



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

**“DESARROLLO DEL SISTEMA ACADÉMICO DEL SINDICATO
DE CHOFERES PROFESIONALES 4 DE OCTUBRE APLICANDO
EL FRAMEWORK JSF”**

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

AUTOR: CRISTIAN GEOVANNY MERINO SANCHEZ
TUTOR: DR. JULIO ROBERTO SANTILLÁN CASTILLO

Riobamba-Ecuador

2017

©2017, Cristian Geovanny Merino Sanchez,

Se autoriza la reproducción 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 técnico: “**DESARROLLO DEL SISTEMA ACADÉMICO DEL SINDICATO DE CHOFERES PROFESIONALES 4 DE OCTUBRE APLICANDO EL FRAMEWOR JSF**”, de responsabilidad del señor Cristian Geovanny Merino Sanchez, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

	FIRMA	FECHA
ING. WASHINGTON LUNA DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
ING. PATRICIO MORENO DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS	_____	_____
DR. JULIO SANTILLÁN DIRECTOR DEL TRABAJO DE TITULACIÓN	_____	_____
ING. EDUARDO VILLA MIEMBRO DEL TRIBUNAL	_____	_____

Yo, Cristian Geovanny Merino Sanchez, soy responsable de las ideas, doctrinas y resultados expuestos en este trabajo y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

Cristian Geovanny Merino Sanchez

DEDICATORIA

Siempre es grato recordar por quién estamos en este mundo, a ti mi Dios padre, mi poder superior es a quien de manera prioritaria quiero dedicar este logro en mi vida profesional, a ti que me das tanto aún sin merecerlo, me has regalado la dicha de la vida, la salud, la familia, que es todo lo que necesito para ser feliz.

A mis queridos padres Luis Merino y Rosita Sanchez, especialmente por el ejemplo latente de superación en solitario ante las adversidades de los caminos de la vida, por su amor, su apoyo incondicional y desinteresado, por inculcarme valores, por enseñarme a enfrentar este inexplicable pero maravilloso mundo, por levantarme ciento uno veces de las cien que caí.

A mis amados hijos, regalitos de Dios, Cristian Voltaire y Ainhoa Viviana, por venir alegrar mis días, porque son la motivación diaria, los que inspiran mi existencia, por ser esas palomas mensajeras que el señor envió del cielo para hablarme de su amor.

A mis extraordinarios hermanos Elizabeth, Janeth, Junior y Jennifer por describir el significado de unión, por ejemplificar el amor de hermanos y por definir correctamente el significado de amistad.

Cristian

AGRADECIMIENTO

A mi Dios padre por su infinito amor, porque nunca te olvidas de mí, aunque yo lo haga, porque nunca sueltas mi mano, por alumbrar mi camino y ser mi acompañante silencioso y hemos llegado hasta acá. Vamos por más.

A mis padres, Dios les pagué por todo y, por tanto, por ser mis ojos en la oscuridad de mis errores, por ser mis piernas cuando me cansé de caminar, por ser el energizante en el momento preciso que sentía no poder dar más. A mis hermanos por su ejemplo y locuras, gracias mi amada familia por seguir alentando en el graderío de mi corazón el juego más importante que he tenido, la vida.

A mi querida ESPOCH por brindarme esta extraordinaria oportunidad de recibir una educación de calidad. Gracias mi querida Poli, espero que Dios y la vida nos vuelva a juntar.

“El que hace lo que ama está benditamente condenado al éxito, que llegará cuando deba llegar porque lo que debe ser será y llegará naturalmente” (Facundo Cabral)

Cristian

TABLA DE CONTENIDO

	Página
RESUMEN	xv
SUMARY	xvi
INTRODUCCIÓN	2
ANTECEDENTES.....	3
FORMULACIÓN DEL PROBLEMA	5
SISTEMATIZACIÓN DEL PROBLEMA.....	5
JUSTIFICACIÓN	5
OBJETIVOS	7
CAPÍTULO I	8
1. MARCO TEÓRICO.....	8
1.1 JavaServer Faces	8
1.1.1 Introducción	8
1.1.2 Historia.....	9
1.1.3 Definición.....	10
1.1.4 Principales Características de JSF	11
1.1.5 Versiones.....	12
1.1.6 Beneficios.....	14
1.1.7 Arquitectura	14
1.1.8 Ciclo de Vida.....	15
1.1.9 Acceso a la Base de Datos.....	17
1.2 Metodología de Desarrollo SCRUM	17
1.2.1 Introducción	17
1.2.2 Los Roles.....	18
1.2.3 Fases de SCRUM	19
1.3 Norma de evaluación ISO/IEC 9126-3.....	20
1.3.1 Usabilidad	21

1.3.2	Funcionalidad	21
1.3.3	Eficiencia	22
CAPÍTULO II		23
2 .	MARCO METODOLÓGICO	23
2.1	Introducción	23
2.2	Aplicación del modelo ágil de desarrollo SCRUM en el proyecto	23
2.3	Planificación del proyecto	23
2.3.1	Personas y Roles del Proyecto	23
2.3.2	Tipos de usuarios y roles del sistema académico	24
2.3.3	Actividades realizadas en el proyecto	25
2.3.4	Requisitos del proyecto	25
2.4	Cronograma de actividades	34
2.5	Desarrollo del proyecto	35
2.5.1	Estandar de codificación	35
2.5.2	Modelado del Sistema UML	35
2.5.3	Descripción de las capas del sistema académico	35
2.5.4	Diseño de la base de datos	36
2.5.5	Diccionario de datos	37
2.5.6	Diseño de interfaces	38
2.5.7	Desarrollo de las historias de usuario	44
2.5.8	Codificación e implementación	46
2.6	Finalización del proyecto	48
2.6.1	Gestión del proyecto	49
CAPÍTULO III		52
3.	MARCO DE ANÁLISIS, DISCUSIÓN Y EVALUACIÓN DE RESULTADOS	52
3.1	Parámetros de calidad	53

3.2	Análisis de la usabilidad del sistema académico	54
3.3	Análisis de la funcionalidad del sistema académico.....	59
3.4	Análisis de la eficiencia del sistema académico	64
3.5	Evaluación de la calidad del sistema académico	67
CONCLUSIONES		71
RECOMENDACIONES		72
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-1:	Versiones de JSF.....	12
Tabla 2-1:	Roles en Scrum	18
Tabla 3-1:	Fases de Scrum	19
Tabla 1-2:	Personas y roles del proyecto	24
Tabla 2-2:	Tipos de usuarios del sistema académico	24
Tabla 3-2:	Actividades realizadas en el proyecto	25
Tabla 4-2:	Niveles de prioridad de los requisitos del proyecto	26
Tabla 5-2:	Unidades de medición de prioridad.....	26
Tabla 6-2:	Asignación estimación- carta	27
Tabla 7-2:	Product Backlog.....	27
Tabla 8-2:	Sprint Backlog	31
Tabla 9-2:	Cronograma de actividades	34
Tabla 10-2:	Diccionario de datos	37
Tabla 11-2:	Interfaz de autenticación	38
Tabla 12-2:	Componentes de interfaz de usuario	39
Tabla 13-2:	Historia de usuario 01 (HU-01) / proceso de autenticación.....	44
Tabla 14-2:	Historia de usuario 01 (HU-01) / Tareas de ingeniería.....	44
Tabla 15-2:	Historia de usuario 01 (HU-01) / Tareas de ingeniería 01	45
Tabla 16-2:	Historia de usuario 01 / Prueba de aceptación 01	46
Tabla 17-2:	Actividades de finalización del proyecto.....	48
Tabla 1-3:	Parámetros para la medición de la evaluación.....	53
Tabla 2-3:	Características y Subcaracterísticas de calidad.....	54
Tabla 3-3:	Métrica de entendimiento.....	55
Tabla 4-3:	Métrica de aprendizaje	56
Tabla 5-3:	Métrica de operabilidad	57

Tabla 6-3:	Métrica de atracción.....	58
Tabla 7-3:	Métrica de conformidad de la usabilidad	59
Tabla 8-3:	Métrica de adecuación	60
Tabla 9-3:	Métrica de exactitud.....	61
Tabla 10-3:	Métrica de interoperabilidad	62
Tabla 11-3:	Métrica de seguridad.....	63
Tabla 12-3:	Métrica de conformidad de la funcionalidad	63
Tabla 13-3:	Métrica de Comportamiento de respuesta	65
Tabla 14-3:	Métrica de la utilización de recursos.....	66
Tabla 15-3:	Métrica de conformidad de la eficiencia	67
Tabla 16-3:	Evaluación de la usabilidad del sistema académico	67
Tabla 17-3:	Evaluación de la funcionalidad del sistema.....	69
Tabla 18-3:	Evaluación de la eficiencia del sistema	70

ÍNDICE DE FIGURAS

Figura 1-1: JavaServer Faces	8
Figura 2-1: Estructura de JavaServer Faces.....	10
Figura 3-1: Arquitectura JavaServer Faces.....	14
Figura 4-1: Ciclo de vida de JSF	15
Figura 5-1: Acceso a la Base de Datos con JSF	17
Figura 6-1: Ciclo de vida de SCRUM	17
Figura 7-1: Roles de Scrum.....	18
Figura 8-1: Norma de evaluación ISO/IEC 9126-3	20
Figura 9-1: Métricas de evaluación de calidad de software ISO/IEC 9126-3	20
Figura 1-2: Planning Póker	26
Figura 2-2: Esquema de la base de datos.....	36
Figura 3-2: Diagrama entidad relación	37
Figura 4-2: Bosquejo de autenticación	39
Figura 5-2: Interfaz de autenticación.....	43
Figura 6-2: Interfaz de administración	43
Figura 1-3: Muestreo aleatorio estratificado desproporcionado	52

ÍNDICE DE ILUSTRACIONES

Gráfico 1-2: Sprint BurnDown Chart	50
Gráfico 1-3: Evaluación de la usabilidad del sistema	68
Gráfico 2-3: Evaluación de la funcionalidad del sistema.....	69
Gráfico 3-3: Evaluación de la eficiencia del sistema.....	70

ÍNDICE DE ABREVIATURAS

ESPOCH:	Escuela Superior Politécnica de Chimborazo.
JSF	JavaServer Faces
XAMPP	SO, APACHE, MySql, PHP y PERL
SO	Sistema Operative
APACHE	Servidor web de HTTP de código abierto.
MYSQL:	My Structured Query Language o Lenguaje de Consulta Estructurado
PHP:	Hypertext Preprocessor
PERL	Es un Lenguaje de progradation
JAVABEANS	Modelo de componentes Java.
SERVLET	Es una clase en el lenguaje de programación Java.
FACELETS	Es un sistema de código abierto de plantillas web.
TOMCAT	Contenedor de servlets.
PRIMEFACES	Es una librería componentes para JSF.
API	Application Programming Interface, Interfaces programación de aplicaciones.
OP	Open source o código abierto
HTTP	Hypertext Transfer Protocol o Protocolo de transferencia de hipertexto
SCRUM	Significa melé, un tipo de jugada de Rugby.
CSS:	Cascading style sheet (Hoja de estilo en cascada).
HTML:	Hypertext markup language (Lenguaje de marcas de hipertexto).
HU:	Historias de usuario.
HT:	Historia técnica.
ISO:	Organización Internacional de Normalización.
LC:	Líneas de Código.
SW:	Software.

RESUMEN

El objetivo del presente trabajo de titulación fue desarrollar el sistema académico para el Sindicato de Choferes Profesionales 4 de Octubre de la ciudad de Penipe aplicando el framework JSF para automatizar el proceso de matrículas y digitalizar la información académica, las herramientas tecnológicas utilizadas para el desarrollo del sistema académico son: Framework JSF, MySQL, APACHE, PHP, TOMCAT, PRIMEFACES, CSS3. La metodología de desarrollo escogida es SCRUM por su enfoque de gestión ágil que facilitó la gestión del presente proyecto, notablemente contribuyó a la creación de un entorno laboral transparente, adaptable, de retroalimentación y mejora continua a un ritmo de trabajo sostenible entre el Product Owner y el Team Scrum para que el proceso de desarrollo sea eficiente, posee tres fases; la fase de planificación mediante reuniones se capturaron los requerimientos denominados historias de usuario del sistema, descrito en un lenguaje no técnico y priorizados por el valor de la institución, la fase de desarrollo se formalizaron las historias de usuarios asociadas en sprints, y la fase de cierre se ejecutaron las pruebas del sistema académico, la documentación y la capacitación a los usuarios del sistema académico. Posteriormente, se sometió a pruebas de usabilidad, funcionalidad y eficiencia, con la ayuda de la encuestas dirigidas a la secretaria, estudiantes y profesores se determinó que el sistema académico es 90,48% usable, un 98.72% funcional, y 92.91% eficiente, obteniendo un promedio general de 94.04% en calidad, finalmente se concluye que el sistema ayudará a la secretaria, estudiantes y profesores del Sindicato de Choferes Profesionales 4 de Octubre agilizar el proceso de matrículas y también a llevar una organización digital de la información académica, por lo que se recomienda a los integrantes de la institución hacer uso del sistema académico para facilitar los procesos académicos.

Palabras claves: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <FRAMEWORK JSF>, <METODOLOGÍA DE DESARROLLO (SCRUM)>, < APACHE (SERVIDOR LOCAL)>, < (PRIMEFACES) LIBRERÍA DE COMPONENTES>, <INFORMACIÓN ACADÉMICA>, <MÉTRICAS DE EVALUACIÓN>.

SUMMARY

The objective of the present titling work was develop the academic system for the Sindicato de Choferes Profesionales 4 de Octubre of the city of Prnipe applying the JSF framework to automate the registration process and digitize the academic information. The technological tools used for the development of the academic system are JSF framework, MySQL, APACHE, PHP, TOMCAT, PRIMEFACES, CSS3. The development methodology chosen is SCRUM, due to its agile management approach that facilitated the management of this project. It notably contributed to the creation of a transparent, adaptable, feedback and continuous improvement at a sustainable work pace between the Product Owner and Team Scrum. For the development process to be efficient, it has three phases, in the planning phase, through meetings were captured the requirements called user stories of the system, described in a non-technical language and prioritized by the value of the institution, in the development phase user stories associated with in sprints were formalizad, and in the closing phase the test of the academic system, the documentation and the captioning to the users of the academic system were executed. Subsequently, it was tested for usability, functionality and efficiency, with the help of surveys directed at the secretary, students and teachers. It was determined that the academic system is 90,48% usable, 98,72% functional, and 92,91% efficient, it was obtained a general average of 94.04% in quality. Finally, it is concluded that the system will help the secretary, students and teachers of the Sindicato de Choferes Profesionales 4 de Octubre to speed up the registration process and also to carry a digital organization of academic information, so it is recommended that the members of the institution make use of the system to facilitate academic processes.

Keywords: <TECNOLOGY AND SCIENCE OF ENGINNERING>, <SOFTWARE ENGINEERING>, <JSF FRAMEWORK>, <SCRUM DEVELOPMENT METODOLOGY>, < APACHE (LOCAL SERVER)>, < (PRIMEFACES) LIBRARY OF COMPONENTS>, < ACADEMIC INFORMATION>, <EVALUATION METRICS>.

INTRODUCCIÓN

El presente trabajo de titulación tiene por objetivo exponer el proceso de análisis, diseño, implementación y evaluación del sistema académico del Sindicato de Choferes Profesionales “4 de Octubre” de la ciudad de Penipe en la provincia de Chimborazo aplicando el Framework JSF, de manera que se pueda demostrar la usabilidad, funcionalidad y eficiencia.

La metodología ágil SCRUM, mediante la cual se realizó el proceso de desarrollo del presente proyecto utilizando los procesos y roles Scrum que estableció una adaptación óptima entre los comprometidos e involucrados para un proceso de desarrollo eficiente, definiendo períodos específicos de trabajo avalando el cumplimiento de los requisitos de usuario del sistema académico. Se presenta una propuesta de software que mejora el proceso de matrículas de la institución además exterioriza la organización digital de la información académica de cualquier institución educativa para mejorar el proceso de matriculación.

El presente entregable contiene tres capítulos; el primer capítulo se refiere a los antecedentes, justificación y objetivos del proyecto además la descripción teórica de la herramienta tecnológica de desarrollo de software utilizada, en el capítulo dos se describe el proceso de desarrollo del sistema académico del Sindicato de Choferes Profesionales 4 de Octubre aplicando el framework JSF, finalmente el capítulo tres expone los resultados generados en la evaluación del sistema académico.

ANTECEDENTES

Las empresas hoy en día tienen la necesidad de estar acorde con las exigencias de la tecnología en especial con aplicaciones web ya que las necesidades de cada uno de ellas así lo ameritan, en los últimos años han surgido diferentes tecnologías para el desarrollo de aplicaciones basadas en la web tanto con licencia y también libres. En cuanto a las herramientas libres han evolucionado significativamente, se destaca a Java como una de las más conocidas para los desarrolladores de aplicaciones web teniendo diferentes ambientes de trabajo que permiten desarrollar de manera moderna y acorde a los requerimientos de la tecnología hoy en día.

Al presente es evidente que en el Sindicato de Choferes Profesionales 4 de Octubre requiere realizar el sistema académico que será gran asistencia tanto para la secretaria, profesores y estudiantes, ya que no existe una senda digital para manejar la información académica en la formación profesional lo cual ha producido mala usabilidad de recursos académicos, por tal razón se busca facilitar el trabajo con la información correspondiente a esta área.

En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto, así como lo definen en Struts 2:

Un framework es un conjunto de bibliotecas, herramientas y normas a seguir que ayudan a desarrollar aplicaciones. Los frameworks los desarrollan los programadores de sistemas. Un framework está compuesto por varios segmentos/componentes que interactúan los unos con los otros. Las aplicaciones pueden escribirse de manera más eficaz si lo utilizamos adaptado al proyecto. Un framework Java proporciona un conjunto de características a partir de una implementación de objeto. En proyectos de desarrollo a gran escala y de diseño en equipo los frameworks son muy útiles, incluso imprescindibles.(Lafosse, 2010, pp. 11–12).

La utilización de un framework en el desarrollo de una aplicación implica un cierto coste inicial de aprendizaje, aunque a largo plazo es probable que facilite tanto el desarrollo como el mantenimiento. Existen multitud de frameworks orientados a diferentes lenguajes y funcionalidades. Aunque la elección de uno de ellos puede ser una tarea complicada, lo más

probable que a largo plazo sólo los mejor definidos (o más utilizados, que no siempre coinciden con los primeros) permanezcan.

Analizado las situaciones presentadas anteriormente para el desarrollo del sistema académico se ha optado por el framework JSF. JavaServer Faces (JSF) es un framework de aplicaciones web basado en Java destinadas a simplificar la integración de desarrollo de interfaces de usuario basadas en web. JavaServer Faces es una tecnología de visualización normalizada que se formalizó en una especificación a través de la comunidad de procesos Java (González, 2008).

Actualmente en el repositorio de datos de la Escuela Superior Politécnica de Chimborazo (Dspace Espoch), podemos encontrar trabajos de tesis relacionados como: Análisis del rendimiento entre framework JavaServer Faces (JSF) y JBOSS-SEAM realizado por Villa Piray en Diciembre del 2014 , donde se comprobó el mayor rendimiento de JSF en relación con el framework Seam con una diferencia de eficiencia del 14.61%, por lo que se optó para el desarrollo de la Aplicación denominado Ficha Económica para el Departamento de Bienestar Politécnico (ESPOCH).

Juntamente podemos constatar el trabajo investigativo que titula: “Análisis de la Tecnología JavaServer Faces como framework de desarrollo de Aplicaciones Web”, que fue desarrollado por Miguel Ángel Baldeón Ordóñez, en junio del 2012, donde se estudió y se analizó la tecnología JavaServer Faces (JSF) para el desarrollo de la aplicación web para procesos de comercialización de productos de la subgerencia de la comercialización de hidrocarburos del Ecuador, en la ciudad de Quito, en el que se pudo mejorar los procesos en línea en un 18%.

En la actualidad según el decreto 1014 con el cual el Software Libre pasa a ser una política de estado para ser adoptado por todos los sistemas educativos y para la gestión de las instancias públicas. La migración paulatina y completa se realizará en un período de cinco años a software libre en todas las agencias públicas(“Decreto_1014_software_libre_Ecuador_c2d0b.pdf,” 2008.).

El auge del desarrollo de Aplicaciones Web Empresariales en ambientes open source es esencialmente por la no existencia de costos de licencia para algún producto, el código abierto tiene mayor flexibilidad, y toda la información. El Framework para implementar aplicaciones web, JSF es de ambiente open source y además es una tecnología sobresaliente que tiene como objetivo principal simplificar el desarrollo de interfaces de usuario utilizando Java.

En el análisis de rendimiento existe la dificultad al momento de elegir un Framework de

desarrollo de software ya que es muy importante su eficiencia al momento del consumo de recursos, es necesario que los tiempos de respuestas de las aplicaciones web sean cortos y de gran velocidad, el rendimiento antes, durante y después del despliegue de la aplicación web, por esta razón se va a seleccionar el Framework JSF (versión 2.2), que consume menos recursos de hardware.

FORMULACIÓN DEL PROBLEMA

¿Se puede mejorar la gestión de información académica del sindicato de choferes profesionales “4 de octubre” a través de implementación del Sistema Académico?

SISTEMATIZACIÓN DEL PROBLEMA

- ✓ ¿Qué Ventajas tiene utilizar el framework JSF?
- ✓ ¿Cuál es la Arquitectura adecuada para el desarrollo del sistema académica del sindicato de choferes profesionales “4 de octubre”?
- ✓ ¿Qué cambios obtendrá el sindicato de choferes profesionales “4 de octubre” al automatizar la gestión de la información académica?
- ✓ ¿Cuáles son las herramientas necesarias para el desarrollo del sistema académico del sindicato de choferes profesionales “4 de octubre”?

JUSTIFICACIÓN

Justificación Teórica

La evolución de tecnologías para desarrollo web es inevitable, definitivamente, estableciendo una competencia entre ellas para sobresaltar ya sea por su optimización de recursos y sobre todo por su rendimiento, ya que este último es la causa principal para el abandono de un sitio web. En la actualidad las empresas tanto públicas como privadas su prioridad de innovación es el desarrollo de aplicaciones web que sean eficientes, que ofrezcan una mayor velocidad de acceso y efectividad.

Por tal motivo, se ha optado por el framework JSF debido a las ventajas y beneficios que sobresalen en comparación a otras tecnologías, ventajas como: el código con el que se crea las vistas es muy parecido al HTML, lo pueden utilizar fácilmente desarrolladores y diseñadores

web, resuelven validaciones, conversiones, mensajes de error e internacionalización además son extensibles, por lo que se pueden desarrollar nuevos componentes a medida.

Definitivamente los beneficios que ofrece son sugestivos ya que permite desarrollar proyectos web de calidad en tiempos muy competitivos y usando metodologías de programación ágiles, es más económico para el que contrata el desarrollo de una aplicación, más económico para la empresa desarrolladora, es sinónimo de Calidad.

Justificación Aplicativa

Con el desarrollo del sistema académico del Sindicato de Choferes Profesionales “4 de Octubre” utilizando el frameworks JSF se planteó mejorar el manejo de la información académica de la institución, referente al proceso de inscripciones y matrículas. El uso de la herramienta tecnológica JSF contribuirá al rendimiento en la aplicación web, donde se ha relacionado módulos para que los usuarios practiquen una interacción distinta e innovadora en contraste a los sistemas ordinarios existentes en la web.

Considerando que en la actualidad está en auge y con el fin de llevar a cabo una solución ágil, el sistema está realizado utilizando la metodología de desarrollo SCRUM, con las siguientes etapas:

- ✓ **Planificación:** Requerimientos, personas y roles, tipos de usuarios, cronograma de actividades, arquitectura del sistema.
- ✓ **Desarrollo:** Realización y ejecución de los Sprints.
- ✓ **Cierre:** Evaluación y pruebas del sistema.(Palacio, 2014)

Para el desarrollo modular se dividirá al sistema en 5 módulos que estará en capacidad de realizar y detallo a continuación:

- ✓ **Autenticación:** Para usuarios del sistema, si se trata de un nuevo estudiante hay el apartado de inscripción en línea, además un link que redireccionara a una interfaz de oferta académica.
- ✓ **Secretaria:** Será el usuario administrador, encargada (o) de gestionar: profesores, estudiantes, cursos, paralelos, asignaturas, horarios, periodos, sección, tipo de licencias.

- ✓ **Matriculación:** Se realizará el ingreso de datos de los estudiantes, previamente si ya realizaron la pre-inscripción en la web, el proceso también podrá realizarlo el usuario administrador. Posteriormente se realizará el asentamiento de matrícula.
- ✓ **Profesor:** Podrá actualizar sus datos personales además visualizar la carga horaria en el respectivo período académico y generará un archivo con su horario
- ✓ **Estudiante:** Podrá actualizar sus datos personales además visualizar generará un archivo con su horario.
- ✓ **Reportes:** Se generará: las listas de estudiantes matriculados, solicitud de inscripción, horarios para profesores y estudiantes.

OBJETIVOS

Objetivo General

Desarrollar el Sistema Académico del Sindicato de Choferes profesionales “4 de Octubre” aplicando el framework JavaServer Faces (JSF).

Objetivos Específicos

- ✓ Estudiar las definiciones, elementos, características y funcionamiento del Frameworks JSF.
- ✓ Diseñar el Sistema Académico que será capaz de; realizar el proceso de matrículas, facilitar la información de la oferta académica incluido para dispositivos móviles.
- ✓ Implementar el Sistema Académico para el proceso de Matriculas.
- ✓ Evaluar el progreso en la gestión de información académica de la Institución.

CAPÍTULO I

1.MARCO TEÓRICO

1.1 JavaServer Faces



Figura 1-1: JavaServer Faces

Fuente: JavaServer Faces-Wikipedia

1.1.1 *Introducción*

Los frameworks en el desarrollo de software se han convertido en más que una alternativa en una necesidad para aquellos que desarrollan software por los beneficios que brindan. Un framework es una asociación de herramientas tecnológicas con bibliotecas y normas que contribuyen al desarrollo de aplicaciones. Son desarrollados por programadores de sistemas que facilita el proceso de construcción de un producto software a los desarrolladores de aplicaciones, por esto está formado por segmentos o componentes que interactúan entre sí. JavaServer Faces (JSF) es un framework web J2EE basado en Java destinadas a simplificar la integración de desarrollo de interfaces de usuario (componentes del lado del servidor), JavaServer Faces es una tecnología de visualización normalizada que se formalizó en una especificación a través de la comunidad de procesos Java. (González, 2008, párras. 1–3)

La herramienta tecnológica JavaServer Faces está basado en el patrón Modelo Vista Controlador (MVC), que utiliza los objetos Javabean como modelos, páginas Facelets como vistas y métodos de esos objetos como controladores (Anónimo, 2012, parr. 16), que ofrece un conjunto de funciones y procedimientos (particularmente en la programación orientada a objetos) conocidas como APIS para tener la interfaz de programación del producto software y

su gestión sea más llevadera en actividades como: controladores, validaciones, además tiene por defecto dos bibliotecas que admiten una comunicación directa con páginas JSP.

En la mayoría de los casos en el proceso de desarrollo de software, se opta por el MVC, patrón de diseño que gracias a su estandarización y facilidad de gestión de recursos en implementaciones de aplicaciones web. El proceso de comunicación dentro de este patrón la inicia el cliente web o usuario mediante solicitudes por medio de las vistas en la interfaz del producto, estas solicitudes son explícitamente llamadas a las funciones o métodos, posteriormente las solicitudes las recibe el controlador quien se encarga de procesar la información llamando al modelo que generalmente están alojadas en las entidades. Se manifiesta en el círculo de desarrollo una clasificación de dos tipos a las aplicaciones basadas en el Modelo Vista Controlador que son:

Primer tipo

Es donde las vistas conocen el camino, es decir, saben que función va a ser invocada en la petición. Regularmente esta función se encuentra cableada en la vista.

Segundo tipo

Es donde el controlador inserta un conjunto de reglas que ubican a las peticiones con las funciones, además controlan el flujo de navegación por la aplicación.

Un ejemplo de aplicaciones de tipo 1 son las que se construyen utilizando JSF o ASP.NET y como ejemplo de tipo 2 sería creadas con Struts, que es una herramienta de soporte para el desarrollo de aplicaciones web bajo el patrón MVC en la plataforma JavaEE. (Munoz, 2012, párrs. 1-8)

1.1.2 Historia

JSF nació en el año 2004, pero no fue hasta 2006 cuando comenzó a ser popular al integrarse con Java EE 5. Actualmente, la última versión disponible de JSF es la versión 2.2, la cual ha cambiado considerablemente desde 2006. JSF difiere en gran medida de los desarrollos con PHP o JSP. Para empezar, se suele separar en dos grupos de archivos, los que contienen la presentación de la página propiamente dicha (xhtml), y los que contienen la lógica de presentación (Java). En los archivos xhtml (*.xhtml) se usan componentes de JSF que gestionan todos los formularios, campos, tablas de datos, así como los estilos y javascript para los mismos.

Mientras tanto, en los archivos java (*.java) configurados, se realizan las peticiones a base de datos, el procesamiento de la información o el registro de acciones, entre otros.

Esta tecnología genera una división física entre la parte puramente estética y la parte puramente lógica que favorece notablemente la legibilidad del código y el mantenimiento del mismo, haciendo su aprendizaje mucho más sencillo. Algunas implementaciones en JSF permite tener utilidades muy atractivas como campos para subir listas de documentos, campos ‘drag&drop’ o tablas expandibles y editables solamente añadiendo un par de líneas de código. Cabe destacar, además, que la mayoría de herramientas de JSF en el mercado son Open Source, por tanto, pueden usarse de manera libre y sin licencia. Por otro lado, tenemos la posibilidad de editarlas y modificarlas a nuestras necesidades. Por estas razones su uso se está incrementando notablemente en los últimos años, dejando en una minoría los desarrollos en tecnologías como ASP.NET que es de pago. (García et al., 2013, pp. 28–29)

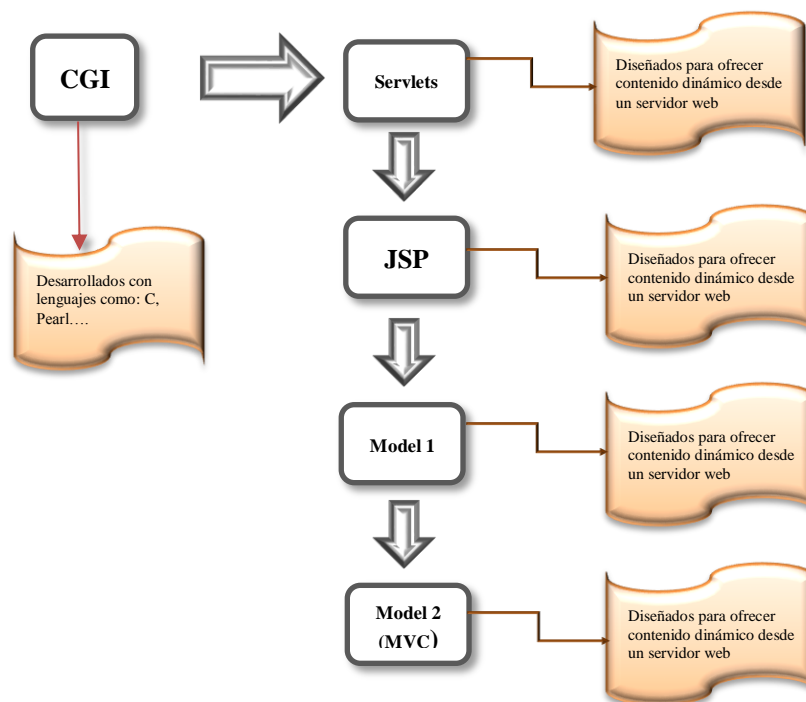


Figura 2-1: Estructura de JavaServer Faces

Realizado por: MERINO Cristian. 2017

1.1.3 Definición

Es una herramienta tecnológica utilizada para la construcción de aplicaciones en la plataforma Java basada en la web, este framework brinda una facilidad relevante en el proceso de desarrollo de interfaces de usuario en aplicaciones JavaEE, hace uso de JavaServer Pages (JSP) para crear páginas web dinámicas, por esto también se puede representar como una analogía de otras

tecnologías como: HTML y XML, es muy similar a PHP (González, 2008, parr. 1). La asociación de APIS para componentes de una interfaz de usuario y también gestionar su estado, manipular eventos, aprobar entrada o validarlas, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

JSF incluye:

- ✓ La asociación de componentes que por defecto vienen integrada para la interfaz de usuario.
- ✓ Las dos bibliotecas de etiquetas personificadas para JavaServer Pages que admiten representar una interfaz JSF dentro de una página JSP.
- ✓ Un modelo de eventos en el lado del servidor.
- ✓ Beans Administrados. (González, 2008, parr. 3)

La especificación de JSF fue desarrollada por la Java Community Process como *JSR 127*, que definía JSF 1.0 y 1.1, *JSR 252* que define JSF 1.2 y *JSR 314* para JSF 2.0. (Anónimo, n.d., parr. 9)

1.1.4 Principales Características de JSF

- ✓ JSF Trae código automático para crear páginas JSP para generar las vistas, agregando la biblioteca de etiquetas propia para generar los elementos de los formularios HTML.
- ✓ Inserta un grupo de etapas en el proceso de la petición, una validación de autenticación, por ejemplo, restablecimiento de la vista, recobro de los valores de los elementos, y más.
- ✓ Relaciona cada vista con formularios un conjunto de objetos java manejados por el controlador (managed beans) que proporcionan la recolección, manejo y visualización de los valores mostrados en los diferentes elementos de los formularios.(Jaramillo, 2013, p. 36)
- ✓ Manipula un fichero de configuración muy sencillo para el controlador, el mismo que está en formato xml.
- ✓ Es prolongable y extensible pudiendo construir de igual modo nuevas características y elementos de la interfaz o actualizar los existentes.
- ✓ Manipular eventos, del lado del servidor validar y convierte datos generando información.
- ✓ Lo más trascendental es que pertenece al estándar J2EE. En efecto, existen una variedad de opciones para construir la presentación y control web en una aplicación de la plataforma java. (Munoz, 2012, parr. 3)

1.1.5 Versiones

La primera versión fue lanzada el 11 de marzo de 2004, desde entonces se ha lanzado cinco versiones más. (De Avila, 2013, p. 3). La **Tabla 1-1** describe la evolución de las versiones de JavaServer Faces (JSF).

Tabla 1-1: Versiones de JSF

Versiones de JavaServer Faces (JSF)			
Versión	Fecha dd/mm/aa	Descripción	Características
JSF 1.0	11/03/2004	Versión Inicial de la especificación JSF. Versión independiente que no forma parte de ninguna versión de Java EE/J2EE	<ul style="list-style-type: none"> ✓ Compatible con Servlet 2.3 y jsp 1.2 ✓ Fue bastante buggy y deficiencias y se mejoró poco después con JSF 1.1
JSF 1.1	27/05/2004	Solucionaba errores. Sin cambios en las especificaciones ni en el rederkit de HTML	<ul style="list-style-type: none"> ✓ Solo corrección de errores. ✓ Incluyen la comprensión de opiniones serializados al cliente. ✓ Soporte de Conmutación por error al almacenar puntos de vista sobre el servidor. ✓ Mejoras generales.
JSF 1.2	11/05/2006	Lanzamiento con mejoras y correcciones de errores a los sistemas centrales y las APL COINCIDENTE CON Java EE5.	<ul style="list-style-type: none"> ✓ Indudable división entre modelo y vista. ✓ Desarrollo en base a componentes en vez de peticiones. ✓ De parte del usuario, las peticiones se adaptan con facilidad al código del lado del servidor. ✓ Construcción de conjuntos de componentes visuales para aligerar el desarrollo. ✓ Brinda muchas posibilidades de opciones entre diversas implementaciones. ✓ Un inconveniente detectado en esta versión es la complejidad en la configuración.

JSF 2.0	12/08/2009	<p>Versión principal para la facilidad de uso. Funcionalidad mejorada y rendimiento. Coincide con Java EE 6.</p> <p>Presenta novedades en cuatro áreas: facilidad de desarrollo, desempeño, adopción y nuevas características.</p>	<ul style="list-style-type: none"> ✓ Facilidad de desarrollo: esta área permitirá la creación de componentes por medio de agregación, reducirá la configuración por medio de descriptores. ✓ Expansión del ciclo de vida del request para brindar soporte a peticiones Ajax. ✓ Desempeño: Se propone rendir deltas de los componentes JSF según se necesite en lugar de un rendimiento completo de los componentes, se mejorará el manejo de eventos mediante PhaseListener y los componentes eran sin estado por defecto. ✓ Se le permite utilizar anotaciones para declarar la navegación.
JSF 2.1	22/10/2010	<p>La versión actual, según versión de mantenimiento de 2.0, cantidad reducida de cambios en especificaciones.</p>	<ul style="list-style-type: none"> ✓ Convierte librerías de etiquetas JSF personalizadas para bibliotecas de etiquetas Facelets.
JSF 2.2	16/04/2013	<p>Propone soporte para HTML5, stateless views, faces flow, y resource, library contrac.</p>	<ul style="list-style-type: none"> ✓ Soporte para plantillas múltiples. ✓ Soporte para declaración de flujos de navegación ✓ Facilitadores para el desarrollo, no será necesaria clase para manipulación de tags personalizadas (tag handler). ✓ Ajuste para bibliotecas Facelets (Abreviando atributos y tag). ✓ Soporte para OSGI (Open Service Gateway Initiative) ✓ Soporte para CDI en toda API ✓ Integración con Portlet. ✓ Bookmarkability ✓ Direccionamiento vía Ajax ✓ Implementación de “component server”

Fuente: Guadalupe y Orlando, 2013 , De Avila, 2013, Jaramillo, 2013

Realizado por: Cristian Merino. 2017

1.1.6 Beneficios

A continuación, se menciona los beneficios relevantes del Framework JSF.

- ✓ Sin duda un considerable beneficio que la herramienta tecnológica JavaServer Faces es que propone un apartamiento entre comportamiento y la presentación. Anteriormente JavaServer Pages (JPS) lograba una separación parcial, en la construcción de aplicaciones web.
- ✓ JavaServer Faces admite crear aplicaciones Web.
- ✓ Provee una arquitectura interesante en la manipulación de componentes, procesamiento de datos, emisión de información, validar entrada del usuario y manejar eventos.
- ✓ Proporciona una ágil y veloz adaptación en desarrolladores nuevos. (Juntadeandalucia, n.d., pp. 1–2)

1.1.7 Arquitectura

Uno de los puntos fuertes hacia los cuales se dirige la arquitectura JSF, definitivamente es la generación de interfaces que puedan ser consumidas por diferentes tipos de dispositivos clientes. Geancarlo Leiva (Leiva, 2014) en su blog de nombre: Sw de Fábrica, por medio de la **Figura 3-1** muestra el diagrama de interacción de dichas interfaces, la mismas que puede verse desde una Pc de Escritorio o una Tablet Pc o un PDA, sin tener que crear nuevas interfaces, ni desarrollar diferentes procesos para cada uno de los ambientes antes mencionados. (Valdiviezo et al., 2012, pp. 80–81).

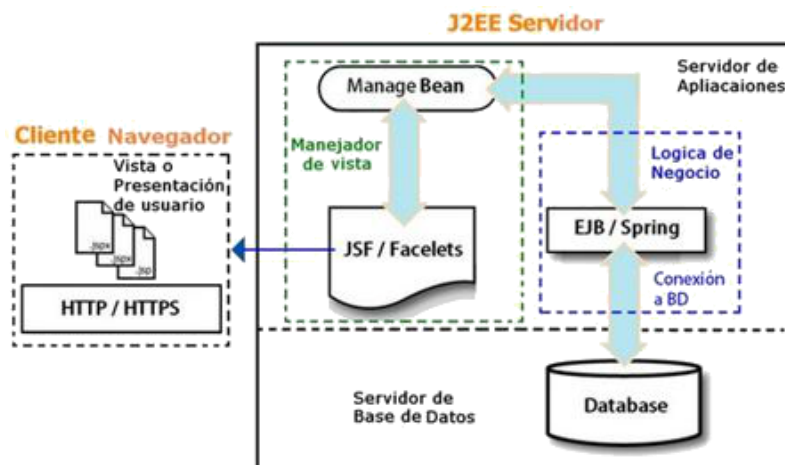


Figura 3-1: Arquitectura JavaServer Faces

Fuente: (Leiva, 2014, parr. 2). La arquitectura JSF

1.1.8 Ciclo de Vida

El ciclo de vida de un sistema o aplicación desarrollado con la herramienta tecnológica JavaServer Faces puntualiza fase a fase todas las acciones aplicadas desde el inicio de comunicación o interacción con el usuario final. En la **Figura 4-1** Jansoblog expone las 6 fases que ejemplifica el ciclo de vida del framework JSF (Jansoblog, 2007, parr. 1).

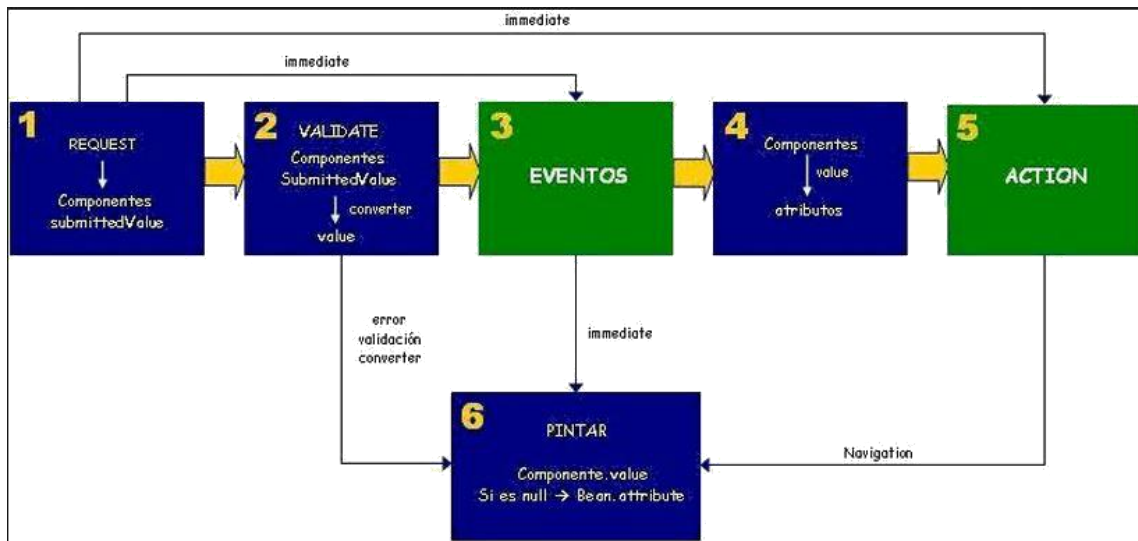


Figura 4-1: Ciclo de vida de JSF

Fuente: (Jansoblog, 2007, parr. 1). Ciclo de Vida en JSF

A continuación, se describe el proceso de inicio con las posibles vías alternativas hasta el final del ciclo de vida del framework JSF:

Primera Fase: Aplicación de los valores de petición

Los componentes del formulario de la interfaz de usuario van recorriendo una por uno hasta consultar todos en orden jerárquico con la filosofía de herencia (padre-hijo) simultáneamente se reciben los valores a partir de parámetros de la petición solicitada al servidor. Al momento de asignar el valor a cada componente lo hace en su propiedad “SubmittedValue” siempre, excepto cuando se presenten don escenarios:

1. El componente esté deshabilitado (disabled = true), ó;
2. El componente tiene una condición que exija su impresión en pantalla (atributo ‘rendered’).
Posteriormente se direccionará a la fase de validaciones o la fase ejecución, esto dependerá del valor que posea el atributo ‘immediate’.

Segunda Fase: Proceso de validaciones

En esta fase tenemos la validación de los datos enviados por el usuario basado en normas de revalidación que son determinadas para el componente. El uso tanto de la validación, así como la de un convertidor es frecuente, el mismo que se aplica a la propiedad SubmittedValue. En consecuencia, si hay ausencia de error en la interacción, SubmittedValue de todos los componentes toma un valor null que se aloja en la propiedad “value” del dato convertido, al contrario, la fase segunda fase finalizará y el ciclo se dirigirá directamente a la fase de presentación de la respuesta.

Tercera Fase: Proceso de eventos

En esta fase JSF realiza los eventos, absolutamente todos con las acciones que haya realizado el usuario.

Cuarta Fase: Actualización de valores del modelo

En esta fase ya cuando haya sido: validada y/o ejecutada los eventos de la fase anterior, se prosigue a la asignación de la propiedad “value” de los componentes situados a los atributos del Javabean.

Quinta fase: Invocar a la aplicación.

En esta fase, tal como muestra la **Figura 4-1** en el recuadro que representa la quinta fase, es la donde se utiliza los valores Javabean para la ejecución de la lógica de la aplicación, y labores que corresponden a las acciones aplicadas de parte del usuario.

Sexta Fase: Renderización de la respuesta

En esta fase para concluir el ciclo de vida de JSF, el framework en autoridad comisiona a JSP para mostrar al usuario la pantalla apropiada, es decir, donde absolutamente todos los campos en función del resultado estén con el valor en ese momento. (Jansoblog, 2007, párras. 1–7)

1.1.9 Acceso a la Base de Datos

La **Figura 5-1** expuesta por Joaquín (Hangalo, n.d., parr. 1) determina el procedimiento de acceso a la Base de datos con el Framework JSF.

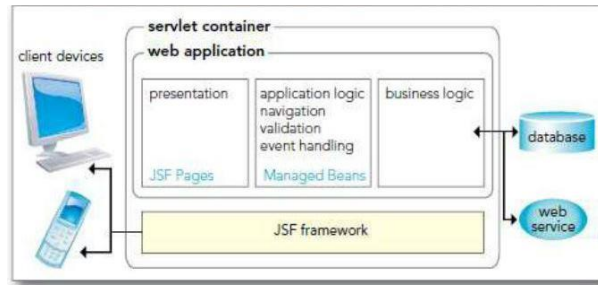


Figura 5-1: Acceso a la Base de Datos con JSF

Fuente: (Hangalo, n.d., parr. 1)

1.2 Metodología de Desarrollo SCRUM

1.2.1 Introducción

El presente proyecto se ha desarrollado con la metodología Scrum. La utilización y aplicación del modelo de procesos SCRUM metodología ágil y flexible, que como característica fundamental es gestionar el desarrollo de software mediante el trabajo en equipo obteniendo mejores resultados de productividad, una Analogía holística o estilo rugby que como metáfora; Scrum todos junto, a la vez. (Hirakata y Ninoxata, 1986). En Scrum un proyecto de desarrollo de software se ejecuta mediante bloques de trabajo temporales cortos o fijos llamados Sprint (cada Sprint dura 2 a 4 semanas). La **Figura 6-1** representa el proceso de la metodología dentro del desarrollo del proyecto software.

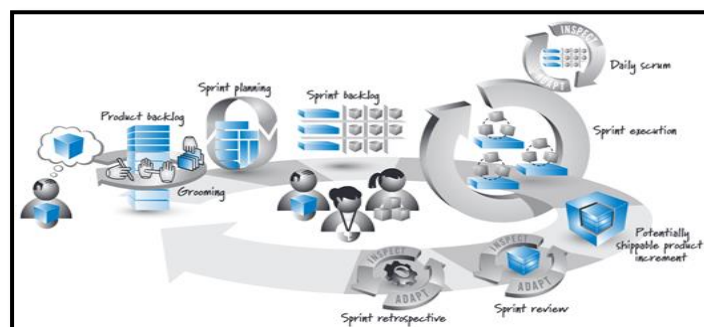


Figura 6-1: Ciclo de vida de SCRUM

Fuente: (Tachoures, 2016, p. 9). Conceptos de Scrum

1.2.2 Los Roles

Los roles se dividen en 2 grupos: cerdos y gallinas, esto surge en el chiste sobre un cerdo y una gallina y su intención de poner un restaurante. Manuel Trigas Gallego representa la interpretación de los roles en la **Figura 7-1** (Trigas, 2012, p. 35).



Figura 7-1: Roles de Scrum

Fuente:(Trigas, 2012, p. 35) Gestión de Proyectos Informáticos Metodología Scrum

En la **Tabla 2-1**, se describe los roles en el proceso de desarrollo Scrum.

Tabla 2-1: Roles en Scrum

GRUPO PERSONA	ROL	DESCRIPCIÓN
LOS CERDOS	ProductOwner	Toma decisiones, escribe ideas, prioriza y las lista.
	ScrumMaster	Encargado de comprobar que el modelo y metodología funciona.
	Equipo de Desarrollado	Grupo de 5-9 personas, todos tienen autoridad y toma de decisiones para conseguir el objetivo.
LAS GALLINAS	Usuarios	Es el receptor final del Producto
	Stakeholders	Es el o los Interesados del producto que les generará un beneficio. Participan en las revisiones del Sprint.
	Managers	Toma las decisiones finales, participa en la selección de objetivos y requisitos funcionales del producto.

Realizado por: Cristian Merino. 2017

1.2.3 Fases de SCRUM

Scrum como metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, así es que se formalizan entre los involucrados el proyecto, estos ciclos que comúnmente se llaman interacciones, en el proceso de desarrollo con Scrum se los conoce con el nombre de “Sprints”. Para comprender mejor el ciclo de desarrollo de Scrum, es preciso conocer y comprender las fases que definen el ciclo de desarrollo ágil (Trigas, 2012, pp. 33-52)

Tabla 3-1: Fases de Scrum

FASES SCRUM	DESCRIPCIÓN	DETALLE
Product Backlog	CONCEPTO	<ul style="list-style-type: none"> ✓ Lista de necesidades del Cliente. ✓ Es elaborado por el Product Owner y las funciones las prioriza dependiendo del nivel de jerarquía que lo clasifique el cliente.
Sprint Backlog	ESPECULACIÓN Y EXPLORACIÓN	<ul style="list-style-type: none"> ✓ Lista de Tareas que se realizan en un Sprint. ✓ El equipo Scrum establece durante la planificación y la duración de cada Sprint.
Sprint Planning Meeting	REVISIÓN	<ul style="list-style-type: none"> ✓ Tiene como finalidad realizar una reunión en la que participará el Product Owner, el Scrum Master y el Equipo Scrum para seleccionar de las listas Backlog del producto. ✓ Antes de comenzar la reunión el Product Owner preparará el Backlog.
Daily SCRUM		<ul style="list-style-type: none"> ✓ Lo realizan los miembros de equipo. ✓ La reunión no podrá consumir más de 15 minutos. ✓ Se responden individualmente tres preguntas: ¿Qué se ha hecho?, ¿Qué se hará hoy?, ¿Qué problema hay para realizarlo?
Sprint Review		<ul style="list-style-type: none"> ✓ Los desarrolladores presentan el producto entregable (Sprint), que han implementado. ✓ El Product Owner analiza y escucha sobre problemas en el proceso, de haberlos. ✓ Máximo 4 horas.
Sprint Retrospective	CIERRE	<ul style="list-style-type: none"> ✓ Aquí el equipo debatirá temas relacionados con el Sprint finalizado. ✓ Se analiza posibles cambios para mejorar el próximo Sprint y que sea más productivo. ✓ Máximo 3 horas.

Realizado por: Cristian Merino. 2017

1.3 Norma de evaluación ISO/IEC 9126-3

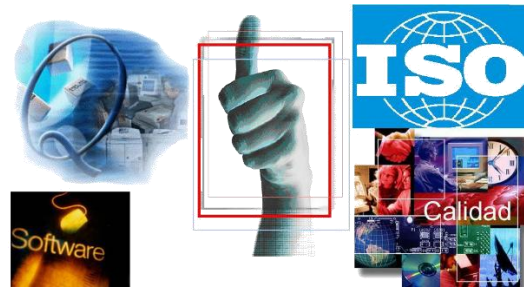


Figura 8-1: Norma de evaluación ISO/IEC 9126-3

Fuente: (Borbón, 2013, parr. 1) Evaluación de Software

Es una norma internacional utilizada para la evaluación de calidad de productos software publicada en el año 1992. Propone seis características generales que son: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad. En el presente proyecto se ha especificado tres: Usabilidad, Funcionalidad y eficiencia debido a que son las métricas internas de calidad relevantes del producto software, vale resaltar que las mencionadas características dividen un conjunto de subcaracterísticas cada una. Así mismo las características: Confiabilidad, Mantenibilidad y Portabilidad con sus subcaracterísticas correspondientes, se manifiestan de manera externa, es decir cuando producto software que se está evaluando va a formar parte de un sistema informático. En la **Figura 9-1** se asocian las características. (García, 2010),(Mena, 2006), (Estayno et al., 2009), (Borbón, 2013), (Lacalle, 2008)

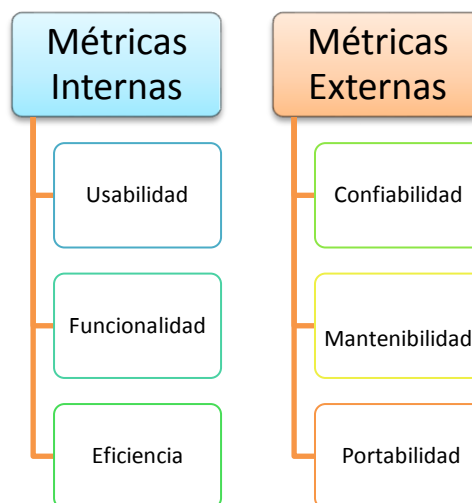


Figura 9-1: Métricas de evaluación de calidad de software ISO/IEC 9126-3

Realizado por: MERINO Cristian. 2017

1.3.1 Usabilidad

Esta característica fue evaluada para medir en porcentaje (%) la capacidad que tiene el sistema académico de ser entendido, aprendido y usado por el usuario final. “como la gente usa el producto software”. A continuación, se explica las subcaracterísticas de la usabilidad.

Entendimiento: Es la capacidad que tiene el producto software para permitir entender al usuario su manipulación de manera sencilla y rápida acatando sus condiciones.

Aprendizaje: Es la capacidad que tiene el producto software para permitir al usuario aprender su usabilidad.

Operabilidad: Es la capacidad que tiene el producto software para permitir al usuario su manipulación y control.

Atracción: Es la capacidad que tiene el producto software de ser atractiva al usuario, es decir, las cualidades relevantes que agradan al frontend, el diseño gráfico es importante.

Conformidad de la usabilidad: Es la capacidad que tiene el producto software para cumplir con los estándares de la usabilidad.

1.3.2 Funcionalidad

Esta característica fue evaluada para medir en porcentaje (%) la capacidad que tiene el sistema académico para realizar los procesos especificados en el proyecto y proveer los resultados esperados. “lo que es capaz de hacer el producto software”. A continuación, se explica las subcaracterísticas de la funcionalidad.

Adecuación: Es la capacidad que tiene el producto software para cumplir con las funciones de manera adecuada el conjunto tareas o requisitos especificados por el usuario.

Exactitud: Es la capacidad que tiene el producto software para realizar procesos y emitir respuestas precisas a peticiones realizadas.

Interoperabilidad: Es la capacidad que tiene el producto software para interactuar con otros sistemas.

Seguridad: Es la capacidad que tiene el producto software para salvaguardar datos e información que contiene o genere el producto software, restringiendo acceso a usuarios o sistemas no autorizados que puedan afectar la integridad de sus datos.

Conformidad de la funcionalidad: Es la capacidad que tiene el producto software para cumplir con los estándares de la funcionalidad.

1.3.3 Eficiencia

Esta característica fue evaluada para medir en porcentaje (%) el desempeño que tiene el sistema académico al utilizar recursos necesarios bajo condiciones establecidas. “lo que consume el producto software”. A continuación, se explica las subcaracterísticas de la eficiencia.

Comportamiento de tiempos: Es la capacidad que tiene el producto software para proporcionar tiempos apropiados de petición, proceso y respuesta.

Utilización de recursos: Es la capacidad que tiene el producto software para hacer uso de recursos de cualquier tipo adecuados, cuando el producto realiza su función sujeta a condiciones establecidas.

Conformidad de la eficiencia: Es la capacidad que tiene el producto software para cumplir con los estándares de la eficiencia.

CAPÍTULO II

2 . MARCO METODOLÓGICO

2.1 Introducción

El presente capítulo, se expone el desarrollo del Sistema Académico del Sindicato de Choferes profesionales 4 de Octubre, para el cual se aplicó la metodología de desarrollo ágil SCRUM, para la planificación, descripción de cada una de las actividades directamente relacionado en el proceso, estimaciones y esfuerzo en el desarrollo del Sistema Académico para las necesidades y requerimientos de parte de los stakeholders representado por el Product Owner, es decir, el normal desarrollo del ciclo de vida de SCRUM, mediante el cumplimiento oportuno para cada uno de los requerimientos simbolizando en los SPRINTS del proyecto.

2.2 Aplicación del modelo ágil de desarrollo SCRUM en el proyecto

SCRUM como metodología de desarrollo del presente proyecto realiza entregables parciales y regulares del producto final, las cuales son priorizadas de parte de los interesados según el beneficio que aportan, de manera que, resuelve situaciones en caso de que no se esté entregando al cliente lo que necesita, sin embargo, cuando las entregas se alargan demasiado, los recursos especialmente económicos se dispara, por lo que la calidad disminuye y optar por tornarse en un estado que no es aceptable, es necesaria la capacidad de reacción ante la competencia, cuando el ánimo y armonía del equipo de desarrollo es baja y los pendientes altos, es necesario identificar y corregir ineficiencias sistemáticamente en el desarrollo del producto. (Colla, 2012, pp. 1–3)

2.3 Planificación del proyecto

2.3.1 *Personas y Roles del Proyecto*

Para la construcción del proyecto de software, hubo la intervención de 3 personas que interactuaron durante el desarrollo del sistema Académico que se nombran a continuación:

Tabla 1-2: Personas y roles del proyecto

PERSONA	ROL
Ing. Jesica Pilco	ProductOwner
Dr. Julio Santillán	ScrumMaster
Sr. Cristian Merino	Desarrollador
Usuarios del Sistema Académico del Sindicato de Choferes Profesionales “4 de Octubre”	Profesor Estudiante

Realizado por: Cristian Merino. 2017

2.3.2 Tipos de usuarios y roles del sistema académico

Aquellas personas que están involucradas directa o indirectamente con el sistema reciben el nombre de usuarios, a continuación, en la **Tabla 2-2** de detalla el tipo, el rol que desempeña en el sistema una breve descripción de usuario y la responsabilidad que se otorga.

Tabla 2-2: Tipos de usuarios del sistema académico

TIPO DE USUARIO	ROL	DESCRIPCIÓN	RESPONSABILIDAD
Administrador	Secretaria	Será quien administre el sistema académico	<ul style="list-style-type: none"> ✓ Administrar funcionalmente el sistema. ✓ Administrar el proceso de inscripción y matriculas. ✓ Administrar profesores, estudiantes, períodos, sección, asignaturas, cursos, paralelos, horarios, nómina, requisitos, reportes, actas y licencias.
Usuario Profesor	Profesor	Será quien actualice calificaciones de los estudiantes	<ul style="list-style-type: none"> ✓ Actualización de datos personales. ✓ Actualización de calificaciones. ✓ Ingreso y actualización de calificaciones.
Usuario Estudiante	Estudiante	Será quien pueda Visualizar sus calificaciones	<ul style="list-style-type: none"> ✓ Actualización de datos personales. ✓ Visualización de calificaciones.

Realizado por: Cristian Merino. 2017

2.3.3 Actividades realizadas en el proyecto

Las actividades realizadas antes, durante y después del proceso de desarrollo del sistema se describe a continuación en la **Tabla 3-2**.

Tabla 3-1: Actividades realizadas en el proyecto

PROYECTO	ACTIVIDADES	HORAS	RESPONSABLE
Desarrollo del Sistema Académico	Recolección de información, requerimientos de usuario.	40	Cristian Merino
	Diseño preliminar del sistema.	120	Cristian Merino
	Diseño de la arquitectura del sistema.		Cristian Merino
	Diseño de la base de datos.		Cristian Merino
	Diseño detallado del lenguaje de modelado unificado (UML).	120	Cristian Merino
	Diseño de interfaces del sistema.	30	Cristian Merino
	Desarrollo, pruebas de aceptación y corrección de errores.	440	Cristian Merino
	Implementación del sistema en un servidor local.	120	Cristian Merino
	Evaluación del sistema y exposición de resultados obtenidos.	120	Cristian Merino
	Documentación del sistema.	60	Cristian Merino
	Capacitación al personal.	30	Cristian Merino
Total		1080	

Realizado por: Cristian Merino. 2017

2.3.4 Requisitos del proyecto

2.3.4.1 Product Backlog

El proceso parte de la lista de tareas o lo que es lo mismo las historias de usuario, que es el conjunto de requisitos funcionales que son definidos por el equipo SCRUM para las actividades a desarrollar en el proceso. Prioriza los requisitos basandose en objetivos, coste , etc., de esta manera el cliente puede maximizar la utilidad de lo que se desarrolla. Se detalla con el siguiente formato: *Como <tipo de usuario (quien)> quiero y/o necesito <objetivo (el qué)> con la finalidad de <poder hacer algo (el por qué)>*.

La priorización de requerimientos se han concretado considerando el criterio de importancia para el desenvolvimiento del sistema académico en acuerdo y aceptación con el Product Owner, Detalle de la Información. (Tabla 4-2).

Tabla 4-2: Niveles de prioridad de los requisitos del proyecto

Criterio de Prioridad	Valor de Prioridad
Alto	1
Medio	2
Bajo	3

Realizado por: Cristian Merino. 2017

2.3.4.2 Unidades de medición del esfuerzo

$$\text{Unidad de medicion} = \frac{\text{Puntos de Historia}}{\text{Horas}} \quad \begin{matrix} \text{(Historias)} \\ \text{(Tareas)*} \end{matrix}$$

Tabla 5-2: Unidades de medición de prioridad

Fibonacci Tradicional	0,1,1,2,3,5,8,13,21,34,55,89,144...
------------------------------	-------------------------------------

Realizado por: Cristian Merino. 2017

La estimación la realiza el equipo SCRUM con la técnica de Planning Poker (Figura 1-2). Se optó por el método fibonacci tradicional con el objetivo de particularizar cada estimación con una carta para cada una de ellas, el detalle de la estimación se detalla en la Tabla 6-2.



Figura 1-2: Planning Póker

Realizado por: Cristian Merino. 2017

Tabla 6-2: Asignación estimación- carta

Estimación	Carta
5	0
10	½
15	1
30	2
40	3
50	5
60	8

Realizado por: Cristian Merino. 2017

La definición y captura de requerimientos para el proyecto software concurren con el PRODUCT OWNER y SCRUM MASTER, mediante reuniones diarias una semana previa al inicio del proyecto, en el que se definieron la funcionalidad del sistema académico, para lograr esto se establecieron sesenta y cinco requerimientos, se fijan ocho requerimientos técnicos y cincuenta y siete requerimientos funcionales los mismos que están definidos a continuación en la **Tabla 7-2**, la cual consta de un identificador de la historia, una descripción, el esfuerzo estimado, el esfuerzo real y la prioridad establecida con base a la **Tabla 4-2**.

Tabla 7-2: Product Backlog

HISTORIAS TÉCNICAS				
ID	DESCRIPCIÓN	ESFUERZO		Prioridad
		Estimado	Real	
HT-01	Como desarrollador del proyecto de software, necesito receptar, capturar y analizar los requisitos del sistema académico.	40	40	1
HT-02	Como desarrollador del proyecto de software, necesito diseñar la base de datos del sistema académico.	30	30	1
HT-03	Como desarrollador del proyecto de software, necesito diseñar la arquitectura del sistema académico.	15	15	1
HT-04	Como desarrollador del proyecto de software, necesito diseñar la interfaz de usuario del sistema académico.	30	30	1
HT-05	Como desarrollador del proyecto de software, necesito implementar las herramientas tecnologías requeridas para el proceso de desarrollo del sistema académico.	5	5	1
HT-06	Como desarrollador del proyecto de software, es necesario implementar el sistema académico en equipos de la institución	30	56	3

	académica.			
HT-07	Como desarrollador del proyecto de software, se requiere elaborar los manuales de usuario y técnico.	60	64	3
HT-08	Como desarrollador del proyecto de software, es necesario capacitar a los usuarios del sistema académico.	30	30	3
HISTORIAS DE USUARIO				
ID	DESCRIPCIÓN	ESFUERZO		Prioridad
		Estimado	Real	
HU-01	Como cliente del proyecto software, necesita un sistema académico de acceso con autenticación.	10	8	1
HU-02	Como cliente del proyecto software, necesita visualizar un ambiente agradable, que muestre en pantalla de autenticación.	10	8	1
HU-03	Como cliente del proyecto software, necesita que los cibernautas visualicen la oferta académica.	5	5	1
HU-04	Como cliente del proyecto software, necesita que estudiantes nuevos realicen su inscripción en la web.	10	8	1
HU-05	Como cliente del proyecto software, necesita un proceso para crear períodos académicos.	10	8	1
HU-06	Como cliente del proyecto software, necesita un proceso para modificar períodos académicos.	5	5	1
HU-07	Como cliente del proyecto software, necesita un proceso para eliminar períodos académicos.	5	5	1
HU-08	Como cliente del proyecto software, necesita un proceso para crear requisitos académicos.	10	5	1
HU-09	Como cliente del proyecto software, necesita un proceso para modificar requisitos académicos.	5	8	1
HU-10	Como cliente del proyecto software, necesita un proceso para eliminar requisitos académicos.	5	5	1
HU-11	Como cliente del proyecto software, necesita un proceso para crear requisitos académicos.	10	8	1
HU-12	Como cliente del proyecto software, necesita un proceso para crear un Profesor.	10	5	1
HU-13	Como cliente del proyecto software, necesita un proceso para modificar un Profesor.	5	5	1
HU-14	Como cliente del proyecto software, necesita un proceso para eliminar un Profesor.	5	7	1
HU-15	Como cliente del proyecto software, necesita un proceso para crear un estudiante.	10	5	1
HU-16	Como cliente del proyecto software, necesita un proceso para modificar un estudiante.	5	5	1
HU-17	Como cliente del proyecto software, necesita un proceso para eliminar un estudiante.	5	8	1
HU-18	Como cliente del proyecto software, necesita un proceso para	10	10	1

	crear un estudiante.			
HU-19	Como cliente del proyecto software, necesita un proceso para matricular a un estudiante.	40	40	1
HU-20	Como cliente del proyecto software, necesita un proceso para asentar la matrícula de un estudiante.	30	35	1
HU-21	Como cliente del proyecto software, necesita un proceso para crear un curso.	10	15	1
HU-22	Como cliente del proyecto software, necesita un proceso para modificar un curso.	5	10	1
HU-23	Como cliente del proyecto software, necesita un proceso para eliminar un curso.	5	5	1
HU-24	Como cliente del proyecto software, necesita un proceso para crear un paralelo.	10	10	1
HU-25	Como cliente del proyecto software, necesita un proceso para crear un paralelo.	5	7	1
HU-26	Como cliente del proyecto software, necesita un proceso para modificar un paralelo.	5	5	1
HU-27	Como cliente del proyecto software, necesita un proceso para eliminar un paralelo.	5	5	1
HU-28	Como cliente del proyecto software, necesita un proceso para crear una asignatura.	10	8	1
HU-29	Como cliente del proyecto software, necesita un proceso para modificar una asignatura.	5	6	1
HU-30	Como cliente del proyecto software, necesita un proceso para eliminar una asignatura.	5	5	1
HU-31	Como cliente del proyecto software, necesita un proceso para crear un horario de clase.	30	24	1
HU-32	Como cliente del proyecto software, necesita un proceso para modificar un horario de clase.	5	5	1
HU-33	Como cliente del proyecto software, necesita un proceso para eliminar un horario de clase.	5	5	1
HU-34	Como cliente del proyecto software, necesita un proceso para crear un periodo académico.	10	8	1
HU-35	Como cliente del proyecto software, necesita un proceso para modificar un periodo académico.	5	5	1
HU-36	Como cliente del proyecto software, necesita un proceso para eliminar un periodo académico.	5	8	1
HU-37	Como cliente del proyecto software, necesita un proceso para crear una sección académica.	10	8	1
HU-38	Como cliente del proyecto software, necesita un proceso para modificar una sección académica.	5	5	1
HU-39	Como cliente del proyecto software, necesita un proceso para eliminar una sección académica.	5	8	1
HU-40	Como cliente del proyecto software, necesita un proceso para	10	5	1

	crear un tipo de licencia.			
HU-41	Como cliente del proyecto software, necesita un proceso para modificar un tipo de licencia.	10	15	1
HU-42	Como cliente del proyecto software, necesita un proceso para eliminar un tipo de licencia.	10	15	1
HU-43	Como cliente del proyecto software, necesita un proceso para emitir un listado de los estudiantes registrados.	40	60	2
HU-44	Como cliente del proyecto software, necesita un proceso para emitir un listado de los Profesor registrados.	30	40	2
HU-45	Como cliente del proyecto software, necesita un proceso para emitir un listado de los estudiantes registrados.	30	40	2
HU-46	Como cliente del proyecto software, necesita un proceso para que un profesor registrado actualice su información personal.	15	4	2
HU-47	Como cliente del proyecto software, necesita un proceso para que un profesor pueda visualizar y descargar un archivo con su materia (s) asignada (s).	40	24	2
HU-48	Como cliente del proyecto software, necesita un proceso para que un profesor pueda visualizar y descargar un archivo con su horario asignado.	40	24	2
HU-49	Como cliente del proyecto software, necesita un proceso para que un estudiante registrado actualice su información personal.	15	10	2
HU-50	Como cliente del proyecto software, necesita un proceso para que un estudiante pueda visualizar y descargar un archivo con horario asignado.	10	8	2
HU-51	Como cliente del proyecto software, necesita un proceso para que un estudiante pueda visualizar y descargar un archivo con sus calificaciones.	40	24	2
HU-52	Como cliente del proyecto software, necesita un proceso para asignar un estudiante a un curso, paralelo, periodo y sección académico.	40	36	2
HU-53	Como cliente del proyecto software, necesita un proceso para asignar un profesor a un curso, paralelo, periodo, asignatura y sección académico.	40	40	2
HU-54	Como cliente del proyecto software, necesita un proceso para ingresar calificaciones de los estudiantes.	40	40	2
HU-55	Como cliente del proyecto software, necesita un proceso para modificar calificaciones de los estudiantes.	40	40	2
HU-56	Como cliente del proyecto software, necesita un proceso para visualizar e imprimir el récord académico de los estudiantes.	40	40	2
TOTAL		1080	1080	

Realizado por: Cristian Merino. 2017

2.3.4.3 Sprint backlog

El proyecto de software posee una duración de 27 semanas, fecha de inicio 01/08/2016 y fecha de finalización 03/02/2017, presentada en SPRINTS organizados y detallados a continuación en la **Tabla 8-2**. Cada sprint está descrito por su historia de usuario, su correspondiente fecha de inicio, fecha de finalización, el esfuerzo empleado, además el esfuerzo del Sprint correspondiente y el responsable. Para cada punto de esfuerzo representado en la planificación de los sprints descritos en la **Tabla 7-2**, cada SPRINT tiene una correspondencia de una hora en la labor empleada.

Tabla 8-2: Sprint Backlog

Sprint	Historia	Fecha		Esfuerzo (HT Y HU)	Esfuerzo (SPRINT)	Responsable
		Inicio	Fin			
1	HT – 01	01/08/2016	05/08/2016	40	120	Cristian Merino
	HT – 02	08/08/2016	11/08/2016	30		Cristian Merino
	HT – 03	11/08/2016	15/08/2016	15		Cristian Merino
	HT – 04	15/08/2016	19/08/2016	30		Cristian Merino
	HT – 05	19/08/2016	19/08/2016	5		Cristian Merino
2	HU – 01	22/08/2016	23/08/2016	10	120	Cristian Merino
	HU – 02	23/08/2016	24/08/2016	10		Cristian Merino
	HU – 03	24/08/2016	25/08/2016	5		Cristian Merino
	HU – 04	25/08/2016	26/08/2016	10		Cristian Merino
	HU – 05	26/08/2016	29/08/2016	10		Cristian Merino
	HU – 06	29/08/2016	29/08/2016	5		Cristian Merino
	HU – 07	30/08/2016	30/08/2016	5		Cristian Merino
	HU – 08	30/08/2016	31/08/2016	10		Cristian Merino

	HU – 09	31/09/2016	01/09/2016	5		Cristian Merino
	HU – 10	01/09/2016	02/09/2016	5		Cristian Merino
	HU – 11	02/09/2016	05/09/2016	10		Cristian Merino
	HU – 12	05/09/2016	06/09/2016	10		Cristian Merino
	HU – 13	06/09/2016	06/09/2016	5		Cristian Merino
	HU – 14	07/09/2016	07/09/2016	5		Cristian Merino
	HU – 15	07/09/2016	08/09/2016	10		Cristian Merino
	HU – 16	08/09/2016	09/09/2016	5		Cristian Merino
3	HU – 17	12/09/2016	12/09/2016	5	120	Cristian Merino
	HU – 18	12/09/2016	13/09/2016	10		Cristian Merino
	HU – 19	14/09/2016	20/09/2016	140		Cristian Merino
	HU – 20	21/09/2016	26/09/2016	30		Cristian Merino
	HU – 21	26/09/2016	27/09/2016	10		Cristian Merino
	HU – 22	28/09/2016	28/09/2016	5		Cristian Merino
	HU – 23	28/09/2016	29/09/2016	5		Cristian Merino
	HU – 24	29/09/2016	30/09/2016	10		Cristian Merino
	HU – 25	30/09/2016	30/09/2016	5		Cristian Merino
4	HU – 26	03/10/2016	03/10/2016	5	120	Cristian Merino
	HU – 27	03/10/2016	04/10/2016	5		Cristian Merino
	HU – 28	04/10/2016	05/10/2016	10		Cristian Merino
	HU – 29	05/10/2016	06/10/2016	5		Cristian Merino
	HU – 30	06/10/2016	06/10/2016	5		Cristian Merino
	HU – 31	06/10/2016	11/10/2016	30		Cristian Merino
	HU – 32	11/10/2016	12/10/2016	5		Cristian Merino

	HU – 33	12/10/2016	13/10/2016	5		Cristian Merino
	HU – 34	13/10/2016	14/10/2016	10		Cristian Merino
	HU – 35	14/10/2016	17/10/2016	5		Cristian Merino
	HU – 36	17/10/2016	17/10/2016	5		Cristian Merino
	HU – 37	18/10/2016	19/10/2016	10		Cristian Merino
	HU – 38	19/10/2016	19/10/2016	5		Cristian Merino
	HU – 39	19/10/2016	20/10/2016	5		Cristian Merino
	HU-40	20/10/2016	21/10/2016	10		Cristian Merino
5	HU – 41	24/10/2016	25/10/2016	10	120	Cristian Merino
	HU – 42	25/10/2016	26/10/2016	10		Cristian Merino
	HU – 43	26/10/2016	02/11/2016	40		Cristian Merino
	HU – 44	02/11/2016	08/11/2016	30		Cristian Merino
	HU – 45	08/11/2016	11/11/2016	30		Cristian Merino
6	HU – 46	14/11/2016	15/11/2016	15	120	Cristian Merino
	HU – 47	15/11/2016	22/11/2016	40		Cristian Merino
	HU – 48	22/11/2016	29/11/2016	140		Cristian Merino
	HU – 49	29/11/2016	01/12/2016	15		Cristian Merino
	HU - 50	01/12/2016	02/12/2016	10		Cristian Merino
7	HU – 51	05/12/2016	09/12/2016	40	120	Cristian Merino
	HU – 52	12/12/2016	16/12/2016	40		Cristian Merino
	HU – 53	19/12/2016	23/12/2016	40		Cristian Merino
8	HU – 54	26/12/2016	30/12/2016	40	120	Cristian Merino
	HU – 55	02/01/2017	06/01/2017	40		Cristian Merino
	HU – 56	09/01/2017	13/01/2017	40		Cristian Merino

9	HT - 06	16/01/2017	19/01/2017	30	120	Cristian Merino
	HT - 07	19/01/2017	31/01/2017	60		Cristian Merino
	HT - 08	31/01/2017	03/02/2017	30		Cristian Merino

Realizado por: Cristian Merino. 2017

2.4 Cronograma de actividades

En este punto se expone el cumplimiento de los sprints que se planificaron para el desarrollo del proyecto software se ha determinado un cronograma de actividades con una duración de 27 semanas con fecha de inicio el 01/08/2016 y fecha de finalización 03/02/2017, trabajando conforme con la jornada laboral en el Ecuador que es de 8 horas diarias de lunes a viernes, incluidos días feriados. Como se puede evidenciar en la **Tabla 9-2**.

Tabla 9-2: Cronograma de actividades

No	Actividad	Fecha		Responsable
		Inicio	Fin	
01	Recolección de información, requerimientos de usuario.	01/08/2016	05/08/2016	Cristian Merino
02	Diseño preliminar del sistema.	08/08/2016	12/08/2016	Cristian Merino
03	Diseño de la arquitectura del sistema.	15/08/2016	19/08/2016	Cristian Merino
04	Diseño de la base de datos.	22/08/2016	26/08/2016	Cristian Merino
05	Diseño detallado del lenguaje de modelado unificado (UML).	29/08/2016	02/09/2016	Cristian Merino
06	Diseño de interfaces del sistema.	05/09/2016	08/09/2016	Cristian Merino
07	Desarrollo, pruebas de aceptación y corrección de errores.	09/09/2016	24/11/2016	Cristian Merino
08	Implementación del sistema en un servidor local.	24/11/2016	15/12/2016	Cristian Merino
09	Evaluación del sistema y exposición de resultados obtenidos.	15/12/2016	05/01/2017	Cristian Merino
10	Documentación del sistema.	05/01/2017	17/01/2017	Cristian Merino
11	Capacitación al personal.	17/01/2017	03/02/2017	Cristian Merino

Realizado por: Cristian Merino. 2017

2.5 Desarrollo del proyecto

En este ítem del presente documento se expone de manera detallada todos los procesos realizados para el **DESARROLLO DEL SISTEMA ACADÉMICO DEL SINDICATO DE CHOFERES PROFESIONALES 4 DE OCTUBRE APLICANDO EL FRAMEWORK JSF**.

2.5.1 Estandar de codificación

El estándar de codificación o redacción de código de programación en un proceso de desarrollo de software es una forma de “normalizar” el proceso de construcción del producto software en la parte de codificación de manera que en dicho proyecto el team scrum tengan acceso y comprenda el código y que le (s) permitirá:

- ✓ Precisar la nomenclatura, escritura y organización del código fuente.
- ✓ Facilitar la actualización del código fuente.
- ✓ Definir la nomenclatura de las variables, clases, objetos, métodos, los comentarios y elementos que el desarrollador crea necesarios.
- ✓ Especificar los datos a incluirse, que sea de fácil comprensión a otro (s) programador (a) el código fuente. desarrollador (William, 2010, parr. 2)

2.5.2 Modelado del Sistema UML

En cuanto al modelado del sistema académico se aplicó el lenguaje unificado de modelado UML, debido a que se trata de un lenguaje gráfico para visualizar, especificar y documentar un sistema independiente de los métodos de análisis y diseño. Asimismo, brinda soporte a cualquier metodología de desarrollo. En el **Anexo F** se detalla los diagramas, los cuales muestran los diferentes aspectos de las entidades representadas.

2.5.3 Descripción de las capas del sistema académico

El sistema académico está desarrollado a base de el modelo vista controlador (MVC) que está basada en una estructura de tres capas y se constituye en la aproximación de origen del framework JavaServer Faces (JSF).

1. **Modelo.-** Utiliza los objetos Javabean como modelos, procesos de conexión a la base de datos y además es donde se ejecutan las sentencias o código.
2. **Vista.-** Utiliza páginas Facelets como vista, que es donde se encuentran las interfaces del sistema.
3. **Controlador.-** Utiliza los métodos de los objetos como controladores, donde se ejecutan sentencias y validaciones de procesos como: autenticación, cargar archivos, etc., restricción de accesos, seguridad y mantener la integridad de datos del sistema.

2.5.4 Diseño de la base de datos

En el diseño de la base de datos del sistema académico previo al análisis de los requisitos funcionales del sistema y la planificación basados en las historias de usuario se procedió a realizar una distribución estructurada de entidades que constituye la lógica del sistema académico, la flexibilidad y crecimientos de un sistema radica en el diseño objetivo de la base de datos permite adecuarse al ambiente o entorno donde lo vaya a desarrollar, se menciona objetivo priorizando la garantía de que los datos sean manejados correctamente y así evitar: redundancia de datos, inconsistencia de datos, normalización inadecuada. (Concha and Tiupul, 2014, p. 136)

En la **Figura 2-2** se muestra el esquema de la base de datos que detalla el diseño y las relaciones establecidas entre las entidades que comprenden las tablas del almacén de datos para el sistema académico. Además, expone el proceso de matriculación que es el requisito prioritario del proyecto, también se puede observar la asignación de entidades que obtendrá el proceso.

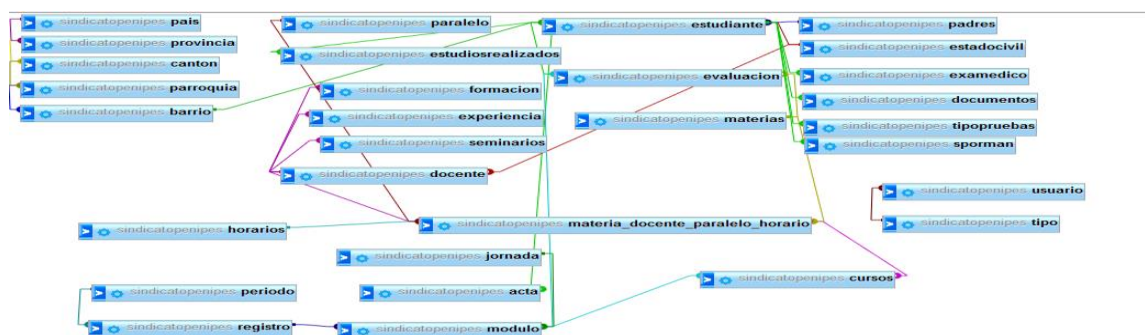


Figura 2-2: Esquema de la base de datos

Realizado por: Cristian Merino. 2017

En la **Figura 3-2** se muestra el diagrama modelo entidad relación, se enfatiza y prioriza el proceso principal del proyecto que es la matriculación. Siendo así que se ha normalizado la base de datos obteniendo de este modo el diagrama lógico de la base de datos que evidencia más detalladamente los objetos, clases, relaciones y demás características de cada entidad, se ha obtenido un total de 37 tablas, 27 tablas relacionadas y normalizadas y 10 tablas normalizadas. Todas las tablas se pueden visualizar en el **Anexo A** (Modelo Físico).

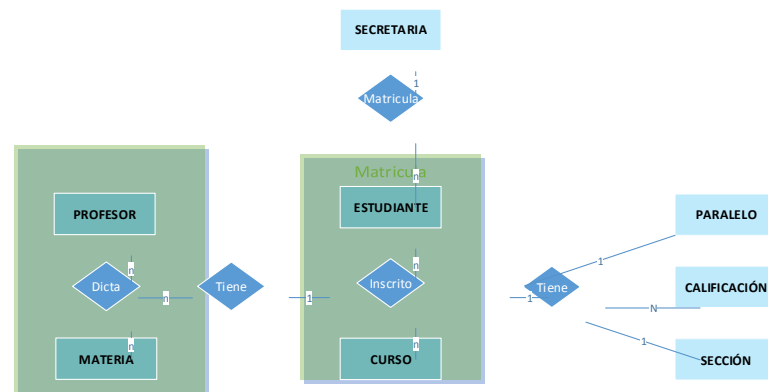


Figura 3-2: Diagrama entidad relación

Realizado por: Cristian Merino. 2017

2.5.5 Diccionario de datos

Es el catálogo donde se aloja la estructura lógica de los datos que interactúan entre ellos y reflejar al frontend los procesos requeridos de la funcionalidad para los usuarios puedan interpretarlos. El diccionario de datos proporciona, además, asistencia para asegurar la existencia y la fácil localización de los flujos de datos del sistema. En la **Tabla 10-2** se ilustra un ejemplo con la entidad estudiante donde se describe el meta data de la tabla estudiante, las demás tablas se han adjuntado al **Anexo B** (Manual Técnico).

Tabla 10-2: Diccionario de datos

TABLA ESTUDIANTE			
Atributo	Tipo de dato	Descripción	Nulo
Cod_Estudiante	Entero	LLAVE PRIMARIA	No
Ci	Varchar (10)	HASTA 10 CARACTERES/UNICODE	No
Apellido	Varchar (50)	HASTA 50 CARACTERES/UNICODE	No
Nombre	Varchar (50)	HASTA 50 CARACTERES/UNICODE	No
Dirección	Varchar (50)	HASTA 50 CARACTERES/UNICODE	Si

Teléfono	Varchar (10)		Si
celular	Varchar (10)		Si
Mail	Varchar (100)	HASTA 100 CARACTERES/UNICODE	Si
Clave	Char (32)	HASTA 32 CARACTERES/UNICODE	No
Código	Entero		No
Fecha_nacimiento	Date		Si
Sexo	Tinyint (1)	M/F	Si
edad	Int (11)		No
Estado civil	Varchar (15)	HASTA 15 CARACTERES/UNICODE	Si
Num_papeleta	Int (10)		Si
foto	Varchar (200)	HASTA 50 CARACTERES/UNICODE	Si
nac_parroquia	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
nac_canton	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
nac_provincia	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
nac_ciudad	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
dom_ciudad	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
dom_parroquia	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
dom_barrio	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si
dom_calle	Varchar (20)	HASTA 20 CARACTERES/UNICODE	Si

Realizado por: Cristian Merino. 2017

2.5.6 Diseño de interfaces

Para el diseño de las interfaces se realizó un esquema en la autenticación y una plantilla estandar para las interfaces de los usuarios del sistema académico, donde varía los módulos dependiendo logicamente de la funcionalidad, procedimientos y actividades que realiza cada uno. En la **Tabla 11-2** se detalla componentes y características aplicadas para la interfaz de autenticación del sistema académico.

Tabla 11-2: Interfaz de autenticación

COMPONENTE	UBICACIÓN	COLOR FONDO /FUENTE
Iconografía	General	Amarillo / Rojo/ Verde
Pestaña	Body	Verde
Botones	Body, centrado	Verde
Mensaje de confirmación	Body, centrado	Celeste/ Blanco
Mensaje de error	Body, centrado	Rojo / Blanco
Input Type	Body	Blanco / Rojo/ Verde

Listado desplegable	Body, alineado a la izquierda	Celeste / Blanco
Información	Footer	Verde

Realizado por: Cristian Merino. 2017

En la **Tabla 12-2** se detalla componentes y características aplicadas para la interfaz de usuario secretaria del sistema académico.

Tabla 12-2: Componentes de interfaz de usuario

COMPONENTE	UBICACIÓN	COLOR FONDO /FUENTE
Menú	Header	Azul/Plata / Blanco
Banner	Header	Amarillo/Blanco / Rojo/ Verde
Pestañas	Body	Blanco
Botones	Body, centrado	Verde
Panel	Body, centrado	Verde degradado
Iconografía	General	Amarillo / Rojo/ Verde
Mensaje de confirmación	Body, centrado	Celeste/ Blanco
Mensaje de error	Body, centrado	Rojo / Blanco
Input Type	Body	Blanco / Rojo/ Verde
Listado desplegable	Body, alineado a la izquierda	Celeste / Blanco
Información	Footer	Plata

Realizado por: Cristian Merino. 2017

A continuación, se muestra el bosquejo de las interfaces de autenticación y la del usuario secretaria quien administrará el sistema por políticas internas y decisión de los stakeholders del proyecto. La **Figura 4-2** muestra el bosquejo de autenticación y el rol de usuario que iniciará sesión.

Figura 4-2: Bosquejo de autenticación

Realizado por: Cristian Merino. 2017

Descripción: Interfaz de autenticación

1. Cédula
2. Clave
3. Seleccione un rol
4. Acceder
5. Inscripción en línea
6. Oferta académica.

Cédula

Para interactuar con la aplicación , el atributo del usuario con el que se inicia sesión es la cédula de identidad. El campo es numérico de 10 caracteres.

Clave

Es el password o contraseña que el usuario digita para la veracidad de autenticación e ingresar al sistema por realizar las tareas o actividades correspondientes según el rol de usuario.

Seleccione un rol

Es un menu donde debemos seleccionar el usuario, los usuario del sistema son: Secretaria, Profesor y estudiante.

Acceder

Es el boton mediante el cual si se ha ingresado correctamente los datos de autenticación nos llevara a la interfaz correspondiente al rol de usuario, caso contrarios, nos mostrara un cuadro de error.

Inscripción en línea

Es un vinculo que nos transala a una interfaz de inscripción en línea, solicitará el número de cédula y al aceptar tendrá que llenar el formilario.

Oferta académica

Éste vinculo nos llevara a una interfaz de información, que muestra la oferta academica de la institución.

Descripción: Interfaz de Secretaria

1. Inicio
2. Administrar
3. Estudiante
4. Docentes
5. Actas
6. Reglamento

Inicio

Apartado estandar para volver a la pantalla de inicio del sistema academico (index)

Administrar

En éste apartado encontramos un desplegable para gestionar: Periodos, Requisitos, Paralelo, Sección, Materias, Licencias, Cursos, Horarios, Nómina.

Estudiantes

En éste apartado tenemos la gestión de estuadiante, aquí encontramos el proceso objetivo principal del desarrollo del presente proyecto de titulación que es la MATRICULACIÓN, encontramos una lista de estudaintes que ponemos buscarlos por sus atributos como: cédula, nombres, telefono, correo electrónico. También tenemos tres botones: el primero para matricular un estudiante, el segundo asentar matrícula y finalmnete un boton para agregar un nuevo estudiante.

Docentes

En éste apartado tenemos la gestión del docente, muestra una lista de docentes que ponemos buscarlos por sus atributos como: cédula, nombres, telefono, correo electrónico, también tenemps un boton para agregar un nuevo docente.

Actas

Este apartado contine las actas que emite la secretaria seleccionando un periodo academico, que son: notas y actas de estudiantes.

Reglamento

Este apartado contiene un texto con el reglamento vigente de la institución.

Reportes

Son un tipo de archivos que según su tipo estructurado y sus elementos se constituyen en archivos de salida, es decir, son una colección de datos procesados y reflejados como información requerida visualizada en dicho REPORTE o ARCHIVO DE SALIDA.

El sistema académico tiene la capacidad de emitir reportes, importando datos almacenados en la base de datos, éstos reportes se crean en formato PDF en concordancia con las historias usuario, los archivos de salida generados tienen características siguientes:

Parte superior del archivo: Consta del escudo institucional, a continuación, está el nombre de la institución académica, posteriormente para el encabezado se encuentra el tipo de licencia que el usuario estudiante se postula. Arial 10, letra mayúscula y negrita es la fuente seleccionada.

Cuerpo del archivo: Consta de todos datos recolectados por el sistema académico para el estudiante, la constancia de pertenecer a la institución. Arial 10 es la fuente seleccionada.

Parte inferior del archivo: Finalmente los apartados de las firmas de constancia por el secretario general de la institución y el estudiante. Arial 10 es la fuente seleccionada. Un ejemplar del reporte de orden de matrícula adjunto en el **Anexo D**.

Despues de realizar los boquejos de las interfaces de sistema académico, se muestra en la **Figura 5-2** la interfaz de autenticación y en la **Figura 6-2** la interfaz del administrador del sistema académico “secretaria”.



Figura 5-2: Interfaz de autenticación

Realizado por: Cristian Merino. 2017



Figura 6-2: Interfaz de administración

Realizado por: Cristian Merino. 2017

Para tener una mejor interacción entre un sistema y el usuario es de suma importancia un diseño claro y conciso que permita que la usabilidad del producto sea eficiente y eficaz. El resto de interfaces del sistema académico están presentes en el manual de usuario el cual adjunto en el **Anexo C** (Manual de Usuario).

En un producto software definitivamente la operatividad funcional será la parte medular de la construcción de software, no obstante, el diseño de interfaces de usuario tiene un significativísimo considerable para la interacción y familiarización entre el sistema y usuario lógicamente, es lo que he denominado como atracción visual al cliente. Todas las interfaces se explican en el **Anexo C** (Manual de Usuario).

2.5.7 Desarrollo de las historias de usuario

Las historias de usuario se constituyen en los requisitos de usuarios elaborados por el Product Owner y plasmados en el Sprint Backlog, las historias de usuario cuentan con un identificador, nombre de la historia, responsable, Sprint en el que se desarrollo , fecha de inicio y fin de desarrollo, descripción y prueba de aceptación. Todos estos parametros garantizan una correcta elaboración de la historia de usuario. En la **Tabla 13-2** se describe un modelo para representar una historia de usuario, la misma que orienta cual guía en el desarrollo de las demás historias de usuario que adjunto en el manual técnico en el **Anexo B** (Manual Técnico).

Tabla 13-2: Historia de usuario 01 (HU-01) / proceso de autenticación

HISTORIA DE USUARIO	
ID: HU-01	Nombre de la Historia: Autenticación de usuario.
Usuario: secretaria / profesor / estudiante	Sprint: 02
Fecha de Inicio: 22/08/2016	Fecha Fin: 23/08/2016
Descripción: Como desarrollador del sistema necesito desarrollar un proceso de autenticación del sistema.	
Pruebas de Aceptación:	
<ul style="list-style-type: none"> ✓ Verificar la autenticidad de los atributos registrados para la autenticación de usuarios. 	

Realizado por: Cristian Merino. 2017

Cumplida la elaboración detallada de las historias de usuario, que son un total de 64 historias, correspondientes 56 a historias de usuario y 8 historias técnicas. En la **Tabla 14-2** se enfoca en dar detalle de un modelo de ingeniería realizada en la historia de usuario 01, modelo que cuenta con: la fecha , descripción de la tarea de ingeniería y el esfuerzo asignado.

Tabla 14-2: Historia de usuario 01 (HU-01) / Tareas de ingeniería

FECHA	TAREAS DE INGENIERÍA	ESFUERZO
22/08/2016	Tarea 1: Definir procedimiento de base de datos para validar información del usuario.	1h
	Creación conexión a la base de datos.	1h
22/08/2016	Tarea 2: Crear clases y paquetes para validación.	1h
	Creación de paquetes organizativos.	1h
	Creación de atributos y métodos para la clase tipo y usuario.	
22/08/2016	Tarea 3: Crear método de la clase Controlador para el proceso de autenticación de usuario.	2h

	Creación de controlador php	2h
23/08/2016	Tarea 4: Crear Vista de Interfaz de usuario para completar la autenticación de usuario.	6h
	Adición de métodos de acceso al controlador para la validación de usuario.	1h
	Preparación de método controlador para recepción y carga preliminar de datos.	4h
	Procesar envío de datos para la validación al servidor.	1h
	Creación de funciones javascript para validar cédula y clave.	

Realizado por: Cristian Merino. 2017

Listadas las tareas de ingeniería, se ha realizado una ficha por cada una de las tareas de ingeniería. Para el cumplimiento con dicha historia de usuario en la **Tabla 15-2**, se describe un modelo de ficha de una tarea de ingeniería modelo que cuenta con: sprint al que pertenece, número de la tarea de ingeniería nombre de la tarea, responsable, tipo de tarea, fecha de inicio y fin, descripción y al final una lista de pruebas de aceptación que culmina dicha tarea de ingeniería.

Tabla 15-2: Historia de usuario 01 (HU-01) / Tareas de ingeniería 01

TAREA DE INGENIERÍA	
Sprint: 02	Número de Tarea: 01
Nombre de la Historia: Autenticación de usuario.	
Nombre de la Tarea: Definir procedimiento de interfaz y base de datos para el proceso información para la validación de datos del usuario.	
Programador Responsable: Desarrollador	Tipo de Tarea: Desarrollo
Fecha Inicio: 22/08/2016	Fecha Fin: 22/08/2016
Descripción: Investigación, codificación y aplicación de un método para la conexión a la base de datos que ayude a la validación de datos en la autenticación de usuario.	
Pruebas de Aceptación: <ul style="list-style-type: none"> ✓ Probar conexión ✓ Validación de Información de usuario. 	

Realizado por: Cristian Merino. 2017

finalizado la descripción de todas las tareas de ingeniería en directa relación a cada historia de usuario obteniendo así 90 tareas de ingeniería. Para finalizar este proceso se ha elaborado un modelo de fichas de pruebas de aceptación. La **Tabla 16-2** describe el mencionado modelo de las pruebas de aceptación que cuenta con: número de prueba, nombre de la historia, nombre de la prueba, responsable, fecha, condiciones de ejecución, pasos de ejecución, resultado y evaluación de la prueba.

Tabla 16-2: Historia de usuario 01 / Prueba de aceptación 01

PRUEBA DE ACEPTACIÓN	
Prueba No: 1	Nombre de la Historia: Autenticación de usuario.
Nombre de la Prueba: Verificación y Validación de la comunicación de datos entre la interfaz con la base de datos.	
Responsable: Desarrollador	Fecha: 23/08/2016
Descripción: Crear vista de interfaz de autenticación usuario para ingresar información del mismo.	
Condiciones de Ejecución: <ul style="list-style-type: none">✓ Configuración adecuada de la clase para carga y búsqueda de datos del paquete modelo para usuario.✓ Instanciación adecuada del objeto para validar la carga de datos.✓ Especificación correcta de las columnas de la tabla del componente en la interfaz de usuario.	
Pasos de Ejecución: <ul style="list-style-type: none">✓ Instanciar objeto de carga de datos desde el controlador.✓ Llamar al método de carga de datos✓ Envío de datos a la interfaz de usuario.	
Resultado: Autenticación correcta de datos de la tabla en la interfaz de usuario.	
Evaluación de la Prueba: <ul style="list-style-type: none">✓ Satisfactoria.	

Realizado por: Cristian Merino. 2017

Para cerrar este punto en el desarrollo del proyecto, se ha realizada todas las pruebas de aceptación dando como resultado un total de 90 pruebas de aceptación, obteniendo una evaluación de carácter satisfactoria, se finiquita que se han desarrollado exitosamente seguidamente se puede decir que todos los requerimientos de software han sido elaborados y probados para el eficiente consumo del producto.

2.5.8 Codificación e implementación

Para el desarrollo del sistema académico se ha tenido en cuenta la estructura, arquitectura y funcionalidad del producto, así también, se ha tenido claro cómo va a ser la interacción que es abstracta entre el sistema y la base de datos, la comunicación entre el sistema y el usuario. Por lo cual se realizó una codificación normalizada o estandarizada.

Lenguaje

El sistema académico para el Sindicato de Choferes Profesionales 4 de Octubre está desarrollado en el lenguaje de programación Java con el framework JSF (JavaServer Faces), el almacén de datos está desarrollado en el sistema de gestión MySQL , además del uso de servicios y herramientas tecnológicas como: IDE NeatBeans, Apache Tomcat, GlassFish Server, necesario para el cumplimiento de los requisitos y proceso de desarrollo del producto, una prerrogativa es gracias al patrón bajo la que está diseñado el presente proyecto MVC (Modelo Vista Controlador) sinónimo de flexibilidad e independencia entre características del sistema tales como: entidades, clases, controladores, funciones, y vistas.

Vista

Las Vistas de un sistema informático se constituyen en el servicio al cliente directo, el desarrollo de las interfaces para la comunicación directa usuario – sistema, muestra después de recibir información por el intérprete (controlador = métodos y funciones) hacia la vista (vista = páginas), este diseño está desarrollada con Facelets.

Controlador

Es el intérprete responsable de dar respuesta a solicitudes generadas con ayuda del modelo (objetos Javabean) y las vistas (páginas Facelets), por medio del envío y recepción de información que se encuentra en el almacén de datos.

Función

Representa el conjunto de instrucciones que realizan una tarea específica y puede retornar un valor, en caso de ser necesario, por tanto, simboliza una gestión de información del sistema.

Clase

Representa el conjunto de atributos que posee una entidad, en la cual podemos almacenar datos y propiedades referentes a la misma.

En esta sección también se realizó las instalaciones de las herramientas tecnológicas en los equipos del Sindicato de Choferes Profesionales, descritos en el Manual Técnico que se halla

anexado en el CD, en la path: SistemaAcadémico/MANUALES/Manual-Técnico-SistemaAcadémico.

2.6 Finalización del proyecto

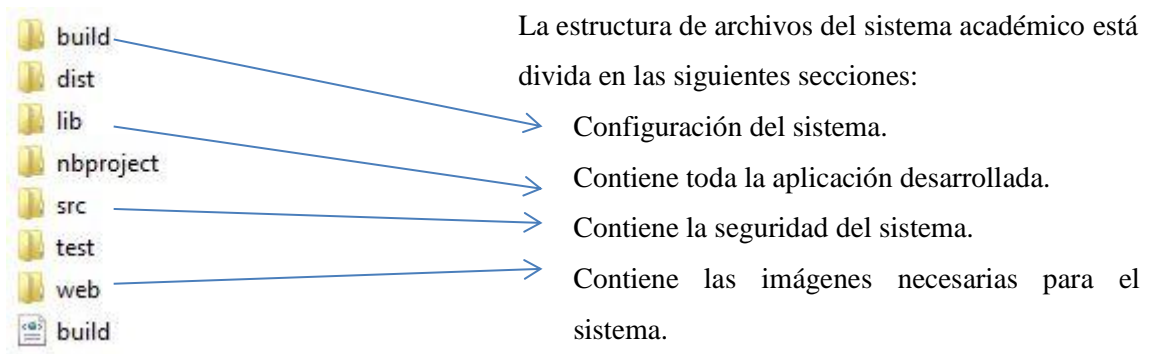
En esta etapa del proyecto se detallan las actividades realizadas para la finalización del desarrollo del sistema académico del Sindicato de Choferes Profesionales “4 de Octubre”, se complementa elaborando el sprint denominado **BurnDown Chart**, que es una gráfica representativa que muestra como fue el proceso de desarrollo del proyecto, si el esfuerzo se sobreestimo o si se subestimo, además de manera explícita refleja que el equipo scrum culminó el proyecto. A continuación, en la **Tabla 17-2** se describe las actividades realizadas en la etapa de finalización del Proyecto.

Tabla 17-2: Actividades de finalización del proyecto

ACTIVIDAD	DESCRIPCION	RESPONSABLE
Evaluación del sistema	<ol style="list-style-type: none"> 1) Someter a pruebas al sistema para medir su calidad. 2) Realizar una encuesta con el objetivo de medir la usabilidad, funcionalidad y eficiencia del sistema 	Desarrollador
Documentación del sistema	<ol style="list-style-type: none"> 1) Elaboración del manual técnico del sistema académico. 2) Elaboración del manual de usuario para el sistema académico. 	Desarrollador
Capacitación de usuario	<ol style="list-style-type: none"> 1) Entrega de manual técnico de sistema al Product Owner. 2) Entrega de manual de usuario al Product Owner. 	Desarrollador

Realizado por: Cristian Merino. 2017

Una vez finalizado el desarrollo del sistema académico del Sindicato de Choferes Profesionales 4 de Octubre, los recursos entregables físicos en cuanto a la codificación y configuración del producto software, se obtuvo la siguiente estructura del paquete desarrollado:



En la carpeta src se almacenan todos los contenidos creados para la administración del sistema donde cada uno tiene sus controladores, entidades, vistas de interfaz de usuario, de esta manera se puede concluir que el total de líneas de código se describen a continuación:

Los controladores utilizados para manejar las vistas es de 337 lc, interfaz de usuario (vistas .html.twig) un total de 587 lc, código JavaScript necesario para la ejecución de los procesos un total de 420 lc, hojas de estilo (.css) un total de 1344 lc, de esta manera se concluye que todas y cada una de las líneas de código, descritas anteriormente han sido desarrolladas y ejecutadas para poder comprobar la veracidad en eficiencia en los resultados de poder exponer la información académica de la institución.

Para los detalles de las líneas de código del sistema académico se anexa el aplicativo y base de datos descritas en path:

- ✓ SistemaAcadémico/sistemaPenipe;
- ✓ SistemaAcadémico/sindicato_penipe, respectivamente.

2.6.1 Gestión del proyecto

El diagrama BurnDown Chart, hace referencia al seguimiento del proyecto durante su desarrollo, habitualmente no ayuda a determinar “el tiempo restante” para concluir las historias de usuarios que pertenecen a un sprint planificado. Las fechas están distribuida en el eje horizontal X en la que se representa el inicio y fin que conciernen a los días comprendidos entre el 01 de agosto de 2016 y 03 de febrero de 2017, por el eje vertical y está representado por el esfuerzo, representado con un total de 1080 horas. La gráfica está representada por dos líneas; la línea de color turquesa determina el desarrollo ideal del proyecto, la línea de color rojizo sustenta el desarrollo real.

En el **Gráfico 1-2** se visualiza que inicialmente se cumplió con lo planificado y entrega de la funcionalidad del primer sprint, durante el desarrollo del proyecto existieron retardos en consecuencia a los scrum meeting review, por parte de Product Owner no podía estar presente. Los puntos de esfuerzo retrasados se los realizó durante el sprint 3 y 4, consecuentemente se concluyó lo planificado.

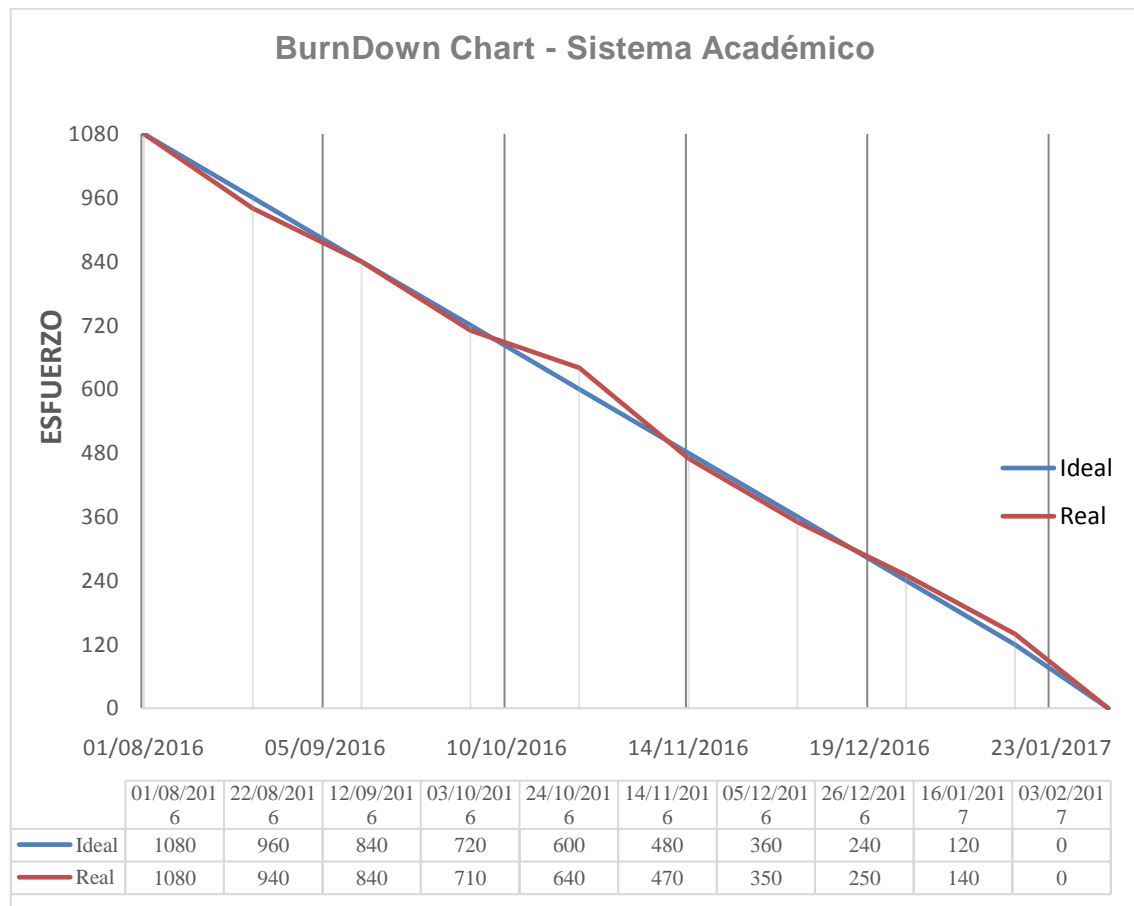


Gráfico 1-2: Sprint BurnDown Chart

Realizado por: Cristian Merino. 2017

En el **Gráfico 1-2** se puede visualizar: en la columna izquierda (eje de las y) muestra un total de 1080 horas de esfuerzo que se ha dedicado en la realización del proyecto, en el eje de las x la fecha de inicio del proyecto que es el día 01/08/2016 y la fecha de culminación que es el 03/01/2017.

El proyecto está dividido en 9 Sprints; en el Sprint 01 que inicia en la fecha 01/08/2016 se nota una leve subestimación en el tiempo de desarrollo de las historias de usuario, en el Sprint 03 que inicia en la fecha 12/09/2016 se nota una equiparación en el desarrollo de las historias de usuario asignadas a este Sprint.

De igual manera, al alcanzar el Sprint 04 que inicia 03/10/2016 se percibe una subestimación minúscula, sin embargo, en el sprint 05 que inicia en la fecha 24/10/2016 se visualiza una sobrestimación considerable el desarrollo de los sprints mencionado. En el Sprint 06 que inicia en la fecha 14/11/2016 se puede observar una compensación de la sobrestimación del sprint anterior, en este punto del desarrollo del proyecto los sprints se ajustan los ideales planificados. En el Sprint 07 que inicia el 05/12/2016 se puede evidenciar que el esfuerzo se ha obtenido nuevamente una ligera subestimación, ya en el sprint 08 que inicia en la fecha 26/12/2016 al inicio se parte con una porción de esfuerzo a favor, pero a medida que avanza el sprint el tiempo queda sobrestimado, y en el Sprint 09 que inicio en la fecha 16/01/2017 se puede observar una sobrestimación del esfuerzo, no obstante, al final se obtiene una compensación en el tiempo establecido.

De esta manera se concluye que, aunque los tiempos de desarrollo de los Sprints han sido subestimados o sobreestimados estos han sido culminados dentro de los tiempos establecidos de inicio y fin del proyecto.

El detalle de las historias de usuario con su respectivo diagrama estadístico BurnDown Chart están desarrollados en el **Anexo B** (Manual Técnico).

CAPÍTULO III

3 . MARCO DE ANÁLISIS, DISCUSIÓN Y EVALUACIÓN DE RESULTADOS

El presente capítulo, expone los resultados obtenidos en la evaluación del “**DESARROLLO DEL SISTEMA ACADÉMICO DEL SINDICATO DE CHOFERES PROFESIONALES 4 DE OCTUBRE APLICANDO EL FRAMEWORK JSF**”, bajo parámetros establecidos que rigen en las normas vigente del estándar ISO/IEC 9126-3. Utilizando las herramientas tecnológicas y metodología que ayudaron a la construcción del producto software, por consiguiente, se realizó la evaluación de la usabilidad, funcionalidad y eficiencia del producto software, para determinar y garantizar la calidad de sistema académico.

Cálculo de la muestra poblacional para la evaluación del sistema académico

Para la evaluación de calidad del sistema académico se utilizó la técnica de encuesta con preguntas cerradas para la recopilación de información acerca de la usabilidad funcionalidad y eficiencia del producto software. además, para el tamaño de la muestra poblacional se hizo uso del muestreo aleatorio estratificado desproporcionado puesto que el sistema académico está dirigido a tres tipos de usuarios: secretaria, profesor, estudiante. En la **Figura 1-3** se ejemplifica la selección la muestra poblacional dando como resulta un total de 10 usuarios encuestados.



Figura 1-3: Muestreo aleatorio estratificado desproporcionado

Realizado por: Cristian Merino. 2017

3.1 Parámetros de calidad

Los parámetros de calidad son un conjunto características con que poseen subcaracterísticas propias de la usabilidad, de la funcionalidad, y también de la eficiencia del producto, para la evaluación de la calidad del sistema académico, según el estándar ISO/IEC 9126-3, se debe establecer parámetros y métricas que permitan identificar el cumplimiento de la calidad del producto. (Mena, 2006, p. 3) (García, 2010, pp. 7–8)

Los criterios para evaluar la calidad del sistema académico han sido tomados en cuenta aspectos tanto funcionales que son las historias de usuario como no funcionales que se refiere a las historias técnicas aplicadas en el proceso de desarrollo. Estas consideraciones son esenciales para garantizar la calidad del producto software.

Entonces para la evaluación del sistema académico respecto a la: usabilidad, funcionalidad y eficiencia, los parámetros son:

- ✓ Alta,
- ✓ Media, y;
- ✓ Baja

A continuación, en la **Tabla 1-3** se detallan:

Tabla 1-3: Parámetros para la medición de la evaluación

JERARQUÍA DE ACEPTACIÓN	CATEGORÍA DE ACEPTACIÓN
Alta	0,71 a 1,00
Media	0,36 a 0,70
Baja	0,00 a 0,35

Realizado por: Cristian Merino. 2017

La siguiente **Tabla 2-3**, especifica los niveles desde los cuales se evaluó la usabilidad, funcionalidad y eficiencia del sistema académico.

Tabla 2-3: Características y Subcaracterísticas de calidad

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	ACEPTACIÓN SOLICITADA
Usabilidad	Entendimiento	Media
	Aprendizaje	Alta
	Operabilidad	Media
	Atracción	Baja
	Conformidad de la usabilidad	Alta
Funcionalidad	Adecuación	Alta
	Exactitud	Alta
	Interoperabilidad	Media
	Seguridad	Alta
	Conformidad de la funcionalidad	Media
Eficiencia	Comportamiento de respuesta	Alta
	Utilización de recursos	Media
	Conformidad de eficiencia	Media

Realizado por: Cristian Merino. 2017

3.2 Análisis de la usabilidad del sistema académico

El análisis de la usabilidad del sistema académico, se ha establecido mediante las métricas de sus características de calidad (entendimiento, aprendizaje, operabilidad, atracción, conformidad de la usabilidad), establecidas por la norma ISO 9126-3 aplicada a una encuesta realizada a usuarios del sistema. **Anexo E**

3.2.1 Métrica de evaluación: Entendimiento

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis del entendimiento del producto software, el detalle se describe en la **Tabla 3-3**.

- M:** Es el número de funciones (o tipo de funciones) evidentes al usuario.
- N:** Es el total de funciones (o tipos de funciones).
- X:** Variable resultante de la fórmula de medición.

Tabla 3-3: Métrica de entendimiento

MÉTRICA DE ENTENDIMIENTO	
Nombre:	Funciones evidentes del sistema académico
Propósito:	¿Cuáles son de las funciones del sistema académico que son evidentes para el usuario?
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Funciones o herramientas evidentes:	<p>Funciones de acceso al sistema (autenticación).</p> <p>Funciones para el proceso de pre matricula.</p> <p>Funciones para la creación, registro, eliminar y actualización de períodos.</p> <p>Funciones para la creación, registro, eliminar y actualización de requisitos.</p> <p>Funciones para la creación, registro, eliminar y actualización de paralelos.</p> <p>Funciones para la creación, registro, eliminar y actualización de sección de horario.</p> <p>Funciones para la creación, registro, eliminar y actualización de materias.</p> <p>Funciones para la creación, registro, eliminar y actualización de licencias.</p> <p>Funciones para la creación, registro, eliminar y actualización de cursos.</p> <p>Funciones para la creación, registro, eliminar y actualización de nómina.</p> <p>Funciones para la creación, registro, eliminar y actualización de horario.</p> <p>Funciones para la creación, registro, eliminar y actualización de usuarios.</p> <p>Funciones para la creación, registro, eliminar y actualización de calificaciones.</p> <p>Funciones para la asignación de cursos, materias, estudiantes, profesores.</p> <p>Funciones para visualización e impresión de reportes.</p>
Medición, formula:	<p>$M = 15$</p> <p>$N = 15$</p> <p>$M = N$</p> <p>$X = (M/N)$</p> <p>$X = (15/15)$</p> <p>$X = 1$</p>
Interpretación:	<p>$0 < X \leq 1$</p> <p>$0 < 1 \leq 1$</p> <p>Entre más cercano a 1, más completa.</p>
Tipo de escala:	Absoluta
Tipo de medida:	<p>$M = \text{count}$</p> <p>$N = \text{count}$</p> <p>$X = \text{count/count}$</p>
Fuente de medición:	<p>Especificación de requisitos</p> <p>Diseño</p> <p>Informe de revisión</p>
Audiencia:	<p>Stakeholders</p> <p>Desarrollador</p>

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de entendimiento, se logró un valor numérico de medición “1” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la usabilidad del sistema.

3.2.2 Métrica de evaluación: Aprendizaje

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis del aprendizaje del producto software, el detalle se describe en la **Tabla 4-3**.

- M:** Es el número de usuarios encuestados.
- N:** Es el número de usuarios que no aprendieron a usar el sistema.
- X:** Es el resultado de la fórmula de medición.

Tabla 4-3: Métrica de aprendizaje

MÉTRICA DE APRENDIZAJE	
Nombre:	Exactitud del sistema académico
Propósito:	¿El aprendizaje del sistema académico fue sencillo?
Medición, fórmula:	M = 10 N = 1 $X = 1 - (N/M) = 1 - (1/10)$ X = 0,90
Interpretación:	$0 \leq X \leq 1$ $0 \leq 0,90 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	X = count/count M = count N = count
Fuente de medición:	Informe de revisión
Audiencia:	Usuario administrador del sistema Usuario estudiante / Usuario profesor

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de aprendizaje, se logró un valor numérico de medición “0,90” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la usabilidad del sistema.

3.2.3 Métrica de evaluación: Operabilidad

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la operabilidad del producto software, el detalle se describe en la **Tabla 5-3**.

- M:** Es el número de usuarios encuestados.
- N:** Es el número de usuarios que no pudieron operar el sistema.
- X:** Es el resultado de la fórmula de medición.

Tabla 5-3: Métrica de operabilidad

MÉTRICA DE OPERABILIDAD	
Nombre:	Inspección del sistema académico
Propósito:	¿Es sencillo el control y operatividad del sistema académico?
Medición, fórmula:	M = 10 N = 2 $X = 1 - (N/M) = 1 - (2/10)$ X = 0,80
Interpretación:	$0 \leq X \leq 1$ $0 \leq 0,70 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	X = count/count M = count M = count
Fuente de medición:	Verificación Informe de revisión
Audiencia:	Usuarios administradores del sistema Usuario estudiante/ Usuario profesor

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de operabilidad, se logró un valor numérico de medición “0,80” con una jerarquía de aceptación “media”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la usabilidad del sistema.

3.2.4 Métrica de evaluación: atracción

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la atracción del producto software, el detalle se describe en la **Tabla 6-3**.

- M:** Es el número de usuarios encuestados.
- N:** Es el número de usuarios que no les agradó el diseño del sistema académico.
- X:** Es el resultado de la fórmula de medición.

Tabla 6-3: Métrica de atracción

MÉTRICA DE ATRACCIÓN	
Nombre:	Diseño de la Interfaz del sistema académico
Propósito:	¿El sistema académico tiene una agradable apariencia visual en su diseño?
Medición, fórmula:	$M = 10$ $N = 3$ $X = 1 - (N/M) = 1 - (3/10)$ $X = 0,7$
Interpretación:	$0 \leq X \leq 1$ $0 \leq 0,70 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ $M = \text{count}$ $N = \text{count}$
Fuente de medición:	Informe de revisión
Audiencia:	Usuario administrador del sistema. Usuario estudiante/ Usuario profesor. Usuario invitado.

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de atracción, se logró un valor numérico de medición “0,70” con una jerarquía de aceptación “media”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la usabilidad del sistema.

3.2.5 Métrica de evaluación: conformidad de la usabilidad

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la conformidad de la usabilidad del producto software, el detalle se describe en la **Tabla 7-3**.

M: Es el número de normas o estándares de diseño.

X: Es el resultado de la fórmula de medición.

Tabla 7-3: Métrica de conformidad de la usabilidad

MÉTRICA DE CONFORMIDAD DE LA USABILIDAD	
Nombre:	Aprobación de la usabilidad del sistema
Propósito:	¿El sistema académico en su diseño de interfaz, considera que se rige a normas o estándares?
Normas o estándares usados en el desarrollo:	Diseño moderno responsivo.
Medición:	$X = (1/M) - 1$ $X = (1/1) - 1$ $X = 1$
Interpretación:	$0 < X \leq 1$ $0 < 1 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = 1/\text{count}$ $M = \text{count}$
Fuente de medición:	Informe de revisión
Audiencia:	Desarrolladores

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de conformidad de la usabilidad, se logró un valor numérico de medición “1” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la usabilidad del sistema.

3.3 Análisis de la funcionalidad del sistema académico

El análisis de la funcionalidad del sistema académico, se ha establecido mediante las métricas de sus características de calidad (adecuada, exactitud, interoperabilidad, seguridad, conformidad de la funcionalidad), establecidas por la norma ISO 9126-3 aplicada a una encuesta (**Anexo E**) realizada a usuarios del sistema, y desarrollador del sistema, se detalla a continuación:

3.3.1 Métrica de evaluación: adecuación

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la adecuación del producto software, el detalle se describe en la **Tabla 8-3**.

- M:** Es el número de requerimientos planteados.
- N:** Es el número de requerimientos faltantes.
- X:** Es el resultado de la fórmula de medición.

Tabla 8-3: Métrica de adecuación

MÉTRICA DE ADECUACIÓN	
Nombre:	Finalización e implementación funcional del sistema
Propósito:	¿Está completa la implementación funcional del sistema académico?
Método de Aplicación	Contabilizar las funciones restantes que han sido descubiertas en la evaluación y realizar la comparación con el número de funciones de requisitos planificados.
Requerimientos faltantes:	No existen requerimientos inconclusos o faltantes.
Medición, formula:	M= 56 N = 0 $X = 1 - (M/N)$ $X = 1 - (0/56)$ X = 1
Interpretación:	$0 \leq X \leq 1$ $0 \leq 1 \leq 1$ Entre más cercano a 1, más completa
Tipo de escala:	Absoluta
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Especificación de requisitos Diseño Código fuente
Audiencia:	Usuario administrador del sistema

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de adecuación, se logró un valor numérico de medición “1” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la funcionalidad del sistema.

3.3.2 Métrica de evaluación: exactitud

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la exactitud del producto software, el detalle se describe en la **Tabla 9-3**.

- M:** Es el número de requerimientos establecidos.
- N:** Es el número de requerimientos que no fueron exactos.
- X:** Es el resultado de la fórmula de medición.

Tabla 9-3: Métrica de exactitud

MÉTRICA DE EXACTITUD	
Nombre:	Exactitud del sistema
Propósito:	¿El sistema académico, realiza los procesos esperados de manera correcta?
Requerimientos no exactos:	Tiempo que tarda la impresión en pantalla de reportes.
Medición, fórmula:	M = 35 N = 1 $X = 1 - (N/M) = 1 - (1/35)$ X = 0,97
Interpretación:	$0 \leq X \leq 1$ $0 \leq 0,96 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida	X = count/count M = count N = count
Fuente de medición:	Informe de revisión
Audiencia:	Usuario administrador del sistema

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de exactitud, se logró un valor numérico de medición “0,97” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la funcionalidad del sistema.

3.3.3 Métrica de evaluación: interoperabilidad

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la interoperabilidad del producto software, el detalle se describe en la **Tabla 10-3**.

- M:** Es el número de sistemas de interacción.
X: Es el resultado de la fórmula de medición.

Tabla 10-3: Métrica de interoperabilidad

MÉTRICA DE INTEROPERABILIDAD	
Nombre:	Interoperabilidad entre sistemas
Propósito:	¿El sistema académico interactúa con otros sistemas?
Sistemas de interacción:	No existen sistemas de interacción.
Medición, fórmula:	M = 0 X = 1 / (M+1) = (1/1) X = 1
Interpretación:	0 <= X <= 1 0 <= 1 <= 1
Tipo de escala:	Absoluta
Tipo de medida	X = 1/count M = count
Fuente de medición:	Informe de revisión
Audiencia:	Desarrollador

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de interoperabilidad, se logró un valor numérico de medición “1” una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la funcionalidad del sistema.

3.3.4 Métrica de evaluación: seguridad

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la seguridad del producto software, el detalle se describe en la **Tabla 11-3**.

- M:** Es el número de accesos no contralados.
X: Es el resultado de la fórmula de medición.

Tabla 11-3: Métrica de seguridad

MÉTRICA DE SEGURIDAD	
Nombre:	Seguridad del sistema académico
Propósito:	¿El sistema académico restringe el acceso a usuarios no autorizados?
Accesos no controlados:	No existen usuarios que no se haya registrado que tengan acceso al sistema.
Medición, fórmula:	M = 0 $X = 1/(M+1) = (1/1)$ X = 1
Interpretación:	0 <= X <=1 0 <= 1 <= 1 Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	M = count X = 1/count
Fuente de medición:	Informe de revisión
Audiencia:	Usuario administrador del sistema Usuario profesor Usuarios estudiante

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de seguridad, se logró un valor numérico de medición “1” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para el funcionamiento del sistema.

3.3.5 Métrica de evaluación: conformidad de la funcionalidad

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la conformidad de la funcionalidad del producto software, el detalle se describe en la **Tabla 12-3**.

M: Es el número de normas y/o estándares de creación.

X: Es el resultado de la fórmula de medición.

Tabla 12-3: Métrica de conformidad de la funcionalidad

MÉTRICA DE CONFORMIDAD DE LA FUNCIONALIDAD	
Nombre:	Conformidad de la funcionalidad del sistema académico

Propósito:	¿El sistema académico está desarrollado rigiéndose a normas y/o estándares?
Normas o estándares usados en el desarrollo:	Patrón Modelo Vista Controlador Estándar de codificación. Estándar de normalización de base de datos.
Medición, fórmula:	$A = 3$ $X = (1/A) - 1$ $X = (1/3) - 1$ $X = 0,66$
Interpretación:	$0 < X \leq 1$ $0 < 0,67 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = 1/\text{count}$ $A = \text{count}$
Fuente de medición:	Especificaciones de requerimientos Informe de revisión Código fuente
Audiencia:	Desarrolladores

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de conformidad de la funcionalidad, se logró un valor numérico de medición “0,67” con una jerarquía de aceptación “media”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la funcionalidad del sistema.

3.4 Análisis de la eficiencia del sistema académico

El análisis de la eficiencia del sistema académico, se ha establecido mediante el uso de métricas de sus características de calidad (comportamiento de respuesta, utilización de recursos, conformidad de la eficiencia), establecidas por la norma ISO 9126-3 aplicada a los tiempos tomados mediante un cronómetro y con la técnica de observación directa.

3.4.1 Métrica de evaluación: Comportamiento de respuesta

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis del tiempo de respuesta del producto software, el detalle se describe en la **Tabla 13-3**.

M: Es el tiempo (calculado o simulado) que se demora con un sistema de escritorio.

N: Es el tiempo (calculado o simulado) que se demora con un sistema web.

X: Es el resultado de la fórmula de medición.

Tabla 13-3: Métrica de Comportamiento de respuesta

MÉTRICA DE COMPORTAMIENTO DE RESPUESTA	
Nombre:	Comportamiento o tiempo de respuesta del sistema académico
Propósito:	¿Cuál es el tiempo estimado que consume una acción?
Método de aplicación	Evaluar la eficiencia de las llamadas al SO y a la aplicación. Estimar el tiempo de respuesta basado en ello. Puede medirse: Todo o partes de las especificaciones de diseño. Probar la ruta completa de una transacción. Probar módulos o partes completas del producto. Producto completo durante la fase de pruebas
Medición, fórmula:	M = 1 hora N = 0.10 hora $X = 1 - (N/M) = 1 - (0.10/1)$ X = 0,90
Interpretación:	$0 < X \leq 1$ $0 < 0,90 \leq 1$ Entre más cercano a 1, Entre más corto mejor.
Tipo de escala:	Proporción
Tipo de medida:	X=Tiempo
Fuente de medición:	Sistema operativo Tiempo que tarda las llamadas al sistema académico
Audiencia:	Desarrollador Stakeholders

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de respuesta, se logró un valor numérico de medición “0.90” con una jerarquía de aceptación “alta”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la eficiencia del sistema.

3.4.2 Métrica de evaluación: Utilización de recursos

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la utilización de recursos del producto software, el detalle se describe en la **Tabla 14-3**.

M: Es el número de hardware o software adicional utilizado con un sistema de escritorio.

N: Es el número de hardware o software adicional utilizado con un sistema web.

X: Es el resultado de la fórmula de medición.

Tabla 14-3: Métrica de la utilización de recursos

MÉTRICA DE LA UTILIZACIÓN DE RECURSOS	
Nombre:	Utilización de recursos en el sistema académico
Propósito:	Los recursos que incluyen en el desarrollo de software, la configuración de hardware y software del sistema, y materiales
Método de aplicación:	Capacidad del sistema académico para usar recursos (cantidades y tipos) cuando éste funciona desde las condiciones establecidas. Pendrive CDs DVDs Hoja de Cálculo Excel
Medición:	M = 6 N=2 $X = 1 - (N/M) = 1 - (2/6)$ X = 0,66
Interpretación:	$0 < X \leq 1$ $0 < 0,67 \leq 1$ Entre más cercano a 1 mejor.
Tipo de escala:	Proporción
Audiencia:	Desarrolladores Stakeholders

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de utilización de recursos, se logró un valor numérico de medición “0,67” con una jerarquía de aceptación “media”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la eficiencia del sistema.

3.4.3 Métrica de evaluación: conformidad de la eficiencia

En este punto de la evaluación, se indica la notación de las variables que han participado en el análisis de la conformidad de la eficiencia del producto software, el detalle se describe en la **Tabla 15-3**.

M: Es el número de normas o estándares de diseño.

X: Variable resultante de la fórmula de medición.

Tabla 15-3: Métrica de conformidad de la eficiencia

MÉTRICA DE CONFORMIDAD DE LA EFICIENCIA	
Nombre:	Conformidad de la eficiencia del sistema académico
Propósito:	¿El sistema académico cumple los requisitos solicitados?
Normas o estándares usados en el desarrollo:	Mejorar los tiempos de respuesta en los procesos IEEE Std 1061-1998 ISO/IEC 15939 Referencia a ISO/IEC 12207
Medición:	$X = (1/M) - 1$ $X = (1/3) - 1$ $X = 0,66$
Interpretación:	$0 < X \leq 1$ $0 < 0,66 \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Fuente de medición:	Informe de revisión
Audiencia:	Desarrollador

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Inmediatamente realizado la métrica de conformidad de la usabilidad, se logró un valor numérico de medición “0,66” con una jerarquía de aceptación “media”, dentro de la categoría establecida en la norma ISO 9126-3 que rige para la eficiencia del sistema.

3.5 Evaluación de la calidad del sistema académico

En la **Tabla 16-3** se expone los valores entre las jerarquías requeridos y obtenidas con sus métricas para realizar la evaluación de la usabilidad del sistema.

Tabla 16-3: Evaluación de la usabilidad del sistema académico

EVALUACIÓN DE LA USABILIDAD DEL SISTEMA ACADÉMICO				
SUB CARACTERÍSTICA	REQUERIDO		OBTENIDO	
	MÉTRICA DE EVALUACIÓN	JERARQUÍA ACEPTACIÓN	MÉTRICA DE EVALUACIÓN	JERARQUÍA ACEPTACIÓN
Entendimiento	1,00	Alta	1,00	Alta
Aprendizaje	1,00	Alta	0,90	Alta

Operabilidad	1,00	Alta	0,70	Media
Atracción	0,35	Baja	0,70	Media
Conformidad de la usabilidad	0,70	Media	0,50	Media
TOTAL	4,20	TOTAL	3,80	
PORCENTAJE TOTAL	100%	PORCENTAJE TOTAL	90,48%	

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Mediante el análisis de la **Tabla 16-3** se concluye que el sistema académico es 90,48% usable, calculo obtenido de la suma de los valores numéricos de las métricas, aplicada a una regla de tres simple directa, con su valor porcentual (Porcentaje Total = $(100 * 3,80) / 4,20$).

A continuación, se muestra el **Gráfico 1-3**, que representa un cuadro estadístico de barras con los valores requeridos en comparación con los valores obtenidos con el fin de tener una visión más amplia de la usabilidad del sistema académico.

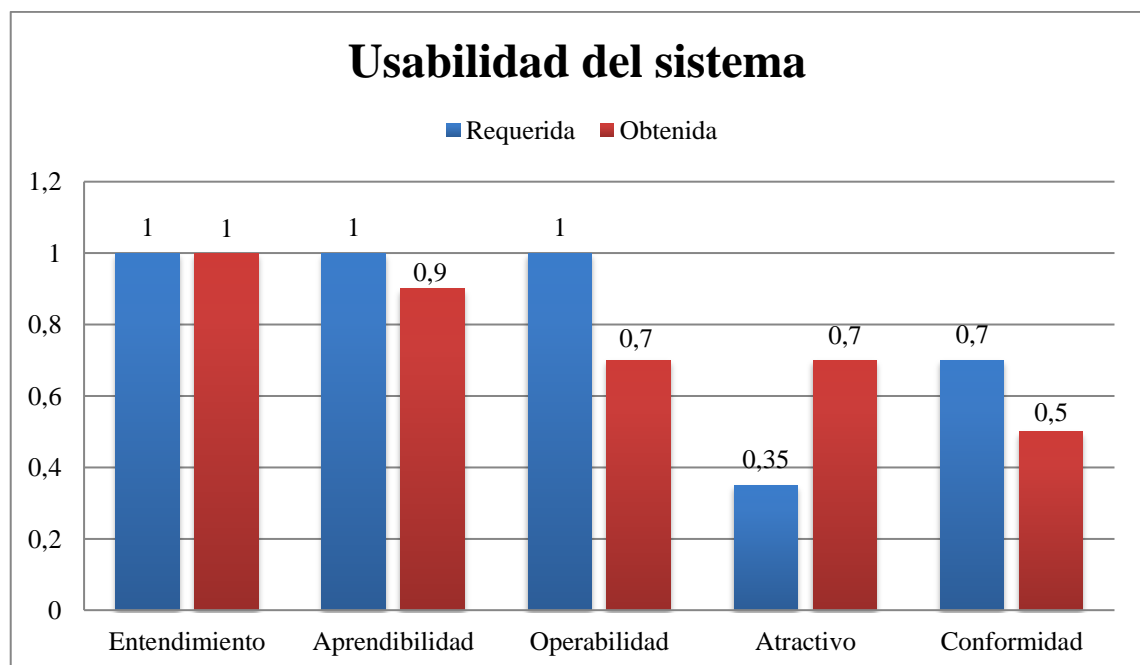


Gráfico 1-3: Evaluación de la usabilidad del sistema

Realizado por: Cristian Merino. 2017

En la **Tabla 17-3** se expone los valores entre las jerarquías requeridos y obtenidas con sus métricas para realizar la evaluación de la funcionalidad del sistema.

Tabla 17-3: Evaluación de la funcionalidad del sistema

EVALUACIÓN DE LA FUNCIONALIDAD DEL SISTEMA				
SUB CARACTERÍSTICA	REQUERIDO		OBTENIDO	
	MÉTRICA DE EVALUACIÓN	JERARQUÍA ACEPTACIÓN	MÉTRICA DE EVALUACIÓN	JERARQUÍA ACEPTACIÓN
Adecuación	1,00	Alta	1,00	Alta
Exactitud	1,00	Alta	0,97	Alta
Interoperabilidad	0.70	Media	1,00	Media
Seguridad	1,00	Alta	1,00	Alta
Conformidad de la funcionalidad	1.00	Media	0,67	Media
TOTAL	4,70	TOTAL	4,64	
PORCENTAJE TOTAL	100%	PORCENTAJE TOTAL	98,72%	

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Mediante el análisis de la **Tabla 17-3** se concluye que el sistema académico es 98,72% funcional, calculo obtenido de la suma de los valores numéricos de las métricas, aplicada a una regla de tres simple directa, con su valor porcentual (Porcentaje Total = $(100 * 4,64) / 4,70$). A continuación, se muestra el **Gráfico 2-3**, que representa un cuadro estadístico de barras con los valores requeridos en comparación con los valores obtenidos con el fin de tener una visión más amplia de la funcionalidad del sistema académico.

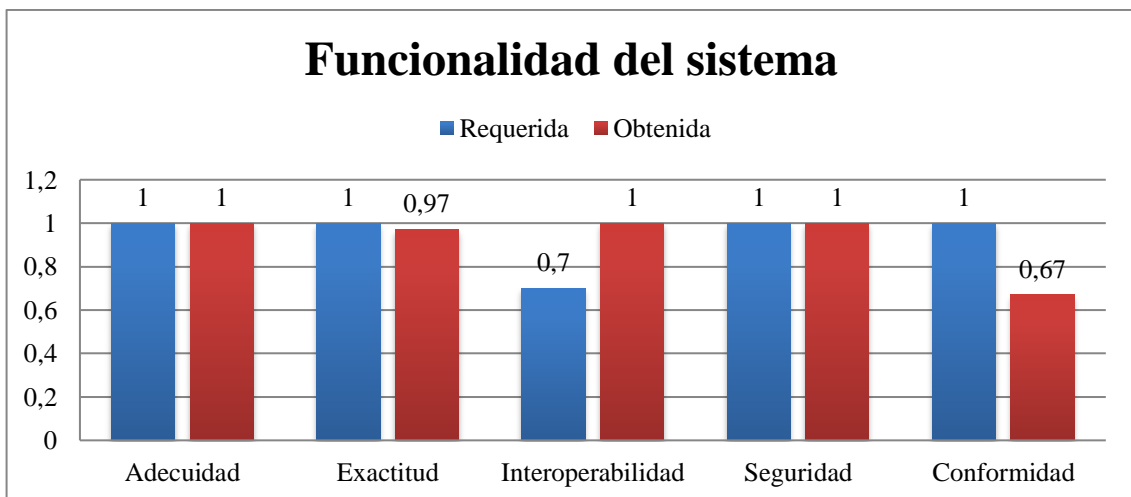


Gráfico 2-3: Evaluación de la funcionalidad del sistema

Realizado por: Cristian Merino. 2017

En la **Tabla 18-3** muestra los valores entre las jerarquías requeridos y obtenidas con sus métricas para realizar la evaluación de la eficiencia del sistema.

Tabla 18-3: Evaluación de la eficiencia del sistema

EVALUACIÓN DE LA EFICIENCIA DEL SISTEMA				
SUB CARACTERÍSTICA	REQUERIDO		OBTENIDO	
	MÉTRICA DE EVALUACIÓN	JERARQUÍA ACEPTACIÓN	MÉTRICA DE EVALUACIÓN	JERARQUÍA ACEPTACIÓN
Tiempo de respuesta	1,00	Alta	0,90	Alta
Utilización de recursos	0.70	Media	0,67	Media
Conformidad de la eficiencia	0,70	Media	0,66	Media
TOTAL	2,40	TOTAL	2.23	
PORCENTAJE TOTAL	100%	PORCENTAJE TOTAL	92,91%	

Realizado por: Cristian Merino. 2017

Fuente: Mena, 2006. (ISO 9126-3: Métricas Internas de la Calidad del Producto de Software)

Mediante el análisis de la **Tabla 18-3** se concluye que el sistema académico es 92,91% eficiente, calculo obtenido de la suma de los valores numéricos de las métricas, aplicada a una regla de tres simple directa, con su valor porcentual (Porcentaje Total = $(100 * 2,23) / 2,40$). A continuación, se muestra el **Gráfico 3-3**, que representa un cuadro estadístico de barras con los valores requeridos en comparación con los valores obtenidos con el fin de tener una visión más amplia de la eficiencia del sistema académico.

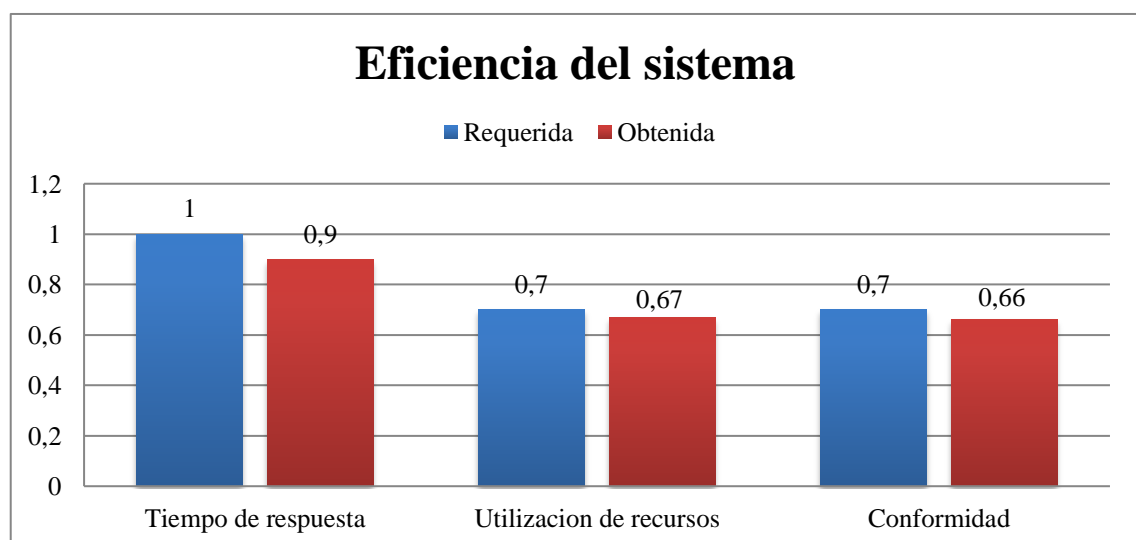


Gráfico 3-3: Evaluación de la eficiencia del sistema

Realizado por: Cristian Merino. 2017

CONCLUSIONES

Finalizado el Desarrollo del Sistema Académico de Choferes Profesionales 4 de Octubre aplicando el framework JSF, previo a la obtención del grado académico de Ingeniero en Sistemas Informáticos, se ha obtenido las conclusiones:

- ✓ Los estudios del framework JSF, basados en proyectos análogos al presente entregable, se ha consolidado la justificación por preferir del framework JSF para el desarrollo de sistemas basados en la web, gracias a sus caracterización y funcionamiento en la construcción del producto software basado en la web.
- ✓ Después de haber analizado la herramienta tecnológica, para el proceso de diseño se utilizó el estándar adoptado a nivel internacional, Lenguaje Unificado de Modelado UML, gracias a que es un lenguaje gráfico se logró identificar actores, entidades, estructuras y comportamiento de los procesos del sistema académico. Conjuntamente se diseñó y se desarrolló el almacén de datos y normalización de la base de datos.
- ✓ El sistema académico se ha desarrollado de acuerdo a la metodología Scrum, que facilitó en el transcurso del tiempo al equipo scrum a equiparar el esfuerzo planificado que se sobrestimó en el sprint 5 con más relevancia, permitió además a realizar los procesos de desarrollo de manera rápida, controlada y sencilla. Posteriormente se implementó el sistema académico.
- ✓ Al finalizar el desarrollo del sistema académico de choferes profesionales “4 de octubre”, se realizó la evaluación con base a la norma internacional de métricas internas de calidad del producto de software ISO 9126-3, consecuentemente se determinó que el sistema académico es 90,48% usable, un 98.72% funcional, y 92.91% eficiente, obteniendo un promedio general de 94.04% en calidad.

RECOMENDACIONES

- ✓ Sugiero al sistema académico del sindicato de choferes profesionales “4 de octubre” crear módulos como: Calificaciones, Asistencia, que se completarían para formar un sistema informático que automatice todos los procesos de la institución.
- ✓ Se recomienda optar por el framework JSF para el desarrollo de aplicaciones web, por su robustez y simplicidad como herramienta tecnológica, facilitando el proceso de desarrollo considerablemente, además, se adapta a un elevado número entorno de construcción de software.
- ✓ Finalmente se recomienda para comprobar y verificar la calidad de producto software, aplicar métricas internas de calidad del producto de software basándose en la norma internacional ISO 9126-3.

BIBLIOGRAFÍA

ABRAN Alain et.al. Software Product Evaluation-Quality Characteristics and Guidelines for their Use. Ottawa, Canadá, noviembre, 2003. ISO/IEC 9126. [Consulta: 06-07-2016 08:23:46]. Disponible en: <http://link.springer.com/article/10.1023/A:1025869312943>

ANÓNIMO. *Historia de Scrum* [En línea]. Proyectosagiles.org, 19 agosto, 2008. [Consulta: 2017-03-22 08:49:19]. Disponible en: <https://proyectosagiles.org/historia-de-scrum/>

ANÓNIMO. *JavaServer Faces(JSF) | Marco de Desarrollo de la Junta de Andalucía* [En línea]. www.juntadeandalucia.es. Ficha Técnica, código: RECU-0101. [Consulta: 2017-03-30 23:59:12]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/101>

BORBÓN, Nuvia. *NORMA DE EVALUACIÓN ISO/IEC 9126 | EVALUACION DE SOFTWARE* [blog]. blogspot.com, 12 marzo, 2008. [Consulta: 2017-04-29 12:27:18]. Disponible en: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>

BURNS, Ed, & SCHALK Chris. *JavaServer Faces 2.0, The Complete Reference* [En línea]. Versión 2.0, McGraw Hill Professional, 2009. ISBN: 978-0-07-162510-4. [Consulta: 19-02-2017 18:23:40]. Disponible en: <https://books.google.com.ec/books?id=aNRq-8GIRYQC>

COLLA, Pedro. “Marco para evaluar el valor en metodología SCRUM” [En línea], 2012, (Argentina), pp. 1-3. [Consulta: 24-02-2017 11:41:12]. Disponible en: http://41jaiio.sadio.org.ar/sites/default/files/086_ASSE_2012.pdf

CONCHA, Laura, & TIUPUL, María. *Análisis comparativo de rendimiento de los frameworks de persistencia JPA y JDO en aplicaciones web JSF. Caso práctico: Sistema Integrado de Información para Control del Parque Automotor ECORAE* [En línea]. (tesis

pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela Ingeniería en Sistemas. Riobamba, Ecuador. 2014. pp. 136- 145[Consulta: 2017-01-23 05:22:48]. Disponible en:

<http://dspace.esPOCH.edu.ec/handle/123456789/3611>

DE AVILA, Neyder. *¿Qué es JSF?* [blog]. Brazil: prezi.com, 2013. [Consulta: 2017-02-22 13:21:39]. Disponible en: <https://prezi.com/2tgutw1n1vvk/que-es-jsf/>

DELPINO, Byron. *Diseño e implementación de una plataforma E-learning para la materia de tecnologías de software para Electrónica* [En línea] (tesis pregrado) Escuela Politécnica del Ejército, Departamento de Electrónica y Electrónica, Carrera de Ingeniería en Electrónica, Redes y Comunicación de Datos. 2013. [Consulta: 2017-02-23 02:01:14]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/6228>

ESCOBAR, Cynthian, Artículo Científico. - “Análisis comparativo de Frameworks JSF 2.0: Icefaces, Primefaces y Richfaces; para la implementación en el desarrollo del sistema de gestión de proyectos ambientales de la empresa KAYMANTA”. [En línea], 2014, (Ecuador). [Consulta: 2017-02-23 01:46:16]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/8162>

ESTAYNO, Marcelo et.al. Modelos y métricas para evaluar calidad de software [en línea]. Buenos Aires, Argentina, 2009. [Consulta: 11-209-2016 14:46:06]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/19762>

GARCÍA, Ana. *Evaluación de métricas de calidad del software sobre un programa Java* (Tesis) (maestría). Universidad Complutense de Madrid, Facultad de Informática, Departamento de Sistemas Informáticos y Computación. Madrid-España. 2010. pp. 7-8 [Consulta: 2017-03-23 10:02:55]. Disponible en: <http://eprints.ucm.es/11487/>

GARCÍA, Carlos, & PÉREZ, Miguel. *Diseño e implementación con tecnologías JSF y JPA de una web para un grupo de investigación en biotecnología* [En línea] (proyecto final de carrera).

Universitat Politècnica de València, Escuela Técnica Superior de Ingeniería Informática. Valencia-España. 2013. pp.28-29[Consulta: 2017-02-22 13:12:58]. Disponible en: <https://riunet.upv.es/handle/10251/31735>

GONZÁLEZ, Gabriel. *JavaServer Faces (JSF)* [blog]. Wordpress.com, 13 octubre, 2008. [Consulta: 2017-02-19 18:27:00]. Disponible en: <https://kalistog.wordpress.com/javaserver-faces-jsf/>

HANGALO, Joaquin. *Processo de desenvolvimento con JSF · Desenvolvimento de aplicações Web com JavaServer Faces* [blog]. [Consulta: 2017-02-28 18:17:22]. Disponible en: <https://joaquinhangalo.gitbooks.io/aplicacoes-web-com-javaserver-faces/content/processo-de-desenvolvimento-com-jsf.html>

JANSOBLOG, s.n. *Ciclo de Vida en JSF* [blog]. Wordpress, 23 octubre, 2007. [Consulta: 2016-12-28 18:17:22]. Disponible en: <https://jansoblog.wordpress.com/2007/10/23/ciclo-de-vida-en-jsf/>

JARAMILLO, Miriam. *Análisis Comparativo sobre los Frameworks MYFACES, ICEFACES y RICHFACES Aplicado al Sistema Nutricional de la ESPOCH* [En línea]. (tesis pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela Ingeniería en Sistemas. Riobamba, Ecuador. 2011. pp. 33-48 [Consulta: 2017-02-23 02:22:47]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/2530>
<http://docslide.net/documents/18t00503.html>

JUNTA DE ANDALUCÍA, s.n. *Marco de Desarrollo de la Junta de Andalucía (MADEJA) / Marco de Desarrollo de la Junta de Andalucía* [blog]. www.juntadeandalucia.es, 01 marzo, 2013. [Consulta: 2016-11-28 16:97:43]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/>

LAFOSSE, Jérôme. Struts 2: El framework de desarrollo de aplicaciones Java EE. Edi-
ediciones, mayo, 2010, ISBN: 978-2-7460-5542-1. [Consulta: 20-02-2017 07:22:00].

LEIVA, Geancarlo. *Sw de fábrica: Nota de arquitecto JEE: el estándar Java Server Faces
para manipular la interfaz gráfica de nuestras aplicaciones web* [En línea]. Blogger, 31 enero,
2014. [Consulta: 2017-02-28 18:01:26]. Disponible en:
<http://swdefabrica.blogspot.com/2014/01/nota-de-arquitecto-jee-el-estandar-java.html>

MENA, Gonzalo. *ISO 9126-3: Métricas Internas de la Calidad del Producto de Software*
[blog]. México: <http://mena.com.mx>, marzo, 2006. [Consulta: 2017-03-23 09:20:49].
Disponible en:
http://mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126-3/

MOLINA, Inma, *Ciclo de Vida en JSF* [blog]. Wordpress, 23 octubre, 2007. [Consulta: 2017-
02-28 18:07:23]. Disponible en:
<https://jansoblog.wordpress.com/2007/10/23/ciclo-de-vida-en-jsf/>

MONTAÑO, Deivid. *SCRUM: Ensayo sobre Scrum* [blog]. Blogger, 20 julio, 2011. [Consulta:
2017-03-22 08:52:00]. Disponible en:
<http://scrum-ing-soft.blogspot.com/2011/07/ensayo-sobre-scrum.html>

MUNOZ, Jose. *JSF, características principales, ventajas y puntos a destacar* [blog].
Wordpress.com, 30 julio, 2012. [Consulta: 2017-03-30 23:05:28] Disponible en:
[https://josemmsimo.wordpress.com/2012/07/30/jsf-caracteristicas-principales-ventajas-y-
puntos-a-destacar/](https://josemmsimo.wordpress.com/2012/07/30/jsf-caracteristicas-principales-ventajas-y-puntos-a-destacar/)

PALACIO, Juan. *Gestión de proyectos Scrum Manager (Scrum Manager I y II)* [En línea].
Versión 2.5. de la edición: Scrum Manager, 2014. pp. 13-69 [Consulta: 26-02-2017 12:46:49].
Disponible en: http://www.scrummanager.net/files/sm_proyecto.pdf

RODRIGUEZ, Ricardo. *Errores comunes de programación: Segunda Parte* [blog]. PMOinformatica.com, 24 octubre, 2012. [Consulta: 2017-03-07 09:40:49]. Disponible en: <http://www.pmoinformatica.com/2012/10/errores-de-programacion-comunes-segunda.html>

SALANOVA, Pilar. *Modelos de calidad web, Clasificación de métricas.* [En línea]. (Proyecto de fin de carrera de Ingeniería Informática). Universidad Nacional de Educación a Distancia, Escuela Técnica Superior de Ingeniería Informática. Madrid- España, 2016. [Consulta: 10-22-2016 08:56:06]. Disponible en: http://www.issi.uned.es/CalidadSoftware/Noticias/PFC_1.doc

TACHOIRES, Rodrigo, *Conceptos de Scrum* [blog]. SlideShare, 19 abril, 2016. [Consulta: 2017-02-28 18:54:15]. Disponible en: <https://www.slideshare.net/RodrigoTachaires/conceptos-de-scrum>

TRIGAS, Manuel. *Metodología Scrum* [En línea]. Ana Cristina Domingo Troncho, 2012. pp. 33-52 [Consulta: 22-02-2017 22:43:39]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

VALDIVIEZO, Patricia, & GUACH, María. *Análisis Comparativo de Tecnologías de Aplicaciones Web en el Entorno JSF y ADF. Caso Práctico: IESS de Riobamba – Chimborazo* [En línea]. (tesis pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela Ingeniería en Sistemas. Riobamba, Ecuador. 2011. pp. 56-57 [Consulta: 2017-02-23 02:22:47]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/1490>

VILLA, Diego. *Análisis del rendimiento entre framework Java Server Face (JDF) y JBOSS-SEAM. Caso práctico en el sistema socioeconómico* [En línea]. (tesis pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela Ingeniería en Sistemas. Riobamba, Ecuador. 2014. [Consulta: 2017-02-22 12:46:31]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/3615>

WILLIAM, s.n. *Estándares de Programación ... Mañana, hoy y siempre ...* [blog]. Blogspot.com, 28 junio, 2010. [Consulta: 2017-03-06 16:35:56]. Disponible en:

<http://yolopuedohacer.blogspot.com/2010/06/estandares-de-programacion-manana-hoy-y.html>

ZAMBON, Giulio, & SEKLER, Michael. *Beginning JSP, JSF and Tomcat Web Development: From Novice to Professional* [en línea]. Edición 4. Berkeley- California: Apress, 03 marzo, 2008. ISBN: 978-1-4302-0465-7. [Consulta: 2017-02-25 02:06:18]. Disponible en: https://books.google.com.ec/books?id=XM0gswEACAAJ&hl=es&source=gbs_book_other_versions

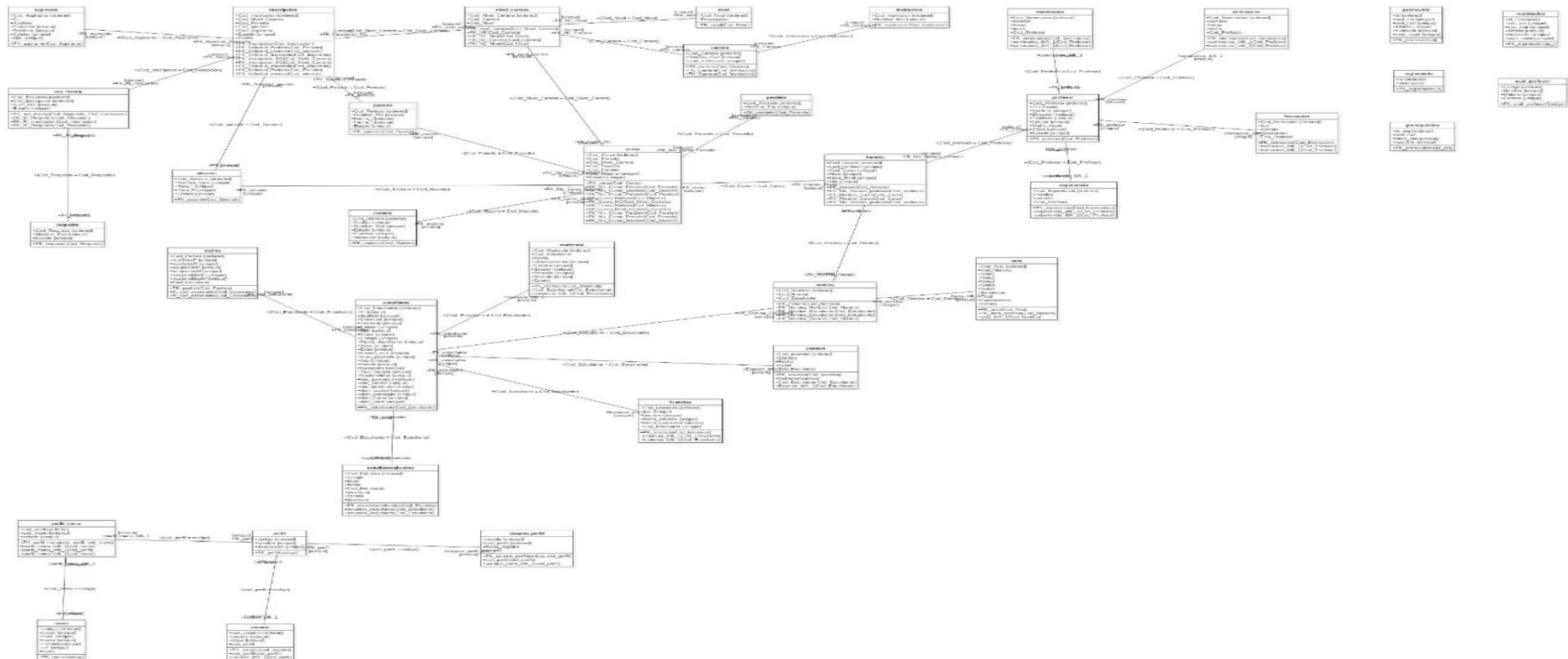
ZAMBON, Giulio. *Beginning JSP, JSF and Tomcat Java Web Development* [En línea]. Berkeley-California: Apress, 2008. ISBN: 978-1-4302-4623-7. [Consulta: 2017-03-28 18:15:39]. Disponible en: http://zambon.com.au/non_fiction/books_and_manuals/jsp_jsf_tomcat_2/

1DA4, Diagramas UML [blog]. SlideShare, 19 octubre, 2009. [Consulta: 2017-03-27 19:39:27]. Disponible en: <https://es.slideshare.net/e1da4/diagramas-uml>

ANEXOS

Anexo A: Modelo físico de la base de datos

En consecuencia, de la implementación del DER (Diagrama Entidad Relación), se ha creado y generado el siguiente esquema:



Realizado por: Merino, Cristian, 2017

Anexo B: Manual técnico

Adjunto el manual técnico del sistema académico en el CD del presente trabajo de titulación con la siguiente ruta: SistemaAcadémico/Manuales/Manual-Técnico

Anexo C: Manual de usuario

Adjunto el manual de usuario del sistema académico en el CD del presente trabajo de titulación con la siguiente ruta: SistemaAcadémico/Manuales/Manual-de-Usuario

Anexo D: Reporte orden de matricula



ESCUELA DE CONDUCTORES PROFESIONALES "4 DE OCTUBRE" DEL CANTÓN PENIPE

CATEGORIA TIPO "C"

MATRÍCULA No: 32

FOLIO No: 32



En la ciudad de Penipe, el día 2016-05-05

El Señor Cristian Geovanny nacido/a en la parroquia Iizarzaburu, cantón Penipe, provincia de Chimborazo, nacido/a el 1986-01-06 con cédula de ciudadanía N° 0603895871, que ha realizado los estudios en ella Maldonado, de la parroquia Maldonado, cantón Riobamba, provincia Chimborazo, es hijo/a del Sr. Luis, de nacionalidad ecuatoriano, de profesión Docente, y de la Sra. Rosita, de nacionalidad ecuatoriana, de profesión Estilista .

Obtiene la presente matricula para entrar al Curso de Formación de Conductores Profesionales Carrera en la Escuela de Conductores Profesionales "4 de Octubre " del Cantón Penipe, Período Académico Período.

Dirección domiciliaria del estudiante:

Calle: Vicente Solano y

Barrio: tapi

Parroquia: Iizarzaburu

Ciudad: Riobamba

Teléfono convencional: 0992593250

Teléfono móvil: 032 601044

SECRETARIO

ESTUDIANTE

Realizado por: Cristian Merino. 2017

Anexo E: Encuesta a usuarios del Sistema

Son las preguntas reflejadas como “propósito” en cada métrica de evaluación del producto software del presente proyecto.

Anexo F: Diagramas UML

DIAGRAMAS DE CASOS DE USOS

Es la representación gráfica de los Casos de Uso, partiendo de la nomenclatura definida en un Lenguaje de Modelo Unificado UML. Representamos actores, los casos de uso propiamente dicho, las interacciones de las entidades u objetos y las relaciones que existen entre los mismos.

1. Caso de uso Esencial

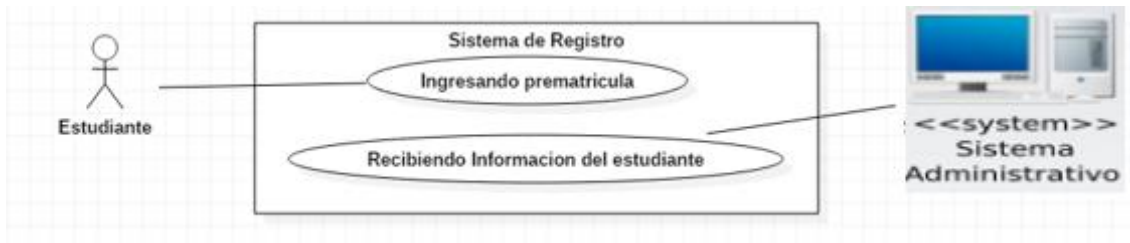
Define o narra a los actores utilizando el sistema de forma general, es decir en base a los requisitos funcionales, los mismos que son redactados en lenguaje natural a partir de las interacciones del usuario.

Planillas de los requerimientos funcional

1.1 El sistema permitirá registrar su pre matricula (Inscripción)

Identificador caso de uso	R1-REGUSER
Nombre del caso de uso	Registro pre matricula
Actores	Estudiante, Sistema de matriculación
Propósito	Ingreso de datos estudiante para su pre matricula
Visión General	El usuario ingresa al sistema para realizar la prematricula. (inscripción online) 1. El estudiante ingresa 2. El sistema de matriculación recibe la información del estudiante para ser procesada
Tipo	Primario

Realizado por: Cristian Merino. 2017

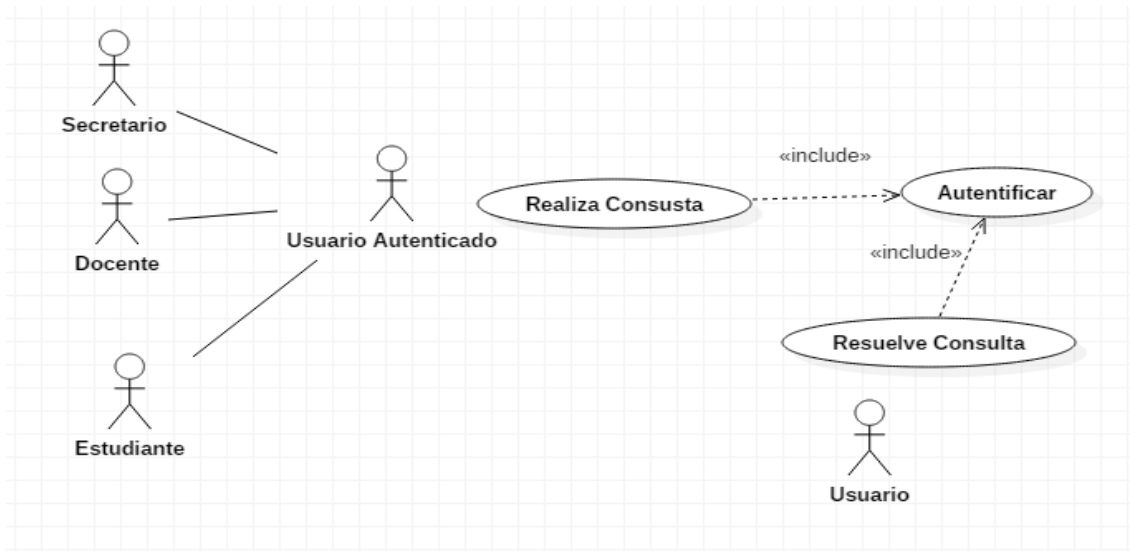


Realizado por: Cristian Merino. 2017

1.2 El sistema permitirá la autenticación en el sistema académico

Identificador caso de uso	R2-AUTENTICATION
Nombre del caso de uso	Inicio de sesión
Actores	Secretaria, Profesor, Estudiante
Propósito	Ingresa al sistema académico.
Visión General	<p>El usuario se identifica, ingresa al sistema.</p> <ol style="list-style-type: none"> 1. Cualquiera de los tipos de usuario entra a inicio de sesión. 2. Ingresa el usuario y la clave 3. Se valida la existencia del usuario 4. Autentifica e inicia la sesión
Tipo	Primario

Realizado por: Cristian Merino. 2017

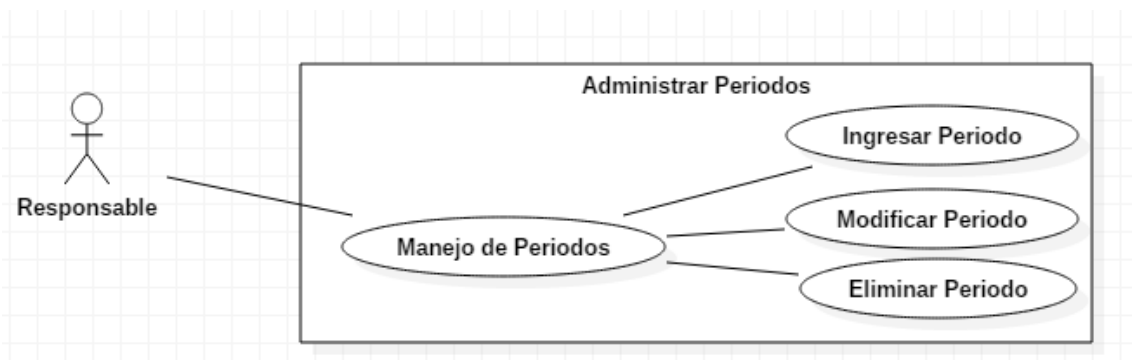


Realizado por: Cristian Merino. 2017

1.3 El sistema permitir administrar los periodos académicos de la institución

Identificador caso de uso	R3-ADMPERIODO
Nombre del caso de uso	Administración de períodos
Actores	Usuario responsable
Propósito	Realizar el plan de períodos
Visión General	<ol style="list-style-type: none"> 1. El responsable entra a manejo de períodos 2. Ingresa nuevo período 3. Agrega, elimina o modifica período
Tipo	Primario

Realizado por: Cristian Merino. 2017

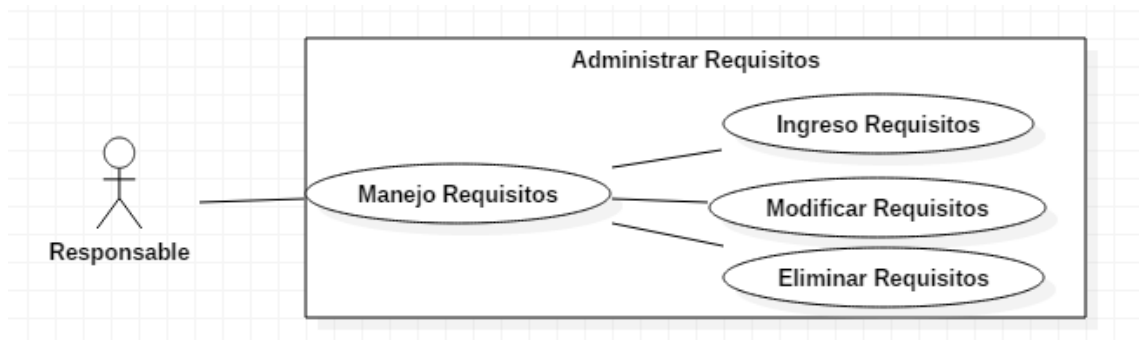


Realizado por: Cristian Merino. 2017

1.4 El sistema permitirá administrar requisitos

Identificador caso de uso	R4-ADMREQUISITO
Nombre del caso de uso	Administrar requisitos
Actores	Usuario responsable
Propósito	Actualización de los requisitos
Visión General	<p>El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de los requisitos.</p> <ol style="list-style-type: none"> 1. El responsable entra a manejo de requisitos 2. Ingresa nuevo requisito 3. Agrega, elimina o modifica requisito
Tipo	Primario

Realizado por: Cristian Merino. 2017

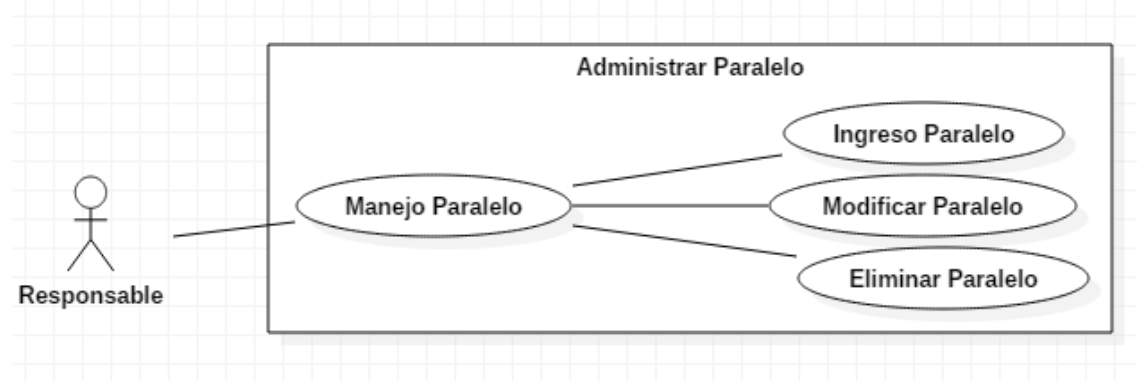


Realizado por: Cristian Merino. 2017

1.5 El sistema permitirá administrar paralelo.

Identificador caso de uso	R5-ADMPARALELO
Nombre del caso de uso	Administración de paralelo
Actores	Usuario responsable.
Propósito	Actualización de paralelo
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de paralelo. 1. El responsable entra a manejo de paralelos 2. Ingresa nuevo paralelo 3. Agrega, elimina o modifica paralelo
Tipo	Primario

Realizado por: Cristian Merino. 2017



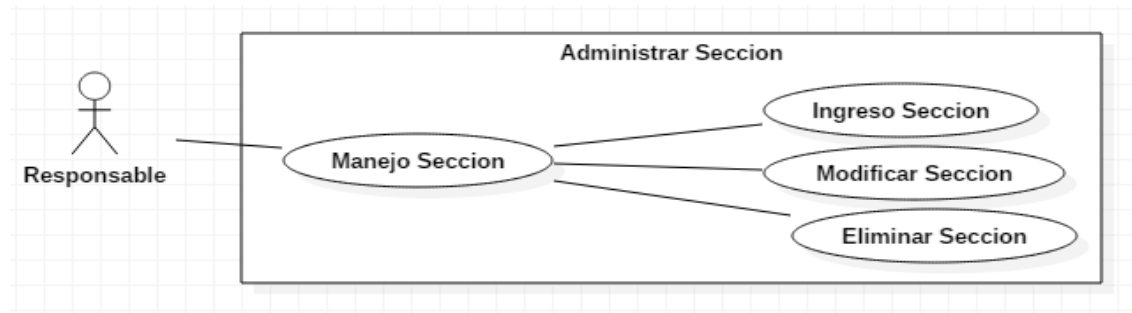
Realizado por: Cristian Merino. 2017

1.6 El sistema permitirá administrar sección de horario

Identificador caso de uso	R6-ADMSECCION
Nombre del caso de uso	Administración de sección
Actores	Usuario responsable.

Propósito	Actualización de sección
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de sección. 1. El responsable entra a manejo de secciones 2. Ingresa nueva sección 3. Agrega, elimina o modifica sección
Tipo	Primario

Realizado por: Cristian Merino. 2017

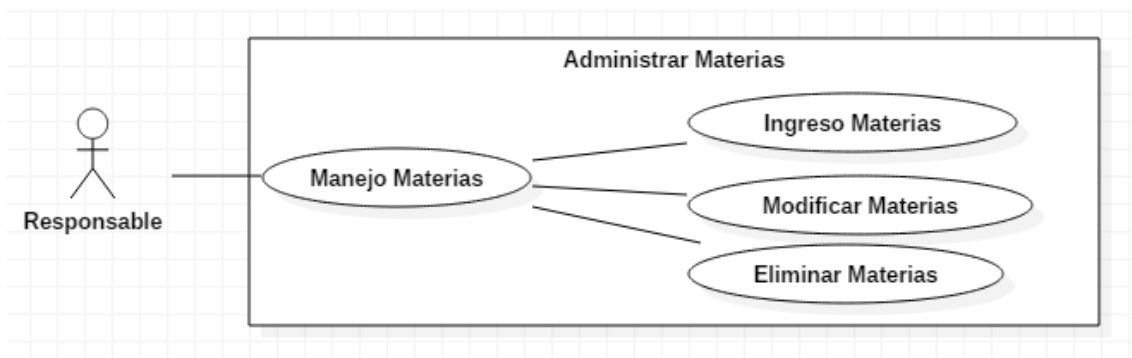


Realizado por: Cristian Merino. 2017

1.7 El sistema permitirá administrar materia

Identificador caso de uso	R7-ADMMATERIA
Nombre del caso de uso	Administración de materia
Actores	Usuario responsable.
Propósito	Actualización de paralelo
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de materia. 1. El responsable entra a manejo de materias 2. Ingresa nueva materia 3. Agrega, elimina o modifica materia.
Tipo	Primario

Realizado por: Cristian Merino. 2017

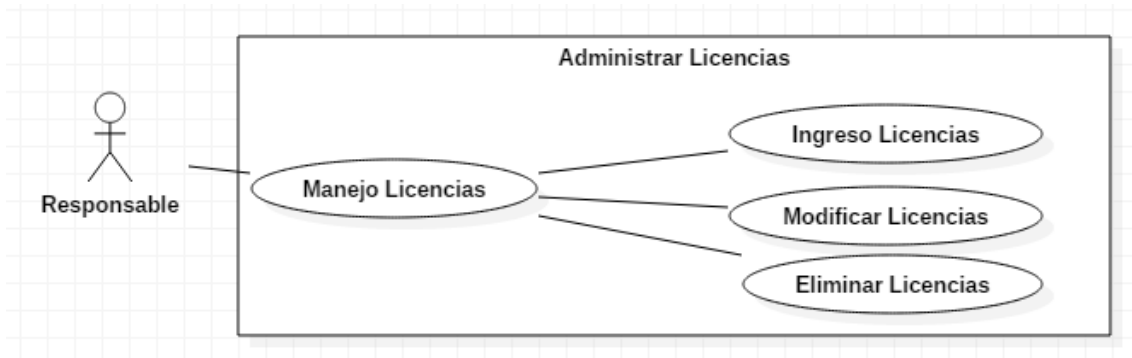


Realizado por: Cristian Merino. 2017

1.8 El sistema permitirá administrar licencia

Identificador caso de uso	R8-ADMLICENCIA
Nombre del caso de uso	Administración de licencia
Actores	Usuario responsable.
Propósito	Actualización de licencia
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de Licencia. <ol style="list-style-type: none"> 1. El responsable entra a manejo de licencias 2. Ingresar nueva licencia 3. Agregar, eliminar o modificar licencia.
Tipo	Primario

Realizado por: Cristian Merino. 2017

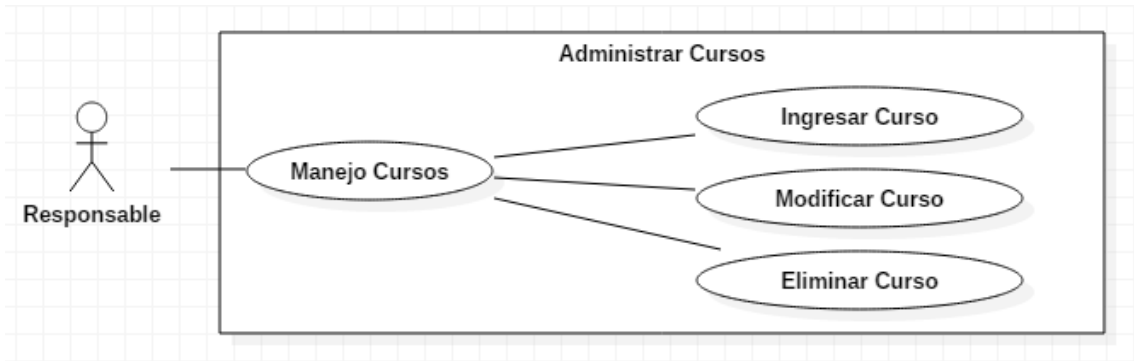


Realizado por: Cristian Merino. 2017

1.9 El sistema permitirá administrar curso

Identificador caso de uso	R9-ADMCURSO
Nombre del caso de uso	Administración de curso.
Actores	Usuario responsable.
Propósito	Actualización de curso
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de curso. <ol style="list-style-type: none"> 1. El responsable entra a manejo de cursos 2. Ingresar nuevo curso 3. Agregar, eliminar o modificar curso
Tipo	Primario

Realizado por: Cristian Merino. 2017

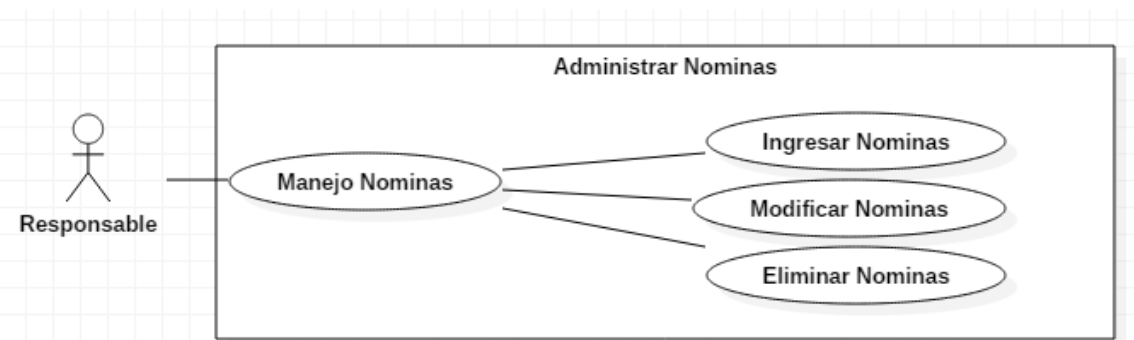


Realizado por: Cristian Merino. 2017

1.10 El sistema permitirá administrar nómina

Identificador caso de uso	R10-ADMNOMINA
Nombre del caso de uso	Administración de nómina.
Actores	Usuario responsable.
Propósito	Actualización de nómina.
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de nómina. 1. El responsable entra a manejo de nóminas 2. Ingresa nueva nómina 3. Agrega, elimina o modifica nómina
Tipo	Primario

Realizado por: Cristian Merino. 2017



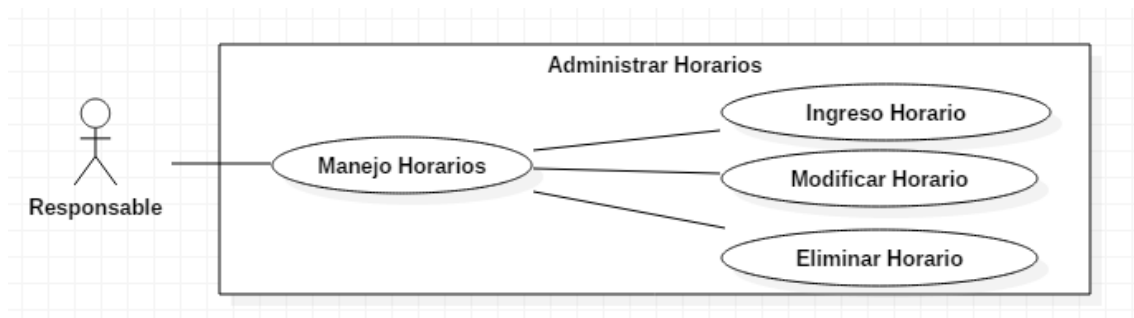
Realizado por: Cristian Merino. 2017

1.11 El sistema permitirá administrar horario.

Identificador caso de uso	R11-ADMHORARIO
Nombre del caso de uso	Administración de horario.

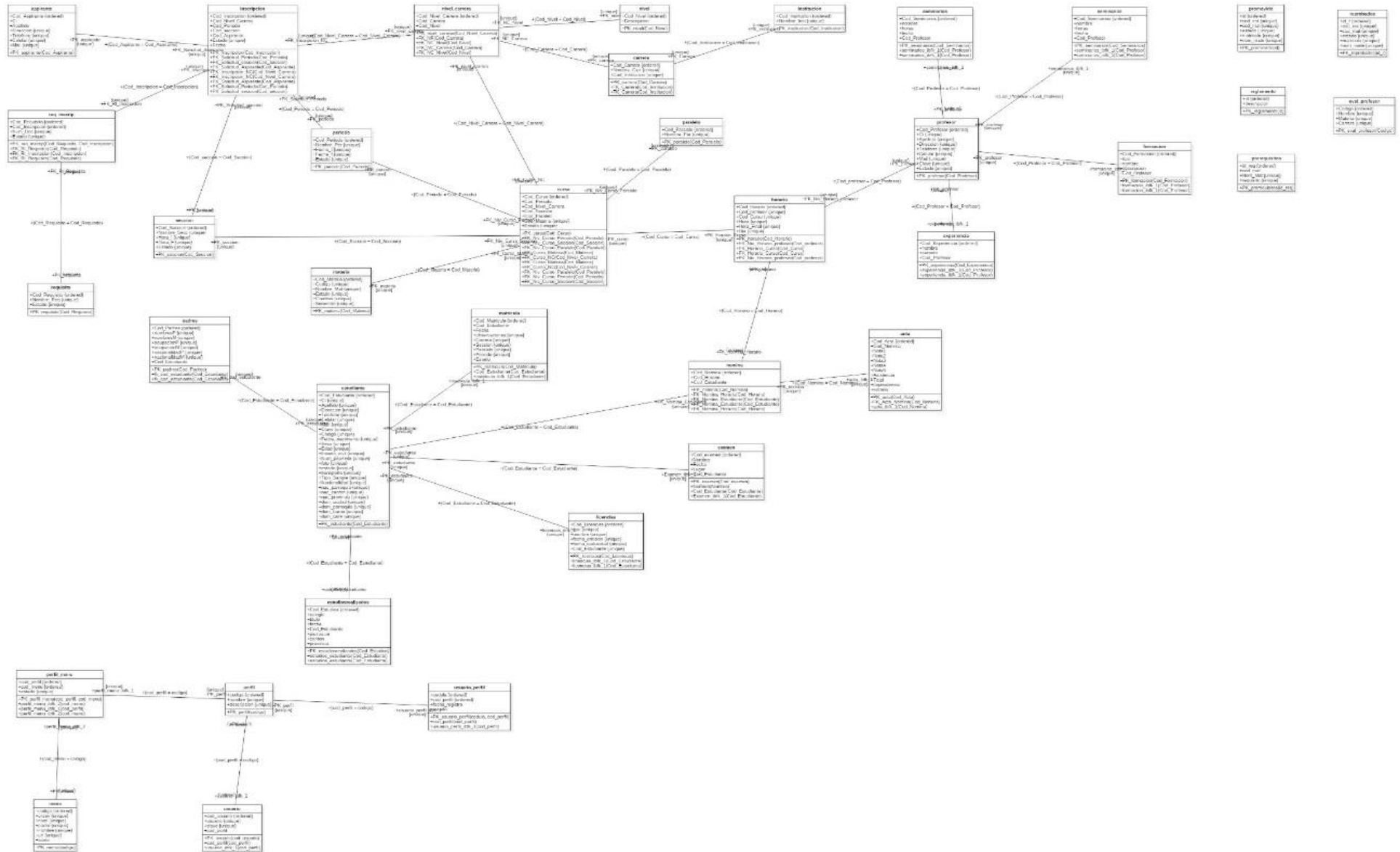
Actores	Usuario responsable.
Propósito	Actualización de horario
Visión General	El usuario ingresa al sistema previa autenticación y autorización, para realizar la actualización de horario. 1. El responsable entra a manejo de horarios 2. Ingresa nuevo horario 3. Agrega, elimina o modifica horario
Tipo	Primario

Realizado por: Cristian Merino. 2017



Realizado por: Cristian Merino. 2017

2. Diagrama de clases



3. Diagrama de objetos

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Acta	Entero	10	PRIMARY			La columna Cod_Acta es clave primaria de acta	1
Cod_Nomina	Entero	10		acta_ibfk_1	nomina	Clave foránea que se vincula con la tabla nomina	1
Nota1	decimal					Almacena la información de Nota1	0.00
Nota2	decimal					Almacena la información de Nota2	0.00
Nota3	decimal					Almacena la información de Nota3	0.00
Nota4	decimal					Almacena la información de Nota4	0.00
Nota5	decimal					Almacena la información de Nota5	0.00
Asistencia	Entero	10				Almacena la información de Asistencia	0
Total	decimal					Almacena la información de Total	0.00
Equivalencia	Cadena	25				Almacena la información de Equivalencia	BAJA
Estado	Entero	10				Almacena la información de Estado	0

Ejemplo de entrada de datos:[Cod_Acta] => "1", [Cod_Nomina] => "1", [Nota1] => "0.00", [Nota2] => "0.00", [Nota3] => "0.00", [Nota4] => "0.00", [Nota5] => "0.00", [Asistencia] => "0", [Total] => "0.00", [Equivalencia] => "BAJA", [Estado] => "0"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Aspirante	Entero	10	PRIMARY			La columna Cod_Aspirante es clave primaria de aspirante	1255
Ci	Cadena	13				Almacena la información de Cédula	2222222222
Apellido	Cadena	100				Almacena la información de Apellido	Perez
Direccion	Cadena	100				Almacena la información de Direccion	Direccion Perez
Telefono	Cadena	10				Almacena la información de Telefono	333333333
Celular	Cadena	10				Almacena la información de Celular	966666666
Mail	Cadena	100				Almacena la información de Mail	perez@servidor.com

Ejemplo de entrada de datos:[Cod_Aspirante] => "1255", [Ci] => "2222222222", [Apellido] => "Perez", [Direccion] => "Direccion Perez", [Telefono] => "0333333333", [Celular] => "0966666666", [Mail] => "perez@servidor.com"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Carrera	Entero	10	PRIMARY			La columna Cod_Carrera es clave primaria de carrera	1
Nombre_Carr	Cadena	50				Almacena la información de Nombre de Carrera	CARRERA 1
Cod_Institucion	Entero	10		FK_Carrera	institucion	Clave foránea que se vincula con la tabla institucion	1

Ejemplo de entrada de datos:[Cod_Carrera] => "1", [Nombre_Carr] => "CARRERA 1", [Cod_Institucion] => "1"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Curso	Entero	10	PRIMARY			La columna Cod_Curso es clave primaria de curso	408
Cod_Periodo	Entero	10		FK_Niv_Curso_Periodo	periodo	Clave foránea que se vincula con la tabla periodo	19
Cod_Nivel_Carrera	Entero	10		FK_Curso_NC	nivel_carrera	Clave foránea que se vincula con la tabla nivel_carrera	1
Cod_Seccion	Entero	10		FK_Niv_Curso_Seccion	seccion	Clave foránea que se vincula con la tabla seccion	5
Cod_Paralelo	Entero	10		FK_Niv_Curso_Paralelo	paralelo	Clave foránea que se vincula con la tabla paralelo	9
Cod_Materia	Entero	10		FK_Curso_Materia	materia	Clave foránea que se vincula con la tabla materia	84
Estado	tinint					Almacena la información de Estado	1

Ejemplo de entrada de datos:[Cod_Curso] => "408", [Cod_Periodo] => "19", [Cod_Nivel_Carrera] => "1", [Cod_Seccion] => "5", [Cod_Paralelo] => "9", [Cod_Materia] => "84", [Estado] => "1"

Columna	Tipo	Tam	PK	FK	TABLA	DESCRIPCION	EJEMPLO DE DATO
Cod_Estudiante	Entero	10	PRIMARY			La columna Cod_Estudiante es clave primaria de estudiante	858
Ci	Cadena	13				Almacena la información de Cédula	0
Apellido	Cadena	100				Almacena la información de Apellido	Perez
Direccion	Cadena	100				Almacena la información de Direccion	Calle Uno 123
Telefono	Cadena	10				Almacena la información de Telefono	999999999
Celular	Cadena	10				Almacena la información de Celular	111111111
Mail	Cadena	100				Almacena la información de Mail	usuario@servidor.com
Clave	Cadena	20				Almacena la información de Clave	12345
Codigo	Cadena	30				Almacena la información de Codigo	1
Fecha_nacimiento	Fecha					Almacena la información de Fecha de nacimiento	32874
Sexo	Entero	10				Almacena la información de Sexo	1
Edad	Entero	10				Almacena la información de Edad	36
Estado_civil	Entero	10				Almacena la información de Estado civil	1
Num_papeleta	Cadena	12				Almacena la información de Número de papeleta de votación	1
foto	Cadena	100				Almacena la información de foto	foto.jpg
estado	bit					Almacena la información de estado	1
transporte	tinint					Almacena la información de transporte	1
Tipo_Sangre	Cadena	4				Almacena la información de Tipo Sangre	A+
Nacionalidad	Cadena	15				Almacena la información de Nacionalidad	ECUATORIANA
nac_parroquia	Cadena	200				Almacena la información de parroquia nacimiento	PARROQUIA X
nac_canton	Cadena	200				Almacena la información de canton nacimiento	CANTON X
nac_provincia	Cadena	200				Almacena la información de provincia nacimiento	PROVINCIA X
dom_ciudad	Cadena	200				Almacena la información de ciudad domicilio	CIUDAD DOMICILIO
dom_parroquia	Cadena	200				Almacena la información de parroquia domicilio	PARROQUIA DOMICILIO
dom_barrio	Cadena	200				Almacena la información de barrio domicilio	BARRIO DOMICILIO
dom_calle	Cadena	200				Almacena la información de calle domicilio	CALLE DOMICILIO

Ejemplo de entrada de datos:[Cod_Estudiante] => "858", [Ci] => "0000000000", [Apellido] => "Perez", [Direccion] => "Calle Uno 123", [Telefono] => "0999999999", [Celular] => "0111111111", [Mail] => "usuario@servidor.com", [Clave] => "12345", [Codigo] => "00001", [Fecha_nacimiento] => "1990-01-01", [Sexo] => "1", [Edad] => "36", [Estado_civil] => "1", [Num_papeleta] => "1", [foto] => "foto.jpg", [estado] => "1", [transporte] => "1", [Tipo_Sangre] => "A+", [Nacionalidad] => "ECUATORIANA", [nac_parroquia] => "PARROQUIA X", [nac_canton] => "CANTON X", [nac_provincia] => "PROVINCIA X", [dom_ciudad] => "CIUDAD DOMICILIO", [dom_parroquia] => "PARROQUIA DOMICILIO", [dom_barrio] => "BARRIO DOMICILIO", [dom_calle] => "CALLE DOMICILIO"

Tabla estudiosrealizados							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Estudios	Entero	10	PRIMARY			La columna Cod_Estudios es clave primaria de estudiosrealizados	17
colegio	Cadena	80				Almacena la información de colegio	Colegio N° 1
titulo	Cadena	100				Almacena la información de titulo	Contador
fecha	Fecha					Almacena la información de fecha	42370
Cod_Estudiante	Entero	10		estudios_estudiante	estudiante	Clave foránea que se vincula con la tabla estudiante	858
parroquia	Cadena	200				Almacena la información de parroquia	Parroquia 1
canton	Cadena	200				Almacena la información de canton	Canton 1
provincia	Cadena	200				Almacena la información de provincia	Provincia 1

Ejemplo de entrada de datos:[Cod_Estudios] => "17", [colegio] => "Colegio N° 1", [titulo] => "Contador", [fecha] => "2016-01-01", [Cod_Estudiante] => "858", [parroquia] => "Parroquia 1", [canton] => "Canton 1", [provincia] => "Provincia 1"

Tabla eval_profesor							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Codigo	Entero	10	PRIMARY			La columna Codigo es clave primaria de eval_profesor	0
Nombre	Cadena	50				Almacena la información de Nombre	Nombre Evaluación
Materia	Cadena	50				Almacena la información de Materia	Materia X
Carrera	Cadena	50				Almacena la información de Carrera	Carrera x

Ejemplo de entrada de datos:[Codigo] => "0", [Nombre] => "Nombre Evaluación", [Materia] => "Materia X", [Carrera] => "Carrera x"

Tabla examen							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_examen	Entero	10	PRIMARY			La columna Cod_examen es clave primaria de examen	7
Nombre	Cadena	30				Almacena la información de Nombre	Examen 1
Fecha	Fecha y Hora					Almacena la información de Fecha	42370,50069
Lugar	Cadena	30				Almacena la información de Lugar	institución
Cod_Estudiante	Entero	10		Examen_ibfk_1	estudiante	Clave foránea que se vincula con la tabla estudiante	858

Ejemplo de entrada de datos:[Cod_examen] => "7", [Nombre] => "Examen 1", [Fecha] => "2016-01-01 12:01:00", [Lugar] => "institución", [Cod_Estudiante] =>

Tabla experiencia							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Experiencia	Entero	10	PRIMARY			La columna Cod_Experiencia es clave primaria de experiencia	1
nombre	Cadena	80				Almacena la información de nombre	Experiencia 1
periodo	Cadena	20				Almacena la información de periodo	2015
Cod_Profesor	Entero	10		experiencia_ibfk_1	profesor	Clave foránea que se vincula con la tabla profesor	86

Ejemplo de entrada de datos:[Cod_Experiencia] => "1", [nombre] => "Experiencia 1", [periodo] => "2015", [Cod_Profesor] => "86"

Tabla formacion							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Formacion	Entero	10	PRIMARY			La columna Cod_Formacion es clave primaria de formacion	28
tipo	Cadena	60				Almacena la información de tipo	Primaria
nombre	Cadena	60				Almacena la información de nombre	Relaciones Humanas
descripcion	Cadena	60				Almacena la información de descripcion	Curso de Relaciones Humanas
Cod_Profesor	Entero	10		formacion_ibfk_1	profesor	Clave foránea que se vincula con la tabla profesor	86

Ejemplo de entrada de datos:[Cod_Formacion] => "28", [tipo] => "Primaria", [nombre] => "Relaciones Humanas", [descripcion] => "Curso de Relaciones Humanas", [Cod_Profesor] => "86"

Tabla horario							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Horario	Entero	10	PRIMARY			La columna Cod_Horario es clave primaria de horario	1396
Cod_profesor	Entero	10		FK_Niv_Horario_profesor	profesor	Clave foránea que se vincula con la tabla profesor	86
Cod_Curso	Entero	10		FK_Horario_Curso	curso	Clave foránea que se vincula con la tabla curso	408
Hora	Hora					Almacena la información de Hora	0,33333333
Hora_Final	Hora					Almacena la información de Hora Final	0,375
Dia	Cadena	10				Almacena la información de Dia	LUNES

Ejemplo de entrada de datos:[Cod_Horario] => "1396", [Cod_profesor] => "86", [Cod_Curso] => "408", [Hora] => "08:00:00", [Hora_Final] => "09:00:00", [Dia] => "LUNES"

Tabla inscripcion							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Inscripcion	Entero	10	PRIMARY			La columna Cod_Inscripcion es clave primaria de inscripcion	1277
Cod_Nivel_Carrera	Entero	10		FK_Inscripcion_NC	nivel_carrera	Clave foránea que se vincula con la tabla nivel_carrera	1
Cod_Periodo	Entero	10		FK_Solicitud_Periodo	periodo	Clave foránea que se vincula con la tabla periodo	19
Cod_seccion	Entero	10		FK_Solicitud_seccion	seccion	Clave foránea que se vincula con la tabla seccion	5
Cod_Aspirante	Entero	10		FK_Solicitud_Aspirante	aspirante	Clave foránea que se vincula con la tabla aspirante	1255
Estado	Entero	10				Almacena la información de Estado	1
Fecha	timestamp					Almacena la información de Fecha	42370

Ejemplo de entrada de datos:[Cod_Inscripcion] => "1277", [Cod_Nivel_Carrera] => "1", [Cod_Periodo] => "19", [Cod_seccion] => "5", [Cod_Aspirante] => "1255", [Estado] => "1", [Fecha] => "2016-01-01 00:00:00"

Tabla institucion							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Institucion	Entero	10	PRIMARY			La columna Cod_Institucion es clave primaria de institucion	1
Nombre_Inst	Cadena	50				Almacena la información de Nombre Institución	INSTITUCION 1

Ejemplo de entrada de datos:[Cod_Institucion] => "1", [Nombre_Inst] => "INSTITUCION 1"

Tabla licencias							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Licencias	Entero	10	PRIMARY			La columna Cod_Licencias es clave primaria de licencias	5
tipo	Cadena	50				Almacena la información de tipo	A
nombre	Cadena	100				Almacena la información de nombre	SPORTMAN
fecha_emision	Fecha					Almacena la información de fecha emision	42370
fecha_caducidad	Fecha					Almacena la información de fecha caducidad	43831
Cod_Estudiante	Entero	10		licencias_ibfk_1	estudiante	Clave foránea que se vincula con la tabla estudiante	858
Ejemplo de entrada de datos:[Cod_Licencias] => "5", [tipo] => "A", [nombre] => "SPORTMAN", [fecha_emision] => "2016-01-01", [fecha_caducidad] => "2020-01-01", [Cod_Estudiante] => "858"							

Tabla materia							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Materia	Entero	10	PRIMARY			La columna Cod_Materia es clave primaria de materia	84
Codigo	Cadena	30				Almacena la información de Codigo	M0001
Nombre_Mat	Cadena	100				Almacena la información de Nombre Materia	Materia de Ejemplo
Estado	tinyint					Almacena la información de Estado	1
Creditos	Entero	10				Almacena la información de Creditos	10
Semestre	Entero	10				Almacena la información de Semestre	10
Ejemplo de entrada de datos:[Cod_Materia] => "84", [Codigo] => "M0001", [Nombre_Mat] => "Materia de Ejemplo", [Estado] => "1", [Creditos] => "10", [Semestre] => "10"							

Tabla matricula							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Matricula	Entero	10	PRIMARY			La columna Cod_Matricula es clave primaria de matricula	36
Cod_Estudiante	Entero	10		matricula_ibfk_1	estudiante	Clave foránea que se vincula con la tabla estudiante	858
Fecha	Fecha y Hora					Almacena la información de Fecha	42370
observaciones	Cadena	500				Almacena la información de observaciones	Ninguna Observación
Carrera	Cadena	100				Almacena la información de Carrera	Carrera 1
Seccion	Cadena	100				Almacena la información de Seccion	Sección 1
Paralelo	Cadena	2				Almacena la información de Paralelo	A
Periodo	Cadena	100				Almacena la información de Periodo	2016
Estado	Entero	10				Almacena la información de Estado	1
Ejemplo de entrada de datos:[Cod_Matricula] => "36", [Cod_Estudiante] => "858", [Fecha] => "2016-01-01 00:00:00", [observaciones] => "Ninguna Observación", [Carrera] => "Carrera 1", [Seccion] => "Sección 1", [Paralelo] => "A", [Periodo] => "2016", [Estado] => "1"							

Tabla menu							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
codigo	Entero	10	PRIMARY			La columna codigo es clave primaria de menu	34
orden	tinyint					Almacena la información de orden	1
nivel	tinyint					Almacena la información de nivel	1
padre	Entero	10				Almacena la información de padre	0
nombre	Cadena	100				Almacena la información de nombre	Inicio
url	Cadena	200				Almacena la información de url	index.php/inicio
icono	Cadena	50				Almacena la información de icono	icono.jpg
Ejemplo de entrada de datos:[codigo] => "34", [orden] => "1", [nivel] => "1", [padre] => "0", [nombre] => "Inicio", [url] => "index.php/inicio", [icono] => "icono.jpg"							

Tabla nivel							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Nivel	Entero	10	PRIMARY			La columna Cod_Nivel es clave primaria de nivel	1
Descripcion	Cadena	50				Almacena la información de Descripcion	PRIMERO
Ejemplo de entrada de datos:[Cod_Nivel] => "1", [Descripcion] => "PRIMERO"							

Tabla nivel_carrera							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Nivel_Carrera	Entero	10	PRIMARY			La columna Cod_Nivel_Carrera es clave primaria de nivel_carrera	1
Cod_Carrera	Entero	10		FK_NC_Carrera	carrera	Clave foránea que se vincula con la tabla carrera	1
Cod_Nivel	Entero	10		FK_NC_Nivel	nivel	Clave foránea que se vincula con la tabla nivel	1
Ejemplo de entrada de datos:[Cod_Nivel_Carrera] => "1", [Cod_Carrera] => "1", [Cod_Nivel] => "1"							

Tabla nomina							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Nomina	Entero	10	PRIMARY			La columna Cod_Nomina es clave primaria de nomina	1
Cod_Horario	Entero	10		FK_Nomina_Horario	horario	Clave foránea que se vincula con la tabla horario	1396
Cod_Estudiante	Entero	10		FK_Nomina_Estudiante	estudiante	Clave foránea que se vincula con la tabla estudiante	858
Ejemplo de entrada de datos:[Cod_Nomina] => "1", [Cod_Horario] => "1396", [Cod_Estudiante] => "858"							

Tabla padres							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Padres	Entero	10	PRIMARY			La columna Cod_Padres es clave primaria de padres	28
nombresP	Cadena	80				Almacena la información de nombres Padre	Juan Perez
nombresM	Cadena	80				Almacena la información de nombres madre	Maria Juma
ocupacionP	Cadena	50				Almacena la información de ocupacion padre	Comerciante
ocupacionM	Cadena	50				Almacena la información de ocupacion madre	Ninguna
nacionalidadP	Cadena	40				Almacena la información de nacionalidad padre	ECUATORIANA
nacionalidadM	Cadena	40				Almacena la información de nacionalidad madre	ECUATORIANA
Cod_Estudiante	Entero	10		fk_cod_estudiante	estudiante	Clave foránea que se vincula con la tabla estudiante	858
Ejemplo de entrada de datos:[Cod_Padres] => "28", [nombresP] => "Juan Perez", [nombresM] => "Maria Juma", [ocupacionP] => "Comerciante", [ocupacionM] => "Ninguna", [nacionalidadP] => "ECUATORIANA", [nacionalidadM] => "ECUATORIANA", [Cod_Estudiante] => "858"							

Tabla paralelo							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Paralelo	Entero	10	PRIMARY			La columna Cod_Paralelo es clave primaria de paralelo	9
Nombre_Par	Cadena	50				Almacena la información de Nombre Paralelo	A

Ejemplo de entrada de datos:[Cod_Paralelo] => "9", [Nombre_Par] => "A"

Tabla perfil							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
codigo	Entero	10	PRIMARY			La columna codigo es clave primaria de perfil	5
nombre	Cadena	10				Almacena la información de nombre	Usuario
descripcion	Cadena	100				Almacena la información de descripcion	Perfil de Usuario

Ejemplo de entrada de datos:[codigo] => "5", [nombre] => "Usuario", [descripcion] => "Perfil de Usuario"

Tabla perfil_menu							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
cod_perfil	Entero	10	PRIMARY	perfil_menu_ibfk_1	perfil	Clave foránea que se vincula con la tabla perfil	5
cod_menu	Entero	10	PRIMARY	perfil_menu_ibfk_2	menu	Clave foránea que se vincula con la tabla menu	34
estado	tinyint					Almacena la información de estado	1

Ejemplo de entrada de datos:[cod_perfil] => "5", [cod_menu] => "34", [estado] => "1"

Tabla periodo							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Periodo	Entero	10	PRIMARY			La columna Cod_Periodo es clave primaria de periodo	19
Nombre_Per	Cadena	50				Almacena la información de Nombre Pereriodo	2016
Fecha_I	Fecha					Almacena la información de Fecha Inicio	42370
Fecha_f	Fecha					Almacena la información de Fecha finalización	42735
Estado	tinyint					Almacena la información de Estado	1

Ejemplo de entrada de datos:[Cod_Periodo] => "19", [Nombre_Per] => "2016", [Fecha_I] => "2016-01-01", [Fecha_f] => "2016-12-31", [Estado] => "1"

Tabla prerequisitos							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
id_req	Entero	10	PRIMARY			La columna id_req es clave primaria de prerequisitos	1
cod_mat	Entero	10				Almacena la información de cod mat	1
Nom_Mat	Cadena	100				Almacena la información de Nom Mat	2
requisito	Entero	10				Almacena la información de requisito	1

Ejemplo de entrada de datos:[id_req] => "1", [cod_mat] => "1", [Nom_Mat] => "0002", [requisito] => "1"

Tabla profesor							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Profesor	Entero	10	PRIMARY			La columna Cod_Profesor es clave primaria de profesor	86
Ci	Cadena	13				Almacena la información de Ci	1111111111
Apellido	Cadena	100				Almacena la información de Apellido	Vasquez
Direccion	Cadena	100				Almacena la información de Direccion	Calle Uno 345
Telefono	Cadena	10				Almacena la información de Telefono	2222222222
Celular	Cadena	10				Almacena la información de Celular	933333333
Mail	Cadena	100				Almacena la información de Mail	profesor@servidor.com
Clave	Cadena	20				Almacena la información de Clave	12345
Estado	tinyint					Almacena la información de Estado	1

Ejemplo de entrada de datos:[Cod_Profesor] => "86", [Ci] => "1111111111", [Apellido] => "Vasquez", [Direccion] => "Calle Uno 345", [Telefono] => "2222222222", [Celular] => "0933333333", [Mail] => "profesor@servidor.com", [Clave] => "12345", [Estado] => "1"

Tabla promovido							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
id	Entero	10	PRIMARY			La columna id es clave primaria de promovido	3830
cod_est	Entero	10				Almacena la información de cod est	1
cod_mat	Entero	10				Almacena la información de cod mat	1
estado	Cadena	1				Almacena la información de estado	1
matricula	Entero	10				Almacena la información de matricula	1
nom_mate	Cadena	50				Almacena la información de nom mate	materia x

Ejemplo de entrada de datos:[id] => "3830", [cod_est] => "1", [cod_mat] => "1", [estado] => "1", [matricula] => "1", [nom_mate] => "materia x"

Tabla reglamento							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
id	Entero	10	PRIMARY			La columna id es clave primaria de reglamento	3
descripcion	text					Almacena la información de descripcion	Ejemplo de Reglamento

Ejemplo de entrada de datos:[id] => "3", [descripcion] => "Ejemplo de Reglamento"

Tabla reprobados							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
id_r	Entero	10	PRIMARY			La columna id_r es clave primaria de reprobados	380
cod_est	Entero	10				Almacena la información de cod est	1
cod_mat	Entero	10				Almacena la información de cod mat	1
estado	Cadena	2				Almacena la información de estado	1
matricula	Entero	10				Almacena la información de matricula	1
nom_mate	Cadena	30				Almacena la información de nombre materia	Materia 1

Ejemplo de entrada de datos:[id_r] => "380", [cod_est] => "1", [cod_mat] => "1", [estado] => "1", [matricula] => "1", [nom_mate] => "Materia 1"

Tabla requisito							
Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Requisito	Entero	10	PRIMARY			La columna Cod_Requisito es clave primaria de requisito	5
Nombre_Req	Cadena	50				Almacena la información de Nombre Requisito	Requisito 1
Estado	Entero	10				Almacena la información de Estado	1

Ejemplo de entrada de datos:[Cod_Requisito] => "5", [Nombre_Req] => "Requisito 1", [Estado] => "1"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Requisito	Entero	10	PRIMARY	FK_RI_Requisito	requisito	Clave foránea que se vincula con la tabla requisito	5
Cod_Inscripcion	Entero	10	PRIMARY	FK_RI_Inscripcion	inscripción	Clave foránea que se vincula con la tabla inscripción	1277
Num_Doc	Cadena	50				Almacena la información de Numero Documento	1
Estado	Entero	10				Almacena la información de Estado	0

Ejemplo de entrada de datos:[Cod_Requisito] => "5", [Cod_Inscripcion] => "1277", [Num_Doc] => "1", [Estado] => "0"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Seccion	Entero	10	PRIMARY			La columna Cod_Seccion es clave primaria de seccion	5
Nombre_Secc	Cadena	50				Almacena la información de Nombre Sección	SECCION PRUEBA
Hora_I	Hora					Almacena la información de Hora Inicio	0,333333333
Hora_F	Hora					Almacena la información de Hora Fin	0,5
Estado	Entero	10				Almacena la información de Estado	1

Ejemplo de entrada de datos:[Cod_Seccion] => "5", [Nombre_Secc] => "SECCION PRUEBA", [Hora_I] => "08:00:00", [Hora_F] => "12:00:00", [Estado] => "1"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
Cod_Seminarios	Entero	10	PRIMARY			La columna Cod_Seminarios es clave primaria de seminarios	1
nombre	Cadena	60				Almacena la información de nombre	SEMINARIO EJEMPLO
horas	Cadena	4				Almacena la información de horas	30
fecha	Fecha					Almacena la información de fecha	42370
Cod_Profesor	Entero	10		seminarios_ibfk_1	profesor	Clave foránea que se vincula con la tabla profesor	86

Ejemplo de entrada de datos:[Cod_Seminarios] => "1", [nombre] => "SEMINARIO EJEMPLO", [horas] => "30", [fecha] => "2016-01-01", [Cod_Profesor] => "86"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
cod_usuario	Entero	10	PRIMARY			La columna cod_usuario es clave primaria de usuario	8
usuario	Cadena	20				Almacena la información de usuario	Administrador
clave	Cadena	20				Almacena la información de clave	12345
cod_perfil	Entero	10		usuario_ibfk_1	perfil	Clave foránea que se vincula con la tabla perfil	5

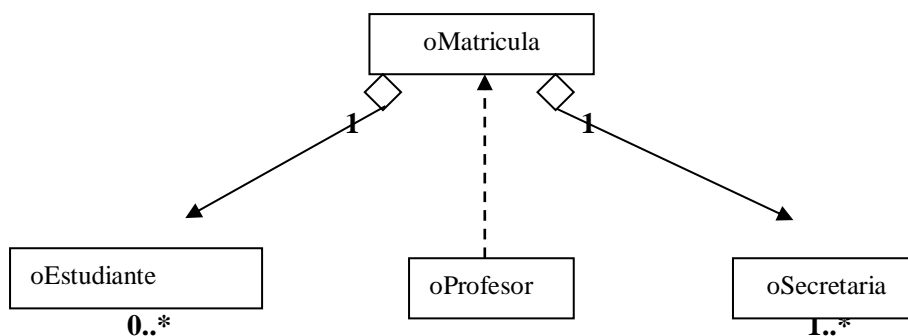
Ejemplo de entrada de datos:[cod_usuario] => "8", [usuario] => "Administrador", [clave] => "12345", [cod_perfil] => "5"

Columna	Tipo	Tam	PK	FK	TABLA REFERENCIA	DESCRIPCION	EJEMPLO DE DATO
cedula	Cadena	13	PRIMARY			La columna cedula es clave primaria de usuario_perfil	
cod_perfil	Entero	10	PRIMARY	usuario_perfil_ibfk_1	perfil	Clave foránea que se vincula con la tabla perfil	5
fecha_registro	Fecha y Hora					Almacena la información de fecha registro	42687
estado	tinyint					Almacena la información de estado	1

Ejemplo de entrada de datos:[cedula] => "", [cod_perfil] => "5", [fecha_registro] => "2016-11-13 00:00:00", [estado] => "1"

Realizado por: Cristian Merino. 2017

Modelo conceptual

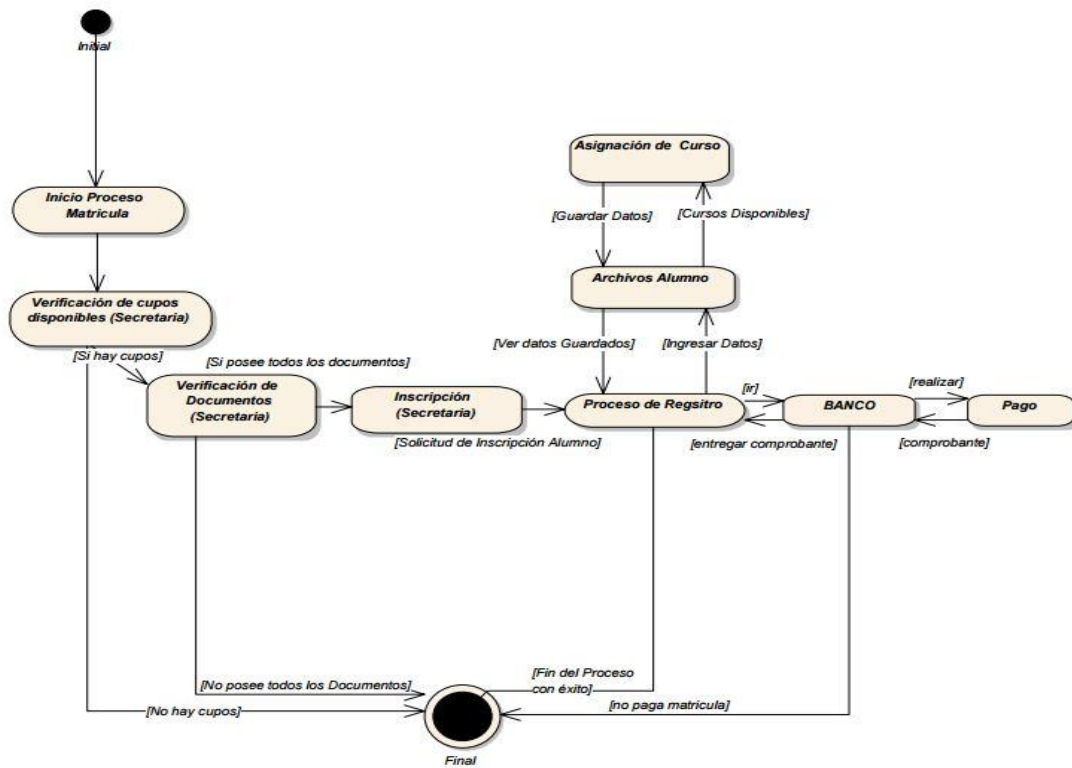


Arquitectura del Sistema

Los Diagramas del sistema proyectan detalladamente a nivel de componentes en guía de implementación.

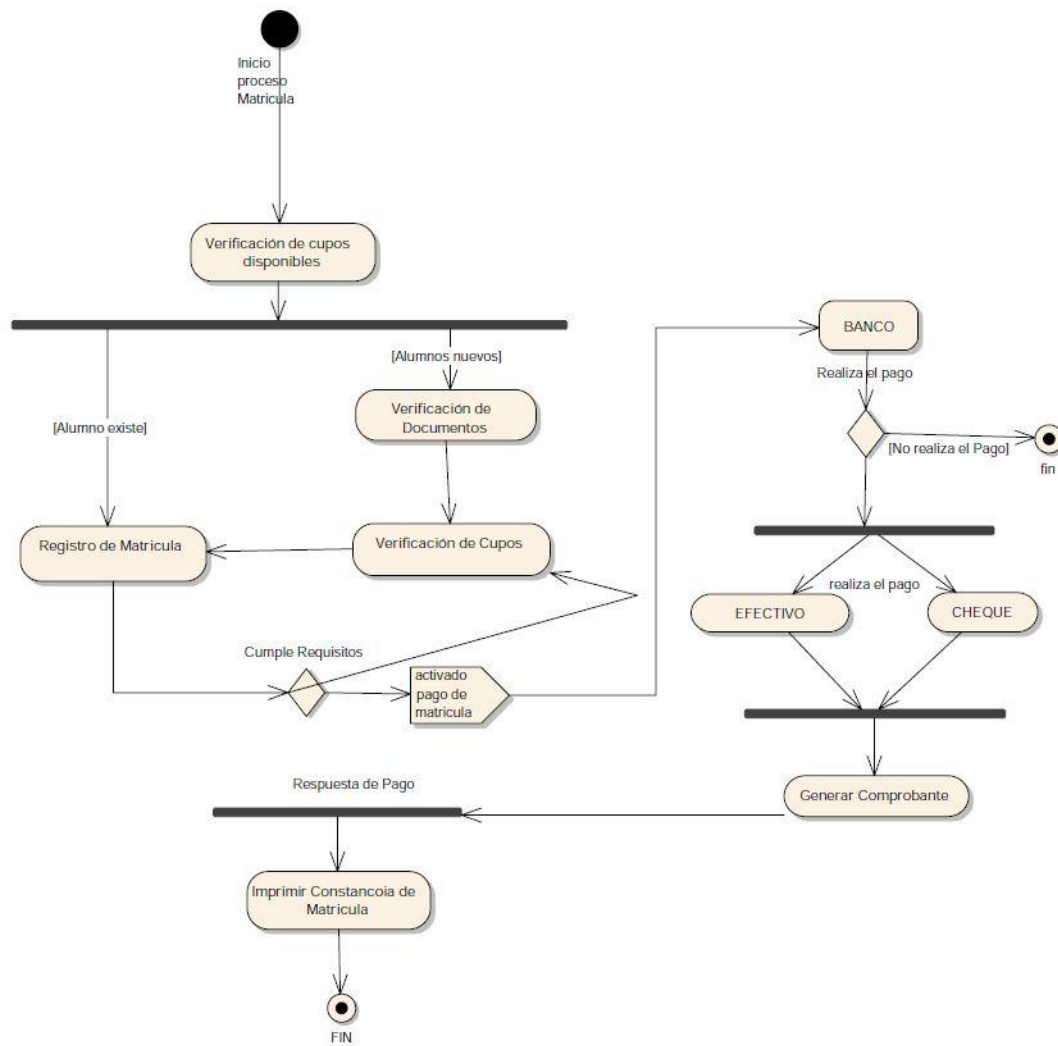
4. Diagrama de Comportamiento

4.1 Diagrama de Estado



Realizado por: Cristian Merino. 2017

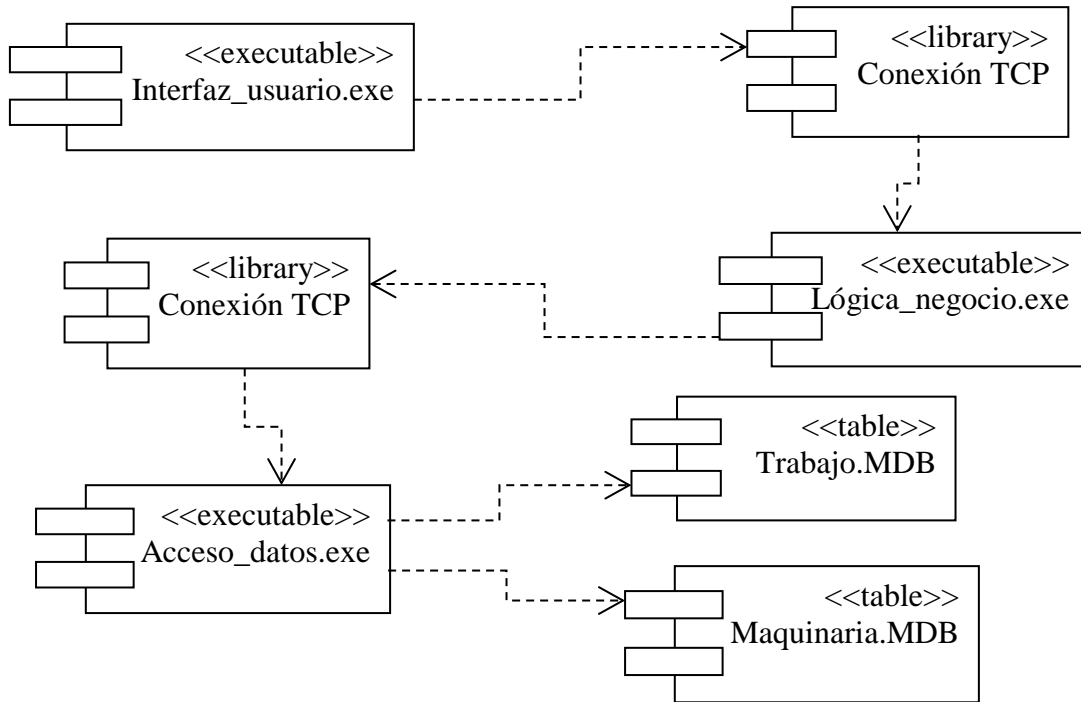
4.2 Diagrama de Actividad



Realizado por: Cristian Merino. 2017

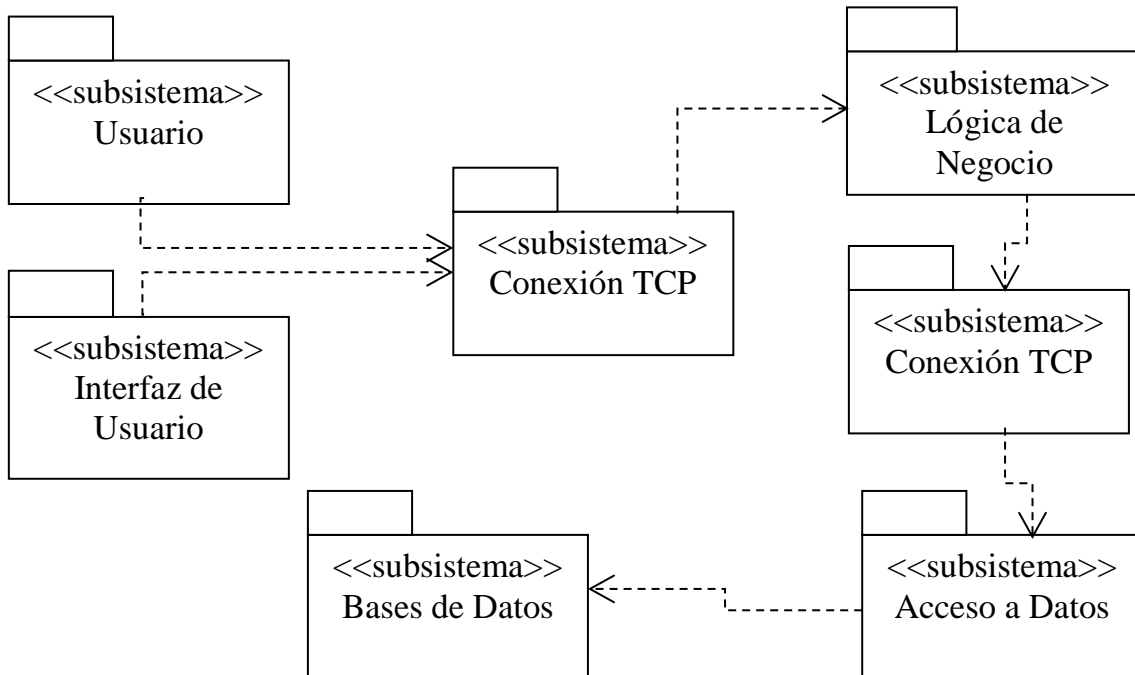
5. Diagramas de Implementación

5.1 Diagrama de componentes



Realizado por: Cristian Merino. 2017

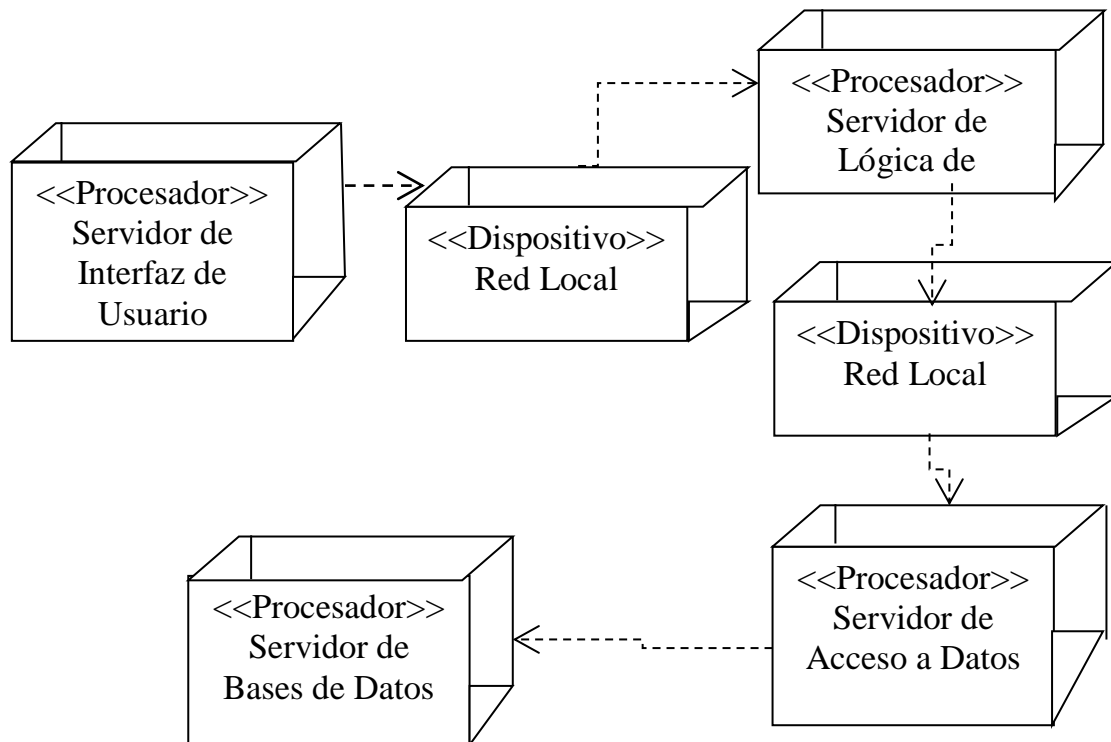
A continuación, se presenta la arquitectura del sistema por medio de los diagramas de componentes que visualizan la distribución de las capas en el sistema, las mismas que se encapsulan algunos componentes del anterior diagrama de componentes en subsistemas.



Realizado por: Cristian Merino. 2017

5.2 Diagrama de despliegue

A continuación, se presenta el diagrama de despliegue el mismo que muestra las capas del sistema sobre hardware y dispositivos de red que conforman la arquitectura del sistema.



Realizado por: Cristian Merino. 2017