

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniera e Ingeniero de Sistemas**

**TEMA:
ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL MÓDULO DE CONTROL DE
UN WORKFLOW PARA EL SEGUIMIENTO DE PASANTÍAS DE LA
CARRERA DE INGENIERÍA DE SISTEMAS – SEDE QUITO.**

**AUTORES:
KAREN VANESSA ACHIG GUACHAMIN
BRYAN ANDRÉS TORRES LARA**

**TUTOR:
RODRIGO EFRAÍN TUFÍÑO CÁRDENAS**

Quito, marzo del 2018

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Achig Guachamin Karen Vanessa con documento de identificación No. 1724247703 y Torres Lara Bryan Andrés con documento de identificación No. 1723450365, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL MÓDULO DE CONTROL DE UN WORKFLOW PARA EL SEGUIMIENTO DE PASANTÍAS DE LA CARRERA DE INGENIERÍA DE SISTEMAS – SEDE QUITO.”, mismo que ha sido desarrollado para optar por el título de: INGENIERA E INGENIERO DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
KAREN VANESSA
ACHIG GUACHAMIN
C.I: 1724247703



.....
BRYAN ANDRÉS
TORRES LARA
C.I: 1723450365

Quito, marzo del 2018

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación con el tema ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL MÓDULO DE CONTROL DE UN WORKFLOW PARA EL SEGUIMIENTO DE PASANTÍAS DE LA CARRERA DE INGENIERÍA DE SISTEMAS – SEDE QUITO. realizado por Achig Guachamin Karen Vanessa y Torres Lara Bryan Andrés, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, marzo del 2018



TUFIÑO CÁRDENAS RODRIGO EFRAÍN

CI: 1717646390

DEDICATORIA

Dedico este proyecto de titulación, primero a Dios por haberme bendecido y llenado de sabiduría y fuerzas para poder culminar con éxito este gran proyecto en mi vida.

A mi padre Roberto Achig que es la persona que más amo en este mundo, gracias papito por enseñarme el verdadero significado del trabajo, de la humildad y del esfuerzo diario, le agradezco por confiar totalmente en mí y por ese gran esfuerzo que realizo por apoyarme en este gran sueño. Eres y serás siempre el hombre de mi vida.

A mi madre Patricia Guachamin que es el regalo mas grande que Dios me pudo dar, gracias mamita por ser mi amiga, mi confidente, mi cómplice porque siempre estas pendiente de como estoy, porque me consientes y me mimas siempre. Te amo infinitamente gracias porque no me has dejado ni un segundo sola en este largo caminar.

A mi hermano Daniel Achig que con su ejemplo de perseverancia y valentía al afrontar la vida me ha enseñado que nada es imposible, gracias por cuidarme, defenderme y quererme siempre. Gracias por nunca dejarme sola y siempre guiarme. Te amo.

A mi sobrino Cristopher Achig que desde que estas en este mundo llenaste mi vida de alegría y amor, gracias porque en los momentos más difíciles me brindas una sonrisa y un abrazo que me hace olvidar todo lo malo te amo eres la alegría de toda nuestra familia.

A mi tía Janeth Cáceres, a mi abuelita Lola Pazmiño y a mi primo Dubany Noboa por ser un gran apoyo siempre y por confiar ciegamente en mí.

Achig Guachamin Karen Vanessa

Dedico este proyecto a Dios por haberme dado la fuerza para seguir avanzando hasta este punto de mi vida, por darme la sabiduría y bendiciones para alcanzar mis metas y por poner a mi lado las personas que me ayudaron a alcanzarlas.

A mis padres, pilares fundamentales en mi vida y los tengo a mi lado. A mi padre Wiston Torres porque fue el primero que creyó en mí e hizo un esfuerzo muy pero muy duro para que yo pueda llegar a este punto de mi carrera. Porque la lucha constante en su trabajo valió la pena. A mi madre Ermita Lara, quien ha sido la guía en mi camino, ya que con su ejemplo y con su amor nunca bajé los brazos y no me di por vencido. A los dos gracias por su apoyo.

A mi hermano Alexis Torres, que ha sido de mucha ayuda para mí y me ha apoyado al igual que a la familia, ¡ahora me toca a mí ayudarlo a él y a mi familia :)!. A mi novia Kati Suarez, que supo apoyarme en los momentos más duros de mi carrera y me alentó a seguir a delante. A mi hermana Mariuxi por sus fuerzas y a mi mamita Piedad porque no dejó de apoyarme para que pueda cumplir mi meta.

A todos mis familiares y amigos que de alguna u otra manera me aportaron su granito de arena muchas gracias!!!.

Torres Lara Bryan Andrés

AGRADECIMIENTOS

Agradezco a mi alma mater la Universidad Politécnica Salesiana, por brindarme durante estos años de estudio un excelente enriquecimiento intelectual y humano, gracias por ser mi templo del saber y guiarme de la mejor manera durante este proceso educativo.

A mi tutor el Ingeniero Rodrigo Tufiño por ser un gran apoyo durante este proceso de titulación, gracias por confiar en mí y brindarme una guía clara para poder desarrollar mi proyecto de titulación, también por ser una gran persona y un excelente mentor ha sido un verdadero honor trabajar a su lado.

A mi ángel Werner que se que desde el cielo estas orgulloso de mi, te agradezco infinitamente por encaminar mi carrera gracias a ti, tome la mejor decisión de mi vida que ha sido seguir esta grandiosa carrera. ¡No te defraude!

A mi compañero de tesis Bryan Torres por creer en mí y por su apoyo incondicional en este duro camino que atravesamos, lo logramos juntos y estoy muy orgullosa de nuestro trabajo.

A mis hermanas de corazón Lissette Villacis y Luz Toscano porque desde que llegaron a mi vida me han apoyado y han confiado en mí incondicionalmente, gracias por su inmenso amor me han enseñado el verdadero significado de una amistad y nunca me han abandonado, las amo gracias por todo.

A mi incondicional amiga Andrea Zambonino por nunca dejarme rendir siempre has tenido las palabras correctas para animarme a seguir adelante, eres una de las mejores cosas que me ha pasado dentro de la universidad. Te quiero mucho.

Achig Guachamin Karen Vanessa

En este punto de mi carrera universitaria doy gracias a mi Dios por la vida y la salud. A mis padres y mi hermano, siempre agradecido con ellos por su gran esfuerzo ya que fueron tiempos muy duros pero juntos salimos adelante. Muchas gracias familia los amooo!!.

A el Ing. Rodrigo Tufiño, por su guía como tutor en este proceso y por creer en nosotros. Una gran persona y un gran y apasionado profesional, de los mejores profesores de la U.P.S.

A mi compañera de proyecto Karen, por creer en mí, por su apoyo en cada paso que dimos que no fueron fáciles pero aquí estamos.

A mis amigos Jairo Flores y Jhon Castillo con los que forme el grupo J2B y tenemos bellos recuerdos de Cuenca, el hackathon y el Just Dance. Gracias por su ayuda friends.

A la U.P.S por todos estos años de educación, por ser mi 2da casa mucho tiempo y por el conocimiento que cada docente me impartió.

Por último, a todos mis compañeros, amigos, familiares, primos que estuvieron y están cuando más los necesito.

Torres Lara Bryan Andrés

ÍNDICE GENERAL

1 Marco Metodológico	4
1.1 Metodologías	4
1.1.1 Metodología Scrum	4
1.1.2 Metodología Xp	6
1.2 Pasantía Universitaria.....	8
1.3 Workflow	10
1.3.1 Definiciones manejadas en un workflow	12
1.3.2 Modelo de referencia.....	13
1.3.3 Tipos de workflow	15
1.3.4 Modelos de workflow	16
1.3.5 Ventajas de un workflow	17
1.4 UML.....	17
1.4.1 Diagrama de Casos de Uso	18
1.4.2 Diagrama de Clases.....	18
1.4.3 Diagrama de Secuencia	18
1.4.4 Diagrama de Estados.....	19
1.4.5 Diagrama de Actividad	19
1.4.6 Diagrama de Objetos.....	19
1.4.7 Diagrama de Interacción	19
1.4.8 Diagrama de Paquetes	19
1.4.9 Diagrama de Colaboración.....	19
1.4.10 Diagrama de Tiempo.....	20
1.4.11 Diagrama de Componentes	20
1.4.12 Diagrama de Despliegue	20
2 Analisis y Diseño.....	21
2.1 Análisis.....	21
2.1.1 Introducción.....	21
2.1.2 Propósito.....	21
2.1.3 Alcance.....	21
2.2 Visión General	22
2.2.1 Diagnóstico Actual del Proceso	22
2.2.2 Modelo Automatizado Propuesto.....	23
2.3 Descripción global	24
2.3.1 Perspectiva del producto	24
2.3.2 Funciones del producto	24
2.3.2.1 Módulo de visita.....	24
2.3.2.2 Módulo de validación final	25
2.4 Restricciones	25
2.5 Dependencia	25
2.6 Requerimientos específicos.....	26
2.6.1 Interfaz del hardware.....	26
2.6.2 Interfaz del software.....	26
2.6.3 Interfaz de comunicaciones	26
2.7 Requerimientos funcionales	26
2.7.1 Módulo de Visita.....	27
2.7.2 Módulo de Validación Final.....	29
2.8 Requerimientos no funcionales	30
2.9 Diseño	33
2.9.1 Diagramas de caso de uso	33

2.9.1.1 Casos de uso 1: Modulo de visita.....	33
2.9.1.2 Casos de uso 2: Modulo validación final	34
2.9.1.3 Diagramas de clases	35
2.10 Base de datos.....	36
2.11 Interfaz de usuario.....	39
2.11.1 Estudiante.....	39
2.11.2 Tutor.....	42
3 Construcción y Pruebas.....	47
3.1 Herramientas	47
3.1.1 BootsFaces.....	47
3.1.2 NetBeans.....	47
3.1.3 Ajax.....	48
3.1.4 Itext 5.0.5.....	48
3.1.5 GlassFish server	48
3.1.6 PostgreSQL.....	49
3.1.7 JavaScript.....	49
3.1.8 MVC.....	50
3.1.9 JQuery.....	50
3.1.10 Bootstrap.....	50
3.1.11 Primefaces.....	51
3.1.12 Hibernate.....	51
3.1.13 GitHub.....	51
3.2 Base de Datos	52
3.3 Diccionario de Datos.....	54
3.4 Historias de usuario.....	59
3.4.1 Módulo de visita.....	59
3.4.2 Módulo de validación final	68
3.5 Código.....	80
3.5.1 Código de configuración.....	80
3.5.2 Proceso de Visita.....	81
3.6 Pruebas	89
3.6.1 Pruebas de Estrés.....	89
3.6.2 Pruebas Unitarias	92
3.6.3 Pruebas contra requerimientos	93
4 Implementación.....	98
4.1 Implementación de HW y SW	98
4.2 Configuración de la base de datos.....	98
4.2.1 Paso 1.....	99
4.2.2 Paso 2.....	100
4.2.3 Paso 3.....	100
4.3 Configuraciones Servidor Web	100
4.4 Diagrama de componentes	102
CONCLUSIONES.....	103
RECOMENDACIONES.....	105
GOSARIO DE TÉRMINOS	106
LISTA DE REFERENCIA.....	107

ÍNDICE DE TABLAS

Tabla 1 Requerimiento funcional 1.1: Envío de alertas al tutor vía e-mail para la visita.....	27
Tabla 2 Requerimiento funcional 1.2: Creación de fecha tentativa para la visita.....	27
Tabla 3 Requerimiento funcional 1.3: Envío de alertas al estudiante vía e-mail para la visita.....	27
Tabla 4 Requerimiento funcional 1.4: Agendar la cita para una visita.....	28
Tabla 5 Requerimiento funcional 1.5: Generar informe de visita del tutor.....	28
Tabla 6 Requerimiento funcional 1.6: Generar informe de seguimiento de la pasantía.....	28
Tabla 7 Requerimiento funcional 1.7: Subir documentos de visita al sistema.....	28
Tabla 8 Requerimiento funcional 2.1: Generar solicitud para validación de pasantía.....	29
Tabla 9 Requerimiento funcional 2.2: Generar hoja de autoevaluación.....	29
Tabla 10 Requerimiento funcional 2.3: Subir documentación de validación al sistema.....	29
Tabla 11 Requerimiento funcional 2.4: Validación de documentación.....	30
Tabla 12 Requerimiento funcional 2.5: Validación información subida al sistema..	30
Tabla 13 Requerimiento no funcional 1.1: Portabilidad (Flores, Castillo y Tufiño, 2018).....	30
Tabla 14 Requerimiento no funcional 1.2: Interfaz gráfica del usuario GUI.(Flores, Castillo y Tufiño, 2018).....	31
Tabla 15 Requerimiento no funcional 1.3: Sistema Responsive2018).....	31
Tabla 16 Requerimiento no funcional 1.4: Mensajes de error en el sistema (POP-UP)(Flores, Castillo y Tufiño, 2018).....	31
Tabla 17 Requerimiento no funcional 1.5: Sesiones concurrentes.....	31
Tabla 18 Requerimiento no funcional 1.6: Tiempo de respuesta 2018).....	32
Tabla 19 Requerimiento no funcional 1.7: Seguridad.....	32
Tabla 20 Requerimiento no funcional 1.8: Disponibilidad (Flores, Castillo y Tufiño, 2018).....	32
Tabla 21 Requerimiento no funcional 1.9: Ventana de ayuda.....	32
Tabla 22 Requerimiento no funcional 1.10: Estado de flujo.....	33
Tabla 23 Tabla de base de datos.....	38
Tabla 24 Descripción de la tabla Perfil.....	54
Tabla 25 Descripción de la tabla Usuario.....	54
Tabla 26 Descripción de la tabla Estudiante.....	55
Tabla 27 Descripción de la tabla Tutor.....	55
Tabla 28 Descripción de la tabla VisitaTutor.....	55
Tabla 29 Descripción de la tabla Proceso.....	56
Tabla 30 Descripción de la tabla DetallePasantía.....	56
Tabla 31 Descripción de la tabla Pasantía.....	57
Tabla 32 Descripción de la tabla Formato.....	57
Tabla 33 Descripción de la tabla Preguntas.....	58
Tabla 34 Descripción de la tabla TipoPregunta.....	58
Tabla 35 Descripción de la tabla Respuesta.....	58
Tabla 36 Descripción de la tabla Datos.....	58
Tabla 37 Historia de usuario (Módulo de visita).....	59
Tabla 38 Descripción historia de usuario 1 (Interacción 1).....	60
Tabla 39 Descripción historia de usuario 1(Interacción 2).....	62

Tabla 40 Descripción historia de usuario 1(Interacción 3	67
Tabla 41 Historia de usuario(Módulo de validación final)	68
Tabla 42 Descripción historia de usuario 2 (Interacción 1)	68
Tabla 43 Descripción historia de usuario 2 (Interacción 2)	73
Tabla 44 Descripción historia de usuario 2 (Interacción 3)	79
Tabla 55: Ejecución Pruebas Unitarias	93
Tabla 56: Tutor.....	94
Tabla 57 Estudiante.....	96
Tabla 58 Coordinador	97

ÍNDICE DE FIGURAS

Figura 1 Ciclo Scrum.(Moreira Delgado, 2006)	5
Figura 2 Proceso de un workflow	11
Figura 3 Modelo de referencia. (Menéndez Domínguez y Castellanos Bolaños, 2016)	14
Figura 4 workflow (Manual) de visita y validación de la pasantía.	23
Figura 5 workflow (Automatizado) SPPP visita y validación final.	24
Figura 6. Caso de uso 1.	34
Figura 7 Caso de uso 2.	35
Figura 8 Diagrama de clases.	36
Figura 9 Diagrama de base de datos conceptual. (Flores, Castillo y Tufiño, 2018).	37
Figura 10 Prototipo de interfaz Avance Estudiante	39
Figura 11 Prototipo de interfaz Confirmación Cita	40
Figura 12 Prototipo de interfaz Solicitud Validación	40
Figura 13 Prototipo de interfaz Formato Autoevaluación	41
Figura 14 Prototipo de interfaz Subir Archivos	41
Figura 15 Prototipo de interfaz Avance Tutor	42
Figura 16 Prototipo de interfaz Citas Tutor	42
Figura 17 Prototipo de interfaz Agendar Cita	43
Figura 18 Prototipo de interfaz Citas Agendadas	43
Figura 19 Prototipo de interfaz Formularios Tutor	44
Figura 20 Prototipo de interfaz Informe Seguimiento	44
Figura 21 Prototipo de interfaz Informe de Tutor	45
Figura 22 Prototipo de interfaz Subir archivos Tutor	46
Figura 23 Repositorio proyecto SPPP	52
Figura 24 Diseño Relacional de la Base de Datos. (Flores, Castillo y Tufiño, 2018).....	53
Figura 25 Pantalla de agendación de cita tutor (Primera iteración)	61
Figura 26 Pantalla informe tutor (Primera interacción)	61
Figura 27 Pantalla de confirmación de cita estudiante (Primera iteración)	62
Figura 28 Pantalla de agendación de cita tutor (Segunda iteración)	64
Figura 29 Pantalla de confirmación y cancelación de cita estudiante(Segunda iteración).....	64
Figura 30 Pantalla informe tutor (Segunda iteración).....	65
Figura 31 Pantalla información de visitas y estado de las mismas. (Segunda iteración)	65
Figura 32 Pantalla de cancelación de cita tutor (Segunda iteración)	66
Figura 33 Pantalla de dashboard tutor (Segunda iteración)	66
Figura 34 Pantalla de dashboard estudiante (Segunda iteración).....	66
Figura 35 Pantalla de subir documentos tutor (Primera iteración)	70
Figura 36 Pantalla de autoevaluación estudiante (Primera iteración).....	70
Figura 37 Pantalla de subir documentos estudiante (Primera iteración).....	71
Figura 38 Pantalla de validación documentos coordinador (Primera iteración)	71
Figura 39 Pantalla para generar reporte coordinador (Primera iteración).....	72
Figura 40 Pantalla para registrar resolución coordinador (Primera iteración)	72
Figura 41 Pantalla de proceso concluido (Primera iteración)	73
Figura 42 Pantalla para subir documentos de validación tutor (Segunda iteración)..	75
Figura 43 Pantalla para autoevaluación estudiante (Segunda iteración).....	76
Figura 44 Pantalla para subir documentos de validación estudiante (Segunda iteración)	76

Figura 45 Pantalla para generar reporte coordinador(Segunda iteración)	77
Figura 46 Pantalla para registro de resolución coordinador(Segunda iteración)	77
Figura 47 Pantalla para dashboard coordinador(Segunda iteración)	78
Figura 48 Pantalla para dashboard secretaria(Segunda iteración)	78
Figura 49 Pantalla para listar reportes secretaria(Segunda iteración)	78
Figura 50 Propiedades de configuración.....	81
Figura 51 Clase para persistir.....	82
Figura 52 Case CitasDaoImp, metodo Find All.....	82
Figura 53 Método Datos de la Visita	83
Figura 54 Método Guardar Datos	84
Figura 55 Método Guardar Datos dinámicos	85
Figura 56 Método Download File Estudiante	86
Figura 57 Método Listar Informe Coordinador	87
Figura 58 Método Create coordinador folder first time	88
Figura 59 Método Guardar Datos pasantía	89
Figura 60 : Estadísticas 1 con 100 users	90
Figura 61 : Estadísticas 2 con 100 users	90
Figura 62: Estadísticas por solicitud	90
Figura 63: Resultados Tiempo de Respuesta	91
Figura 64: Estadísticas 1 con 250 users	91
Figura 65: Estadísticas 2 con 250 users	91
Figura 66: Estadísticas por solicitud	92
Figura 67: Resultados Tiempo de Respuesta	92
Figura 68: Diagrama de componentes.....	102

RESUMEN

Este proyecto de titulación, describe el proceso que se está llevando a cabo en la carrera de Ingeniería de Sistemas Sede Quito con lo que respecta a pasantías, actualmente dicho procedimiento se lo maneja de forma manual. Por lo tanto, en este proyecto se presenta una solución para automatizarlo.

El objetivo principal del proyecto ha sido diseñar un sistema que permita a los docentes de la carrera de Ingeniería de Sistemas, manejar el proceso de pasantías de una manera automatizada, para ello se ha desarrollado dos módulos uno de visita y otro de validación final, los que permitirán controlar el proceso mediante un flujo de trabajo.

El sistema desarrollado denominado "Sistema de pasantías y prácticas pre profesionales(SPPP)" es una aplicación Web que ha sido implementada usando herramientas y plataformas de software libre, está orientado para estudiantes, docentes y personal administrativo de la carrera, los mismos que podrán interactuar dentro del proceso dependiendo como vaya avanzando el flujo de trabajo.

A continuación, se muestra el desarrollo del sistema de una manera secuencial empezando con el Análisis de requerimientos, el Diseño, la Construcción del sistema y pruebas, terminando con la implementación del mismo.

ABSTRACT

This degree project, describes the process that is taking place in the career of Systems Engineering headquarters in Quito, with regard to internships, currently this procedure it is managed manually. Therefore, this project presents a solution to automate.

The main objective of the project has been to design a system that permits to teachers in the career of Systems Engineering, managing the process of internships in an automated way, has been developed for this two modules one of visit and another for final validation which will allow controlling the process through a workflow.

The system developed called "System of internships and pre-professional practices(SPPP)" is a Web application that has been implemented using tools and platforms of free software, is designed for students, teachers and administrative staff of the race, the same that may interact within the process depending on how to move the flow of work.

Below, shows the development of the system in a sequential manner starting with the requirements analysis, design, construction and testing, ending with the implementation of the same.

INTRODUCCIÓN

Antecedentes

En la actualidad se han desarrollado varios sistemas con el propósito de solucionar diferentes problemas cotidianos del ser humano, con la intención de lograr algún avance significativo. Para ello, la automatización ha sido el detonante con el que han surgido más necesidades en relación con la mejora de procesos manuales convirtiéndolos en procesos automatizados, facilitando así el trabajo que el usuario realiza y a su vez optimizando tiempo y recursos.

Con este objetivo en mira se efectuó una investigación preliminar realizada con el propósito de determinar la implementación y uso que tiene la tecnología workflow dentro del desarrollo de un sistema.

En la tesis de Freire Aillón, Teresa Milena Villacís López, Juan Fernando para la PUCE (Freire Aillón, T. M. & J. F. Villacís López , 2015), se implementó la tecnología workflow logrando un mejor control dentro de los procesos y permitiendo dar un continuo seguimiento a cada uno de los usuarios participantes dentro del sistema. A su vez en la tesis realizada por Meneses Carrasco y Vargas Carrillo para la Universidad de las Fuerzas Armadas (Meneses Carrasco, S. & J. Vargas Carrillo , 2005) se desarrolló un sistema para el manejo documental en donde se utiliza la tecnología workflow logrando tener un control total de todos los procesos, de tal manera que se conozca las actividades que realizan en un determinado momento las personas involucradas dentro del mismo.

Basados en esta investigación se procedió a implementar dicha tecnología en el desarrollo de este proyecto de titulación.

Problema

Los docentes y estudiantes de la Carrera de Ingeniería de Sistemas no cuentan con un sistema automatizado de pasantías, que permita organizar la información de manera medible y cuantificable de dichos procesos para la Dirección de Carrera.

Para los estudiantes de la Carrera de Ingeniería de Sistemas es un proceso manual muy confuso y tedioso, ya que cada vez hay más requerimientos para realizar las pasantías y no son informados correctamente, no existe un formato definido de los pasos que se deben seguir, ni de los formularios que deben realizar para que puedan efectuar la pasantía. Además de ello en cada etapa del procedimiento hay un tiempo de espera considerable, por lo que en muchas ocasiones los estudiantes pueden tener inconvenientes con las empresas.

También para los docentes de la Carrera de Ingeniería de Sistemas el proceso se torna lleno de inconvenientes ya que los estudiantes no proporcionan los datos necesarios (horario, fechas tentativas de inicio y de fin de la pasantía, tipo de empresa, etc.) que permitan realizar el seguimiento correcto de la pasantía, tampoco se posee un registro de los docentes asignados a realizar la visita al estudiante a la empresa por lo que el procedimiento de pasantías se tarda más. (Achig & Torres, 2017)

Justificación

En la actualidad, la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Campus Sur Sede Quito, se encuentra realizando un proceso manual de pasantías. Por lo tanto, la Dirección de Carrera necesita un control automatizado de las actividades de seguimiento, realizadas por los docentes a los estudiantes cuando se encuentren ejecutando sus pasantías, ya que es de vital importancia automatizar este procedimiento para controlar la asignación de docentes y el seguimiento que

hacen los mismos a los estudiantes.

Este producto será una herramienta informática para automatizar el seguimiento y control de los estudiantes, que facilite la asignación de docentes y permita obtener información del paso en el que se encuentra el estudiante en el proceso de pasantías, dicho producto guiará al estudiante y no hará ver el proceso confuso ni tedioso de lo contrario será fácil e intuitivo tanto como para el estudiante como para los docentes.

Objetivo General

Automatizar el proceso de control de pasantías estudiantiles en la Carrera de Ingeniería de Sistemas – Sede Quito.

Objetivos Específicos

Diagnosticar la situación actual del proceso manual de pasantías.

Definir un procedimiento automatizado que permita controlar y hacer el seguimiento de las pasantías.

Diseñar una aplicación Web para los docentes y estudiantes

Desarrollar la aplicación Web utilizando las herramientas disponibles en el mercado (free).

Implementar el sistema en la infraestructura de la Universidad.

Capítulo 1

1 Marco Metodológico

1.1 Metodologías

1.1.1 Metodología Scrum

Scrum es una de las metodologías ágiles más importantes que se han desarrollado en los últimos 20 años en el campo de la gestión y construcción del software.

Scrum.org define a “Scrum como un framework (marco de trabajo) para la gestión de productos, proyectos y servicios complejos que facilita un desarrollo mantenido e incremental.” (Scrum org, 1993)

Scrum es una metodología que propone una manera en la cual el equipo que desarrolla el trabajo pueda organizarse, trabajar y presentar resultados, de tal manera que se de herramientas y libertad para poder realizar o encontrar un sistema que sea auto organizado. Scrum permite que exista dentro del sistema una mejora continua y que el trabajo sea de calidad, ya que fomenta el trabajo en equipo con la delegación de responsabilidades y la realización del daily meeting o el seguimiento al tablero de tareas.

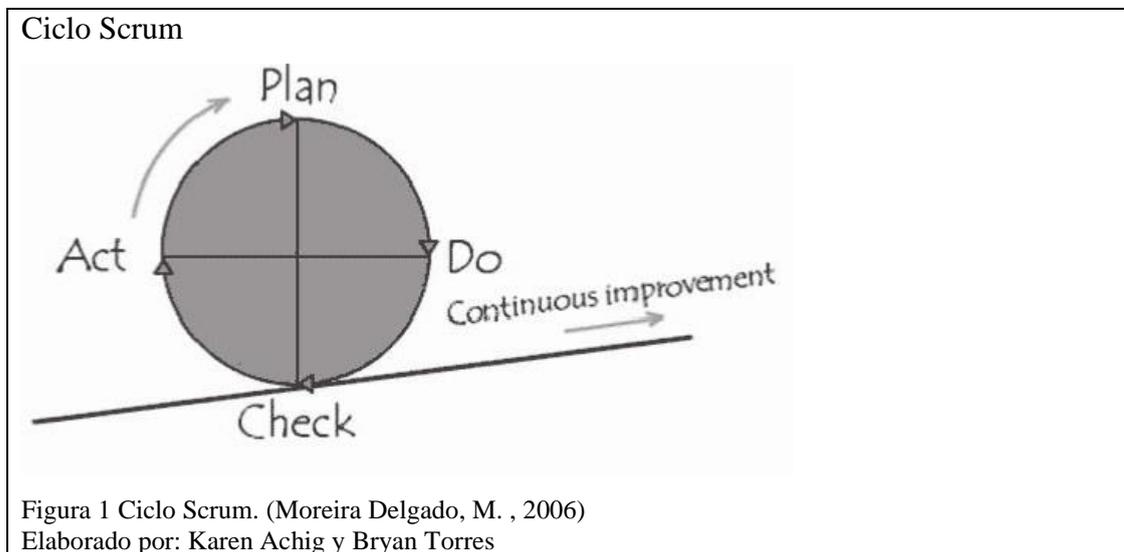
Scrum se basa en iteraciones rápidas que se realizan de manera directa con el cliente en donde el mismo da continuamente su opinión, de tal manera que el desarrollo del sistema se vuelve descentralizado simple, adaptativo y flexible, con ello se puede lograr una alta predisposición y respuesta al cambio. (Monte Galiano, J., 2016) La metodología Scrum está directamente ligada a procesos de mejora continua, que se lo puede resumir a partir del ciclo Deming. (Moreira Delgado, M. , 2006)

Planifica: Se refiere a que hacer y cómo hacerlo

Ejecuta: Hacer lo que se ha planificado

Comprueba: Comprobar si se ha podido hacer o actuar de la manera en la que se planifico

Actuá: como se puede mejorar para la siguiente iteración o ciclo.



La metodología Scrum propone lo siguiente:

Tres roles: Product owner, Scrum master y Scrum team.

Dos artefactos: Product backlog y Sprint backlog.

Tres actividades: Daily meeting, Sprint review y Sprint retrospective

La metodología Scrum dentro de su ejecución tiene claramente definido sus roles, empezando con el Scrum Master que es la persona encargada de dar apoyo a todo el grupo de desarrollo con la finalidad de que los objetivos que se plantearon puedan ser cumplidos en su totalidad, luego el product owner o dueño del producto el cual es el interesado de que el producto se desarrolle y es el que proporciona los recursos necesarios para que se pueda ejecutar el proyecto y por último el Scrum team que es un grupo de desarrolladores que deben cumplir con las tareas que el Scrum Master les asigne.

Dentro de la definición de las tareas Scrum propone dos artefactos, primero se comienza con el product backlog que es un documento en donde se registra una lista de requerimientos que son priorizados de acuerdo a su importancia, de tal manera que se pueda definir un requisito, estimar su inicio, una nota y un solicitante con lo que se delegan funcionalidades, una vez que se tenga este documento se logra conocer de que manera se va a desarrollar el proyecto y así se puede desglosar el desarrollo en tareas pequeñas que no sobrepasan a los 2 días las cuales son delegadas a un desarrollador, de tal forma se da inicio al Sprint que es un ejecutable o un hito de las tareas antes fraccionadas y que dura en su desarrollo aproximadamente 2 y 4 semanas. (Higuera, J. A., C. M. , Duran Camelo , & O. Torres Ced, 2014)

Scrum se maneja con tres actividades una de ellas es el daily meeting que es una reunión diaria que se realiza con el fin de definir las tareas que se van a desarrollar al día siguiente y también se analizan los problemas u inconvenientes que se han presentado en el cumplimiento de las tareas del día anterior, la siguiente actividad es el sprint review este proporciona información sobre cómo va el progreso del proyecto cuando finaliza un sprint y con esto se puede ir realizando cambios u adaptaciones dentro del proyecto, la última actividad es el Sprint retrospective en donde el Scrum Master realiza una reunión para que el equipo discuta el último sprint finalizado y se determine que puede ser mejorado o cambiado. (Scrum org, 1993)

1.1.2 Metodología Xp

La metodología Xp (Programación Extrema) es uno de los métodos ágiles más conocidos y utilizados, debido que para conseguir un desarrollo de software exitoso potencia las relaciones interpersonales de tal manera que se promueva el trabajo en equipo, brindando un excelente ambiente de trabajo y preocupándose por el aprendizaje de sus desarrolladores. (Joskowicz, J. , 2008)

Xp se fundamenta en la comunicación y retroalimentación continua entre el cliente y el equipo de desarrollo, para que exista una correcta comunicación entre todos los participantes de tal manera que se pueda encontrar simplicidad en las soluciones y se pueda enfrentar los cambios rápidamente. Xp es una metodología que se la debe utilizar cuando existan proyectos que posean requisitos imprecisos y muy cambiantes o en proyectos donde pueda existir un alto riesgo técnico.

En Xp cuando se realiza las especificaciones y establecimiento de prioridades de los requisitos del sistema los clientes son una pieza clave ya que están directamente implicados en el proceso, es decir los clientes del sistema son una parte fundamental del equipo de desarrollo. A partir de ello se crean futuros escenarios y story card que son tarjetas de historia en donde se plasman las necesidades que tiene el cliente.

(Gomez Ruedas, J. , 2016)

En esta metodología el desarrollo se lo lleva de una manera incremental en donde se realizan frecuentes entregas pequeñas, para que de ahí se fundamente el proceso de planificación. Dentro de Xp se prioriza el interés de las personas antes que los procesos mediante una programación en parejas para evitar largas jornadas de trabajo sino más bien que sea proceso de desarrollo sostenible. (Letelier Torres, P. & E. Sanchez Lopez , 2003)

Se pueden describir algunos roles dentro de la metodología Xp:

Programador Es la persona encargada de desarrollar el código fuente del sistema y escribe las pruebas unitarias que se van a realizar en el Sistema.

Cliente Es la persona encargada de asignar la prioridad a las historias de usuario, realiza las pruebas funcionales para validar el correcto funcionamiento de su sistema.

Encargado de pruebas (Tester) Es la persona encargada de ejecutar las pruebas funcionales de manera regular y comunica los resultados obtenidos al equipo de

trabajo, además ayuda al cliente al momento de plantear las pruebas funcionales.

Encargado de seguimiento (Tracker) Es la persona que realiza un seguimiento del progreso de cada iteración, verifica el porcentaje de acierto entre las estimaciones que fueron realizadas con el tiempo real con ello da una retroalimentación al equipo de trabajo.

Entrenador (Coach) Es la persona que debe dar guías al equipo para que se sigan las prácticas de la metodología Xp correctamente, es decir es el responsable del proceso global.

Consultor Es una persona que es ajeno al proyecto pero que tiene conocimientos en temas específicos que pueden ser necesarios en el proyecto.

1.2 Pasantía Universitaria

Las pasantías Universitarias, antes llamadas sistema nacional de pasantías educativas, se crearon en 1992, año en el cual se definía por pasantía a la extensión del sistema educativo a instituciones públicas o privadas para la realización de prácticas por parte de estudiantes y maestros de acorde con su formación, bajo el control de la institución de enseñanza a la que pertenecen durante un tiempo.

Dichas pasantías en esta ley no reconocían ningún tipo de relación jurídica entre el pasante y la empresa en donde está. Antes las pasantías eran apodadas como prácticas voluntarias sin pago alguno, en donde el pago por parte de los empleadores era opcional. Su duración máxima era cuatro años, con una actividad diaria mínima de dos horas y máxima de ocho horas.

En el año de 1999, las pasantías se limitaron a una duración de mínimo 2 meses y máximo un año, con una actividad semanal no mayor a los cinco días, cumpliendo jornadas de hasta 4 horas, y ya se le daba un estímulo por las tareas realizadas.

(Adamini, M , 2011)

De acuerdo con la LOES - Ley Organica De Educación Superior, en el art. 87, las prácticas o pasantías preprofesionales se consideran como un requisito previo a la obtención del título, las cuales deben ser realizadas en el campo de especialidad de los o las estudiantes que realizan dichas actividades. (Art.87, 2010)

En el reglamento del régimen académico del CES en su capítulo tercero y Art.88. Consta como prácticas preprofesionales o pasantías a las actividades de aprendizaje en las cuales el estudiante desarrolla conocimientos, destrezas y habilidades específicas lo cual le servirá para un mejor desempeño al momento de ejercer su profesión.

Nos aclara que las pasantías son importantes en el record académico de cada estudiante y que en la universidad cada carrera atribuirá 400 horas por lo bajo para esta práctica.

Reglamento interno de régimen académico de la Universidad Politécnica Salesiana

En el Reglamento Interno de Régimen Académico de la Universidad Politécnica Salesiana el capítulo 4 Art. 136, brinda un concepto de prácticas preprofesionales resaltando que las mismas permiten al estudiante potenciar su proceso de aprendizaje en el nivel correspondiente a su profesión y las cuales impulsan la relación de la Universidad con la comunidad.

En su Art. 138. todos los proyectos académicos de grado incorporan en su currículo al menos 400 horas de pasantías preprofesionales en el ciclo formativo, de estas un mínimo de 160 hora corresponden a servicios a la comunidad de manera obligatoria.

(Universidad Politécnica Salesiana, 2014)

Reglamento de vinculación con la sociedad de la Universidad Politécnica Salesiana

Dentro de la Universidad Politécnica Salesiana en Vinculació'n con la Sociedad también intervienen las actividades de pasantías y prácticas preprofesionales y en el reglamento de Vinculació'n con la Sociedad en el capítulo 4 Art.14 y 15.

Tenemos que el ingreso de estudiantes en estas prácticas genera una contribución

efectiva en el desarrollo de las organizaciones, también son un medio para completar la formación de dichos estudiantes ya que se los vincula al mundo laboral. (Salesiana, 2012)

1.3 Workflow

Workflow es un sistema que permite automatizar y administrar un conjunto de actividades, en donde las tareas se manejan como un flujo de trabajo ya que son pasadas de un participante a otro, en donde cada participante realiza un procesamiento para llegar a un objetivo en común, para este procesamiento se establecen un conjunto de reglas. (WFMC, 1993) (Rusinkiewicz, M. & A. Sheth , 1994)

En Workflow una tarea o proceso es la que va a definir el trabajo que se debe realizar, a partir de ello va a existir una entidad procesadora la cual va ser la responsable de realizar este trabajo. Dentro de este sistema no se especifica qué tipo de entidad procesadora debe ser, por lo tanto, puede ser una máquina, una persona, etc. Por ello es posible aplicar esta tecnología a cualquier tipo de procesos en el que se necesite la colaboración entre personas o entre personas y maquinas. (Geronimo, G. & V. Canseco , 2002)

La realización de un workflow sirve para determinar las funciones del personal que maneja el sistema, mejorar la utilización de los recursos y tener un mejor control de los procesos del negocio. Los beneficios de implementar un workflow son muchos entre los más importantes están el mejor uso y difusión de la documentación y que permite conocer en qué estado se encuentra alguna tarea o proceso.

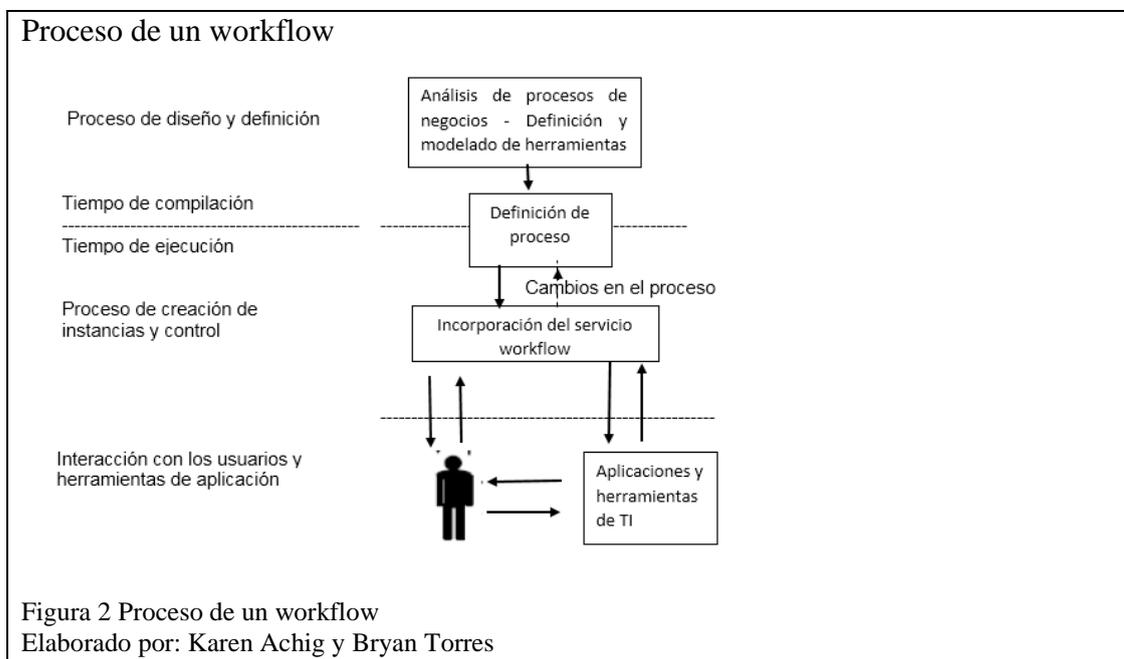
Uno de los puntos fundamentales en el desarrollo de un workflow es el tiempo de transferencia de la información entre las tareas, el cual se disminuye con dicho desarrollo porque en un workflow los procesos se ejecutan en orden correcto y llegan a la persona designada.

El objetivo más claro de un workflow es unir las actividades y aplicaciones dentro del sistema que correspondan a un mismo proceso. Workflow posee ciertas características que le proporciona una capacidad de integración e interoperabilidad, y a su vez puede servir de apoyo en tres áreas funcionales como lo describe figura 2:

Las funciones de tiempo de construcción, con relación a la definición y modelización del proceso de flujo de trabajo.

Las funciones de control de tiempo de ejecución, con relación a la gestión de los procesos de flujo de trabajo en un entorno operativo y también las diversas actividades a tratar como parte de cada proceso.

Las funciones que tienen relación con las iteraciones en tiempo de ejecución con los usuarios humanos y las herramientas de aplicación de TI para el procesamiento de las etapas de las actividades. (Caro Herrero, J. L., A. Guevara Plaza, & A. Aguayo M, 1999)



Existe una gran variedad con respecto a las características que pueden describir a un workflow, sin embargo, se puede simplificar las mismas en los siguiente: (Naranjos Arcos, M. & L. Villarroel Guano , 2002)

Rutas dentro de workflow es indispensable especificar el flujo que va a poseer cualquier tipo de objeto que intervenga dentro del proceso tal como documentos, formularios, datos, aplicaciones, etc. Una vez que se haya realizado la especificación debe ser capaz de moverse por las diferentes rutas secuenciales que existan, con el fin de crear una ruta única.

Reglas es la característica que prima por su importancia dentro del proceso de workflow, ya que con ella es posible establecer y determinar cuál es la información que debe dirigirse a través de una ruta y a quien va ir dirigida.

Roles es determinante dentro del proceso de workflow establecer las funciones que deben desarrollar las personas o procesos.

Procesos se debe lograr identificar los procesos establecidos dentro de la empresa ya que de ellos depende el desarrollo del workflow.

Políticas se refiere a expresiones redactadas de manera formal, escritas con el propósito de establecer la forma en la que se deben manejar ciertos procesos.

Prácticas son consideradas como actividades que ocasionan que se infrinjan las reglas y que hacen que el proceso realmente funciones.

1.3.1 Definiciones manejadas en un workflow

Para el manejo de un workflow se deben manejar los siguientes conceptos:

Tarea Mediante una correcta identificación de las tareas, es posible organizar de manera más ágil un sistema workflow. Estas tareas nacen del estudio del flujo de trabajo, en donde se especifica por quien van a ser ejecutadas.

Proceso Conjunto de recursos que deben ejecutarse para cumplir un objetivo. Permite por medio de procesos ya definidos la creación de otros, por lo tanto, un proceso a más de estar constituido por tareas está compuesto por macro-procesos y sub-procesos.

Agente Lo agentes son entidades, ya sean humanas o computacionales, las cuales están

encargadas de llevar a cabo las actividades de un proceso, las cuales no son automáticas.

Rol Es un conjunto de privilegios y responsabilidades encargadas a una persona.

Ruta Es el orden de pasos que un sistema debe seguir en un sistema workflow.

Dato Los datos son cualquier forma de información (archivos, bdd, imágenes, eléctricas, etc.).

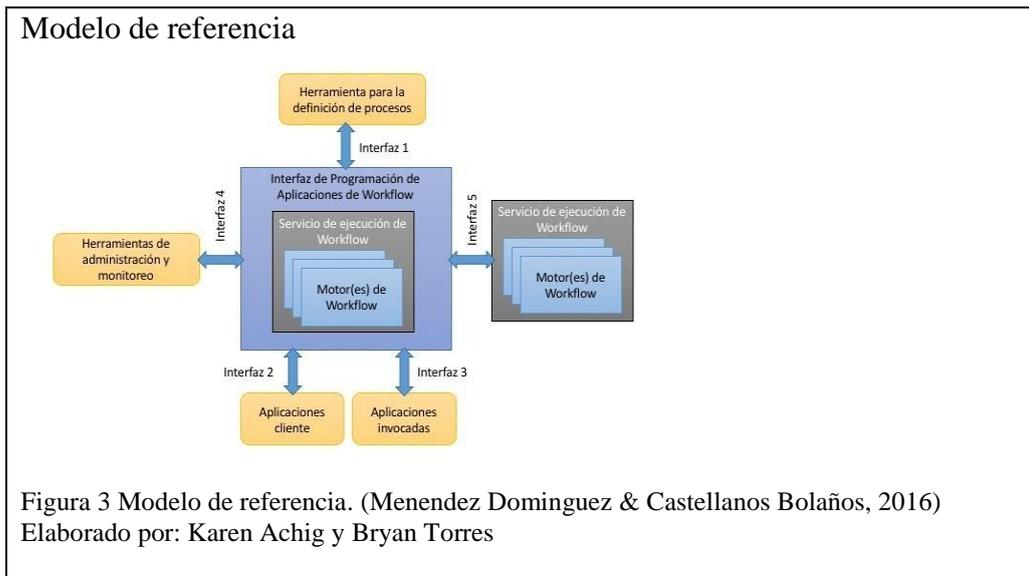
Eventos Son sucesos que ocurren en un sistema, los cuales tienen un solo origen con uno o varios destinatarios y pueden ser disparados por un usuario voluntaria- mente o por medio de un proceso de manera automática.

1.3.2 Modelo de referencia

Workflow posee un modelo de referencia el cual fue elaborado por WFMC con el propósito de tener una estructura genérica o un estándar para el desarrollo de aplicaciones de workflows. Este modelo se basa a partir de estructuras genéricas de aplicaciones de workflow, para que de esta manera sea posible identificar las interfaces con estas estructuras y que así permita comunicarse a distintos niveles. Para lograr tener interoperabilidad entre los diversos productos de workflow, se necesita definir un conjunto de interfaces y formatos para que se pueda realizar el intercambio de datos entre componentes.

A continuación, se describe el modelo de referencia de workflow:

Motor de Workflow El motor de workflow es un software que provee dentro del ambiente de ejecución un control sobre una instancia de workflow, es decir brinda facilidades para la interpretación de la definición de procesos, soporte de interacción con el usuario, control de datos al usuario o hacia aplicaciones, invocación de aplicaciones externas y control de las instancias de los procesos(creación, activación, terminación). (Hollingsworth, D. , 1995)



Servicio de representación de Workflow El servicio de representación de workflow es un componente que permite realizar la interpretación de la descripción de los procesos y además de ello permite controlar las instancias de los procesos o secuencia de actividades. También adiciona elementos a la lista de trabajo de los usuarios y llama a las aplicaciones que sean necesarias. Cuando sea necesaria la intervención del usuario en alguna iteración, se crea una lista de trabajo que es en donde el motor de workflow deposita los elementos que van hacer ejecutados por cada usuario.

Interface de Programación de Aplicaciones de Workflow (WAPI) Las WAPI permiten que se realice una interacción del servicio de representación de workflow con otros recursos existentes y aplicaciones.

Herramientas de definición de procesos (interfaz 1) La interfaz 1 contiene todas las herramientas que se necesiten para realizar la definición del proceso de negocio y a su vez propone un formato estándar para el mismo.

Aplicaciones clientes (interfaz 2) En la interfaz 2 se maneja la interacción entre las aplicaciones cliente y el motor de workflow, esto se lo realiza mediante la lista de trabajo, con la información que esta contiene se le indica al manejador que

aplicaciones hay que invocar.

Esta interfaz debe permitir flexibilidad en lo siguiente:

Los identificadores de procesos y actividades.

Estructuras de datos.

Diferentes alternativas de comunicación.

Aplicaciones invocadas (interface 3) En la interfaz 3 podemos encontrar componentes que permitan invocar funciones de otras aplicaciones, las mismas deben poder interactuar de manera directa con el motor de workflow.

Funciones de interoperabilidad WAPI (Interfaz 4) La interfaz 4 contiene los componentes que permiten el manejo de la interacción entre maquinas que sean heterogéneas.

Herramientas de administración y monitoreo (Interfaz 5) La interfaz 5 contiene los componentes que permiten establecer una vista completa de todo el flujo de trabajo y a su vez se puede establecer auditorias sobre los datos del sistema.

1.3.3 Tipos de workflow

Se establece una clasificación de workflow ya que existen gran variedad de procesos de negocio dentro de una empresa.

Workflow de producción El workflow de producción es el segmento más grande en el mercado ya que esta referenciado a la ejecución de transacciones, permite la automatización de procesos de negocio que tienen a ser repetitivos, que manejan gran volumen de información y además que tiene una buena estructura.

Workflow de colaboración El workflow de colaboración permite establecer una estructura o semi estructura de procesos de negocio en donde participan personas con la finalidad de lograr un objetivo en común. En este workflow se utiliza un documento como un instrumento fundamental, en donde se va a guardar toda la información

esencial, con la finalidad de seguirlos paso a paso. Debido a la importancia de estos documentos es imprescindible que exista integridad en dichos documentos.

Workflow de administración El workflow de administración maneja los procesos administrativos de la empresa como por ejemplo órdenes de compra, reportes de venta, etc.

1.3.4 Modelos de workflow

Estos modelos se desarrollan para representar estructuras de procesos mediante gráficos, determinando las dependencias entre actividades la cuales ayudan al usuario a actuar con efectividad.

Estos modelos pueden desarrollarse de la siguiente manera:

Modelo basado en comunicaciones Este modelo representa la interacción que se da en un workflow entre el cliente y el servidor. Dicha comunicación que se da en el workflow consiste en: pedir, negociar, recibir y aceptar.

Modelo basado en correo electrónico Este modelo es adecuado para aplicaciones de envío de archivos a través de rutas, por ejemplo, la aprobación de un documento.

Modelo basado sobre actividades Este tipo de modelo está basado en la división del proceso en tareas ordenadas.

Modelo estructurado de procesos En este modelo los diagramas de flujo de datos se utilizan para representar los elementos que están involucrados en el sistema de workflow.

Modelo orientado a objetos Estos modelos se centran en la integración de roles y estados, requiriendo fundamentos de especificación formales relacionados con los bloques funcionales críticos.

Modelo de procesos de Comportamiento Este modelo se centra en el control de actividades y ejecución de dependencias entre procesos.

Modelo orientado a Negocios En este modelo se representan de mejor manera los conceptos de comunicación tanto para actores como para la organización.

1.3.5 Ventajas de un workflow

La implementación de un workflow permite tener diversas ventajas, como las siguientes:

Controlar y monitorear tareas

Asignar roles al personal

Incrementar la productividad

Disminuir el tiempo de respuesta entre tareas

Eficiencia en los procesos

Estandarización de procesos

Recursos disponibles cuando sean necesarios

Reducción de costos y recursos

1.4 UML

El UML (Unified Modeling Language) en español Lenguaje Unificado de Modelado, es un lenguaje gráfico que permite visualizar, construir y documentar los componentes de los diferentes sistemas de software, también permite definir todas las vistas necesarias para desarrollar software. (Cabot Sagrera, J. , 2013)

En el UML se usan símbolos, los cuales transmiten significado, este lenguaje es esencial para resolver problemas utilizando imágenes, el uso de imágenes pueden transmitir más información que una gran cantidad de texto. En sí, un modelo es para el software lo que un plano azul es para una casa. (Kimmel, P., 2008)

Elementos que forman parte de la notación UML:

1.4.1 Diagrama de Casos de Uso

El diagrama de casos de uso permite visualizar el conjunto de requisitos del software, este diagrama está definido por un conjunto de casos de uso y cada uno representa una funcionalidad del sistema. Otro elemento importante en el diagrama son los actores, como actor tenemos al elemento externo al sistema el cual tiene algún tipo de interacción con el sistema. Un actor puede ser humano, por ejemplo, el usuario del software o también puede ser otro sistema externo con el que el nuestro se tenga que comunicar. (Cabot Sagrera, J. , 2013)

1.4.2 Diagrama de Clases

El diagrama de clases define los datos que guarda el software con el fin de dar respuesta a las peticiones del usuario, dicho diagrama se usa para mostrar las clases de un sistema y las relaciones entre ellas. Para cada clase se define una serie de atributos y cada atributo representa un pedazo de información que se quiere conocer. (Cabot Sagrera, J. , 2013)

1.4.3 Diagrama de Secuencia

El diagrama de secuencia permite modelar el comportamiento dinámico del sistema, permite definir la interacción y colaboración que hay entre los diferentes elementos del software. En el diagrama de secuencia se muestran el conjunto de mensajes que se generan desde el momento en que el actor empieza la ejecución de la funcionalidad hasta que finaliza, mostrando el orden temporal de los mismos. (Cabot Sagrera, J. , 2013)

1.4.4 Diagrama de Estados

El diagrama de estados como su nombre lo dice, permite ver los diferentes estados por los que pasa un objeto a lo largo de su ciclo de vida. Un diagrama de estados se modela para una clase concreta en el que cada estado representa un mismo objeto en diferentes espacios de tiempo. (Cabot Sagrera, J. , 2013)

1.4.5 Diagrama de Actividad

El diagrama de Actividad sirve como una variante del diagrama de estados, sus elementos más importantes son las actividades y sus transacciones. Tal diagrama puede definir sincronizaciones entre flujo, también puede generar actividades diferentes dependiendo de una condición. (Cabot Sagrera, J. , 2013)

1.4.6 Diagrama de Objetos

El diagrama de objetos, como su nombre lo dice, muestra un conjunto de objetos y sus relaciones. Tal diagrama expresa la parte estática de una interacción.

1.4.7 Diagrama de Interacción

Los diagramas de interacción muestran una interacción concreta, es decir, un conjunto de objetos y sus relaciones, junto con los mensajes que se transfieren entre ellos.

1.4.8 Diagrama de Paquetes

El diagrama de paquetes se usa para reflejar la organización de los paquetes y sus componentes.

1.4.9 Diagrama de Colaboración

Los diagramas de colaboración distribuyen los objetos los cuales permiten observar

adecuadamente una interacción de un objeto con respecto de los demás. Estos diagramas se pueden transformar automáticamente en un diagrama de secuencia.

1.4.10 Diagrama de Tiempo

El diagrama de tiempo se usa para mostrar el cambio en el estado o valor de los elementos en el tiempo, también puede mostrar la interacción entre los eventos de tiempo.

1.4.11 Diagrama de Componentes

El diagrama de componentes es una organización lógica de la implementación de un sistema, como su nombre lo dice, muestra los componentes y dependencias entre ellos.

1.4.12 Diagrama de Despliegue

El diagrama de despliegue es la configuración del sistema en tiempo de ejecución, tal diagrama muestra los nodos de procesamiento y componentes.

Los diagramas UML más usados en el desarrollo de un sistema informático son:

Diagrama de Casos de Uso

Diagrama de Clases

Diagrama de Secuencia

Diagrama de Estados

Diagrama de Colaboración

Capítulo 2

2 Análisis y Diseño

En este capítulo se describen los ciclos de análisis y diseño del sistema, en donde se analiza el proceso manual y automático de la pasantía, lo que ayuda a conocer los requerimientos para el desarrollo del software.

2.1 Análisis

2.1.1 Introducción

Este documento describe las especificaciones de requerimientos para el desarrollo del Sistema de pasantías y prácticas pre profesionales el cual contiene los módulos de visita y validación final.

Por consiguiente, se determina a detalle las características con respecto a la funcionalidad del sistema y se especifica las operaciones que realiza cada usuario.

2.1.2 Propósito

Sistema de pasantías y prácticas pre profesionales es una aplicación que va ligada directamente a un proceso manual que se lleva a cabo dentro de la Universidad Politécnica Salesiana, principalmente orientada a estudiantes que cursan los últimos niveles de la carrera de Ingeniería de Sistemas. La aplicación tiene como propósito ser un mecanismo que facilite el procedimiento de pasantías tanto para los estudiantes como para los docentes.

2.1.3 Alcance

El sistema permite realizar el proceso de visita y validación de pasantías de la Universidad Politécnica Salesiana, tanto por los estudiantes como por los docentes de

la misma.

Específicamente, en el sistema el estudiante confirma una cita para la visita con el docente tutor que se le haya sido asignado, una vez culminada su pasantía puede validar la misma dentro del sistema.

Por otra parte, en el sistema el docente tutor agenda una cita para la visita con el estudiante que se le haya sido asignado, también puede subir toda la documentación que valida la cita realizada.

Por último, se debe mencionar que el Sistema requiere de firmas electrónicas para la documentación del proceso de visita y validación, pero esta característica no forma parte del desarrollo de proyecto, por lo que no se la considera dentro de los requerimientos.

2.2 Visión General

2.2.1 Diagnóstico Actual del Proceso

A continuación, se muestra el modelo del proceso de pasantías que se lleva a cabo actualmente en la Universidad Politécnica Salesiana, el cuál se realiza de forma manual teniendo procesos tediosos que se pueden suprimir reduciendo las tareas de cada actor en el proceso.

Workflow manual

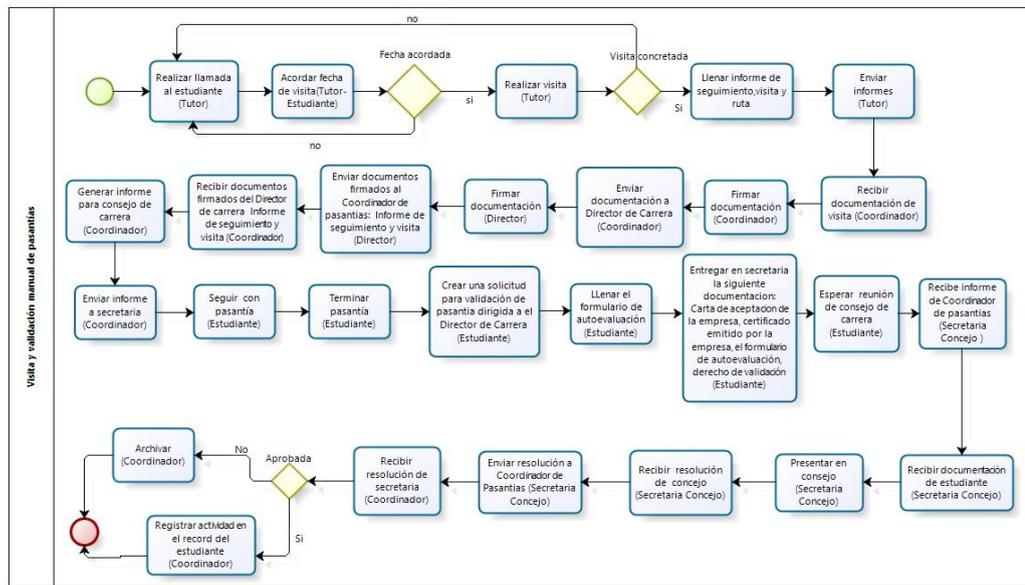
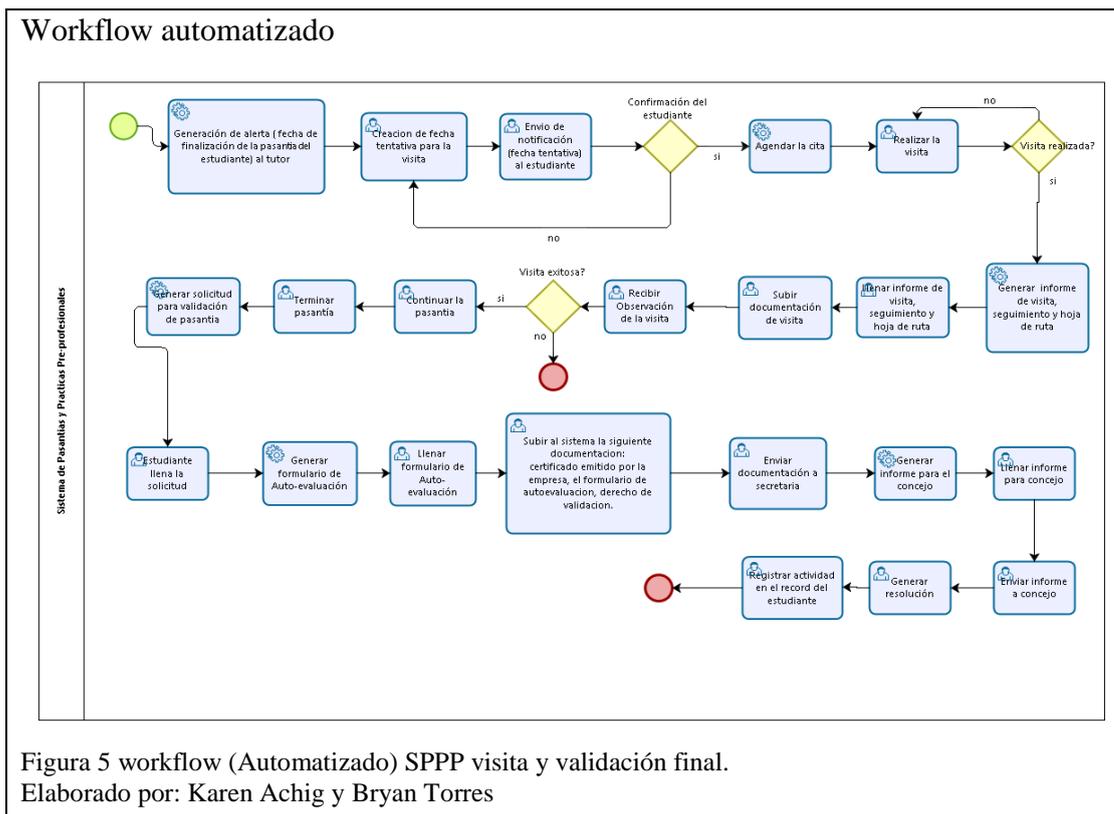


Figura 4 workflow (Manual) de visita y validación de la pasantía.
Elaborado por: Karen Achig y Bryan Torres

2.2.2 Modelo Automatizado Propuesto

A continuación, se muestra el modelo del proceso ya automatizado, el cuál fue propuesto después de un análisis con diferentes docentes de la Universidad Politécnica Salesiana, el mismo que suprime pasos y ayuda a optimizar recursos.



2.3 Descripción global

2.3.1 Perspectiva del producto

El sistema de pasantías y prácticas pre profesionales está diseñado para que tanto estudiantes como docentes, lleven a cabo el proceso de pasantías mediante una interfaz Web amigable e intuitiva, ya que en esta estarán establecidos los pasos a seguir para culminar con éxito dicho procedimiento. El sistema precisa una base de datos para almacenar la información y un servidor.

2.3.2 Funciones del producto

2.3.2.1. Módulo de visita

En este módulo se implementan todos los pasos que se deben llevar a cabo para el desarrollo de la visita tanto para el tutor asignado como el estudiante.

Se realiza lo siguiente:

Enviar alertas al tutor vía e-mail.
Crear fecha tentativa para la visita (Tutor).
Enviar alertas al estudiante vía e-mail.
Agendar la cita para una visita.
Generar informe de visita del tutor.
Generar informe de seguimiento
Subir documentos de visita al sistema.

2.3.2.2. Módulo de validación final

Este módulo está enfocado directamente con el estudiante ya que le va a permitir realizar la validación de la pasantía.
Generar solicitud para validación de la pasantía.
Generar hoja de autoevaluación.
Subir documentación de validación al sistema.
Validación de documentación.
Validación de información subida al sistema.

2.4 Restricciones

Se debe tomar en cuenta que el acceso al sistema tiene que ser a través de un navegador web, además el sistema será instalado en un servidor de aplicaciones.

2.5 Dependencia

El sistema depende del Sistema de pasantías y prácticas pre profesionales módulos:

Módulo de carga de información

Módulo de registro de pasantía

Módulo de validación de documentación

Módulo de activación de inicio de pasantía.

2.6 Requerimientos específicos

Requerimiento de interfaz externo Interfaz del usuario

El sistema provee al usuario interfaces web intuitivas con la finalidad de ofrecer al usuario una interacción agradable y lo más minimalista posible.

2.6.1 Interfaz del hardware

El sistema de pasantías y prácticas pre profesionales es una aplicación standalone ya que ninguna de sus funcionalidades requiere de alguna comunicación con sistemas externos. La pantalla en la que se ejecute el sistema debe ser capaz de soportar una resolución mínima de 1024*768.

2.6.2 Interfaz del software

El sistema debe permitir la comunicación con los módulos: módulo de carga de información, de registro de pasantía, de validación de documentación y de activación de inicio de pasantía del sistema de pasantías y prácticas pre profesionales.

2.6.3 Interfaz de comunicaciones

El sistema de pasantías y prácticas pre profesionales trabaja sobre el protocolo HTTP.

2.7 Requerimientos funcionales

Los requerimientos funcionales se han dividido en dos, de acuerdo con los módulos que conforman el sistema. En una primera instancia se describen los requerimientos del módulo de visita y a continuación se encuentran los requerimientos del módulo de Validación final.

2.7.1 Módulo de Visita

Tabla 1 Requerimiento funcional 1.1: Envío de alertas al tutor vía e-mail para la visita.

Descripción	Envío automático de alertas por medio de correo electrónico al tutor.
Precondición	Tener un estudiante asignado.
Entrada	Fecha de culminación de pasantías del estudiante.
Proceso	El sistema genera una notificación para el tutor, con la cual se alerta al mismo que ya debe generar una fecha tentativa de visita.
Salida	Alertas en el correo institucional del tutor.

Nota: Esta tabla contiene información acerca del envío de correos al tutor.

Tabla 2 Requerimiento funcional 1.2: Creación de fecha tentativa para la visita.

Descripción	Creación automática de fecha para una posible visita.
Precondición	Haber recibido la alerta de visita anteriormente.
Entrada	Estudiante asignado, fecha de culminación de pasantías del estudiante, horario tutor.
Proceso	El tutor puede generar en el sistema una posible fecha para realizar la visita, la cual debe ser confirmada por el estudiante.
Salida	Fecha tentativa para la visita.

Nota: Esta tabla contiene información acerca de la creación de la fecha para la visita.

Tabla 3 Requerimiento funcional 1.3: Envío de alertas al estudiante vía e-mail para la visita.

Descripción	Envío automático de alertas por medio de correo electrónico.
Precondición	Haber empezado la pasantía.
Entrada	Fecha de inicio y culminación de la pasantía del estudiante.
Proceso	Generación de notificación al estudiante con la posible fecha de visita.
Salida	Alerta de posible visita en el correo institucional del estudiante.

Nota: Esta tabla contiene información acerca del envío de correos al estudiante.

Tabla 4 Requerimiento funcional 1.4: Agendar la cita para una visita.

Descripción	Agendación de cita en el sistema.
Precondición	Confirmación de cita por medio del estudiante.
Entrada	Fecha de visita.
Proceso	Una vez que el estudiante confirma la fecha para la visita, esta queda agendada en el sistema.
Salida	Fecha de visita agendada.

Nota: Esta tabla contiene información acerca de la agendación de la cita.

Tabla 5 Requerimiento funcional 1.5: Generar informe de visita del tutor.

Descripción	Permite generar el informe de visita automáticamente.
Precondición	Fecha de visita confirmada.
Entrada	Carta compromiso, tipo de actividad, datos tutor, datos del estudiante, valoración del tutor, observaciones.
Proceso	El tutor llena el informe en el momento de la visita.
Salida	Archivo PDF.

Nota: Esta tabla contiene información acerca de la generación de formatos del tutor.

Tabla 6 Requerimiento funcional 1.6: Generar informe de seguimiento de la pasantía.

Descripción	Permite generar el informe de seguimiento automáticamente.
Precondición	Haber empezado la pasantía.
Entrada	Carta compromiso, tipo de actividad, datos tutor, datos del estudiante, datos tutor de la empresa y responsable de la empresa, actividades realizadas, horas asignadas a cada actividad, criterios de desempeño, horas totales, horas parciales, fecha de visita.
Proceso	El tutor llena el informe y lo sube al sistema.
Salida	Archivo PDF.

Nota: Esta tabla contiene información acerca de la generación de formatos del tutor.

Tabla 7 Requerimiento funcional 1.7: Subir documentos de visita al sistema.

Descripción	El tutor puede subir la documentación de la visita, la cual valide el cumplimiento de la misma.
Precondición	Visita realizada.
Entrada	Documentos llenos.
Proceso	Una vez llenos los documentos el tutor los debe subir al sistema, cada documento puede tener un peso máximo de 2Mb y estar en formato PDF.
Salida	Documentos subidos al servidor.

Nota: Esta tabla contiene información acerca de la subida de documentación por parte del tutor.

2.7.2 Módulo de Validación Final

Tabla 8 Requerimiento funcional 2.1: Generar solicitud para validación de pasantía.

Descripción	Se genera automáticamente el formato de solicitud para la validación de la pasantía.
Precondición	Pasantía terminada.
Entrada	Información estudiante.
Proceso	El estudiante al momento de culminar su pasantía ingresa al sistema y llena la solicitud de validación pasantía, la cual es enviada al director de carrera automáticamente.
Salida	Solicitud registrada.

Nota: Esta tabla contiene información acerca de la solicitud realizada por el estudiante.

Tabla 9 Requerimiento funcional 2.2: Generar hoja de autoevaluación

Descripción	Generar formato de autoevaluación para validación de pasantías.
Precondición	Pasantía terminada.
Entrada	Datos del estudiante, carta compromiso, tipo de actividad (Práctica o pasantía), resolución inicial del concejo, valoración del estudiante.
Proceso	Llenar el formato de autoevaluación basándose en un nivel de madurez evaluado por cada estudiante.
Salida	Archivo PDF.

Nota: Esta tabla contiene información acerca de la autoevaluación llenada por el estudiante.

Tabla 10 Requerimiento funcional 2.3: Subir documentación de validación al sistema.

Descripción	Permite subir la documentación con la cual se valida el proceso de pasantías.
Precondición	Pasantía terminada.
Entrada	Certificado emitido por la empresa, formato de autoevaluación, derecho de validación.
Proceso	El estudiante sube al sistema la documentación en formato PDF, la cual puede tener un peso máximo de 2MB por cada archivo.
Salida	Documentación guardada en el servidor.

Nota: Esta tabla contiene información acerca de la subida de documentación por parte del estudiante.

Tabla 11 Requerimiento funcional 2.4: Validación de documentación.

Descripción	Se puede revisar que la documentación subida por el estudiante sea la correcta.
Precondición	Ninguna.
Entrada	Documentos subidos al sistema.
Proceso	El coordinador revisa la documentación que subió el estudiante para observar la presencia o no presencia de anomalías.
Salida	Notificación al estudiante de documentación correcta o incorrecta.

Nota: Esta tabla contiene información acerca de la validación de los documentos subidos por el estudiante.

Tabla 12 Requerimiento funcional 2.5: Validación información subida al sistema

Descripción	El coordinador valida la información que sube el estudiante.
Precondición	Ninguna
Entrada	Información subida por el estudiante.
Proceso	En la etapa de validación el coordinador podrá corroborar si la información que sube el estudiante es la correcta o no la es.
Salida	Notificación al estudiante de información correcta o incorrecta.

Nota: Esta tabla contiene información acerca de la validación de información por parte del coordinador.

2.8 Requerimientos no funcionales

Tabla 13 Requerimiento no funcional 1.1: Portabilidad

Descripción	El sistema debe ejecutarse en diferentes plataformas.
Razón	Es necesario que el sistema pueda ser instalado en más de un sistema operativo.
Criterio de medición	Instalación correcta en diferentes sistemas operativos.

Nota: Esta tabla contiene información acerca de la portabilidad del sistema. (Flores, Castillo, & Tufiño, 2018)

Tabla 14 Requerimiento no funcional 1.2: Interfaz gráfica del usuario GUI.

Descripción	En la interfaz el usuario puede apreciar lo necesario, para comprender el flujo de trabajo del sistema.
Razón	Una interfaz gráfica correcta logra que los procesos se lleven de una manera adecuada.
Criterio de medición	Pruebas con usuarios.

Nota: Esta tabla contiene información acerca de la interfaz del sistema. (Flores , Castillo, & Tufiño, 2018)

Tabla 15 Requerimiento no funcional 1.3: Sistema Responsive

Descripción	El sistema debe ser adaptativo para que permita unavvisualización correcta en cualquier plataforma.
Razón	Un sistema responsive permite al usuario una experiencia más agradable en cualquier ambiente.
Criterio de medición	Pruebas de adaptabilidad en varios dispositivos como: laptops, computadores y celulares.

Nota: Esta tabla contiene información acerca de la propiedad responsive. (Flores , Castillo, & Tufiño, 2018)

Tabla 16 Requerimiento no funcional 1.4: Mensajes de error en el sistema (POP- UP)

Descripción	En el sistema se despliegan mensajes de error que le permiten al usuario informarse de una mala acción.
Razón	Cuando ocurra alguna mala acción el sistema debe ser capaz de informarle dicha acción al usuario y en qué punto está fallando.
Criterio de medición	Pruebas con estudiantes y docentes

Nota: Esta tabla contiene información acerca de los cuadros de dialogo. (Flores , Castillo, & Tufiño, 2018)

Tabla 17 Requerimiento no funcional 1.5: Sesiones concurrentes

Descripción	El sistema debe ser capaz de operar correctamente con hasta 100 usuarios con sesiones concurrentes.
Razón	El sistema opera adecuadamente mientras se encuentren 100 usuarios con sesiones concurrentes.
Criterio de medición	Pruebas de carga.

Nota: Esta tabla contiene información acerca de las sesiones dentro del sistema. (Flores , Castillo, & Tufiño, 2018)

Tabla 18 Requerimiento no funcional 1.6: Tiempo de respuesta

Descripción	La rapidez con la que el sistema realiza una acción o petición que realiza el usuario.
Razón	El tiempo que espera el usuario para que se concrete una acción debe ser máximo de 2 segundos.
Criterio de medición	Pruebas con JUnit.

Nota: Esta tabla contiene información acerca del tiempo de respuesta del sistema.

Tabla 19 Requerimiento no funcional 1.7: Seguridad

Descripción	La información manejada por el sistema está protegida de acceso no autorizado o divulgación de datos.
Razón	Los respaldos son almacenados en el Storage. Y el único que puede editar los permisos es el administrador del sistema.
Criterio de medición	Pruebas de integridad de datos.

Nota: Esta tabla contiene información acerca de la seguridad del sistema.

Tabla 20 Requerimiento no funcional 1.8: Disponibilidad

Descripción	La plataforma debe soportar 24/7 de funcionamiento ininterrumpido.
Razón	Se requiere una disponibilidad permanente 24/7, no solo laboral.
Criterio de medición	Verificación de disponibilidad.

Nota: Esta tabla contiene información acerca de la disponibilidad del sistema. (Flores , Castillo, & Tufiño, 2018)

Tabla 21 Requerimiento no funcional 1.9: Ventana de ayuda

Descripción	Se detalla todos los pasos del procedimiento y preguntas frecuentes.
Razón	Requerimiento de ayuda de parte del estudiante o tutor.
Criterio de medición	Número de transacciones completas

Nota: Esta tabla contiene información acerca de la ventana de ayuda dentro del sistema. (Flores , Castillo, & Tufiño, 2018)

Tabla 22 Requerimiento no funcional 1.10: Estado de flujo.

Descripción	El tutor o el estudiante pueden observar en que parte del proceso se encuentra.
Razón	Visualizar el estado en el que se encuentra un proceso.
Criterio de medición	Indicador de estado actual.

Nota: Esta tabla contiene información acerca del estado del proceso en el que se encuentra el estudiante.

2.9 Diseño

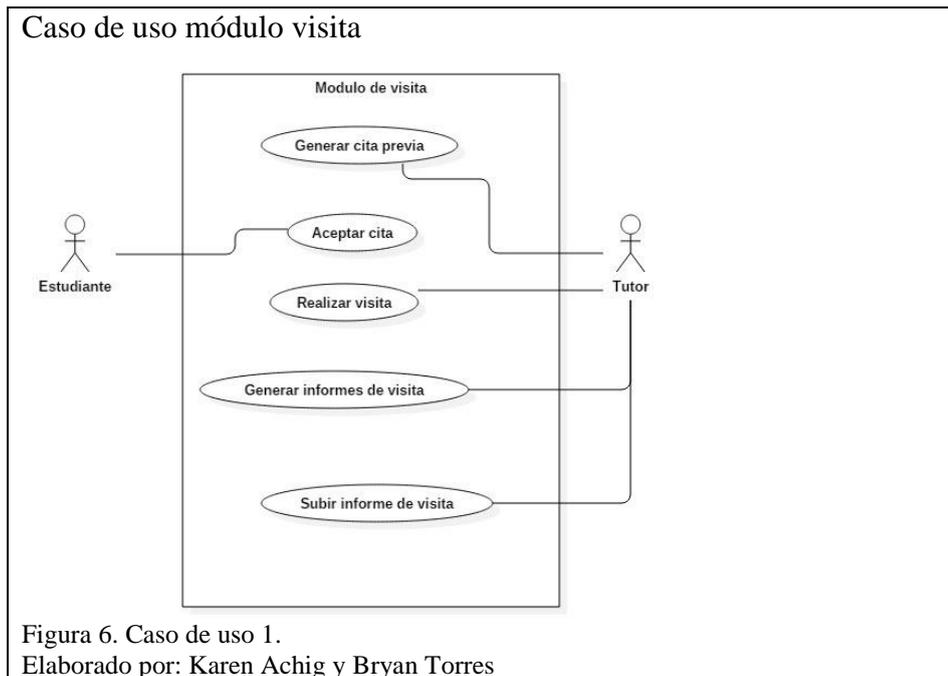
El diseño del sistema de pasantías y prácticas pre profesionales plantea la solución para que dicho proceso se lo lleve de manera automática, aquí se define las tareas que puede realizar el usuario y la estructura con la que cuenta el sistema, este proceso se lleva a cabo mediante los diagramas de casos de uso, los diagramas de clases, el mapa conceptual de la base de datos, interfaces de usuario y las historias de usuario.

2.9.1 Diagramas de caso de uso

El diagrama de casa de uso permite mostrar las principales tareas que pueden realizar los diferentes actores del sistema:

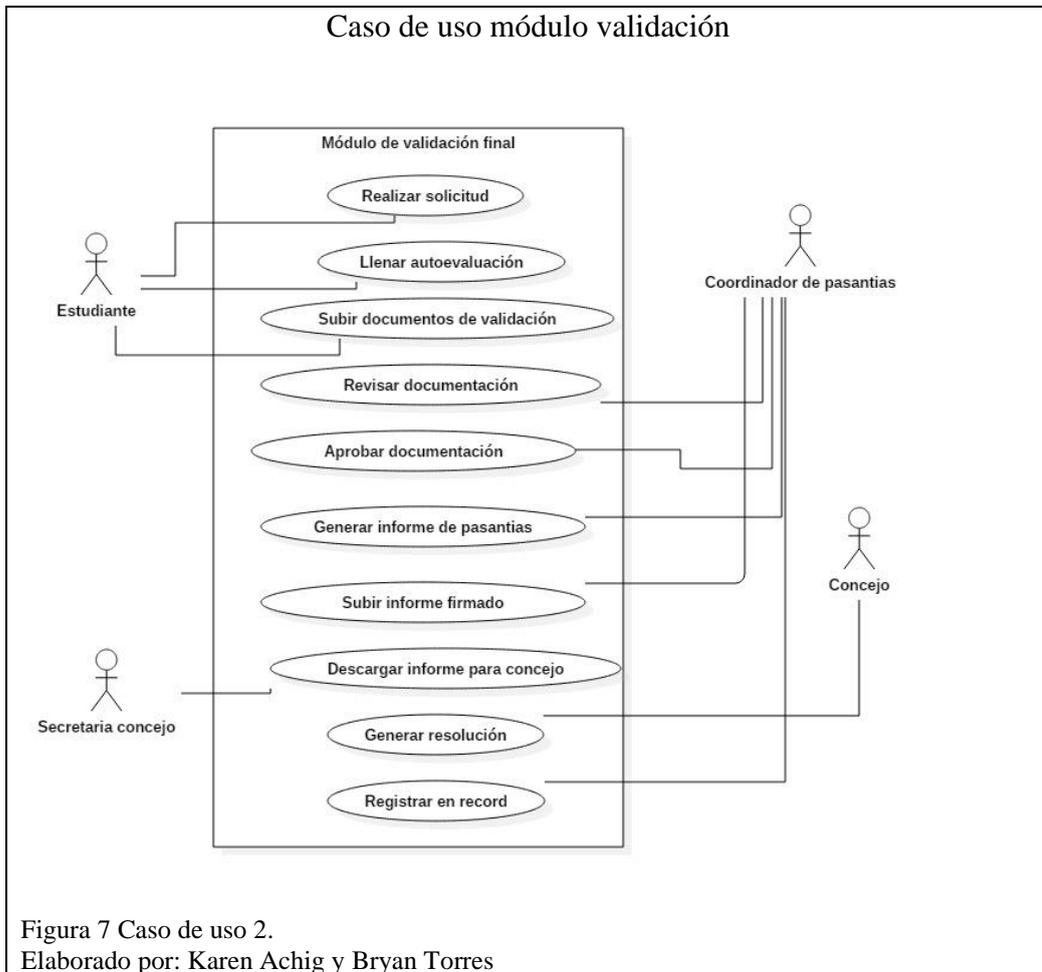
2.9.1.1. Casos de uso 1: Modulo de visita

En este caso de uso los actores que interactúan son el estudiante y el tutor, las operaciones a las que pueden acceder están relacionadas con el módulo de vista.



2.9.1.2. Casos de uso 2: Modulo validación final

En este caso de uso los actores que interactúan son el estudiante, tutor, coordinador de pasantías, director de carrera, secretaria de concejo y el concejo, las operaciones a las pueden acceder están relacionadas con el módulo de validación final.



2.9.1.3. Diagramas de clases

En la siguiente sección se muestran las clases que el sistema posee, las mismas que son usadas para cumplir con los casos de uso:

Diagrama de clases

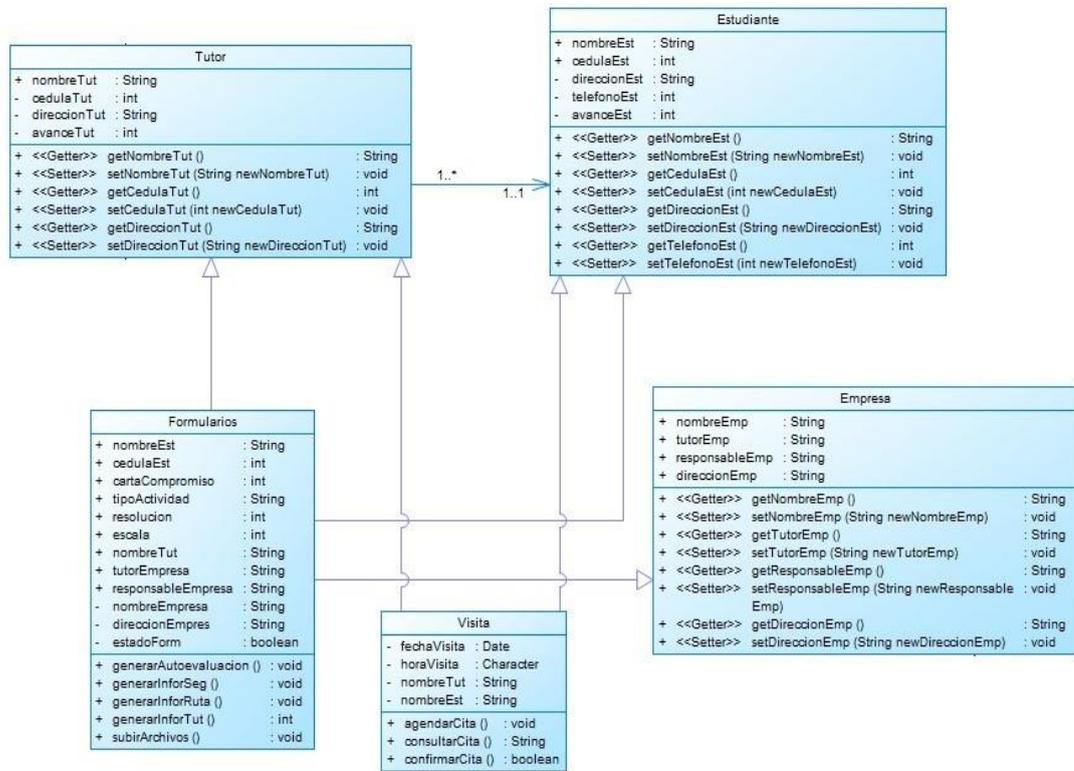


Figura 8 Diagrama de clases.

Elaborado por: Karen Achig y Bryan Torres

2.10 Base de datos

La base de datos se desarrolló teniendo en cuenta los requerimientos planteados en el SRS, con el presente diagrama se muestra las tablas que se han utilizado para diseñar el sistema

Tabla 23 Tabla de base de datos

TABLA	DESCRIPCIÓN
tb-perfil	Contiene la información sobre los perfiles que maneja el sistema.
tb-estudiante	Contiene la información específica del estudiante.
tb-tutor	Contiene la información específica del tutor.
tb-usuario	Contiene la información de cada usuario.
tb-proceso	Contiene la información de cada proceso que se lleva a cabo dentro del sistema.
tb-visitator	Contiene la información acerca de la visita a realizarse.
tb-formato	Contiene la información sobre los formatos manejados en el sistema.
tb-preguntas	Contiene la información de las preguntas que tiene cada formato.
tb-tipo-pregunta	Contiene la información sobre el tipo de pregunta que contiene cada formato.
tb-respuesta	Contiene la información sobre las respuestas que tiene cada pregunta.
tb-datos	Contiene los valores obtenidos de las respuestas, así como los detalles de la pasantía.
tb-detalle-pasantia	Contiene la información detallada de cada proceso que se lleva a cabo dentro de la pasantía.
tb-pasantia	Contiene la información específica del proceso de pasantía.
tb-periodo	Contiene la información sobre periodo en que se desarrolla el proceso de pasantía.
tb-empresa	Contiene la información sobre la empresa en donde se lleva a cabo la pasantía.
tb-encargado-empresa	Contiene la información del encargado de la empresa.

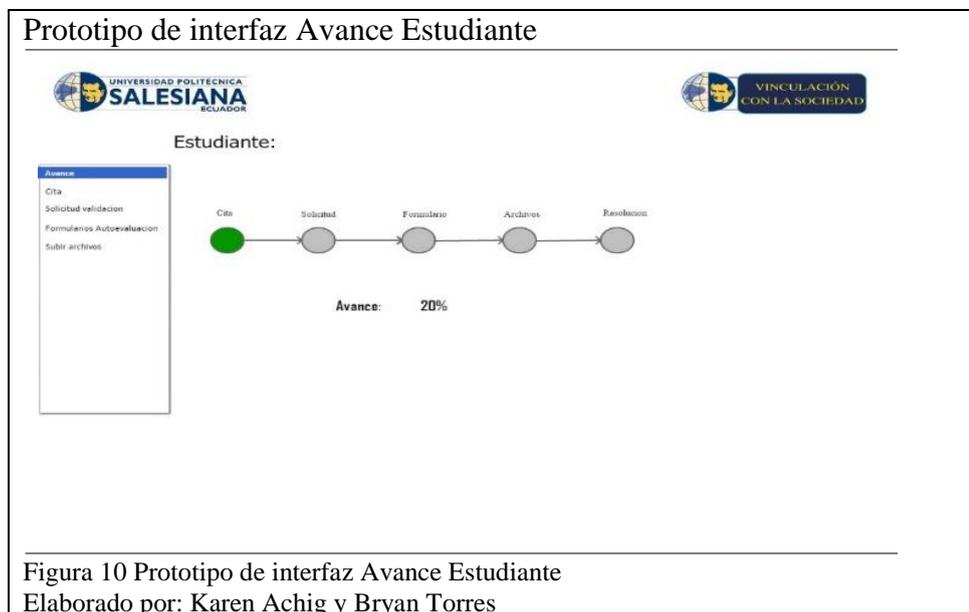
Nota: Esta tabla contiene información acerca de los campos de la base de datos.

2.11 Interfaz de usuario

Para el modelamiento de las interfaces del usuario se ha tomado en cuenta criterios simples, intuitivos, agradables para el usuario con un diseño minimalista.

2.11.1 Estudiante.

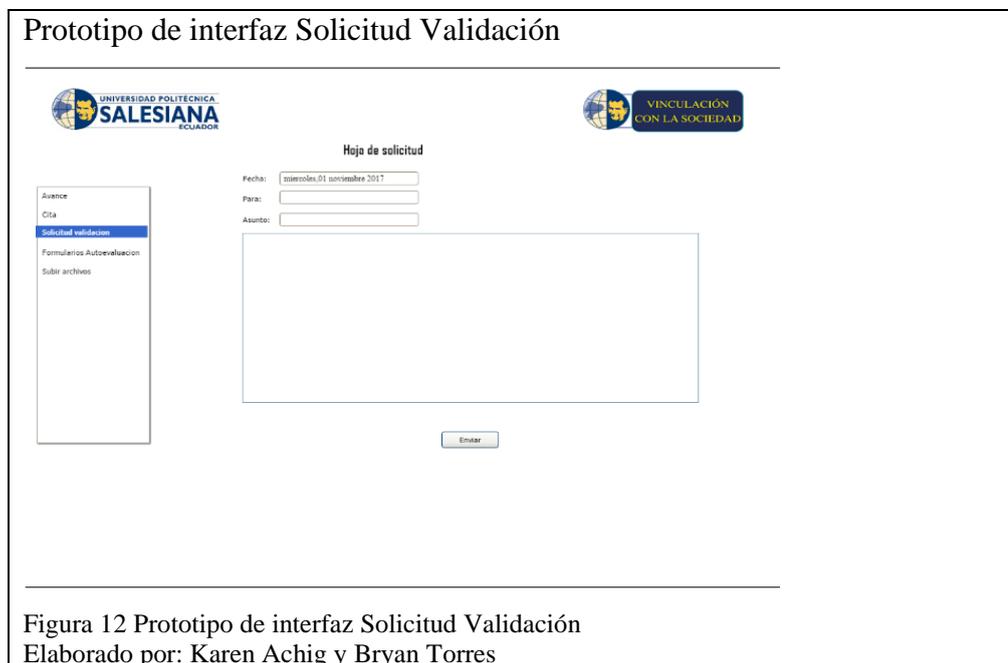
Avance: En esta pantalla se puede visualizar el avance del proceso que esta llevando a cabo el estudiante.



Cita En esta pantalla se visualiza la posible cita, la cual puede ser confirmada o negada.



Solicitud validación En esta pantalla se visualiza el formulario a llenar para realizar la validación del proceso de pasantías.



Formulario autoevaluación En esta pantalla se visualizará el formulario de autoevaluación a llenar, necesario para cumplir con los requisitos para la validación.

Prototipo de interfaz Formato Autoevaluación

Autoevaluación

Carta compromiso: Tipo de actividad:

Resolución de consejo de inicio de actividad:

Estudiante: Codigo de ciudadanía:

Auto- evalúe con honestidad su grado de participación durante el tiempo que efectuó en la institución la pasantía

Muy satisfactorio 5
Satisfactorio 4
Aceptable 3
Deficiente 2
Mala 1

ASPECTOS Y EVIDENCIAS	ESCALA				
	1	2	3	4	5
a. Asistencia y puntualidad durante la extensión					
b. Responsabilidad, disposición y cumplimiento en la					
c. En las relaciones con el personal de la Institución ha					
d. Utilización adecuada de procedimientos					
e. Conocimientos teóricos y prácticos de la cámara.					

Firma del Estudiante: _____

Figura 13 Prototipo de interfaz Formato Autoevaluación
Elaborado por: Karen Achig y Bryan Torres

Subir archivos En esta pantalla se puede subir los siguientes archivos: carta de aceptación, certificado de la empresa, formulario de autoevaluación y derecho, los mismos que son requisitos para el proceso de validación de pasantías (dichos archivos deben ser subidos en formato PDF y pueden tener un tamaño máximo de 2Mb).

Prototipo de interfaz Subir Archivos

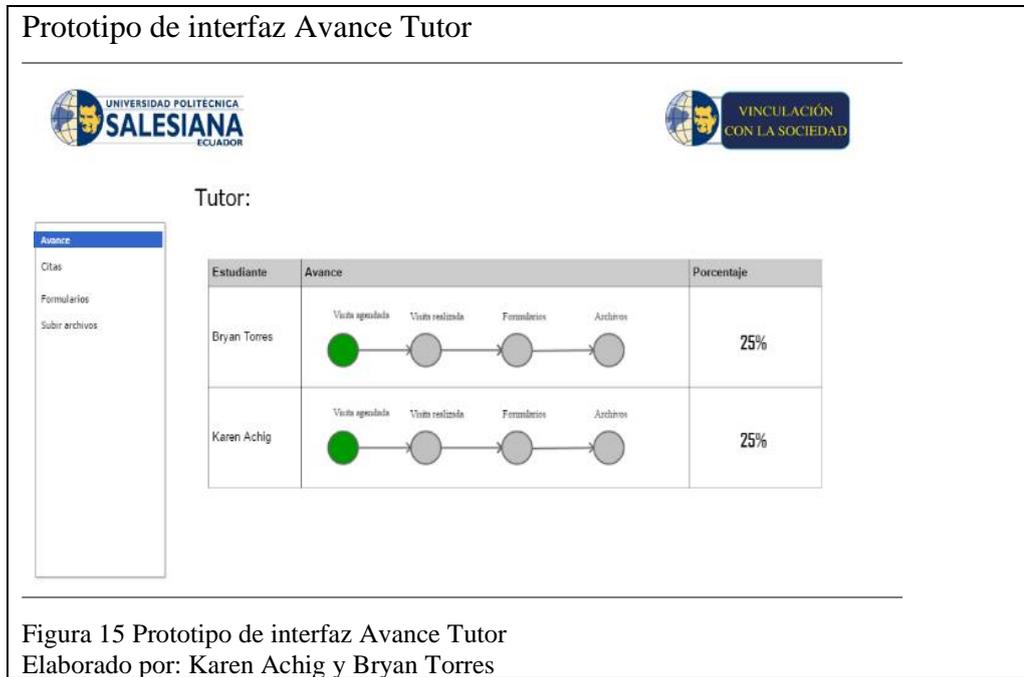
Certificado de la empresa

Search

Figura 14 Prototipo de interfaz Subir Archivos
Elaborado por: Karen Achig y Bryan Torres

2.11.2 Tutor

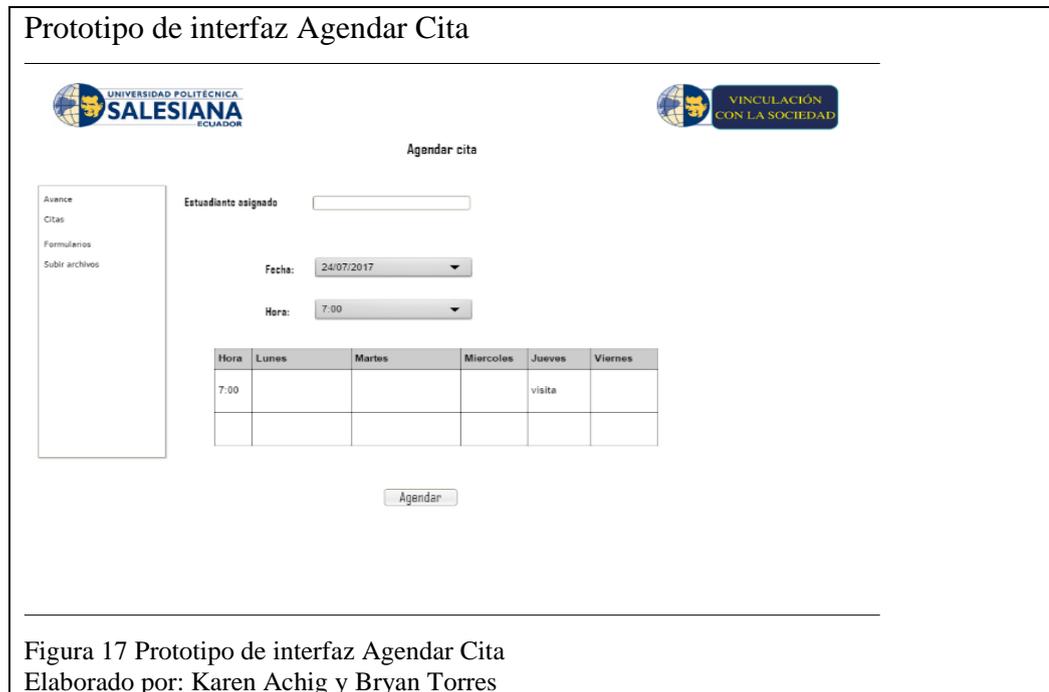
Avance En esta pantalla se visualiza el avance de cada estudiante asignado al tutor.



Citas En esta pantalla se puede elegir entre las opciones de: agendar una cita y citas agendadas con respecto al proceso de visita.



Agendar cita En esta pantalla se pueden establecer una fecha y hora tentativa para la visita, la misma que se visualiza en el horario del tutor.



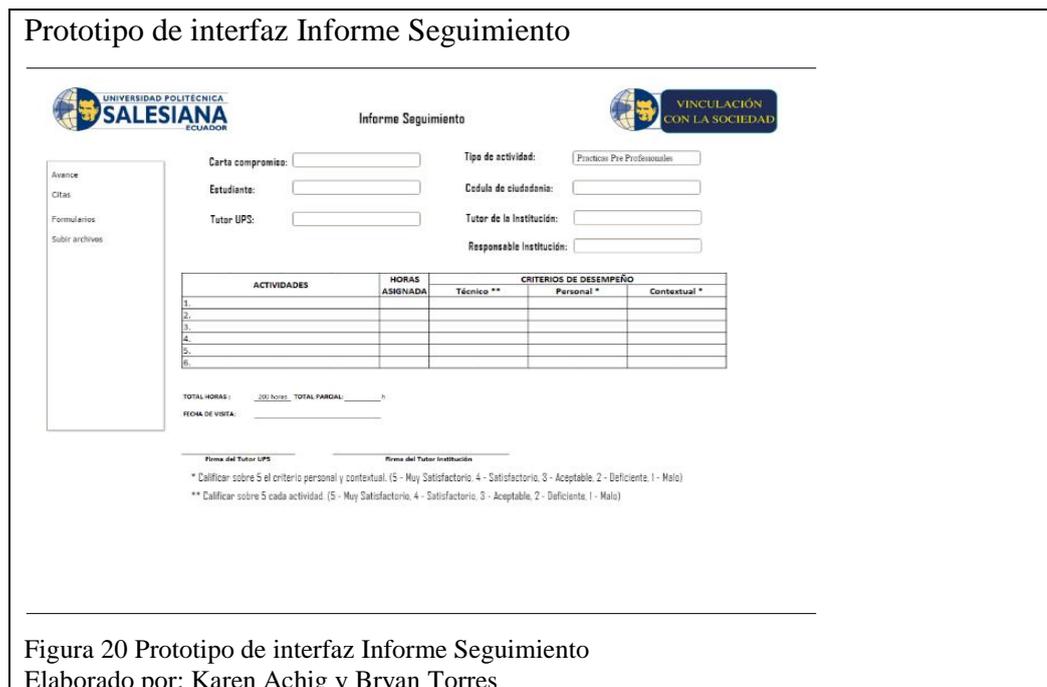
Citas agendadas En esta pantalla se visualiza una lista en donde consta los datos de la visita y la prioridad de la misma (Rojo: la cita esta próxima, verde: la cita esta lejana y azul: el estudiante ha sido visitado).



Formularios En esta pantalla se puede elegir entre los formularios de: informe seguimiento, informe tutor y hoja de ruta con respecto al proceso de validación de visita.



Informe de seguimiento En esta pantalla se visualiza el formato a llenar del informe de seguimiento.



Informe del tutor En esta pantalla se visualiza el formato a llenar del informe del tutor.

Prototipo de interfaz Informe de Tutor



**UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR**



**VINCULACIÓN
CON LA SOCIEDAD**

Informe tutor

Avance

Citas

Formularios

Subir archivos

Carta compromiso:

Tutor:

Estudiante:

Tipo de actividad:

Cedula de ciudadanía:

Muy satisfactorio 5

Satisfactorio 4

Aceptable 3

Deficiente 2

Malo 1

CUESTIONARIO	ESCALA				
	1	2	3	4	5
a. ¿Se cumplió con las actividades propuestas?					
b. ¿Satisficieron los resultados a la labor?					
c. ¿El estudiante tuvo la información necesaria?					
d. La calidad de los productos ofrecidos fueron:					
e. El comportamiento del estudiante en la					
f. La destreza demostrada del estudiante en					
g. El nivel de información proporcionada por la					
h. La relación del estudiante con el tutor de la					
i. La relación del estudiante con el tutor de la					

En caso de tener observaciones, inquietudes y/o sugerencias, detallar a continuación:

Firma Docente Tutor _____

Figura 21 Prototipo de interfaz Informe de Tutor
Elaborado por: Karen Achig y Bryan Torres

Subir archivos En esta pantalla se puede subir los siguientes archivos: informe de seguimiento, informe de tutor y hoja de ruta (dichos archivos deben ser subidos en formato PDF y pueden tener un tamaño máximo de 2Mb).



Capítulo 3

3 Construcción y Pruebas

En este capítulo se va a describir el proceso que se realizó para la construcción del sistema, también se detalla las herramientas utilizadas para el desarrollo del mismo y las pruebas a las que se sometió el sistema para corroborar su correcto funcionamiento.

3.1 Herramientas

3.1.1 BootsFaces

Es un framework JSF que está basado en Bootstrap 3 y jQuery, que se caracteriza por ser liviano y muy potente, con el mismo se puede desarrollar aplicaciones de forma rápida y sencilla. BootsFaces se centra principalmente en la estructura y en el diseño de la página con el objetivo de permitir al desarrollador obtener páginas con un buen diseño y con poco esfuerzo.

BootsFaces permite aprovechar totalmente el sistema de Grid de Bootstrap y al mismo tiempo ofrece los beneficios de un marco JSF, aunque es más expresivo y conciso en comparación a la programación nativa que ofrece Bootstrap. (BootsFaces., 2013)

3.1.2 NetBeans

NetBeans es un IDE (Entorno de desarrollo integrado) que permite de una manera fácil y rápida desarrollar diferentes tipos de aplicaciones ya sea de escritorio, móviles o web. También ofrece variedad en la vista de sus datos como: ventanas de proyección múltiple, herramientas de configuración para las aplicaciones, entre otras.

NetBeans facilita la profundización de datos de una manera rápida y sencilla, además ayuda en la integración de código con sus herramientas integradoras como Git,

mercurial, entre otras. Una de sus principales características es que ofrece un análisis estático y un depurador con el cual se puede identificar errores de una manera más sencilla. (NetBeans, 1997)

3.1.3 Ajax

Ajax (acrónimo de JavaScript y XML) se puede definir a Ajax como una técnica que ayuda a facilitar el desarrollo web, permitiendo crear aplicaciones intuitivas e interactivas ya que es una tecnología asíncrona, es decir que se solicita al servidor los datos adicionales y los mismos se cargan sin interrumpir a la página que se está ejecutando haciendo que para el usuario sea un proceso transparente.

Una de las características más destacable de Ajax es que es compatible con todos los exploradores haciéndolo así más versátil y útil. (AJAX, 2011)

3.1.4 Itext 5.0.5

Itext es una librería open source que permite la creación de archivos en formato pdf, rtf y html.

Esta librería ayuda a simplificar los procesos de documentación durante el desarrollo de una aplicación ya que facilita la organización de la información de manera adecuada dentro de un documento integrando características como: márgenes, tablas, tipos de letras, entre otras. (Itext, 2010)

3.1.5 GlassFish server

GlassFish es un servidor de aplicaciones open source desarrollado por Sun Microsystems que implementa las tecnologías en la plataforma Java EE, permite la ejecución de aplicaciones que siguen la misma especificación referente a la tecnología.

Este servidor proporciona un soporte para la gestión ya sea remota multi-maquina o

multi-dominio segura, con esto se busca mejorar la escalabilidad y la disponibilidad de sistemas de alto volumen y servicios web. (Ponce, A. J. & R. C. Choque, 2014)

3.1.6 PostgreSQL

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python). (Rodríguez González & María Elena. , 2013)

3.1.7 JavaScript

JavaScript fue introducido en 1995 como una forma de agregar programas a páginas web en el navegador Netscape Navigator. El lenguaje ha sido adoptado desde entonces por todos los otros principales navegadores web gráficos. Ha hecho posibles aplicaciones web modernas: aplicaciones con las que puedes interactuar directamente, sin necesidad de recargar la página para cada acción. Pero también se utiliza en sitios web más tradicionales para proporcionar diversas formas de interactividad y habilidad. Es importante tener en cuenta que JavaScript no tiene casi nada que ver con el lenguaje de programación llamado Java. El nombre similar fue inspirado por consideraciones de la comercialización, más bien que buen juicio. Cuando se introdujo JavaScript, el lenguaje Java se estaba comercializando fuertemente y estaba ganando popularidad. Alguien pensó que era una buena idea tratar de montar a lo largo de este éxito. Ahora estamos atrapados con el nombre. (Mohedano, J., J. M. Saiz , & R. P Salazar , 2012)

3.1.8 MVC

Modelo-Vista-Controlador es un patrón utilizado en diferentes frameworks en el que la funcionalidad se separa de la interfaz de usuario, con lo cual se incrementa la reutilización del código. Teniendo así que: El modelo (M), representa el almacenamiento, consulta y modificación de los datos. La vista (V), toma los datos del modelo y los convierte en información para ser mostrada al usuario. El controlador (C), maneja todos los eventos de entrada advirtiendo al modelo que el usuario está ejecutando una acción. (Sarasa, C. A, 2017)

3.1.9 JQuery

Es una biblioteca de JavaScript, la cual permite ahorrar tiempo en la interacción con el documento y desarrollo de animaciones. Con esta librería el programador puede manejar el contenido y los eventos de la página de una manera más simple, es de mucha ayuda ya que contiene componentes visuales como el calendario y cuadros de dialogo. (Vara, M. J. M., , S. M. López , & D. Granada , 2014)

3.1.10 Bootstrap

Es una herramienta que ofrece al programador una serie de elementos gráficos de serie, como ejemplo de estos tenemos: botones, tablas, menús etc. Es un framework con el cual se pueden crear interfaces web full responsive para el usuario, utiliza librerías CSS, JavaScript plugins, etc. Además de esto Bootstrap ha llegado a ser compatible con la mayoría de los navegadores web conocidos y en la actualidad es la herramienta más popular para front-end. (Bootstrap., 2016)

3.1.11 Primefaces

Es una herramienta de código abierto compuesto de diversos elementos gráficos, utilizada en el desarrollo con JSF que facilita la creación y diseño de interfaces web. Permite la integración con otros componentes como RichFaces, JBoss y sirve para desarrollar aplicaciones web para teléfonos móviles Android, IOs. Una de las características fundamentales es que tiene múltiples temas para la interfaz de usuario y no necesita de configuraciones extras ni dependencias. (Rios, Sergio , 2015)

3.1.12 Hibernate

Es un software libre que permite el mapeo de una base de datos, dicho software ofrece el lenguaje HQL (Hibernate Query Lenguaje), el cual sirve para la consulta de datos a una base de datos ya existente. Hibernate contiene dos clases que son muy importantes dentro del desarrollo de una aplicación: las clases java que sirven para representar los objetos que contienen información almacenada en un sistema relacional y el fichero de mapeo (hbm.xml) en el que se encuentra el mapeo de la clase y los campos de una base de datos relacional. (Córcoles, & Montero , 2014)

3.1.13 GitHub

GitHub es una herramienta de código abierto y gratuito, utilizada dentro del desarrollo de este proyecto porque ofrece un control de una manera distribuida y organizada del código, además de ello permite manejar flujos de trabajo.

Una de sus principales características es que posee un sistema de ramificación y fusión, con lo que se pueden tener múltiples sucursales un de un mismo proyecto y ser totalmente independientes unas de las otras, facilitando así la integración del trabajo.

A continuación, se puede observar el repositorio con el que se trabajó todo el desarrollo del proyecto SPPP. (GitHub, 2008)

Repositorio proyecto SPPP

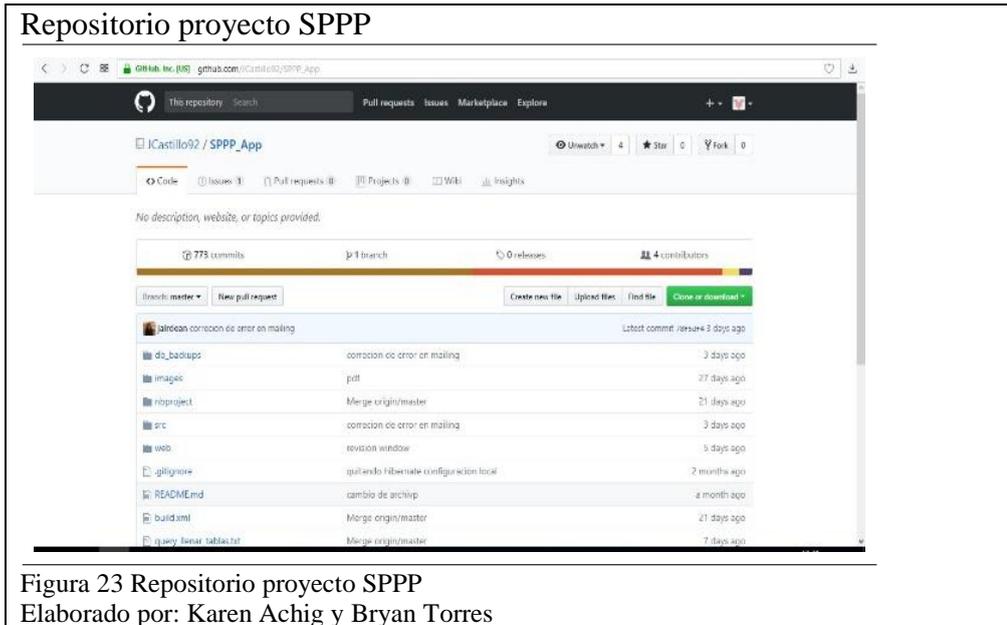


Figura 23 Repositorio proyecto SPPP
Elaborado por: Karen Achig y Bryan Torres

3.2 Base de Datos

A continuación, se muestra el diseño relacional de la base de datos utilizada en el desarrollo del sistema.

Base de datos relacional

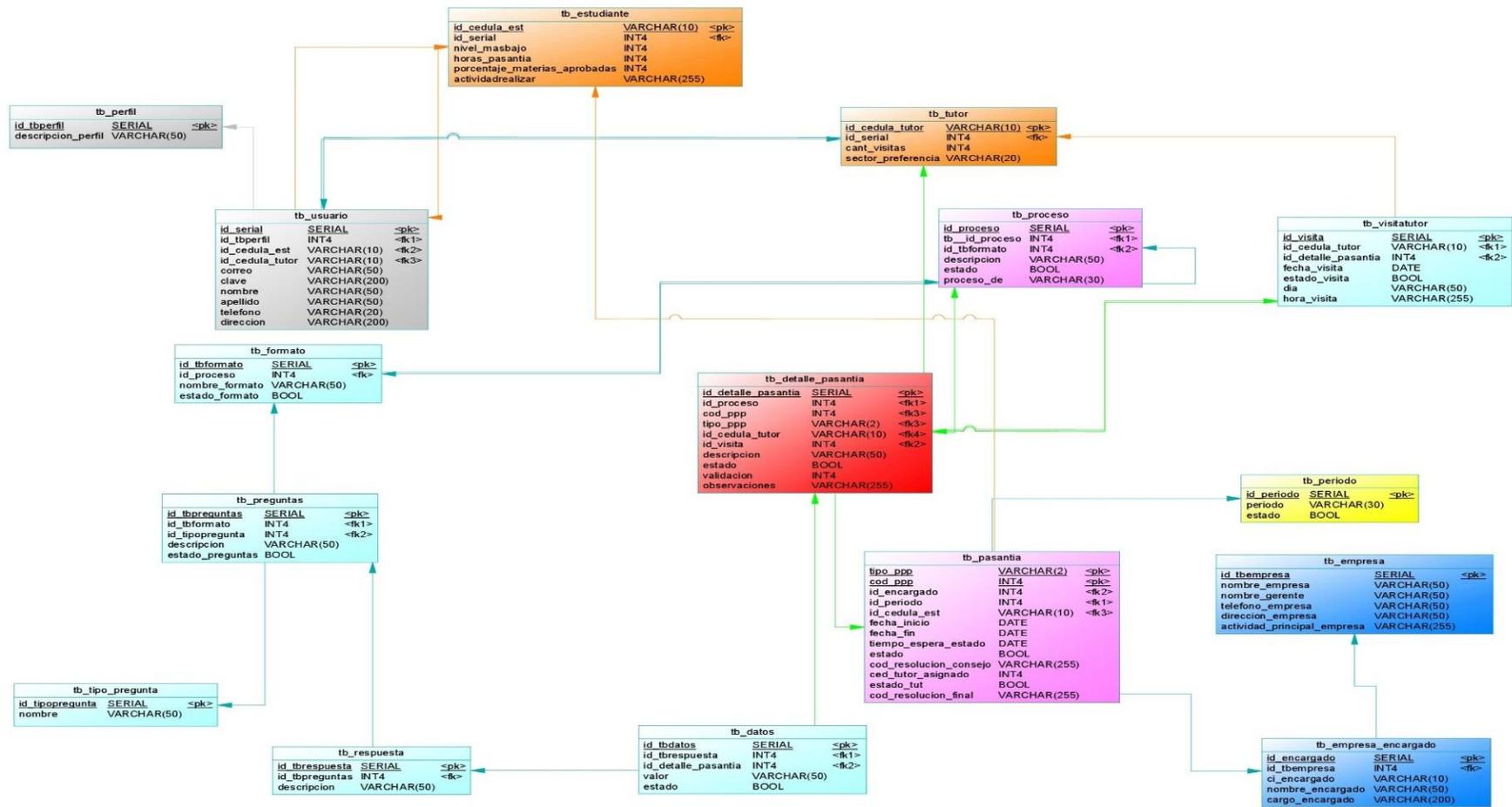


Figura 24 Diseño Relacional de la Base de Datos. (Flores, Castillo y Tufiño, 2018)
Elaborado por: Karen Achig y Bryan Torres

3.3 Diccionario de Datos

En la siguiente sección se describe una reseña del uso y de la utilidad de las diferentes tablas de la base de datos usada para el desarrollo del sistema.

Tabla 24 Descripción de la tabla Perfil

Descripción de tabla: Contiene la información sobre los perfiles que maneja el sistema.			
Campo	Tipo	Descripción	Requerido
id-perfil	Serial	Campo automático incremental (primary key)	Si
descripción-perfil	varchar(50)	Campo para describir el perfil de cada actor del sistema	No

Nota: Esta tabla contiene información acerca de la tabla Perfil.

Tabla 25 Descripción de la tabla Usuario

Descripción de tabla: Contiene la información de cada usuario.			
Campo	Tipo	Descripción	Requerido
id-serial	Serial	Campo automático incremental (primary key)	Si
id-cedula-tutor	varchar(10)	Campo (foreign key) de la tabla tutor	Si
Correo	varchar(50)	Campo para almacenar el correo del usuario	No
Clave	varchar(20)	Campo para clave de acceso al sistema	No
nombre	varchar(50)	Campo para almacenar el nombre del usuario	No
apellido	varchar(50)	Campo para almacenar el apellido del usuario	No
telefono	varchar(20)	almacenar el teléfono del usuario	No
dirección	varchar(20)	almacenar la dirección del usuario	No

Nota: Esta tabla contiene información acerca de la tabla Usuario.

Tabla 26 Descripción de la tabla Estudiante

Descripción de tabla: Contiene la información específica de cada estudiante.			
Campo	Tipo	Descripción	Requerido
id- cedula- est	varchar(10)	Campo automatico incremental (primary key)	Si
nivel- masbajo	int	Campo para especificar el nivel en que se encuentra el estudiante	Si
horas- pasantia	int	Campo para almacenar las horas de la pasantía del estudiante	No
porcenta je- materias aprobada	int	Campo para almacenar el porcentaje de las materias que lleva aprobadas el estudiante.	Si
actividad realizar	varchar(255)	Campo en el que se guarda la actividad que realiza el estudiante en la empresa	No

Nota: Esta tabla contiene información acerca de la tabla Estudiante.

Tabla 27 Descripción de la tabla Tutor

Descripción de tabla: Contiene la información específica del tutor.			
Campo	Tipo	Descripción	Requerido
id- cedula- tutor	varchar(10)	Campo automático incremental (primary key)	Si
cant- visitas	int	Campo para especificar la cantidad de visitas que posee el tutor	Si
sector- preferen cia	varchar(20)	Campo para almacenar el sector preferencial del tutor	Si

Nota: Esta tabla contiene información acerca de la tabla Tutor.

Tabla 28 Descripción de la tabla VisitaTutor

Descripción de tabla: Contiene la información acerca de la visita a realizarse.			
Campo	Tipo	Descripción	Requerido
id- visita	serial	Campo automático incremental (primary key)	Si
id- cedula- tutor	varchar(10)	Campo (foreign key) de la tabla tutor	No
fecha- visita	Date	Campo para guardar la fecha de la visita	No
estado- visita	varchar(255)	Campo para ver en qué estado se encuentra la visita	No
hora- visita	varchar(255)	Campo para guardar la hora de la visita	No

Nota: Esta tabla contiene información acerca de la tabla VisitaTutor.

Tabla 29 Descripción de la tabla Proceso

Descripción de tabla: Contiene la información de cada proceso que se lleva a cabo dentro del sistema.			
Campo	Tipo	Descripción	Requerido
id-proceso	serial	Campo automático incremental (primary key)	Si
id-tb-proceso	int	Campo de autorelación (subprocesos)	Si
id-tbformato	Date	Campo (foreign key) de la tabla formato	No
descripcion	varchar(50)	Campo para ver la descripción de un proceso	No
Estado	boolean	Campo booleano para saber el estado en que se encuentra el proceso	Si

Nota: Esta tabla contiene información acerca de la tabla Proceso.

Tabla 30 Descripción de la tabla DetallePasantia

Descripción de tabla: Contiene la información detallada de cada proceso que se lleva a cabo dentro de la pasantía.			
Campo	Tipo	Descripción	Requerido
id-detalle-pasantia	serial	Campo automático incremental (primary key)	Si
id-proceso	int	Campo (foreign key) de la tabla proceso	Si
cod-ppp	int	Campo (foreign key) de la tabla pasantía	Si
tipo-ppp	Varchar(2)	Campo (foreign key) de la tabla pasantía	Si
id-cedula-tutor	Varchar(10)	Campo (foreign key) de la tabla tutor	No
id-visita	Int	Campo (foreign key) de la tabla visitatutor	No
descripcion	varchar(50)	Campo para guardar la descripción del proceso	No
Estado	Boolean	Campo booleano para guardar el estado del proceso	No
Validación	Int	Campo para evidenciar el proceso en el que se encuentra el estudiante	No
observaciones	varchar(255)	Campo para observaciones de la pasantía	No

Nota: Esta tabla contiene información acerca de la tabla DetallePasantia.

Tabla 31 Descripción de la tabla Pasantía

Descripción de tabla: Contiene la información específica del proceso de pasantías.			
Campo	Tipo	Descripción	Requerido
tipo-ppp	varchar(2)	Campo (primary key)	Si
cod-ppp	int	Campo (primary key)	Si
id-encargado	int	Campo (foreign key) de la tabla empresa-encargado	No
id-periodo	int	Campo (foreign key) de la tabla periodo	Si
id-cedula-est	varchar(10)	Campo (foreign key) de la tabla estudiante	Si
fecha-inicio	int	Campo para guardar la fecha de inicio de la pasantía	No
fecha-fin	Date	Campo para guardar la fecha de fin de la pasantía	No
tiempo-espera-estado	integer	Campo para guardar el tiempo de espera entre tramites	No
Estado	boolean	Campo para saber el estado en que se encuentra la pasantía	No
cod-resolucion-consejo	varchar(255)	Campo para guardar la resolución de inicio de pasantía	No
ced-tutor-asignado	int	Campo para guardar la cédula del tutor asignado a la visita	No
estado-tut	boolean	Campo para guardar el estado del tutor asignado	No
cod-resolucion-final	varchar(255)	Campo para guardar la resolución final	No

Nota: Esta tabla contiene información acerca de la tabla Pasantia.

Tabla 32 Descripción de la tabla Formato

Descripción de tabla: Contiene la información sobre los formatos manejados en el sistema.			
Campo	Tipo	Descripción	Requerido
id-tbformato	serial	Campo automático incremental (primary key)	Si
id-proceso	int	Campo (foreign key) de la tabla proceso	No
nombre-formato	varchar(50)	Campo para guardar el formato	No
estado-formato	boolean	Campo para ver en qué estado se encuentra el formato	No

Nota: Esta tabla contiene información acerca de la tabla Formato.

Tabla 33 Descripción de la tabla Preguntas

Descripción de tabla: Contiene la información de las preguntas que tiene cada formato.			
Campo	Tipo	Descripción	Requerido
id-tbpreguntas	serial	Campo automático incremental (primary key)	Si
id-teformato	int	Campo (foreign key) de la tabla formato	Si
id-tipopregunta	int	Campo (foreign key) de la tabla tipopregunta	Si
descripcion	varchar(50)	Campo para guardar la pregunta	No
estado-preguntas	boolean	Campo para guardar el estado de cada pregunta	No

Nota: Esta tabla contiene información acerca de la tabla Preguntas.

Tabla 34 Descripción de la tabla TipoPregunta

Descripción de tabla: Contiene la información sobre el tipo de pregunta que contiene cada formato.			
Campo	Tipo	Descripción	Requerido
id-tipopregunta	serial	Campo automático incremental (primary key)	Si
nombre	varchar(50)	Campo para guardar el tipo de pregunta	No

Nota: Esta tabla contiene información acerca de la tabla TipoPregunta.

Tabla 35 Descripción de la tabla Respuesta

Descripción de tabla: Contiene la información sobre la respuesta que tiene cada pregunta.			
Campo	Tipo	Descripción	Requerido
id-tbrespuesta	Serial	Campo automático incremental (primary key)	Si
id-tbpreguntas	Int	Campo (foreign key) de la tabla preguntas	Si
descripcion	varchar(50)	Campo para guardar la respuesta	No

Nota: Esta tabla contiene información acerca de la tabla Respuesta.

Tabla 36 Descripción de la tabla Datos

Descripción de tabla: Contiene los valores obtenidos de las respuestas, así como los detalles de la pasantía.			
Campo	Tipo	Descripción	Requerido
id-tbdatos	Serial	Campo automático incremental (primary key)	Si
id-tbrespuestas	Int	Campo (foreign key) de la tabla respuestas	Si

id-detalle-pasantia	Int	Campo (foreign key) de la tabla detallepasantia	Si
Valor	varchar(50)	Campo para guardar las respuestas de cada formato	No
Estado	Boolean	Campo para saber el estado del dato guardado	No

Nota: Esta tabla contiene información acerca de la tabla Datos.

3.4 Historias de usuario

En esta sección se describen las historias de usuario más importantes, debido a que el desarrollo del sistema se lo realizó con la metodología xp.

3.4.1 Módulo de visita

Tabla 37 Historia de usuario (Módulo de visita)

Historia de usuario	
Número de historia:	1
Nombre historia:	Módulo de visita
Tipo de tarea:	Desarrollo
Iteración asignada:	3
Programador responsable:	Karen Achig y Bryan Torres
<p>Descripción: Se solicita la creación de un módulo que permita al tutor agendar una cita previa de sus estudiantes asignados para el proceso de visita, que exista una confirmación por parte del estudiante y posteriormente que el tutor pueda completar formularios acerca de la visita</p>	
<p>Observaciones: Se debe tener en cuenta que en este proceso se interactuá entre el tutor de visita y el estudiante que está dentro de la pasantía. Confirmado con el cliente.</p>	

Nota: Esta tabla contiene información acerca de la historia de usuario para el módulo de visita.

Desarrollo

Para el desarrollo de la historia de usuario número 1 se especifica las funciones y objetivos del mismo, para tener en claro su correcta construcción.

Tabla 38 Descripción historia de usuario 1 (Interacción 1)

Historia de usuario: 1	Módulo visita
Funciones:	Creación de cita tentativa de visita por tutor Confirmación de cita por estudiante Cancelar cita por estudiante Completar formatos de visita por tutor
Objetivos:	Diseñar la pantalla para creación de la fecha tentativa Diseñar la pantalla para la confirmación de cita. Diseñar la pantalla para la cancelación de cita. Habilitar formatos para ingreso de información de la visita
Razón	Se realiza para que el tutor y estudiante puedan establecer una fecha para la realización de la visita Se realiza la cancelación de la visita por si el estudiante no puede en la fecha establecida por el tutor. Se realiza para que el tutor pueda evaluar los resultados obtenidos por el estudiante durante la actividad

Nota: Esta tabla contiene información acerca primera interacción de la historia de usuario.

Primera iteración

Para el diseño de la pantalla de creación de fecha tentativa se realizó un menú al que puede acceder el tutor aquí encuentra la opción de creación de citas tentativas y a su vez los formularios de visita.

En esta página puede escoger un estudiante, fecha y una hora, al momento de ser enviada el estudiante recibe la fecha para su posterior aceptación.

Pantalla de agendación de cita tutor (Primera iteración)

SPPP - Sistemas

Tutor

Avances

Citas

Agendar cita

Citas agendadas

Formularios

Subir Documentos

Comentarios

Ayuda

Contactos

Agendar cita

Estudiante asignado: Karen Achig

Karen Achig

Bryan Torres

Andrea

Fecha:

Hora:

Enviar

Figura 25 Pantalla de agendación de cita tutor (Primera iteración)
Elaborado por: Karen Achig y Bryan Torres

Para la habilitación de formato para el ingreso de datos con respecto a la visita se diseñó un formulario en el cual el tutor puede calificar a cada estudiante que se le haya asignado.

Pantalla informe tutor (Primera interacción)

Informe tutor

Datos:

Carta compromiso: Disabled input

Tipo de actividad: Disabled input

Tutor: Disabled input

Cedula: Disabled input

Estudiante: Disabled input

Muy satisfactorio 5

Satisfactorio 4

Aceptable 3

Deficiente 2

Malo 1

Cuestionario	Escala				
	1	2	3	4	5
.....

Figura 26 Pantalla informe tutor (Primera interacción)
Elaborado por: Karen Achig y Bryan Torres

Para el diseño de la pantalla de confirmación y cancelación de cita se realizó un menú al que puede acceder el estudiante y visualizar dichas opciones. En esta pantalla el estudiante confirma la cita o la cancela y esto es comunicado a su tutor.



Segunda iteración

El módulo de visita para la revisión se encontraba funcional, en la reunión con el usuario se establecieron modificaciones que se verían involucradas tanto en el diseño como en el comportamiento funcional y para ello se realiza un cuadro comparativo entre la primera iteración y la segunda iteración.

Tabla 39 Descripción historia de usuario 1(Interacción 2)

Historia de usuario: 1	Módulo visita	Estado
Funciones:	Creación de cita tentativa de visita por tutor Confirmación de cita por estudiante Cancelar cita por estudiante Completar formatos de visita por tutor Crear tabla informativa con datos de la visita Cancelar cita por tutor Dashboard tutor Dashboard estudiante	Cumplido "Modificaciones" Cumplido "Modificaciones" Cumplido "Modificaciones" Cumplido "Modificaciones" Requerimiento nuevo Requerimiento nuevo Requerimiento nuevo Requerimiento nuevo

Objetivos:	Diseñar la pantalla para creación de la fecha tentativa Diseñar la pantalla para la confirmación de cita. Diseñar la pantalla para la cancelación de cita. Habilitar formatos para ingreso de información de la visita Diseñar una tabla donde se visualice los datos de la visita agendada Diseñar una pantalla que le permita al tutor cancelar la cita que agendo previamente Diseñar una pantalla que le permita al tutor saber cuáles son sus actividades pendientes Diseñar una pantalla que le permita al estudiante saber cuáles son sus actividades pendientes	Cumplido "Cambios de diseño" Cumplido "Cambios de diseño" Cumplido "Cambios de diseño" Cumplido "Cambios de diseño" Crear una tabla para información de visita Crear una pantalla para cancelación de visita por parte del tutor Crear una pantalla para que el tutor conozca en la actividad que se encuentra Crear una pantalla para que el estudiante conozca en la actividad que se encuentra
------------	--	--

Nota: Esta tabla contiene información acerca de la interacción 2 de la historia de usuario1.

Tomando en cuenta los cambios propuestos por el usuario se realizó diferentes modificaciones con respecto al diseño, pero manteniendo la funcionalidad. Entre las principales correcciones están:

En la página de agendación de cita se cambió los estilos de inputs y labels para que sea más agradable para el tutor.

En la página de confirmación de cita se reubicó la tabla y se colocó de manera vertical para que el estudiante pueda visualizar todos los datos de mejor manera.

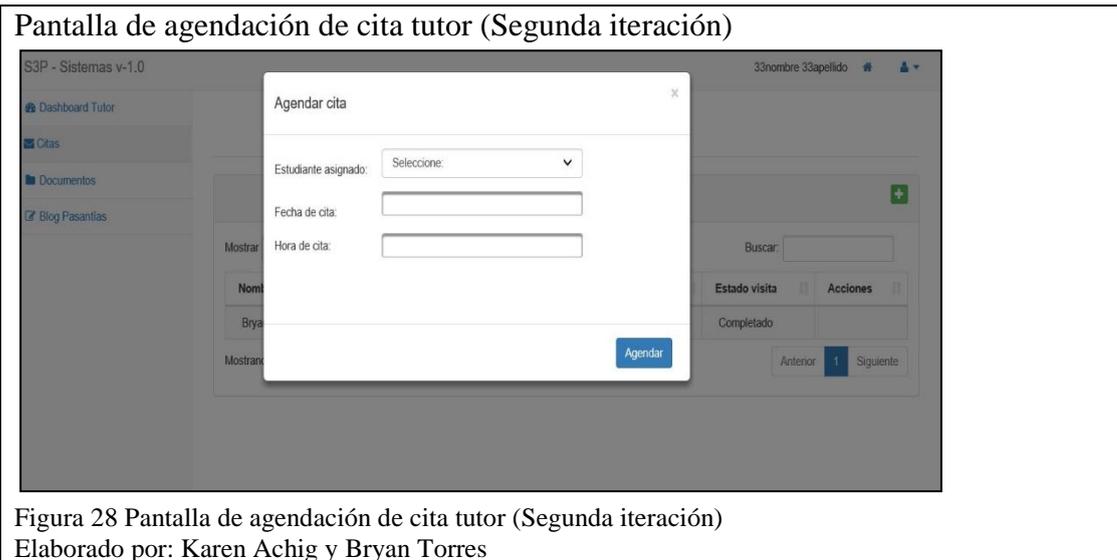
En la página de formato del tutor se agregó la validación para que el mismo pueda escoger una sola opción entre la escala de calificación y que los radio buttons estén inicializados en 5 dentro de la escala. Se diseño una tabla en donde el tutor podrá visualizar los datos de las visitas y sus diferentes estados.

Se diseño una pantalla en donde el tutor podrá cancelar la cita y el estudiante será notificado de la cancelación.

Se diseño una pantalla con el dashboard de tutor indicando que procesos tiene

pendientes y también se incluyeron estadísticas acerca de los procesos en los que se encuentran los estudiantes dentro del workflow.

Se diseñó una pantalla con el dashboard de estudiante indicando los procesos en los que se encuentran dentro del workflow y que debe ir llenando para poder validar su pasantía.



Pantalla informe tutor (Segunda iteración)

Informe tutor

Datos:

Carta Compromiso: PP1 Tipo de actividad: práctica pre profesional

Tutor: 33nombre

Estudiante: Bryan Andres Cedula estudiante: 123

Muy satisfactorio 5
Satisfactorio 4
Aceptable 3
Deficiente 2
Malo 1

Cuestionario	Escala				
	1	2	3	4	5
¿Se cumplió con las actividades propuestas en la Carta Compromiso?	●	●	●	●	●
¿Satisficieron los resultados a la labor institucional?	●	●	●	●	●
¿El estudiante tuvo la información necesaria del proceso de pasantías, prácticas pre profesionales ó extensiones?	●	●	●	●	●
La calidad de los productos entregados fueron:	●	●	●	●	●
El comportamiento del estudiante en la institución externa fue:	●	●	●	●	●

Figura 30 Pantalla informe tutor (Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Pantalla información de visitas y estado de las mismas. (Segunda iteración)

Citas agendadas

Mostrar: 10 registros Buscar:

Nombre estudiante	Dia visita	Fecha visita	Hora visita	Estado visita	Acciones
Bryan Andres Torres Lara	Viernes	2018-01-19	14:00	Completado	

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Figura 31 Pantalla información de visitas y estado de las mismas. (Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Pantalla de cancelación de cita tutor (Segunda iteración)

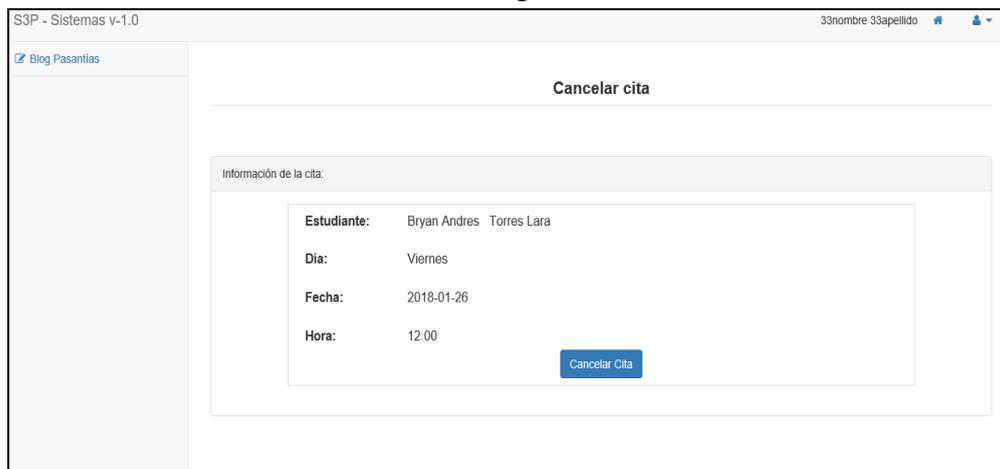


Figura 32 Pantalla de cancelación de cita tutor (Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Pantalla de dashboard tutor (Segunda iteración)

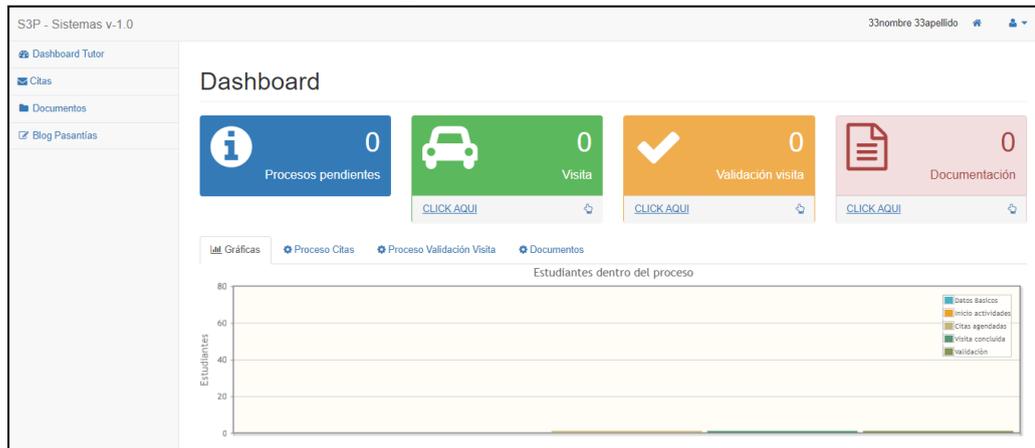


Figura 33 Pantalla de dashboard tutor (Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Pantalla de dashboard estudiante (Segunda iteración)

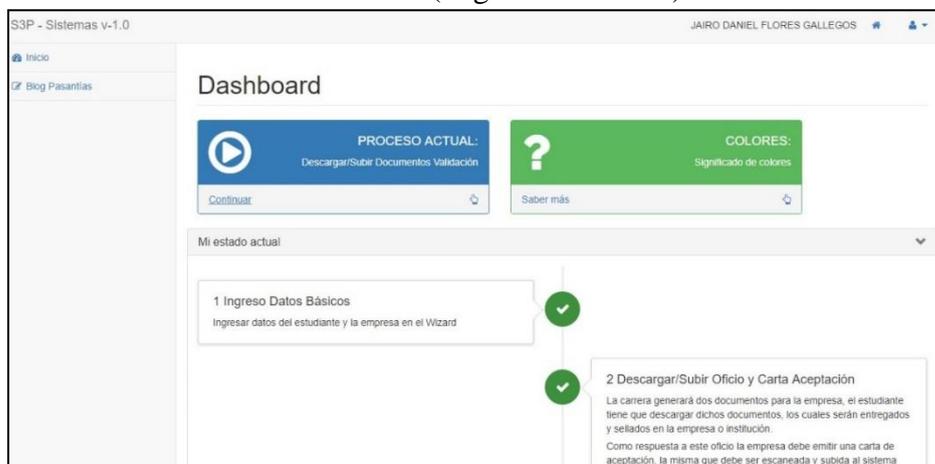


Figura 34 Pantalla de dashboard estudiante (Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Tercera iteración

El módulo de visita en esta iteración ha sido aceptado por el usuario, ya que se cumplieron todos los requisitos y todas las funcionalidades están operativas. En la siguiente tabla se puede apreciar el estado de las funciones del módulo:

Tabla 40 Descripción historia de usuario 1(Interacción 3

Historia de usuario: 1	Módulo visita	Estado
Funciones:	Creación de cita tentativa de visita por tutor Confirmación de cita por estudiante Cancelar cita por estudiante Completar formatos de visita por tutor Crear tabla informativa con datos de la visita Cancelar cita por tutor Dashborad tutor Dashborad estudiante	Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido
Objetivos:	Diseñar la pantalla para creación de la fecha tentativa Diseñar la pantalla para la confirmación de cita. Diseñar la pantalla para la cancelación de cita. Habilitar formatos para ingreso de información de la visita Diseñar una tabla donde se visualice los datos de la visita agendada Diseñar una pantalla que le permita al tutor cancelar a cita que agendo previamente Diseñar una pantalla que le permita al tutor saber cuales son sus actividades pendientes Diseñar una pantalla que le permita al estudiante saber cuáles son sus actividades pendientes	Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido

Nota: Esta tabla contiene información acerca de la interacción 3 de la historia de usuario1.

3.4.2 Módulo de validación final

Tabla 41 Historia de usuario(Módulo de validación final)

Historia de usuario	
Número de historia:	2
Nombre historia:	Módulo de validación final
Tipo de tarea:	Desarrollo
Iteración asignada:	3
Programador responsable:	Karen Achig y Bryan Torres
Descripción: Se solicita la creación de un módulo que permita al tutor subir documentos de validación de las visitas realizadas a cada estudiante, a su vez que le permita al estudiante pedir que se le valide la pasantía que ha realizado, también que permita que el coordinador valide los documentos que han sido llenados por el estudiante y que se determine la culminación del proceso de pasantía del mismo.	
Observaciones: Se debe tener en cuenta que en este proceso van a interactuar más de dos usuarios ya que es un flujo de trabajo en el cual se necesita una validación, en dicho proceso estarán involucrados los siguientes usuarios: estudiante, tutor, coordinador y secretaria. Confirmado con el cliente.	

Nota: Esta tabla contiene información acerca de la historia de usuario para el módulo de validación final.

Desarrollo

Para el desarrollo de la historia de usuario número 2 se especifica las funciones y objetivos del mismo, para tener en claro su correcta construcción.

Tabla 42 Descripción historia de usuario 2 (Interacción 1)

Historia de usuario: 2	Modulo validación final
Funciones:	Subir archivos validación tutor Llenar formularios autoevaluación estudiante Subir archivos para validación final estudiante Validación de archivos del estudiante Generar reporte de estudiantes que culminaron el proceso de pasantía. Registrar número de resolución Registrar la culminación de pasantía estudiante

Objetivos:	<p>Diseñar una pantalla para que el tutor pueda subir documentos de validación de la visita realizada a cada estudiante</p> <p>Diseñar un formulario para que el estudiante pueda realizar la autoevaluación de la actividad realizada.</p> <p>Diseñar una pantalla donde el estudiante pueda subir los documentos necesarios para validar su pasantía.</p> <p>Diseñar una pantalla en donde el coordinador pueda generar un pdf con los datos de los estudiantes que culminaron la pasantía.</p> <p>Diseñar una pantalla en donde el coordinador pueda registrar la resolución de cada estudiante.</p> <p>Diseñar una pantalla para que el estudiante pueda evidenciar la culminación de su proceso</p>
Razón	<p>Se realiza para que el tutor pueda acabar con el proceso de visita, validando con documentos la realización de la misma</p> <p>Se realiza para que el estudiante pueda evaluar su pasantía y complete los documentos para poder realizar la validación de la misma</p> <p>Se realiza porque es indispensable que el estudiante suba todos los documentos requeridos para que la actividad pueda ser validada.</p> <p>Se realiza para que el coordinador pueda corroborar que el estudiante ha subido correctamente la documentación requerida.</p> <p>Se realiza el reporte por parte del coordinador con la finalidad de dicho reporte sea recibido por la secretaria y pueda llevarlo a consejo para su posterior aprobación</p> <p>Se realiza para que el coordinador pueda registrar la resolución del estudiante que ha culminado sus pasantías.</p> <p>Se realiza para que el estudiante pueda ver como su proceso ha culminado y que es su flujo se refleje que la actividad culmino.</p>

Nota: Esta tabla contiene información acerca de la interacción 1 de la historia de usuario2.

Primera iteración

Para diseñar el módulo de validación final se realizó una pantalla en donde el tutor puede descargar el informe que fue realizado en el módulo anterior y subir la documentación completa del estudiante.

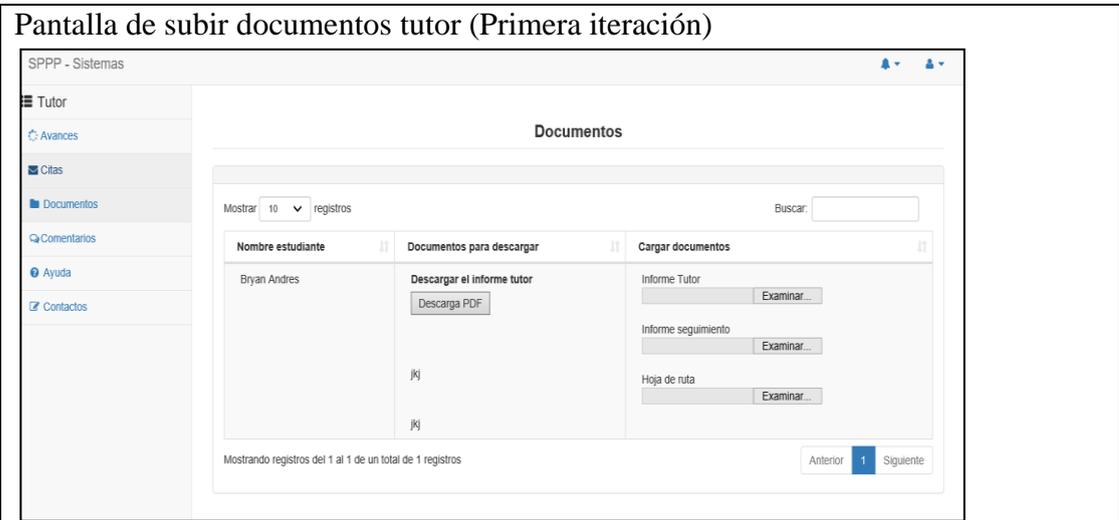


Figura 35 Pantalla de subir documentos tutor (Primera iteración)
Elaborado por: Karen Achig y Bryan Torres

Se desarrollo una pantalla en donde se ubica el formato de autoevaluación que el estudiante debe realizar para poder validar su pasantía.

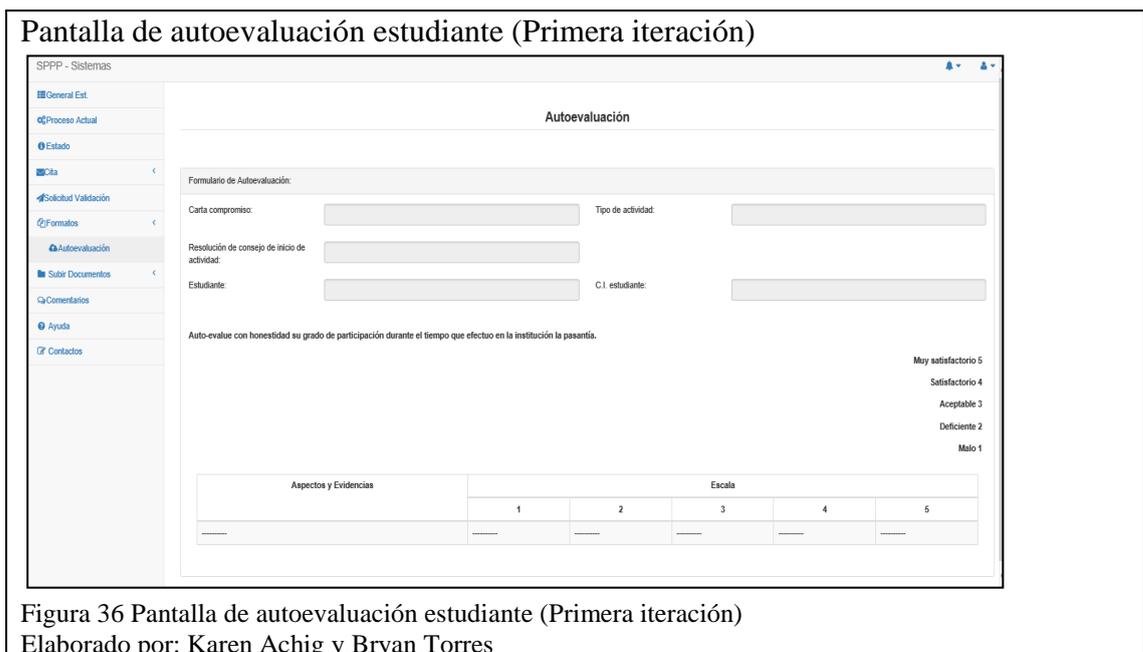
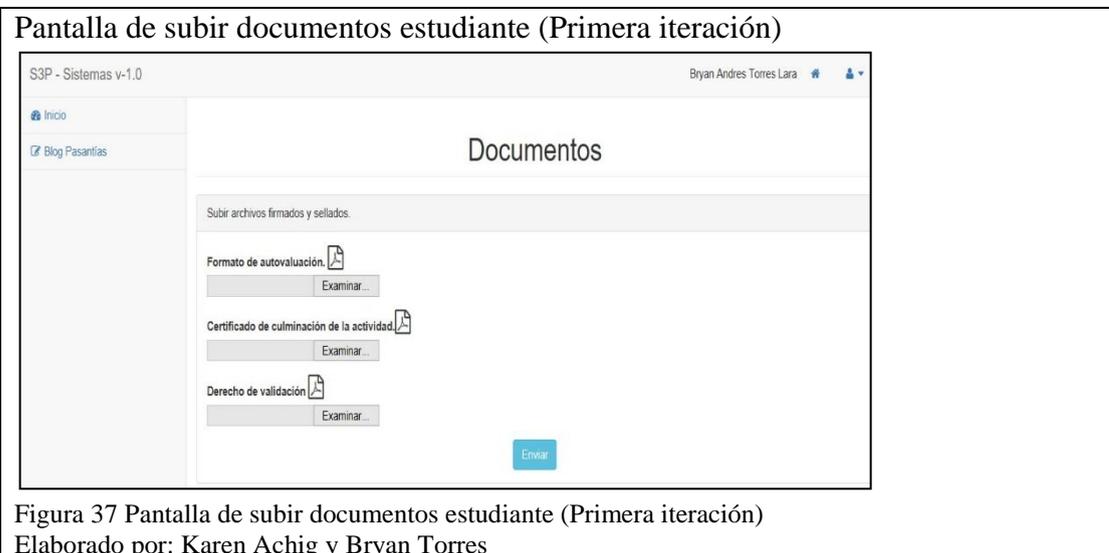
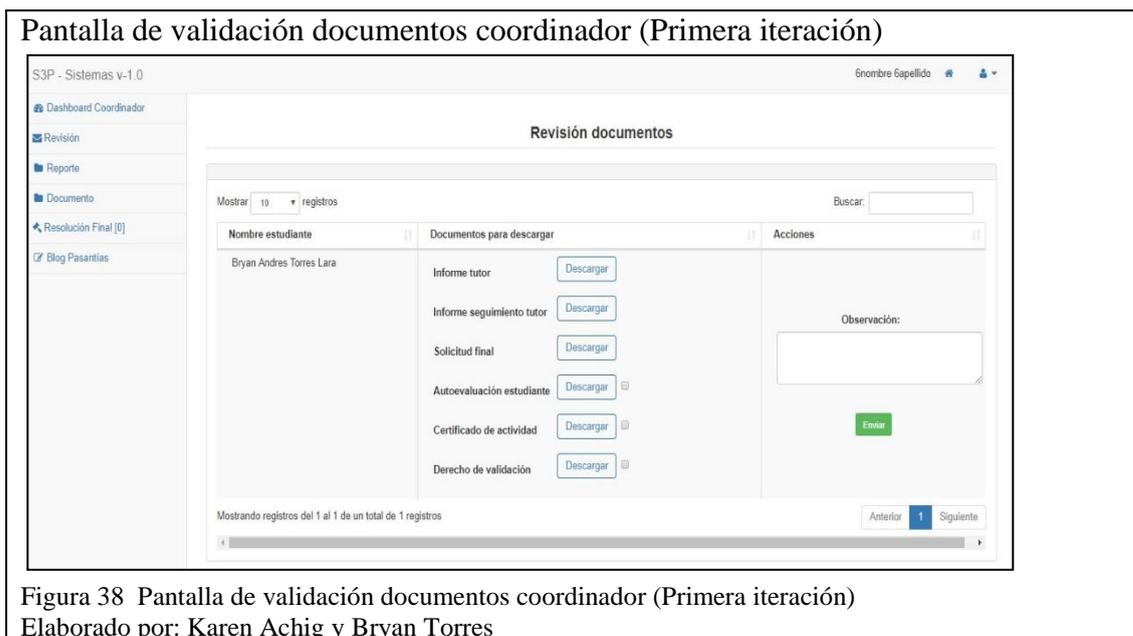


Figura 36 Pantalla de autoevaluación estudiante (Primera iteración)
Elaborado por: Karen Achig y Bryan Torres

Se diseñó una pantalla donde el estudiante puede subir los documentos requeridos para realizar la validación de pasantías.



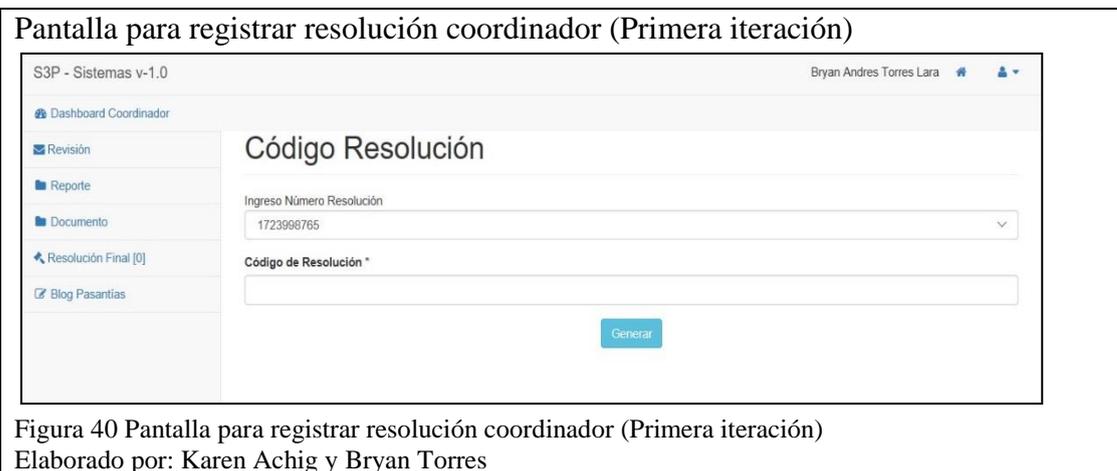
Se diseñó una pantalla para el coordinador donde puede visualizar los documentos de validación subidos por cada estudiante, en el caso de que estén correctos lo valida caso contrario enviara una observación y es estudiante debe volver a subir su documentación



Se desarrollo una pantalla para que el coordinador pueda generar un reporte en donde indique los estudiantes que culminaron con éxito su pasantía y el mismo se enviada a la secretaria para que el documento sea aprobado en consejo de carrera



Se diseño una pantalla en donde el coordinador pueda registrar el número de resolución de cada estudiante que ya haya concluido todo el proceso de validación y haya sido aprobado por el consejo de carrera.



Se desarrollo una pantalla que indique a el estudiante que ha concluido el proceso de pasantía, su estado cambiara cuando el coordinador haya registrado su resolución correspondiente.

Pantalla de proceso concluido (Primera iteración)



Figura 41 Pantalla de proceso concluido (Primera iteración)

Elaborado por: Karen Achig y Bryan Torres

Segunda iteración

El módulo de validación final para la revisión se encontraba funcional, en la reunión con el usuario se establecieron modificaciones que se verían involucradas tanto en el diseño como en comportamiento funcional y para ello se realiza un cuadro comparativo entre la primera iteración y la segunda iteración.

Tabla 43 Descripción historia de usuario 2 (Interacción 2)

Historia de usuario: 2	Modulo validación final	Estado
Funciones :	Subir archivos validación tutor Llenar formularios autoevaluación estudiante Subir archivos para validación final estudiante Validación de archivos del estudiante Generar reporte de estudiantes que culminaron el proceso de pasantía. Registrar número de resolución Registrar la culminación de pasantía estudiante Dashboard para coordinador y secretaria Lista de archivos generados por coordinador para uso de secretaria	Cumplido "Modificaciones" Cumplido "Modificaciones" Cumplido "Modificaciones" Cumplido Cumplido "Modificaciones" Cumplido "Modificaciones" Cumplido Requerimiento nuevo Requerimiento nuevo

Objetivos :	Diseñar una pantalla para que el tutor pueda subir documentos de validación de la visita realizada a cada estudiante	Cumplido” Cambios de diseño”
	Diseñar un formulario para que el estudiante pueda realizar la autoevaluación de la actividad realizada.	Cumplido ”Cambios de diseño”
	Diseñar una pantalla donde el estudiante pueda subir los documentos necesarios para validar su pasantía.	Cumplido ”Cambios de diseño”
	Diseñar una pantalla en donde el coordinador pueda generar un pdf con los datos de los estudiantes que culminaron la pasantía.	Cumplido
	Diseñar una pantalla en donde el coordinador pueda registrar la resolución de cada estudiante.	Cumplido ”Cambios de diseño”
	Diseñar una pantalla para que el estudiante pueda evidenciar la culminación de su proceso	Cumplido Creación de dashboard para coordinador y secretaria
	Diseñar una pantalla en donde el coordinador pueda registrar la resolución de cada estudiante. Diseñar una pantalla para que el estudiante pueda evidenciar la culminación de su proceso	Creación de página para listar reportes

Nota: Esta tabla contiene información acerca de la interacción 2 de la historia de usuario2.

Tomando en cuenta los cambios propuestos por el usuario se realizó diferentes modificaciones con respecto al diseño, pero manteniendo la funcionalidad. Entre las principales correcciones están:

En la página de subir archivos validación tutor se cambió la apariencia de botones y labels para ofrecer una interfaz amigable.

En la página de autoevaluación se modificó la escala de evaluación se lo realizo mediante radio buttons para evitar ingreso de datos erróneos.

En la página de subir archivos de validación de estudiante se agregó la opción de descarga de documento de autoevaluación para que el estudiante pueda acceder a dicho documento.

En la página de generar reporte de estudiantes que culminaron el proceso de pasantía se cambió los estilos en botones para tener una interfaz más agradable

En la página de registro de resolución se agregó un panel para mejorar la organización de los componentes

Se diseño una pantalla con el dashboard de coordinador indicando que procesos tiene pendientes y también se incluyeron estadísticas acerca de los procesos en los que se encuentran los estudiantes dentro del workflow.

Se diseño una pantalla con el dashboard de secretaria donde podrá redirigirse a documentos en donde encuentra los reportes enviados por el coordinador

Se diseño una pantalla donde se listara los documentos de la secretaria que son los reportes enviados por el coordinador.



Figura 42 Pantalla para subir documentos de validación tutor (Segunda iteración)
Elaborado por: Karen Achig y Bryan Torres

Pantalla para autoevaluación estudiante (Segunda iteración)

S3P - Sistemas v-1.0 Bryan Andres Torres Lara

Autoevaluación

Formulario de Autoevaluación:

Carta compromiso: Tipo de actividad:

Resolución de consejo de inicio de actividad: Cédula estudiante:

Estudiante:

Auto-evalúe con honestidad su grado de participación durante el tiempo que efectuó en la institución la pasantía.

Muy satisfactorio 5
 Satisfactorio 4
 Aceptable 3
 Deficiente 2
 Malo 1

Aspectos y Evidencias	Escala				
	1	2	3	4	5
Asistencia y puntualidad durante la extensión universitaria.	<input type="radio"/>				
Responsabilidad, disposición y cumplimiento en la ejecución de tareas.	<input type="radio"/>				
En las relaciones con el personal de la institución ha predominado la cortesía, el buen trato y la amabilidad.	<input type="radio"/>				

Figura 43 Pantalla para autoevaluación estudiante (Segunda iteración)
Elaborado por: Karen Achig y Bryan Torres

Pantalla para subir documentos de validación estudiante (Segunda iteración)

[Blog Pasantías](#)

Documentos

Descarga de Archivo

Formato autoevaluación

Subir archivos firmados y sellados.

Formato de autoevaluación en formato .pdf hasta 2MB

Certificado de culminación de la actividad. (La empresa deberá emitir el certificado especificando fechas y horas cumplidas) en formato .pdf hasta 2MB

Derecho de validación (Subir comprobante de pago realizado en tesorería) en formato .pdf hasta 2MB

Figura 44 Pantalla para subir documentos de validación estudiante (Segunda iteración)
Elaborado por: Karen Achig y Bryan Torres

Pantalla para generar reporte coordinador(Segunda iteración)

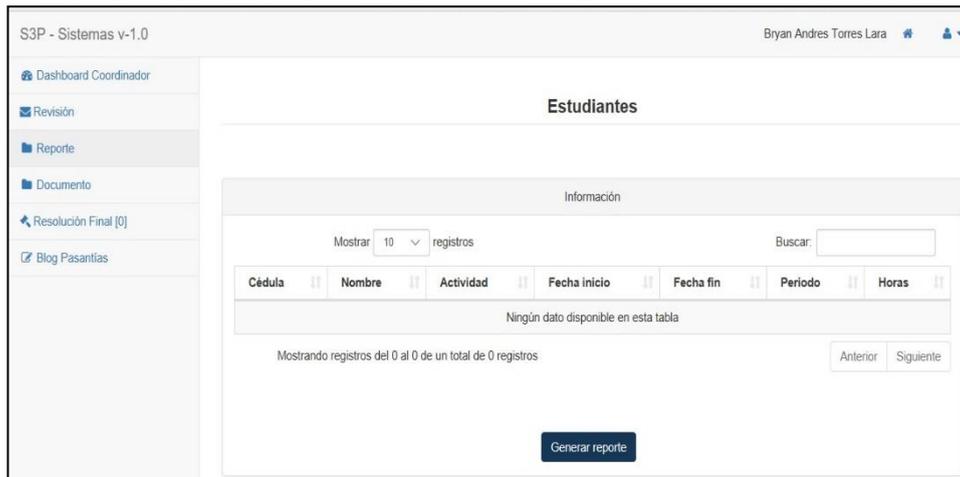


Figura 45 Pantalla para generar reporte coordinador(Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Pantalla para registro de resolución coordinador(Segunda iteración)



Figura 46 Pantalla para registro de resolución coordinador(Segunda iteración)

Elaborado por: Karen Achig y Bryan Torres

Pantalla para dashboard coordinador(Segunda iteración)

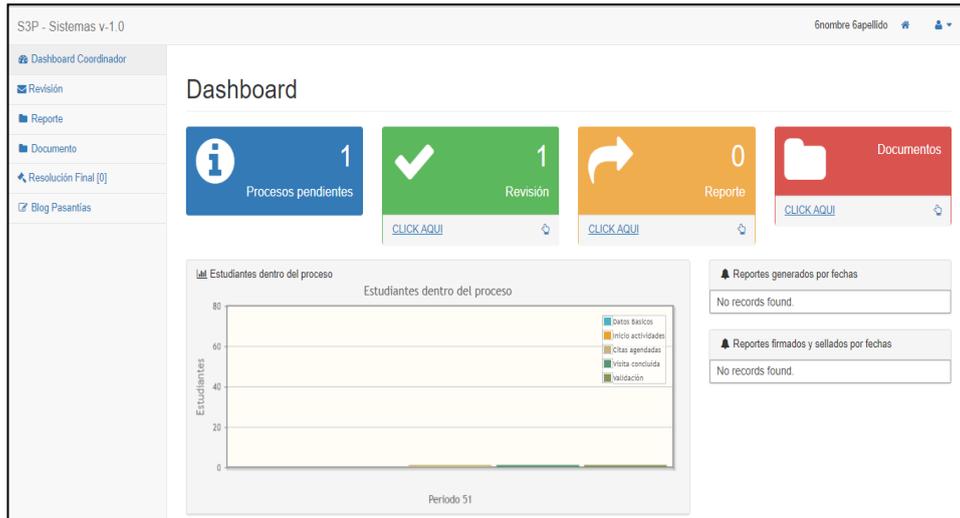


Figura 47 Pantalla para dashboard coordinador(Segunda iteración)
Elaborado por: Karen Achig y Bryan Torres

Pantalla para dashboard secretaria(Segunda iteración)



Figura 48 Pantalla para dashboard secretaria(Segunda iteración)
Elaborado por: Karen Achig y Bryan Torres

Pantalla para listar reportes secretaria(Segunda iteración)

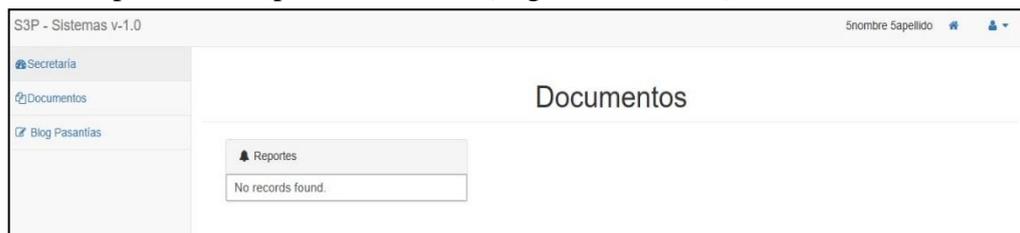


Figura 49 Pantalla para listar reportes secretaria(Segunda iteración)
Elaborado por: Karen Achig y Bryan Torres

Tercera iteración

El módulo de validación final en esta iteración ha sido aceptado por el usuario, ya que se cumplieron todos los requisitos y todas las funcionalidades están operativas. En la siguiente tabla se puede apreciar el estado de las funciones del módulo.

Tabla 44 Descripción historia de usuario 2 (Interacción 3)

Historia de usuario: 2	Módulo validación final	Estado
Funciones :	Subir archivos validación tutor Llenar formularios autoevaluación estudiante Subir archivos para validación final estudiante Validación de archivos del estudiante Generar reporte de estudiantes que culminaron el proceso de pasantía. Registrar número de resolución Registrar la culminación de pasantía estudiante Dashboard para coordinador y secretaria Lista de archivos generados por coordinador para uso de secretaria	Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido
Objetivos:	Diseñar una pantalla para que el tutor pueda subir documentos de validación de la visita realizada a cada estudiante Diseñar un formulario para que el estudiante pueda realizar la autoevaluación de la actividad realizada. Diseñar una pantalla donde el estudiante pueda subir los documentos necesarios para validar su pasantía. Diseñar una pantalla en donde el coordinador pueda generar un pdf con los datos de los estudiantes que culminaron la pasantía. Diseñar una pantalla en donde el coordinador pueda registrar la resolución de cada estudiante. Diseñar una pantalla para que el estudiante pueda evidenciar la culminación de su Proceso Diseñar una pantalla en donde el coordinador pueda registrar la resolución de cada estudiante. Diseñar una pantalla para que el estudiante pueda evidenciar la culminación de su proceso	Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido Cumplido

Nota: Esta tabla contiene información acerca de la interacción 3 de la historia de usuario2.

3.5 Código

En esta sección se describe el código fuente que se implementó en el sistema, detallando los métodos más importantes los cuales cumplen con el objetivo del seguimiento de la pasantía.

3.5.1 Código de configuración

XML: hibernate-cfg.xml

En este fichero xml se encuentra la configuración global de hibernate que se usa en el sistema, ya que es necesario configurarlo para su correcto funcionamiento. Trabaja con diferentes tags como: property y mapping, los cuales podemos ver el siguiente código:

Propiedades de configuración.

```
<hibernate-configuration>
<session-factory>
<property
name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
<property
name="hibernate.connection.driver_class">org.postgresql.Driver</property>
<property
name="hibernate.connection.url">jdbc:postgresql://localhost:5432/db_sp3
</property>
<property name="hibernate.connection.username">postgres</property>
<property name="hibernate.connection.password">1234</property>
<property name="hbm2ddl.auto">update</property>
<property name="show_sql">>false</property>
<property name="format_sql">>true</property>
<property
name="hibernate.current_session_context_class">thread</property>
<property name="hibernate.enable_lazy_load_no_trans">>true</property>
```

Aquí se encuentran las propiedades de configuración, todas con el tag property.

```
<mapping resource="com/sppp/beans/perfil.hbm.xml"/>
<mapping resource="com/sppp/beans/usuario.hbm.xml"/>
<mapping resource="com/sppp/beans/estudiante.hbm.xml"/>
<mapping resource="com/sppp/beans/proceso.hbm.xml"/>
<mapping resource="com/sppp/beans/detallePasantia.hbm.xml"/>
<mapping resource="com/sppp/beans/pasantia.hbm.xml"/>
```

```

<mapping resource="com/sppp/beans/encargado.hbm.xml"/>
<mapping resource="com/sppp/beans/empresa.hbm.xml"/>
<mapping resource="com/sppp/beans/datos.hbm.xml"/>
<mapping resource="com/sppp/beans/respuesta.hbm.xml"/>
<mapping resource="com/sppp/beans/preguntas.hbm.xml"/>
<mapping resource="com/sppp/beans/tipoPregunta.hbm.xml"/>
<mapping resource="com/sppp/beans/formato.hbm.xml"/>
<mapping resource="com/sppp/beans/periodoTutor.hbm.xml"/>
<mapping resource="com/sppp/beans/horarioTutor.hbm.xml"/>
<mapping resource="com/sppp/beans/tutor.hbm.xml"/>
<mapping resource="com/sppp/beans/visitaTutor.hbm.xml"/>
<mapping resource="com/sppp/beans/periodo.hbm.xml"/>
</session-factory>
</hibernate-configuration>

```

Figura 50 Propiedades de configuración.
Elaborado por: Karen Achig y Bryan Torres

3.5.2 Proceso de Visita

En esta parte se analizan los métodos más importantes que se usan en el sistema al momento que se agenda una cita para la visita a la empresa en la que el estudiante realiza su pasantía.

XML principal para la Visita: visitaTutor-hbm-xml.

Para cada clase que se desee persistir se crea un fichero xml, este fichero xml contiene el mapeado de la tabla tb_visitatutor que se encuentra en la base de datos con la que se trabaja en el sistema, además de la creación de cada campo con su tipo y las respectivas relaciones con las diferentes tablas de la base.

Clase para persistir

```

<hibernate-mapping package="com.sppp.beans">
<class name="VisitaTutor" table="tb-visitatutor">
<!-- creacion de campos-->
<id name="id_visita" column="id_visita" type="long">
<generator class="identity" ></generator>
</id>
<property name="dia" type="string"></property>
<property name="fecha_visita" type="date"></property>
<property name="hora_visita" type="string"></property>
<property name="estado_visita" type="string"></property>
<!-- Comienzo de relaciones-->

```

```

<!--RECIBO LAS CLAVES FORANEAS -->
<many-to-one name="tutor" class="Tutor">
<column name="cedula_tut" not-null="false"></column>
</many-to-one>
<!-- RELACION 1 A 1 constrained="true" Aqui se le setea como primaria-->
<many-to-one name="estudiante" class="Estudiante" unique="true"
column="cedula_est" not-null="false" >
</many-to-one>
<!-- Fin de relaciones-->
</class>
</hibernate-mapping>

```

Figura 51 Clase para persistir
Elaborado por: Karen Achig y Bryan Torres

Clase: CitasDaoImp

Esta clase contiene métodos con diferentes consultas a la base de datos a continuación, se listan los más importantes:

Método: Find All

El objetivo de este método es obtener un listado de todos los campos de la tabla VisitaTutor para listarlos en diferentes interfaces que componen la visita.

```

Case CitasDaoImp, metodo Find All

public List<VisitaTutor> findAll(String id) {
SessionFactory sf = HibernateUtil.getSessionFactory(); Session sesion =
sf.openSession();
Transaction tx = null;
//creacion de lista de tipo Visitatutor para guardar la
//informacion de la consulta List<VisitaTutor> listado = null; Long
id1=Long.parseLong(id);
//consulta sql donde me retorne todos los campos de la tabla
//especificando el tutor.
String sql = "FROM VisitaTutor WHERE cedula_tut =:id"; try {
tx = sesion.beginTransaction();
listado = sesion.createQuery(sql).setParameter("id", id1).list(); tx.commit();
} catch (RuntimeException e) { tx.rollback();
throw e;
}finally{
sesion.flush(); sesion.close();
}
//resultado return listado;
}

```

Figura 52 Case CitasDaoImp, metodo Find All
Elaborado por: Karen Achig y Bryan Torres

Clase: VisitaDAO

Esta clase contiene métodos para actualizar y guardar campos de la visita en la tabla VisitaTutor a continuación, se mencionan los métodos más importantes:

Método: Guardar Datos de la Visita.

El objetivo de este método es guardar la información que ingresa el tutor en el momento que agenda la cita y el estudiante la confirma.

Método Datos de la Visita

```
public void guardarDatosVisita(VisitaTutor visita){ sessionFactory
sf=HibernateUtil.getSessionFactory(); Session sesion=sf.openSession();
Transaction tx=null;
try {
//Transaccion para guardar datos en la tabla VisitaTutor tx =
sesion.beginTransaction(); sesion.saveOrUpdate(visita);
tx.commit();
System.out.println("===== PASO LA INFO
=====");
} catch (Exception e) { e.printStackTrace();
Logger.getAnonymousLogger().log(Level.SEVERE, e.getLocalizedMessage());
System.out.println("===== NO PASO LA INFO
=====");
if (tx != null){ tx.rollback();
}
}
finally{
//para cerrar seesion sesion.close();
}}
```

Figura 53 Método Datos de la Visita
Elaborado por: Karen Achig y Bryan Torres

Clase: ControllerBean

Esta clase Bean contiene los getters y setters en los que se guardan los valores de las diferentes interfaces del tutor y el estudiante al momento de la visita.

Método: Guardar Datos

El objetivo de este método es obtener los datos que ingresa el tutor para guardarlos en la tabla VisitaTutor.

Método Guardar Datos

```
public void guardarDatos(String id,String id2,String dia){
//guardar datos en el momento que el tutor agenda la cita Set<VisitaTutor>
setVisita = new LinkedHashSet<>();
//Obtener cedula del tutor que agenda la cita Long id1=Long.parseLong(id);
tutor.setCedula(id1);
//Obtener la cedula del estudiante receptor de la cita Long
id3=Long.parseLong(id2);
Estudiante est = new Estudiante(); est.setCedula(id3);
VisitaTutor vt = new VisitaTutor();
vt.setId_visita(id_visita); vt.setDia(dia);// dia de la visita
vt.setFecha_visita(data);// fecha de la visita vt.setHora_visita(horaConFormato);
// hora de la visita vt.setEstado_visita("Enviada");//estado de la visita
/al momento de la asignacion vt.setTutor(tutor);//guardar cedula de tutor
vt.setEstudiante(est);//guardar cedula del estudiante
setVisita.add(vt); // agregar datos a la LinkedHashSet<>()
VisitaDAO vi= new VisitaDAO();
vi.guardarDatosVisita(vt);// utilizar metodo para guardar datos
System.out.println(vt.getId_visita());
try {
} catch (Exception e) {}}
```

Figura 54 Método Guardar Datos

Elaborado por: Karen Achig y Bryan Torres

Clase: DatosTutDao

Esta clase contiene métodos que sirven para guardar los datos en la tabla Datos de los formularios que el tutor llena después de la visita.

Método: Guardar Datos Informe Tutor

El objetivo de este método es obtener la información que el tutor ingresa en dicho informe y guardarla en la tabla Datos que se encuentra en la base de datos que usamos en el sistema. Para posteriormente los mismos datos presentarlos en un formato Pdf en el momento que el tutor se desee descargar el informe.

Método Guardar Datos dinámicos

```
public void guardarDatosInformeTut(String at1,String at2,String at3,String
at4,String at5,String at6,String at7,String at8,String at9,String observaciones,
DetallePasantia dp){
Respuesta idRespuesta = new Respuesta();
//GUARDO LA DATA EN TABLA DATOS
SessionFactory sf = HibernateUtil.getSessionFactory(); Session sesion =
sf.openSession();
Transaction tx = null;
try {
tx = sesion.beginTransaction();
//setear el id de la respuesta a la que va concatenada la informacin del formulario.
idRespuesta.setId_tbrespuesta(57);
//Guardr en tabla Datos la informacion y el id de la respuesta
antes mencionada
Datos d30 = new Datos(at1, dp, idRespuesta,true); sesion.saveOrUpdate(d30);
idRespuesta.setId_tbrespuesta(58);
Datos d31 = new Datos(at2, dp, idRespuesta,true); sesion.saveOrUpdate(d31);
idRespuesta.setId_tbrespuesta(59);
Datos d32 = new Datos(at3, dp, idRespuesta,true); sesion.saveOrUpdate(d32);
idRespuesta.setId_tbrespuesta(60);
Datos d33 = new Datos(at4, dp, idRespuesta,true); sesion.saveOrUpdate(d33);
tx.commit();
} catch (NumberFormatException e) { System.out.println("=====
ERROR ACA=====");
e.printStackTrace(); if (tx != null) {
tx.rollback();
}
} finally {
//para cerrar seesion sesion.close();}}
```

Figura 55 Método Guardar Datos dinámicos
Elaborado por: Karen Achig y Bryan Torres

Proceso de Validación

En esta parte se analizan los métodos más importantes usados en el sistema en el momento en que el coordinador valida la documentación subida por el usuario.

Clase: UploadFile

En esta clase contiene métodos que sirven para la creación y descarga de archivos pdf de los diferentes actores que forman parte del sistema.

Método: Download File Estudiante.

Este método tiene como objetivo descargar los formatos del estudiante con el uso de servlets, mandando a crear primero el archivo para posteriormente descargarlo.

Método Download File Estudiante

```
public void download_file_est(int opcion) {///aqui recibir nombre de archivo ejem
103.pdf
try {
AlmacenamientoPDF obj_crearpdf = new AlmacenamientoPDF();
HttpSession session = SessionUtils.getSession();
long id;
id = (long) session.getAttribute("id");
FacesContext facesContext = FacesContext.getCurrentInstance();
ExternalContext context = facesContext.getExternalContext();
HttpServletRequest request = (HttpServletRequest)
context.getRequest();
HttpServletResponse response = (HttpServletResponse) context.getResponse();
//mando a crear el archivo pdf, para que sea lo mas actual posible.
obj_crearpdf.pdf_autoevaluacion(id, opcion);
//mando a llamar al mismo archivo pdf en la aplicacion, para que se pueda
descargar
response.sendRedirect(request.getContextPath() +
"/faces/user/estudiantes/download/" + opcion + ".pdf");
//response.sendRedirect("index.jsf");
} catch (IOException e) { e.printStackTrace();
}
} //end of DOWNLOAD_FILE
```

Figura 56 Método Download File Estudiante
Elaborado por: Karen Achig y Bryan Torres

Clase: CitasDaoImp

Método: Listar Informe Coordinador

El objetivo de este método es obtener un listado de todos los estudiantes que ya estén en la etapa de validación para posteriormente verificar la documentación de forma individual.

```

Método Listar Informe Coordinador
@Override
public List<VisitaTutor> listarInformeCoor() { SessionFactory sf =
HibernateUtil.getSessionFactory();
Session sesion = sf.openSession(); Transaction tx = null;
//Lista de estudiantes que se ira llenando List<VisitaTutor> listado = null;
//Consulta los estudiantes que ya estan en el proceso de validación
String sql = "FROM VisitaTutor WHERE estado_visita =:visitado";
String estado="Validacion";
try {
tx = sesion.beginTransaction();
// Query query = sesion.createQuery(sql);
// listado=query.list();
listado = sesion.createQuery(sql).setParameter("visitado", estado).list();
//query.setInteger("id", id);
tx.commit();
} catch (RuntimeException e) { tx.rollback();
throw e;
}finally{
sesion.flush(); sesion.close();
}
return listado;
}

```

Figura 57 Método Listar Informe Coordinador
Elaborado por: Karen Achig y Bryan Torres

Clase: AlmacenamientoPDF

Esta clase contiene métodos como lo son: la creación de directorios y archivos de los diferentes actores que interactúan en el sistema.

Método: Create coordinador folder first time

El objetivo de este método es la creación de un directorio personal en el cual se guardan diferentes archivos del coordinador de la pasantía.

```

Método Create coordinador folder first time

public boolean create_coordinador_folder_first_time(long cedula){
//NOTA el path /home/SPPP_PDF/ ya debe estar creado, para que luego se
proceda a crear cada carpeta con la ci
//Path para guardar el directorio con cedula del coordinador
File dir = new File(local_path+cedula+"/"+ "Subidos");
exitoalguardar=false;
if(!dir.exists()){
System.out.println("intentando crear nuevo directorio por primera y unica
vez"+dir.getName());

```

```

try{
//creacion del directorio para el coordinador dir.mkdir();// attempt to create the
directory here exitoalguardar=true;
}
catch(SecurityException se){ exitoalguardar=false;
}
}else{
exitoalguardar=true;
}
//comprobacion local if(exitoalguardar){System.out.println("directory was created
successfully");
}else{System.out.println("failed trying to create the directory");}
return exitoalguardar;
}

```

Figura 58 Método Create coordinador folder first time
Elaborado por: Karen Achig y Bryan Torres

Clase: CitasAgendadas

Esta clase contiene métodos que se usan para obtener la información de las consultas relacionadas a la validación.

Método: Guardar Datos

El objetivo de este método es que se verifiquen si todos los archivos subidos por el estudiante no contienen errores y pueda seguir normalmente el proceso de validación de su pasantía.

Método Guardar Datos

```

public String guardarDatos(long est,long visita,String correo){
System.out.println("bbbb"+archivo1+archivo2+archivo3);
//Compruebo si cumple todos los checks
//Compara si los archivos del estudiante de manera correcta y sin errores
if(archivo1 && archivo2 && archivo3 ){ VisitaDAO vi = new VisitaDAO();
vi.resolucion(est);//Ingresar el proceso 'correcto' de validacion en
//la tabla DetallePasantia
//Envio de correo (Documentacion correcta) al estudiante.
MailingMain por = new MailingMain();
por.mensajes(2, correo, obs);
//Estado de validacion Completado ControllerBean con = new ControllerBean();
con.updateListoVisita(visita);
}else {
String msg=obs + "\n"+errores;
//Envio de correo (Documentacion erronea) al estudiante.
MailingMain por = new MailingMain();

```

```
por.mensajes(1, correo, msg);  
//ingresar el proceso actual de validacion en la tabla DetallePasantia  
VisitaDAO vi = new VisitaDAO(); vi.resolucion2(est);  
}  
return "review_window";//retornar a ventana emergente  
}
```

Figura 59 Método Guardar Datos pasantía
Elaborado por: Karen Achig y Bryan Torres

3.6 Pruebas

A continuación, se muestra el escenario de pruebas que sirven para demostrar el correcto funcionamiento del sistema, validando que los requerimientos estén totalmente funcionales. Para ello se ha inclinado por tres tipos de pruebas:

Pruebas de estrés.

Pruebas Unitarias.

Pruebas contra requerimientos.

3.6.1 Pruebas de Estrés

En el desarrollo de estas pruebas se plantean 2 escenarios, cada uno con diferentes parámetros en donde se simula el ingreso concurrente de usuarios al sistema, realizando peticiones al servidor con el fin de saturarlo.

Como herramienta para la realización de estas pruebas se usó Apache JMeter, aplicación diseñada especialmente para pruebas de estrés sobre cualquier software.

Escenario de simulación 1

Se ingresaron 100 usuarios para comprobar el requerimiento no funcional 1.5, en el cual, el sistema opera correctamente con hasta 100 usuarios con sesiones concurrentes.

Tenemos 2 procesos: request y post al login.

Resultados escenario de simulación 1

El reporte permite validar que en 100 muestras ejecutadas en la simulación, se presentó un 0.00 % de errores.

Estadísticas 1 con 100 users

Label	# Samples	Average	Min	Max	Std. Dev.
Initial Login Request	100	344	18	636	174,66
Auth	100	492	55	700	184,22
TOTAL	200	418	18	700	194,01

Figura 60 : Estadísticas 1 con 100 users
Elaborado por: Karen Achig y Bryan Torres

Estadísticas 2 con 100 users

Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
0,00%	61,3/sec	302,53	0,00	5049,6
0,00%	45,3/sec	1195,47	0,00	27005,0
0,00%	89,4/sec	1398,72	0,00	16027,3

Figura 61 : Estadísticas 2 con 100 users
Elaborado por: Karen Achig y Bryan Torres

Gráficas resultantes

La siguiente muestra ayuda a comprender mejor los resultados:

Estadísticas por solicitud

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connetd Time(ms)
1	01:43:20.006	Grupo de Hilos 1-1	Initial Login Request	29	✓	5050	0	29	0
2	01:43:20.016	Grupo de Hilos 1-2	Initial Login Request	18	✓	5048	0	18	0
3	01:43:20.026	Grupo de Hilos 1-3	Initial Login Request	26	✓	5052	0	26	0
4	01:43:20.036	Grupo de Hilos 1-4	Initial Login Request	25	✓	5048	0	25	0
5	01:43:20.046	Grupo de Hilos 1-5	Initial Login Request	36	✓	5050	0	36	0
6	01:43:20.038	Grupo de Hilos 1-1	Auth	55	✓	27001	0	55	0
7	01:43:20.057	Grupo de Hilos 1-6	Initial Login Request	35	✓	5048	0	35	0
8	01:43:20.038	Grupo de Hilos 1-2	Auth	71	✓	27011	0	71	0
9	01:43:20.054	Grupo de Hilos 1-3	Auth	61	✓	27001	0	60	0
10	01:43:20.077	Grupo de Hilos 1-8	Initial Login Request	58	✓	5052	0	58	0
11	01:43:20.067	Grupo de Hilos 1-7	Initial Login Request	88	✓	5046	0	86	0
12	01:43:20.064	Grupo de Hilos 1-4	Auth	107	✓	27001	0	106	0
13	01:43:20.087	Grupo de Hilos 1-9	Initial Login Request	83	✓	5050	0	83	0
14	01:43:20.084	Grupo de Hilos 1-5	Auth	94	✓	27011	0	93	0
15	01:43:20.087	Grupo de Hilos 1-10	Initial Login Request	85	✓	5050	0	85	0
16	01:43:20.117	Grupo de Hilos 1-12	Initial Login Request	81	✓	5050	0	81	0
17	01:43:20.107	Grupo de Hilos 1-11	Initial Login Request	96	✓	5052	0	96	0
18	01:43:20.129	Grupo de Hilos 1-13	Initial Login Request	94	✓	5046	0	94	0
19	01:43:20.094	Grupo de Hilos 1-6	Auth	135	✓	27001	0	135	0
20	01:43:20.138	Grupo de Hilos 1-14	Initial Login Request	91	✓	5052	0	91	0
21	01:43:20.148	Grupo de Hilos 1-15	Initial Login Request	91	✓	5050	0	91	0
22	01:43:20.137	Grupo de Hilos 1-8	Auth	120	✓	27006	0	120	0
23	01:43:20.158	Grupo de Hilos 1-16	Initial Login Request	99	✓	5048	0	99	0
24	01:43:20.189	Grupo de Hilos 1-17	Initial Login Request	100	✓	5050	0	99	0
25	01:43:20.179	Grupo de Hilos 1-18	Initial Login Request	103	✓	5052	0	103	0
26	01:43:20.172	Grupo de Hilos 1-9	Auth	121	✓	27006	0	120	0
27	01:43:20.189	Grupo de Hilos 1-19	Initial Login Request	140	✓	5050	0	140	0
28	01:43:20.189	Grupo de Hilos 1-20	Initial Login Request	153	✓	5048	0	153	0

Figura 62: Estadísticas por solicitud
Elaborado por: Karen Achig y Bryan Torres

Como se puede observar, la trama de color verde se mantiene constante por debajo de los 160 ms y sin problemas, visto esto se puede juzgar que el servidor solventa sin problemas el número de peticiones ingresadas.

Tiempo de respuesta

Resultados Tiempo de Respuesta

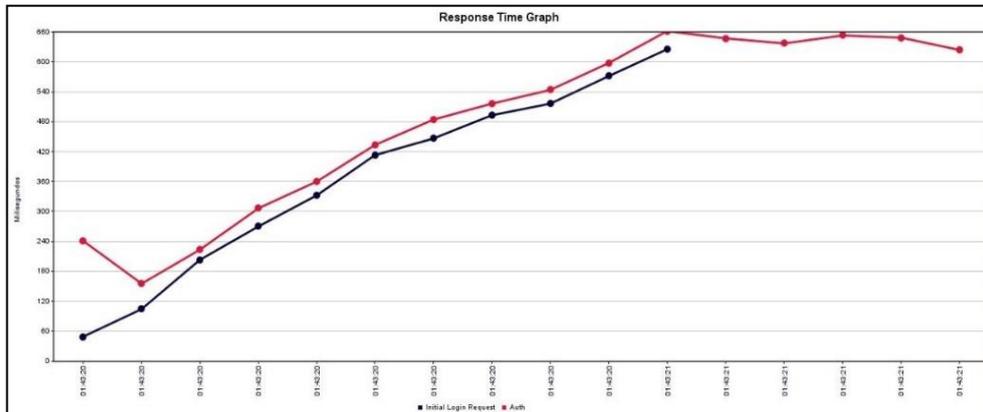


Figura 63: Resultados Tiempo de Respuesta
Elaborado por: Karen Achig y Bryan Torres

Escenario de simulación 2

Se dispone de 250 usuarios con sesiones concurrentes, los mismos que desean ingresar a la página principal del sistema.

Estadísticas 1 con 250 users

Label	# Samples	Average	Min	Max	Std. Dev.
Initial Login Request	250	653	24	1205	334,13
Auth	250	1059	71	1547	375,86
TOTAL	500	856	24	1547	409,53

Figura 64: Estadísticas 1 con 250 users
Elaborado por: Karen Achig y Bryan Torres

El siguiente reporte permite validar que, en 250 muestras ejecutadas en la simulación, se presentó un 0.00 % de errores teniendo un promedio de tiempo de respuesta (Average) de 653 ms para el login request y de 1059 ms para la autenticación, lo que quiere decir que no hay tiempos de espera tan largos que molesten al usuario.

Estadísticas 2 con 250 users

Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
0,00%	110,7/sec	545,73	0,00	5049,6
0,00%	66,7/sec	1757,29	0,00	26984,8
0,00%	131,8/sec	2061,93	0,00	16017,2

Figura 65: Estadísticas 2 con 250 users
Elaborado por: Karen Achig y Bryan Torres

Estadísticas por solicitud

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Send Bytes	Latency	Conned Time(ms)
1	01:25:21.623	Grupo de Hilos 1-3	Initial Login Request	32	✓	5050	0	32	0
2	01:25:21.631	Grupo de Hilos 1-5	Initial Login Request	24	✓	5048	0	24	0
3	01:25:21.614	Grupo de Hilos 1-1	Initial Login Request	41	✓	5052	0	41	0
4	01:25:21.627	Grupo de Hilos 1-4	Initial Login Request	28	✓	5050	0	28	0
5	01:25:21.618	Grupo de Hilos 1-2	Initial Login Request	33	✓	5050	0	33	0
6	01:25:21.656	Grupo de Hilos 1-10	Initial Login Request	24	✓	5048	0	24	0
7	01:25:21.639	Grupo de Hilos 1-7	Initial Login Request	41	✓	5050	0	41	0
8	01:25:21.635	Grupo de Hilos 1-6	Initial Login Request	46	✓	5050	0	46	0
9	01:25:21.647	Grupo de Hilos 1-9	Initial Login Request	35	✓	5048	0	35	0
10	01:25:21.643	Grupo de Hilos 1-8	Initial Login Request	44	✓	5052	0	44	0
11	01:25:21.658	Grupo de Hilos 1-11	Initial Login Request	49	✓	5050	0	49	0
12	01:25:21.658	Grupo de Hilos 1-4	Auth	71	✓	26991	0	76	0
13	01:25:21.658	Grupo de Hilos 1-1	Auth	72	✓	26991	0	72	0
14	01:25:21.658	Grupo de Hilos 1-3	Auth	91	✓	26981	0	90	0
15	01:25:21.660	Grupo de Hilos 1-12	Initial Login Request	89	✓	5050	0	89	0
16	01:25:21.664	Grupo de Hilos 1-13	Initial Login Request	91	✓	5050	0	91	0
17	01:25:21.659	Grupo de Hilos 1-2	Auth	98	✓	26991	0	97	0
18	01:25:21.658	Grupo de Hilos 1-5	Auth	108	✓	26976	0	105	0
19	01:25:21.681	Grupo de Hilos 1-17	Initial Login Request	95	✓	5050	0	95	0
20	01:25:21.668	Grupo de Hilos 1-14	Initial Login Request	110	✓	5050	0	110	0
21	01:25:21.677	Grupo de Hilos 1-16	Initial Login Request	116	✓	5048	0	116	0
22	01:25:21.672	Grupo de Hilos 1-15	Initial Login Request	136	✓	5050	0	136	0
23	01:25:21.683	Grupo de Hilos 1-9	Auth	135	✓	26986	0	135	0
24	01:25:21.683	Grupo de Hilos 1-10	Auth	136	✓	26986	0	135	0
25	01:25:21.683	Grupo de Hilos 1-7	Auth	140	✓	26986	0	139	0
26	01:25:21.694	Grupo de Hilos 1-20	Initial Login Request	166	✓	5050	0	166	0
27	01:25:21.685	Grupo de Hilos 1-18	Initial Login Request	164	✓	5050	0	164	0

Figura 66: Estadísticas por solicitud
Elaborado por: Karen Achig y Bryan Torres

Tiempo de respuesta

Resultados Tiempo de Respuesta

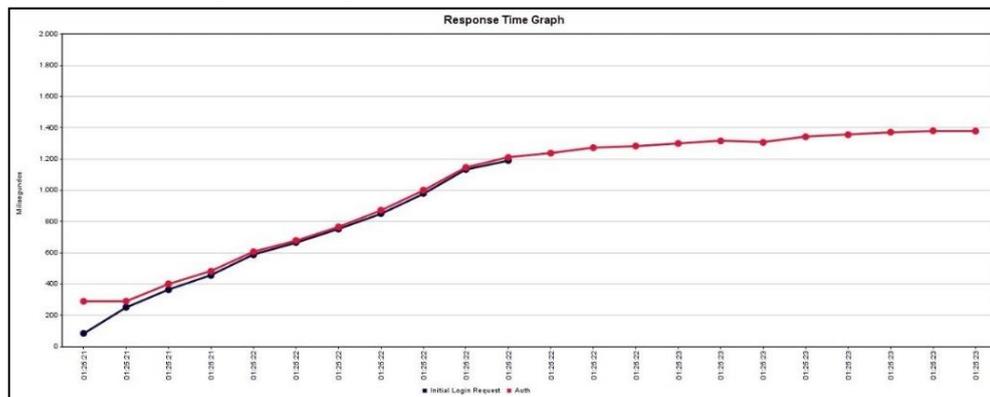


Figura 67: Resultados Tiempo de Respuesta
Elaborado por: Karen Achig y Bryan Torres

3.6.2 Pruebas Unitarias

Las pruebas unitarias sirven para probar componentes de un programa, garantizando que dichos componentes funcionen aisladamente. En este caso se evaluán diferentes métodos los cuales producen un conjunto de salidas que sirven para validar si son correctas o no.

Como herramienta para la realización de estas pruebas se usó JUnit, el mismo que viene ya precargado en Netbeans que es el IDE de desarrollo del Sistema SPPP.

A continuación, se presentan las pruebas:

Tabla 45: Ejecución Pruebas Unitarias

Nro.	Paquete.	Clase	Método	Entada	Tiempo / s	Resp.
1	com.sppp.beans	ControllerBean	guardarDatos	C.I tutor, C.I est, Dia	3.76	OK
2	com.sppp.beans	ControllerBean	updateCantidadVisita	Id visita	3.903	OK
3	com.sppp.mailing	MailingMain	Mensajes	Cod. mensaje, correo est, mensaje	2.751	OK
4	com.sppp.DAO	CitasDaoImp	listarVisitados	C.I. est	3.886	OK
5	com.sppp.DAO	VisitaDAO	validacion-docs	C.I. est	4.53	OK
6	com.sppp.DAO	VisitaDAO	validacionDocs	Ninguna	0.237	OK
7	com.sppp.DAO	DashboardTutDAO	ListarCoordinador-Resolucion	Ninguna	3.89	OK
8	com.sppp.classes	CitasAgendadas	getCedula-est	Ninguna	0.114	OK
9	com.sppp.classes	CitasAgendadas	getVisitasRealizadas	C.I. tutor	4.247	OK
10	com.sppp.classes	DatosFormatos	getPasantia	C.I. est	4.179	OK

Nota: Esta tabla contiene información acerca de la prueba con JUnit realizada al sistema.

Las pruebas anteriores se desarrollaron teniendo en cuenta métodos muy importantes dentro del sistema y de los cuales se han obtenido respuestas favorables al momento de la evaluación.

3.6.3 Pruebas contra requerimientos

Estas pruebas validaron que todos los requerimientos solicitados por diferentes actores del sistema están aptos, por lo cual a continuación se creó una tabla especificando dichos requerimientos.

Requerimientos funcionales

Tabla 46: Tutor

Nombre de la prueba	Req.	Pasos	Ok	Observaciones
Inicio de actividades del estudiante	1.1	Acceder al sistema con perfil gestor. Seleccionar menú: Val.Solicitud Resolucion.Escoger estudiante a validar. Ingresar numero de resolución del estudiante. Guardar registro.	Si	
Confirmar fecha de visita	1.1	Acceder al sistema con perfil estudiante. Seleccionar cita en el panel de notificaciones. Confirmar fecha de visita.	Si	
Cancelación de cita	1.1	Acceder al sistema con perfil estudiante. Seleccionar cita en el panel de notificaciones. Ingresar observación. Cancelar cita.	Si	
Crear fecha de visita	1.2	Acceder al sistema con perfil tutor. Seleccionar menú: Citas. Click en el plus button. Escoger estudiante e ingresar datos de visita. Guardar registro.	Si	
Llenar información en Informe del tutor	1.5	Acceder al sistema con perfil tutor. Seleccionar menú: Citas. Click en el botón Visitado. Ingresar datos en el informe. Guardar registro.	Si	

Descargar Informe tutor	1.5	Acceder al sistema con perfil tutor. Seleccionar menú: Documentos. Click en Descargar informe tutor.	Si	
Llenar información en informe de la pasantía	1.6	Acceder al sistema con perfil tutor. Seleccionar menú: Citas. Click en el botón Visitado.	No	El tutor prefiere llevar la hoja física y llenarla manualmente.
Subir informe del tutor al sistema	1.7	Acceder al sistema con perfil tutor. Seleccionar menú: Documentos. Click en examinar (Informe tutor). Guardar archivo.	Si	
Subir informe de seguimiento al sistema	1.7	Acceder al sistema con perfil tutor. Seleccionar menú: Documentos. Click en examinar (Informe de seguimiento). Guardar archivo.	Si	

Nota: Esta tabla contiene información acerca de los requerimientos funcionales Tutor

Tabla 47 Estudiante

Nombre de la prueba	Req.	Pasos	Ok	Observaciones
Envío de fecha para visita	1.3	Acceder al sistema con perfil tutor. Seleccionar menú: Citas. Click en el plus button. Escoger estudiante e ingresar datos de visita. Guardar registro.	Si	
Cancelación de cita	1.3	Acceder al sistema con perfil tutor. Seleccionar cita en el panel de notificaciones. Click en el botón Cancelar de un estudiante en específico. Click en cancelar cita.	Si	
Aceptar Cita para la visita	1.4	Acceder al sistema con perfil estudiante. Seleccionar cita en el panel de notificaciones. Confirmar fecha de visita.	Si	
Llenar datos para solicitud de validación	2.1	Acceder al sistema con perfil estudiante. Seleccionar Autoevaluación en el panel Llenar datos de solicitud.	No	La solicitud para validar las pasantías se la llena en el portal de la U.P.S. (Sistema de solicitudes)
Descargar formato de solicitud para validación final	2.1	Acceder al sistema con perfil estudiante. Seleccionar Descargar/Subir Documentos Validación en el panel de notificaciones.	No	La solicitud se la descarga en el portal de la U.P.S. (Sistema de solicitudes)
Llenar autoevaluación	2.2	Acceder al sistema con perfil estudiante. Seleccionar Autoevaluación en el panel Llenar datos autoevaluación. Guardar registro.	Si	
Descargar formato de autoevaluación	2.2	Acceder al sistema con perfil estudiante. Seleccionar Descargar/Subir Documentos Validación en el panel de notificaciones. Click en el botón Descargar.	Si	
Subir solicitud de validación final	2.3	Acceder al sistema con perfil estudiante. Seleccionar Descargar/Subir Documentos Validación en el panel de notificaciones.	Si	

		Click en examinar (Solicitud). Guardar Archivo(Global).		
Subir autoevaluación	2.3	Acceder al sistema con perfil estudiante. Seleccionar Descargar/Subir Documentos Validación en el panel de notificaciones. Click en examinar (Autoevaluación). Guardar Archivo(Global).	Si	
Subir certificado de culminación de pasantía	2.3	Acceder al sistema con perfil estudiante. Seleccionar Descargar/Subir Documentos Validación en el panel de notificaciones. Click en examinar (Certificado de culminación). Guardar Archivo(Global).	Si	

Nota: Esta tabla contiene información acerca de los requerimientos funcionales Estudiante.

Tabla 48 Coordinador

Nombre de la prueba	Req.	Pasos	Ok	Observaciones
Validar documentación	2.4	Acceder al sistema con perfil coordinador. Seleccionar Revisión en el panel de notificaciones. Escoger estudiante a validar documentación. Descargar archivos.	Si	
Validar información subida por el estudiante	2.5	Acceder al sistema con perfil coordinador. Seleccionar Revisión en el panel de notificaciones. Escoger estudiante a validar documentación. Descargar archivos. Check documentos revisados. Ingresar observación. Guardar registro.	Si	

Nota: Esta tabla contiene información acerca de los requerimientos funcionales Coordinador.

Capítulo 4

4 Implementación

En este capítulo se explica el levantamiento del sistema en un servidor, así como el de la base de datos y la información de la tecnología utilizada en dicho proceso.

A continuación, se presentan las diferentes configuraciones realizadas para la correcta implementación del sistema.

4.1 Implementación de HW y SW

Se utilizó una computadora personal como servidor a continuación, sus características:

Sistema Operativo: Windows.

Versión Sistema: Windows 8.1 Pro

Bits Del Sistema Operativo: 64 bits

CPU-Marca: DELL

CPU-Modelo: OPTIPLEX 9020

Procesador: Intel Core i7

Ram: 8gb

Disco Duro: 464 GB

Particiones: C de 270 GB y E de 194 GB.

Monitor (No Touch): 14 pulgadas.

Tipo de Conexión a internet: Cableado

4.2 Configuración de la base de datos

Gestor de BDD: PostgreSQL 9.4

Pre requisitos para la instalación de PostgreSQL: Microsoft Visual C++ 2013

Información de configuración PostgreSQL:

Username: postgres

Pwd: s3p@2018

Port: 5432

Información de Base de Datos:

Nombre: db_sp3

Owner: postgres

Nro. tablas: 18

Configuraciones para conexión remota

4.2.1 Paso 1

Firewall de Windows. Para permitir el acceso total al puerto 5432 se realiza la siguiente configuración:

Ir a inicio.

Panel de Control.

Buscar: Firewall de Windows.

Ir a configuración avanzada.

Seleccionar: Reglas de Entrada, dar Click derecho y escoger: Nueva regla.

Escoger Puerto, click en siguiente y en la ventana se tiene por defecto TCP y en puertos locales específicos ingresar: 5432.

Dar click en siguiente y escoger: Permitir la conexión.

Click en siguiente y escoger: Privado.

Por último, poner un nombre y descripción a la regla y finalizar el proceso.

4.2.2 Paso 2

Fichero postgresql.conf. Para que se acepten todas las peticiones configuramos este archivo, para ello buscamos la siguiente línea: listen addresses = localhost y cambiamos el localhost por *, lo que da como resultado: listen addresses = *. Guardamos el archivo y listo.

4.2.3 Paso 3

Fichero pg.hba.conf. Para permitir todas las conexiones entrantes hay que configurar el archivo de autenticaciones, para ello se abre el archivo e ingresamos la siguiente línea dentro de las conexiones: host all all 0.0.0.0/0 md5. Después de ello se reinicia el servicio y listo.

4.3 Configuraciones Servidor Web

Servidor Web: Glassfish 4.1.1

Requerimientos: Instalar Java

Descargar el instalador desde la página de Oracle en el siguiente link:

<http://download.oracle.com/glassfish/4.1.1/release/index.html>

Instalación Glassfish

Descomprimir la carpeta glassfish4 en el directorio C.

Inicializar Glassfish

Abrir la consola de Windows

Ir al siguiente path: C:/glassfish4/bin

Ejecutar el siguiente comando: asadmin start-domain y listo

Consola de Administración Glassfish

En el navegador ingresar lo siguiente: localhost:4848

Subir el .war del Proyecto

Ya en la consola de administración dar click en: Aplicaciones.

Click en: Deploy

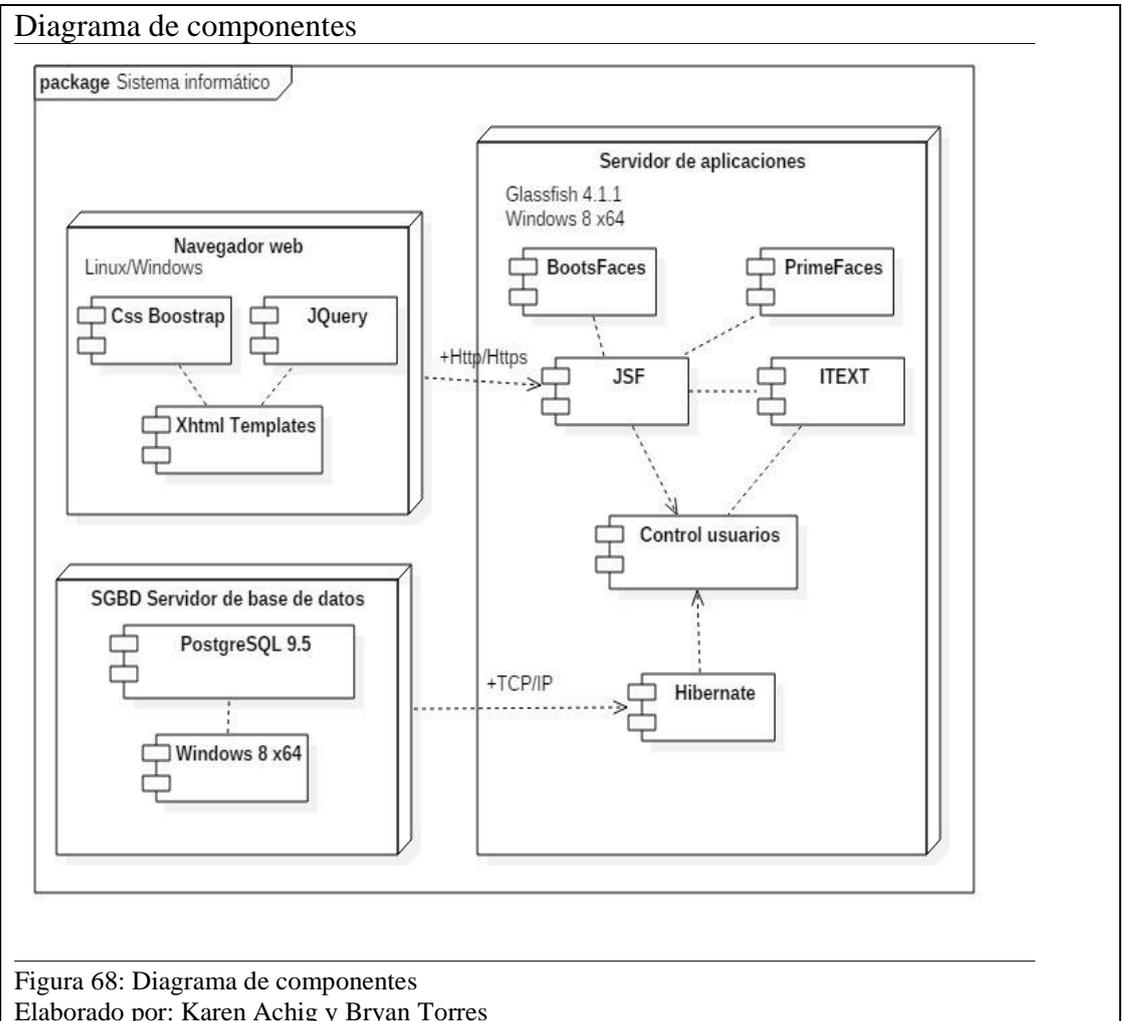
Seleccionar el archivo .war y dar click en aceptar.

Ejecutar el proyecto

En la consola de administración dar click en: Aplicaciones.

En la lista de proyectos buscar: SPPP-App y dar click en Launch

4.4 Diagrama de componentes



CONCLUSIONES

- SPPP es un sistema web amigable e intuitivo para el seguimiento y control de los estudiantes que se encuentran en el proceso de pasantías, que permite a estudiantes, tutores y administrativos llevar a cabo el flujo de trabajo con respecto a dicho procedimiento, cumpliendo a cabalidad todos los requerimientos establecidos por el usuario.
- Utilizar BootsFaces y PrimeFaces integrados dentro del sistema permitió brindar una mejor experiencia para el usuario ya que se pudo desarrollar ventanas amigables e intuitivas logrando realizar el diseño establecido en el modelado del sistema.
- La metodología XP utilizada en la construcción del sistema permitió la integración del usuario dentro del desarrollo, consiguiendo de esta manera una construcción de los módulos más rápida y eficaz, puesto que en cada iteración se integraba requerimientos nuevos, cambios de diseño o funcionalidad y su aprobación de cada elemento.
- En las pruebas de funcionalidad realizadas con estudiantes y docentes, se consiguieron resultados de funcionamiento óptimo lo que muestra que el sistema se encuentra bien estructurado y está en condiciones de realizar un crecimiento o integración.
- GitHub es una herramienta que facilitó el proceso de integración con los módulos iniciales del sistema ya que mediante esta se pudo consumir la información de dichos módulos y lograr que el sistema funcione como uno solo. (GitHub, 2008)

- Durante el desarrollo del sistema se utilizó Hibernate para el mapeo de la base de datos, con lo que se pudo evidenciar que al hacerlo de manera automática muchas relaciones no se establecían correctamente por lo que se tuvo que realizar el mapeo manualmente para que no se vea afectada la construcción del sistema.
- JMeter es una herramienta con la que se realizó pruebas de estrés al sistema sometiéndolo con 250 usuarios concurrentes, obteniendo un error del 0.00 % y un tiempo de respuesta de 1059 ms, lo cual es favorable ya que el tiempo promedio que espera el usuario una petición es de 2000 ms.
- El proceso de pasantías y prácticas pre profesionales con la implementación del sistema SPPP logró la optimización de tiempo y recursos, ofreciendo a estudiantes y docentes una información clara del flujo de trabajo que se debe seguir durante todo este procedimiento.

RECOMENDACIONES

- Para una nueva versión del sistema se recomienda integrar mapas referenciales de las empresas en las que los estudiantes estén realizando la pasantía, con la finalidad de que los tutores puedan reconocer con exactitud la dirección de dicha empresa y puedan agendar con más precisión la visita, optimizando así tiempo y recursos.
- Es necesario realizar respaldos de la base de datos al finalizar cada periodo, con el propósito de que se pueda rescatar la información de los estudiantes que tengan procesos inconclusos y poder integrarla con la nueva base de datos del presente periodo.
- Se recomienda que para una nueva versión del sistema se desarrolle un módulo de administración, con la finalidad de implementar funciones que no se encuentran especificadas dentro del flujo de trabajo y de esta manera conseguir resolver problemas de una manera más sencilla.
- Durante el desarrollo del sistema se presentaron problemas en relación con el flujo de trabajo que se lleva para el proceso de pasantías, por lo que se recomienda evaluar nuevamente el flujo de trabajo y se pueda añadir estas nuevas funcionalidades dentro de una nueva versión del sistema.

GOSARIO DE TÉRMINOS

SPPP: Sistema de pasantías y prácticas pre profesionales.

Workflow: Permite controlar un conjunto de actividades, considerándolo como un flujo de trabajo que debe pasar de un participante a otro.

LISTA DE REFERENCIA

- Achig, K. V., & Torres, B. A. (2017). *Entrevista a Ricardo Albarrin docente encargado del proceso de pasantías universitarias*. Universidad Politécnica Salesiana.
- Adamini, M. (2011). *La precarización laboral oculta en las pasantías universitarias. Cambios y continuidades en su nueva ley*. *Derecho y Ciencias Sociales*. No. 4, *Red Instituto de Cultura Jurídica y Maestría en Sociología Jurídica*.
- AJAX. (2011). *Microsoft Ajax*. . <http://www.ajaxpro.info>.
- Art.87. (2010). *Ley Orgánica De Educación Superior*. .
- BootsFaces. (2013). *Bootsfaces*. www.bootsfaces.org.
- Bootstrap. (2016). *Acens*. Obtenido de <https://www.acens.com>
- Cabot Sagrera, J. . (2013). *Ingeniería del software*. UOC.
- Caro Herrero, J. L., A. Guevara Plaza, & A. Aguayo M. (1999). *Tecnología workflow aplicada a los sistemas informáticos de gestión hotelera*. *Secretaría de Estado de Comercio*.
- Córcoles, , T., & Montero , S. (2014). *Acceso a datos*. RA-MA.
- Flores , J., Castillo, J., & Tufiño, E. (2018). *Análisis, diseño y construcción del módulo de registro de un workflow para el seguimiento de pasantías de la carrera de Ingeniería de Sistemas - Sede Quito.(Tesis pregrado)*. Ecuador: Universidad Politécnica Salesiana.
- Freire Aillón, T. M. , & J. F. Villacís López . (2015). *Gestión de procesos de atención a usuarios mediante un BPM*. Pontificia Universidad Católica del

Ecuador.

- Geronimo, G. , & V. Canseco . (2002). *Breve introducción a los sistemas colaborativos: Groupware workflow*.
- GitHub. (2008). *GitHub*. Obtenido de <https://github.com/github>.
- Gomez Ruedas, J. . (2016). *Dirección y gestión de proyectos de tecnologías de la información en la empresa*. Fc.
- Higuera, J. A., C. M. , Duran Camelo , & O. Torres Ced. (2014). *Scrum a través de una aplicacion movil*.
- Hollingsworth, D. . (1995). *Workflow Management Coalition The Workflow Reference Model*. Crown Walk.
- Itext. (2010). *Itext Software*. <https://itextpdf.com/>.
- Joskowicz, J. . (2008). *Reglas y practicas en extreme programming*. . España.
- Kimmel, P. (2008). *Manual de UML*. McGraw-Hill Interamericana.
- Letelier Torres, P. , & E. Sanchez Lopez . (2003). *Metodologías agiles en el desarrollo de software*. ISSI.
- Menendez Dominguez, V., & Castellanos Bolaños, M. (2016). *Los Sistemas Gestores de Flujos de Trabajo en la gestion de procesos de software*. Mexico: Universidad Autonoma de Yucatan .
- Meneses Carrasco, S. , & J. Vargas Carrillo . (2005). *Desarrollo de un sistema de manejo documental a través de una intranet corporativa utilizando oracle workflow*.). Ecuador: Universidad de las Fuerzas Armadas .
- Mohedano, J., J. M. Saiz , & R. P Salazar . (2012). *Iniciación a javascript*. Madrid: Ministerio de Educaci´on de España.
- Monte Galiano, J. (2016). *Implantar scrum con exito*. . Editorial UOC.
- Moreira Delgado, M. . (2006). *La gestion por procesos en las instituciones de*

información.

Naranjos Arcos, M. , & L. Villarroel Guano . (2002). *Diseño e implementación de procesos y controles de administración de documentación basado en herramientas informáticas.* Ecuador: Universidad de las Fuerzas Armadas.

NetBeans. (1997). *NetBeans IDE.* . <https://netbeans.org/>.

Ponce, A. J., & R. C. Choque. (2014). *Propuesta de integración de google web tool-kit con glassfish y enterprise java bean para desarrollar un sistema web y la administración de informacion. ciencia de la ingeniería de sistemas.*

Rios, Sergio . (2015). *PrimeFaces 5 with JAX-WS y EJB'S.* Sergio Rios. .

Rodríguez González, & María Elena. . (2013). *Gestión de datos: bases de datos y sistemas gestores de bases de datos.* UOC.

Rusinkiewicz, M. , & A. Sheth . (1994). *Specification and execution of transactional workflows en Modern Database Systems.* Cambridge, Massachussets: KIM.

Salesiana, U. P. (2012). *Reglamento De Vinculación Con La Sociedad, Art. 14 y 15.*

Sarasa, C. A. (2017). *Gestión de la información web usando python.* . UOC. .

Scrum org. (1993). *Metodología Scrum.* Obtenido de <https://www.scrum.org/why-scrumorg>.

Universidad Politécnica Salesiana. (2014). *Reglamento Interno de Régimen Académico de la Reglamento Interno de Régimen Académico de la Universidad Politécnica Salesiana Art. 136- 138.*

Vara, M. J. M., , S. M. López , & D. Granada . (2014). *Desarrollo web en entorno cliente.*

WFMC. (1993). *Workflow Management Coalition.* . <http://www.wfmc.org>. .