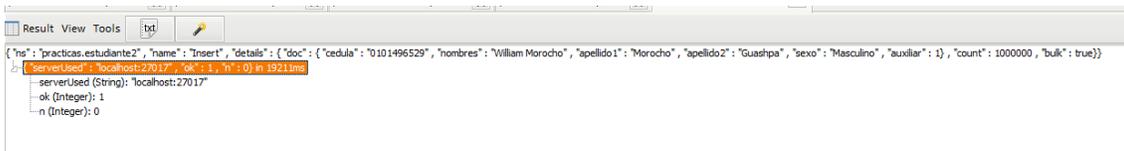
```
Result View Tools txt
{"ns": "practicass.estudiante2", "name": "Insert", "details": {"doc": {"cedula": "0101496529", "nombres": "William Morocho", "apellido1": "Morocho", "apellido2": "Guashpa", "sexo": "Masculino", "auxiliar": 1}, "count": 10000, "bulk": true}}
  serverUsed (String): localhost:27017, "ok": 1, "n": 0 in 150ms
  serverUsed (String): localhost:27017
  ok (Integer): 1
  n (Integer): 0
```

100,000 registros en 1468 milisegundos



```
Result View Tools txt
{"ns": "practicass.estudiante2", "name": "Insert", "details": {"doc": {"cedula": "0101496529", "nombres": "William Morocho", "apellido1": "Morocho", "apellido2": "Guashpa", "sexo": "Masculino", "auxiliar": 1}, "count": 100000, "bulk": true}}
  serverUsed (String): localhost:27017, "ok": 1, "n": 0 in 1468ms
  serverUsed (String): localhost:27017
  ok (Integer): 1
  n (Integer): 0
```

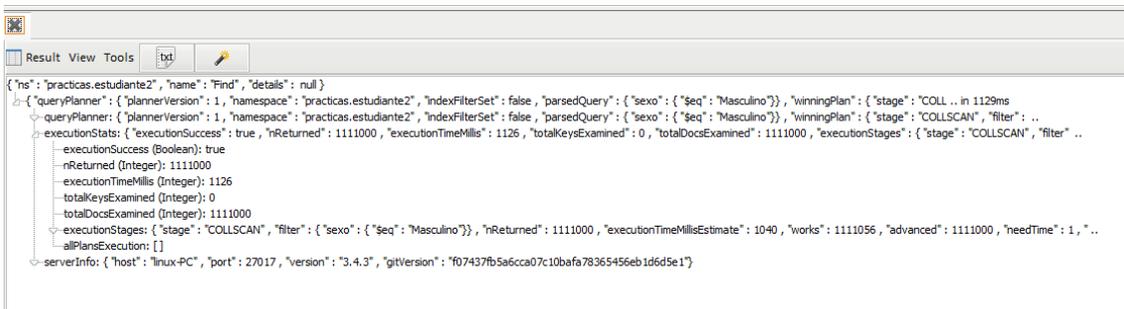
1000,000 de registros en 19221 milisegundos



```
Result View Tools txt
{"ns": "practicass.estudiante2", "name": "Insert", "details": {"doc": {"cedula": "0101496529", "nombres": "William Morocho", "apellido1": "Morocho", "apellido2": "Guashpa", "sexo": "Masculino", "auxiliar": 1}, "count": 1000000, "bulk": true}}
  serverUsed (String): localhost:27017, "ok": 1, "n": 0 in 19221ms
  serverUsed (String): localhost:27017
  ok (Integer): 1
  n (Integer): 0
```

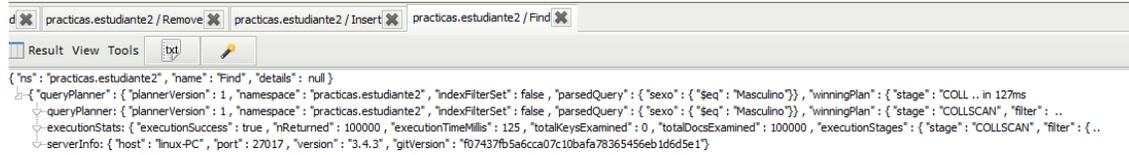
Consultas en mongoDB

1,111,000 registros consultados



```
Result View Tools txt
{"ns": "practicass.estudiante2", "name": "Find", "details": null}
  queryPlanner: {"plannerVersion": 1, "namespace": "practicass.estudiante2", "indexFilterSet": false, "parsedQuery": {"sexo": {"$eq": "Masculino"}}, "winningPlan": {"stage": "COLLSCAN", "filter": ..}}
  executionStats: {"executionSuccess": true, "nReturned": 1111000, "executionTimeMillis": 1126, "totalKeysExamined": 0, "totalDocsExamined": 1111000, "executionStages": {"stage": "COLLSCAN", "filter": ..}}
  executionSuccess (Boolean): true
  nReturned (Integer): 1111000
  executionTimeMillis (Integer): 1126
  totalKeysExamined (Integer): 0
  totalDocsExamined (Integer): 1111000
  executionStages: {"stage": "COLLSCAN", "filter": {"sexo": {"$eq": "Masculino"}}, "nReturned": 1111000, "executionTimeMillisEstimate": 1040, "works": 1111056, "advanced": 1111000, "needTime": 1, "allPlansExecution": []}
  serverInfo: {"host": "linux-PC", "port": 27017, "version": "3.4.3", "gitVersion": "f07437fb5a6cca07c10bafa78365456eb1d6d5e1"}
```

100,000 registros consultados



```
{ "ns": "practicas.estudiante2", "name": "Find", "details": null }
  - { "queryPlanner": { "plannerVersion": 1, "namespace": "practicas.estudiante2", "indexFilterSet": false, "parsedQuery": { "sexo": { "$eq": "Masculino" } }, "winningPlan": { "stage": "COLL .. in 127ms
    - queryPlanner: { "plannerVersion": 1, "namespace": "practicas.estudiante2", "indexFilterSet": false, "parsedQuery": { "sexo": { "$eq": "Masculino" } }, "winningPlan": { "stage": "COLLSCAN", "filter": ..
    - executionStats: { "executionSuccess": true, "nReturned": 100000, "executionTimeMills": 125, "totalKeysExamined": 0, "totalDocsExamined": 100000, "executionStages": { "stage": "COLLSCAN", "filter": { ..
    - serverInfo: { "host": "linux-PC", "port": 27017, "version": "3.4.3", "gitVersion": "f07437fb5a6cca07c10bafa78365456eb1d6d5e1" }
```

10,000 registros consultados



```
{ "ns": "practicas.estudiante2", "name": "Find", "details": null }
  - { "queryPlanner": { "plannerVersion": 1, "namespace": "practicas.estudiante2", "indexFilterSet": false, "parsedQuery": { "sexo": { "$eq": "Masculino" } }, "winningPlan": { "stage": "COLL .. in 26ms
    - queryPlanner: { "plannerVersion": 1, "namespace": "practicas.estudiante2", "indexFilterSet": false, "parsedQuery": { "sexo": { "$eq": "Masculino" } }, "winningPlan": { "stage": "COLLSCAN", "filter": ..
    - executionStats: { "executionSuccess": true, "nReturned": 10000, "executionTimeMills": 24, "totalKeysExamined": 0, "totalDocsExamined": 10000, "executionStages": { "stage": "COLLSCAN", "filter": { "s ..
    - serverInfo: { "host": "linux-PC", "port": 27017, "version": "3.4.3", "gitVersion": "f07437fb5a6cca07c10bafa78365456eb1d6d5e1" }
```

1,000 registros consultados



```
{ "ns": "practicas.estudiante2", "name": "Find", "details": null }
  - { "queryPlanner": { "plannerVersion": 1, "namespace": "practicas.estudiante2", "indexFilterSet": false, "parsedQuery": { "sexo": { "$eq": "Masculino" } }, "winningPlan": { "stage": "COLL .. in 2ms
    - queryPlanner: { "plannerVersion": 1, "namespace": "practicas.estudiante2", "indexFilterSet": false, "parsedQuery": { "sexo": { "$eq": "Masculino" } }, "winningPlan": { "stage": "COLLSCAN", "filter": ..
    - executionStats: { "executionSuccess": true, "nReturned": 1000, "executionTimeMills": 2, "totalKeysExamined": 0, "totalDocsExamined": 1000, "executionStages": { "stage": "COLLSCAN", "filter": { "sexo ..
    - serverInfo: { "host": "linux-PC", "port": 27017, "version": "3.4.3", "gitVersion": "f07437fb5a6cca07c10bafa78365456eb1d6d5e1" }
```

Consultas en PostgreSQL

1.111.000 registros consultados

Query - jasje on postgres@localhost:5432

	idest integer	cedula character varying(10)	nombres character varying(70)	apellidop character varying(50)	apellidom character varying(50)	exo character varying(50)	auxiliar integer
1	570009	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
2	570010	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
3	570011	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
4	570012	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
5	570013	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
6	570014	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
7	570015	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
8	570016	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
9	570017	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
10	570018	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
11	570019	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
12	570020	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
13	570021	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
14	570022	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
15	570023	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
16	570024	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
17	570025	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1
18	570026	0604569327	Williaml Patricio	Morocho	Guashpa	masculino	1

OK. Unix Ln 1, Col 1, Ch 1 1110000 rows. 62840 ms

100.000 registros consultados

Query - jasje on postgres@localhost:5432

SQL Editor

```
SELECT idestudiante, cedula, nombres, apellidop, apellidom
FROM estudiante;
```

Output pane

	idestudiante integer	cedula character varying(10)	nombres character varying(70)	apellidop character varying(50)	apellidom character varying(50)
1	1680009	0604569327	William Morocho	Morocho	Guashpa
2	1680010	0604569327	William Morocho	Morocho	Guashpa
3	1680011	0604569327	William Morocho	Morocho	Guashpa
4	1680012	0604569327	William Morocho	Morocho	Guashpa
5	1680013	0604569327	William Morocho	Morocho	Guashpa

OK. Unix Ln 1, Col 1, Ch 1 100000 rows. 5561 ms

10.000 registros consultados

Query - jaspe on postgres@localhost:5432

File Edit Query Favourites Macros View Help

jaspe on postgres@localhost:5432

SQL Editor Graphical Query Builder

Previous queries Delete Delete All

```
SELECT idestudiante, cedula, nombres, apellidop, apellidom, sexo, aux
FROM estudiante;
```

Scratch pad

Output pane

Data Output Explain Messages History

	idestudiante integer	cedula character varying(10)	nombres character varying(70)	apellidop character varying(50)	apellidom character varying(50)	sexo character varying(10)
1	1856633	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
2	1856634	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
3	1856635	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
4	1856636	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
5	1856637	0604569327	Williaml Patricio	Morocho	Guashpa	masculino

OK. Unix Ln 1, Col 1, Ch 1 10000 rows. 571 ms

1.000 registros consultados

The screenshot shows a PostgreSQL query editor window titled "Query - jaspe on postgres@localhost:5432". The SQL Editor contains the following query:

```
SELECT idestudiante, cedula, nombres, apellidop, apellidom, sexo, aux
FROM estudiante;
```

The Output pane displays the results in a table with the following columns: idestudiante (integer), cedula (character varying(10)), nombres (character varying(70)), apellidop (character varying(50)), apellidom (character varying(50)), and sexo (character varying(?)). The table contains 5 rows of data:

	idestudiante integer	cedula character varying(10)	nombres character varying(70)	apellidop character varying(50)	apellidom character varying(50)	sexo character varying(!)
1	1866633	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
2	1866634	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
3	1866635	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
4	1866636	0604569327	Williaml Patricio	Morocho	Guashpa	masculino
5	1866637	0604569327	Williaml Patricio	Morocho	Guashpa	masculino

The status bar at the bottom indicates "OK.", "Unix", "Ln 1, Col 1, Ch 1", "1000 rows.", and "61 ms".

Eliminar registros en PostgreSQL

100000 registros eliminados.

The screenshot shows the same PostgreSQL query editor window. The SQL Editor now contains the following query:

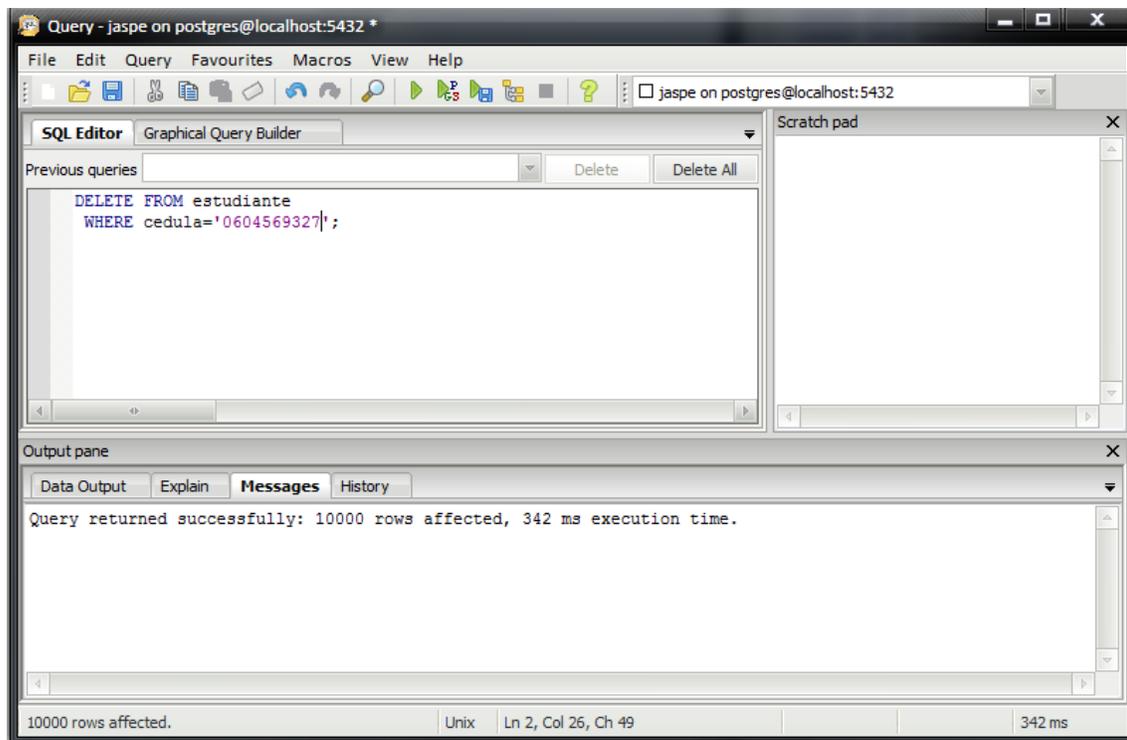
```
DELETE FROM estudiante
WHERE cedula='0604569327';
```

The Output pane is set to "Messages" and displays the following message:

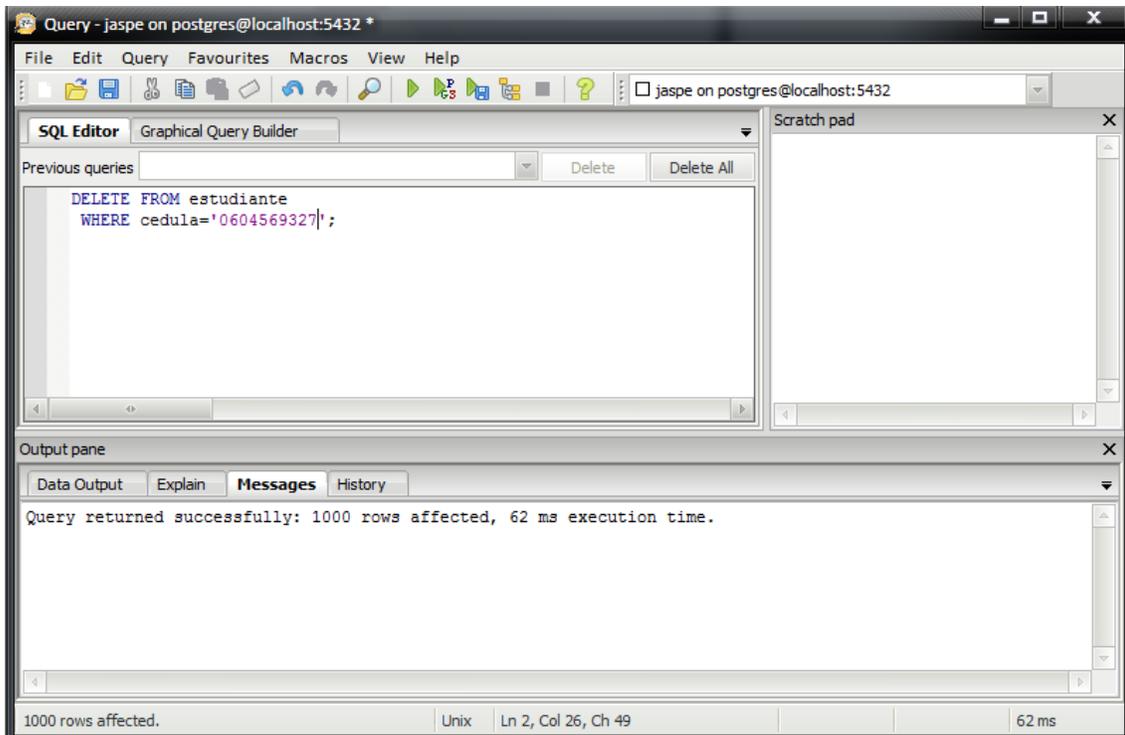
```
Query returned successfully: 100000 rows affected, 3805 ms execution time.
```

The status bar at the bottom indicates "100000 rows affected.", "Unix", "Ln 2, Col 26, Ch 49", and "3805 ms".

10.000 registros eliminados

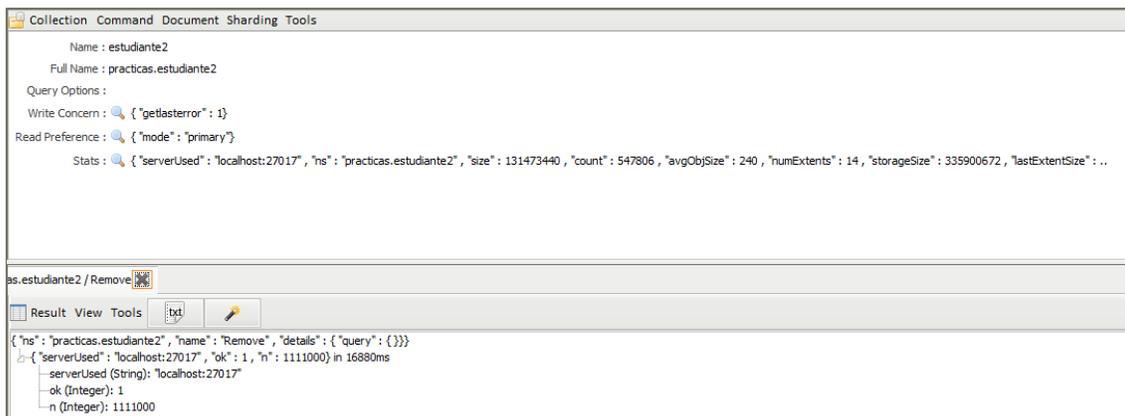


1.000 registros eliminados



Eliminar en mongoDB

1.111.000 registros eliminados



100.000 registros eliminados

Collection Command Document Sharding Tools

Name : estudiante2
Full Name : practicas.estudiante2

Query Options :
Write Concern : { "getLastError" : 1 }
Read Preference : { "mode" : "primary" }

Stats : { "serverUsed" : "localhost:27017", "ns" : "practicas.estudiante2", "size" : 2400000, "count" : 100000, "avgObjSize" : 240, "numExtents" : 14, "storageSize" : 335900672, "lastExtentSize" : ..

practicas.estudiante2 / Remove

Result View Tools

```
{ "ns" : "practicas.estudiante2", "name" : "Remove", "details" : { "query" : {} } }
  { "serverUsed" : "localhost:27017", "ok" : 1, "n" : 100000 } in 1198ms
    -serverUsed (String): localhost:27017
    -ok (Integer): 1
    -n (Integer): 100000
```

10.000 registros eliminados

Collection Command Document Sharding Tools

Name : estudiante2
Full Name : practicas.estudiante2

Query Options :
Write Concern : { "getLastError" : 1 }
Read Preference : { "mode" : "primary" }

Stats : { "serverUsed" : "localhost:27017", "ns" : "practicas.estudiante2", "size" : 0, "count" : 0, "numExtents" : 14, "storageSize" : 335900672, "lastExtentSize" : 9.2585984E7, "paddingFactor" : 1 ..

practicas.estudiante2 / Remove | practicas.estudiante2 / Insert | practicas.estudiante2 / Remove

Result View Tools

```
{ "ns" : "practicas.estudiante2", "name" : "Remove", "details" : { "query" : {} } }
  { "serverUsed" : "localhost:27017", "ok" : 1, "n" : 10000 } in 162ms
    -serverUsed (String): localhost:27017
    -ok (Integer): 1
    -n (Integer): 10000
```

1000 registros eliminados

Collection Command Document Sharding Tools

Name : estudiante2
Full Name : practicas.estudiante2

Query Options :
Write Concern : { "getLastError" : 1 }
Read Preference : { "mode" : "primary" }

Stats : { "serverUsed" : "localhost:27017", "ns" : "practicas.estudiante2", "size" : 0, "count" : 0, "numExtents" : 14, "storageSize" : 335900672, "lastExtentSize" : 9.2585984E7, "paddingFactor" : 1 ..

practicas.estudiante2 / Remove | practicas.estudiante2 / Insert | practicas.estudiante2 / Remove | practicas.estudiante2 / Insert | practicas.estudiante2 / Remove

Result View Tools

```
{ "ns" : "practicas.estudiante2", "name" : "Remove", "details" : { "query" : {} } }
  { "serverUsed" : "localhost:27017", "ok" : 1, "n" : 1000 } in 22ms
    -serverUsed (String): localhost:27017
    -ok (Integer): 1
    -n (Integer): 1000
```