



UNIDAD ACADÉMICA:

OFICINA DE INVESTIGACIÓN Y POSTGRADOS

TEMA:

DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE
PROCESOS DE GRADUACIÓN DE UNA UNIDAD DE POSTGRADOS

**Proyecto de Investigación y desarrollo previo la obtención del título de
Magister en Gerencia Informática**

Línea de Investigación, Innovación y Desarrollo principal:

Ingeniería de Software y/o Plataformas Educativas

Caracterización técnica del trabajo:

Desarrollo

Autora:

Gabriela Fabiola De la Torre Sánchez

Director:

José Marcelo Balseca Manzano, Mg.

Ambato – Ecuador

Enero 2018

Desarrollo de una Aplicación Web para la Administración de Procesos de Graduación de una Unidad de Postgrados

Informe de Trabajo de Titulación
presentado ante la
Pontificia Universidad Católica del Ecuador
Sede Ambato

por

Gabriela Fabiola De la Torre Sánchez

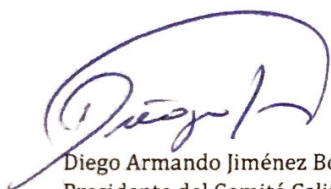
En cumplimiento parcial de
los requisitos para el Grado de
Magister en Gerencia Informática
con mención en Desarrollo De
Software y Redes



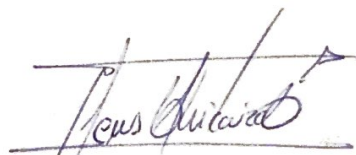
Oficina de Investigación y Postgrados
Enero 2018

Desarrollo de una Aplicación Web para la Administración de Procesos de Graduación de una Unidad de Postgrados

Aprobado por:



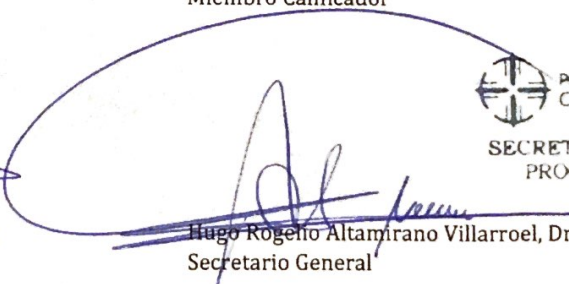
Diego Armando Jiménez Bosquez, Prof.
Presidente del Comité Calificador
Coordinador de la oficina de
Investigación y Posgrados



Dennis Vinicio Chicaiza Castillo, Mg.
Miembro Calificador



José Marcelo Balseca Manzano, Mg
Miembro Calificador
Director de Proyecto

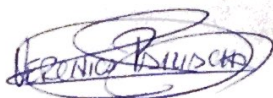


Hugo Rogelio Altamirano Villarroel, Dr.
Secretario General



Pontificia Universidad
Católica del Ecuador

SECRETARÍA GENERAL
PROCURADURÍA



Verónica Maribel Pailiacho Mena, Mg
Miembro Calificador

Fecha de aprobación:
Enero 2018



Pontificia Universidad
Católica del Ecuador

BIBLIOTECA

Ficha Técnica

Programa: Magister en Gerencia Informática

Tema: “Desarrollo de una Aplicación Web para la Administración de Procesos de Graduación de una Unidad de Postgrados”

Tipo de trabajo: Proyecto de Investigación y Desarrollo

Clasificación técnica del trabajo: Desarrollo

Autor: Gabriela Fabiola De la Torre Sánchez

Director: José Marcelo Balseca Manzano, Mg

Líneas de Investigación, Innovación y Desarrollo

Principal: Ingeniería de Software y/o Plataformas Educativas

Resumen Ejecutivo

Mediante el análisis objetivo de los procesos de graduación que tiene lugar actualmente en el Departamento de Investigación y Posgrados (DIP) de la Pontificia Universidad Católica del Ecuador Sede Ambato (PUCESA), se ha logrado identificar múltiples procedimientos engorrosos que ocasionan pérdidas innecesarias de tiempo, dinero y otros recursos tanto al personal del DIP como a los estudiantes, docentes y demás actores del proceso.

Para optimizar dichos procedimientos y agilizar todo el proceso en general, se ha elaborado una aplicación web que permite cumplir con la mayor parte de los requerimientos del proceso de graduación de posgrados bajo la modalidad on-line a través del uso de un computador con conexión a Internet. Dicha aplicación contempla tanto los procedimientos que deben ser ejecutados por los maestrantes así como por los docentes, tanto en calidad de directores así como de lectores de los proyectos, y por el (la) director(a) del DIP.

Hay que mencionar que, si bien la presente aplicación ha sido desarrollada bajo una óptica de unidades de posgrados en general, se ha utilizado como modelo el proceso de graduación de la PUCESA, y los datos preexistentes en sus bases de datos a través de la habilitación de webservices, gracias a la cooperación de su departamento de Informática.

Declaración de Originalidad y Responsabilidad

Yo, Gabriela Fabiola De la Torre Sánchez, portadora de la cédula de ciudadanía y/o pasaporte No. 1802622421, declaro que los resultados obtenidos en el proyecto de titulación y presentados en el informe final, previo a la obtención del título de Magister en Gerencia Informática con mención en Desarrollo De Software y Redes, son absolutamente originales y personales. En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto, y luego de la redacción de este documento, son y serán de mi sola y exclusiva responsabilidad legal y académica.

Gabriela Fabiola De la Torre Sánchez

Gabriela Fabiola De la Torre Sánchez

1802622421



BIBLIOTECA

*Dedico el presente trabajo en primer
lugar a Dios, por su infinita Bondad e
innumerables Gracias*

*A mis padres, por haber sido siempre un
apoyo incondicional en todas las áreas
de mi vida*

*A mi esposo, por ser mi compañero de
vida en las buenas y en las malas*

*Y a mi hermana, sobrinos y resto de mi
familia por traer siempre alegría a mi
vida*

Reconocimientos

Mi sincero reconocimiento va, en primer lugar, para la PUCESA como institución, por haberme otorgado la beca de excelencia académica que me resultó de gran ayuda y estímulo para poder culminar con éxito la malla curricular de mi Maestría.

Para la elaboración del presente trabajo de desarrollo fue de gran ayuda la colaboración del Ing. Mg. José Marcelo Balseca Manzano como director del proyecto, quien con su experiencia en el campo académico ha sabido guiarme acertadamente a través de cada etapa durante el desarrollo del mismo.

De igual forma, ha sido vital la ayuda del Ing. Mg. José Fabián Enríquez, quien en representación del Departamento de Informática de la PUCESA tuvo la amabilidad de poner en mi conocimiento los lineamientos generales sobre la estructura y funcionamiento del Academics y sus bases de datos, que guarda relación directa con la aplicación que he desarrollado, así como también la apertura de habilitarme el acceso necesario a dichas bases de datos para posibilitar el desarrollo del presente proyecto.

Además, mi especial reconocimiento a la PhD. Varna Hernández Junco, quien en su calidad de Directora del Departamento de Investigación y Posgrados de la PUCESA me ha brindado el apoyo necesario que he requerido de su parte para poder concluir con el desarrollo del presente proyecto.

Y finalmente, mi reconocimiento y agradecimiento a todas las personas internas y externas a la PUCESA que de una u otra manera me han brindado su apoyo y colaboración para poder ejecutar el presente proyecto de desarrollo de software.

Resumen

La presente investigación tiene por objeto desarrollar una aplicación web que sirve como herramienta de apoyo para el personal de la Oficina de Investigación y Posgrados (OIP) de la Pontificia Universidad Católica del Ecuador Sede Ambato (PUCESA), con la finalidad de automatizar el manejo de la información correspondiente a dicha oficina y generar la información estadística que su personal requiere de manera urgente, dada la carencia de una aplicación que satisfaga dichos requerimientos. Siendo éste un proyecto de desarrollo de software específicamente diseñado para ser usado por el personal administrativo de la OIP, se utilizan como métodos de recolección de información la observación directa y las entrevistas con dicho personal. Además, como metodología de desarrollo de software se usa Extreme Programming (XP), ya que es la metodología que mejor se adapta a las características técnicas y a la urgencia de implementación de la aplicación a desarrollarse. Además, como herramientas de desarrollo de software y de almacenamiento de información se recurre a Microsoft Visual Studio 2010 y Microsoft SQL Server 2008 R2 respectivamente, con la finalidad de guardar total compatibilidad con las plataformas de hardware y software ya existentes en la PUCESA. Como resultado se obtiene una aplicación web que es debidamente evaluada y probada tanto desde el punto de vista funcional como técnico, pudiéndose implementar y poner en producción sin ningún inconveniente en los servidores de la PUCESA, y que satisface cabalmente los requerimientos del personal de la OIP.

Palabras clave: Aplicación, Web, Desarrollo, Software, Administración, Estadísticas, Posgrados

Abstract

This investigation aims to the development of a web application that serves as a support tool for the staff of the Investigation and Postgraduate Office (OIP) of the Pontifical Catholic University of Ecuador in Ambato (PUCESA). To this end, automated information will be handled by the office, in addition to immediate statistical information that it may require, since there is not any application that meets those requirements. As this is a software development project specifically designed to be used by OIP administrative staff, direct observation and interviews were applied as information gathering. Moreover, Extreme Programming (XP) is used as software development methodology, as it is the methodology that best meets the technical characteristics and the high importance to implement the application. Furthermore, Microsoft Visual Studio 2010 and Microsoft SQL Server 2008 R2 are used as software development and information storage tools respectively, in order to keep full compatibility with the hardware and software platforms that are already implemented at PUCESA. As a result, a web application is created, which was formally assessed and tested in both functional and technical criteria. Finally, the application was put into use without any issues in the servers of PUCESA, which entirely meets the OIP staff requirements.

Keywords: application, web, development, software, management, statistics, postgraduate

Tabla de Contenidos

| | |
|---|-------------|
| Ficha Técnica | iii |
| Resumen Ejecutivo | iii |
| Declaración de Originalidad y Responsabilidad..... | iv |
| Reconocimientos | vi |
| Resumen | vii |
| Abstract..... | viii |
| Tabla de Contenidos..... | ix |
| Lista de Tablas..... | xiv |
| Lista de Figuras | xvi |
| CAPÍTULOS | |
| 1. Introducción | 1 |
| 1.1. Presentación del trabajo..... | 2 |
| 1.2. Descripción del documento..... | 2 |
| 2. Planteamiento de la Propuesta de Trabajo..... | 5 |
| 2.1. Información técnica básica..... | 6 |
| 2.2. Descripción del problema | 6 |
| 2.3. Preguntas básicas | 7 |
| 2.4. Formulación de meta | 7 |
| 2.5. Objetivos | 7 |
| 2.6. Delimitación funcional | 8 |
| 3. Marco Teórico | 11 |
| 3.1. Definiciones y conceptos | 11 |
| 3.1.1. Ingeniería de software | 11 |
| 3.1.1.1. El Proceso del software | 12 |

| | | |
|--------------|---|-----------|
| 3.1.1.2. | La estructura del proceso del software..... | 12 |
| 3.1.2. | Metodologías de desarrollo de software..... | 16 |
| 3.1.2.1. | Metodologías tradicionales vs. metodologías ágiles | 17 |
| 3.1.2.2. | Metodología XP (Extreme Programming) | 22 |
| 3.1.2.2.1. | Elementos básicos de Extreme Programming..... | 22 |
| 3.1.3. | Internet y las Aplicaciones WEB..... | 25 |
| 3.1.3.1. | Arquitectura de la World Wide Web | 25 |
| 3.1.3.1.1. | Hipertexto e hipermedia | 26 |
| 3.1.3.1.2. | Protocolos y estándares de la World Wide Web | 29 |
| 3.1.3.1.2.1. | HTTP..... | 29 |
| 3.1.3.1.2.2. | HTTPS | 30 |
| 3.1.3.1.2.3. | HTML..... | 32 |
| 3.1.3.1.2.4. | CSS | 34 |
| 3.1.3.2. | Servidores WEB o HTTP | 35 |
| 3.1.3.2.1. | Internet Information Services | 36 |
| 3.1.3.2.2. | Páginas web dinámicas..... | 37 |
| 3.1.3.2.3. | Aplicaciones web | 37 |
| 3.1.3.2.4. | ASP.NET | 39 |
| 3.1.3.3. | Web services | 41 |
| 3.1.4. | Gestión Documental..... | 42 |
| 3.1.4.1. | PDF | 43 |
| 3.2. | Estado del Arte..... | 45 |
| 4. | Metodología | 49 |
| 4.1. | Diagnóstico | 50 |
| 4.2. | Método(s) aplicado(s) | 52 |
| 4.2.1. | Desarrollo de software mediante la metodología Extreme Programming (XP) | 53 |

| | | |
|------------------|--|------------|
| 4.2.1.1. | Fase de planeación del proyecto..... | 61 |
| 4.2.1.1.1. | Historias de usuario..... | 62 |
| 4.2.1.1.2. | Plan de entregas..... | 67 |
| 4.2.1.2. | Fase de diseño del proyecto..... | 69 |
| 4.2.1.2.1. | Metáfora de la aplicación..... | 69 |
| 4.2.1.2.2. | Tarjetas Clase – Responsabilidad – Colaboración (CRC)..... | 71 |
| 4.2.1.3. | Fase de codificación y desarrollo del proyecto..... | 74 |
| 4.2.1.3.1. | Refactorización y clarificación del código..... | 74 |
| 4.2.1.3.2. | Diseño e implementación de bases de datos y clases..... | 75 |
| 4.2.1.3.3. | Diagrama de clases..... | 76 |
| 4.2.1.3.4. | Ejemplos de la codificación..... | 77 |
| 4.2.1.4. | Pruebas funcionales..... | 81 |
| 4.3. | Materiales y herramientas..... | 85 |
| 5. | Resultados..... | 87 |
| 5.1. | Producto final del proyecto de titulación..... | 87 |
| 5.2. | Evaluación preliminar..... | 121 |
| 5.3. | Análisis de resultados..... | 121 |
| 5.3.1. | Validación técnica y funcional de la aplicación..... | 122 |
| 6. | Conclusiones y Recomendaciones..... | 126 |
| 6.1. | Conclusiones..... | 126 |
| 6.2. | Recomendaciones..... | 127 |
| APÉNDICES | | |
| A. | Descripción de tablas y campos..... | 128 |
| A.1. | Bases de datos creadas para la aplicación..... | 128 |
| A.2. | Tablas creadas para la base de datos “Graduación”..... | 128 |
| A.2.1. | Tabla “TemasdeTesis”..... | 128 |

| | | |
|----------|--|-----|
| A.2.1.1. | Campos de la tabla “TemasdeTesis” | 129 |
| A.2.1.2. | Posibles estados de un tema de proyecto de titulación de posgrados | 129 |
| A.2.2. | Tabla “MensajesTemasdeTesis” | 131 |
| A.2.2.1. | Campos de la tabla “MensajesTemasdeTesis” | 132 |
| A.2.2.2. | Tipos de autores que pueden generar mensajes para un tema de proyecto de titulación de Posgrados | 132 |
| A.2.2.3. | Tipos de mensajes que pueden ser generados para un tema de proyecto de titulación de posgrados | 133 |
| A.2.3. | Tabla “PlanesdeTesis” | 134 |
| A.2.3.1. | Campos de la tabla “PlanesdeTesis” | 134 |
| A.2.3.2. | Posibles estados de un plan de titulación de posgrados | 138 |
| A.2.4. | Tabla “MensajesPlandeTesis” | 139 |
| A.2.4.1. | Campos de la tabla “MensajesPlandeTesis” | 139 |
| A.2.4.2. | Tipos de autores que pueden generar mensajes para un plan de titulación de Posgrados | 140 |
| A.2.4.3. | Tipos de mensajes que pueden ser generados para un plan de titulación de posgrados | 140 |
| A.2.5. | Tablas auxiliares a la tabla “PlanesdeTesis” | 141 |
| A.2.6. | Tabla TesisPosgrados | 143 |
| A.2.6.1. | Campos de la tabla TesisPosgrados | 144 |
| A.2.6.2. | Posibles estados de un registro de la tabla “TesisPosgrados” | 144 |
| A.2.7. | Tabla AvancesTesis | 145 |
| A.2.7.1. | Campos de la tabla AvancesTesis | 145 |
| A.2.7.2. | Notas sobre la tabla AvancesTesis | 147 |
| A.2.8. | Tabla “MensajesAvancesTesis” | 147 |
| A.2.8.1. | Campos de la tabla “MensajesAvancesTesis” | 148 |
| A.2.9. | Tabla CorreccionesLectoresTesis | 149 |

| | | |
|-----------|---|------------|
| A.2.9.1. | Campos de la tabla CorreccionesLectoresTesis | 149 |
| A.2.9.2. | Notas sobre la tabla CorreccionesLectoresTesis | 150 |
| A.2.10. | Tabla “MiembrosTribunalPosgrados” | 151 |
| A.2.10.1. | Campos de la tabla “MiembrosTribunalPosgrados” | 151 |
| B. | Descripción de Clases..... | 153 |
| B.1. | Clase “UsuarioGraduacionAtencion.cs” | 153 |
| B.1.1. | Miembros de la Clase “UsuarioGraduacionAtencion.cs” | 153 |
| B.2. | Clase “General.cs” | 154 |
| B.3. | Clases de tipo “Managers” | 155 |
| C. | Descripción de Webservices | 157 |
| C.1. | GetCarrerasPosgrados | 157 |
| C.2. | GetDatosDocentes | 157 |
| C.3. | GetDatosEstudiantes | 158 |
| C.4. | GetEstudiantesdePosgrado | 159 |
| C.5. | GetTodoslosEstudiantes | 159 |
| C.6. | EspecificarPerfilUsuario | 160 |
| D. | Encuestas de satisfacción aplicadas al personal del DIP | 161 |
| D.1. | Encuestas reales respondidas personalmente por el personal administrativo del DIP | 161 |
| | Referencias | 166 |

Lista de Tablas

| | |
|--|-----|
| 1. Diferencias entre metodologías ágiles y tradicionales | 21 |
| 2. Equipo de trabajo para desarrollo de la presente aplicación | 56 |
| 3. Historia de usuario 1, solicitud de aprobación del tema | 62 |
| 4. Historia de usuario 2, elaboración del plan..... | 63 |
| 5. Historia de usuario 3, solicitud de docente informante..... | 63 |
| 6. Historia de usuario 4, solicitud de docente director | 64 |
| 7. Historia de usuario 5, matrícula para desarrollo del proyecto..... | 64 |
| 8. Historia de usuario 6, solicitud de docentes revisores..... | 65 |
| 9. Historia de usuario 7, solicitud de tribunal calificador | 65 |
| 10. Historia de usuario 8, revisión en biblioteca para empastado | 66 |
| 11. Historia de usuario 9, solicitud de fecha y hora de defensa | 66 |
| 12. Historia de usuario 10, defensa del proyecto | 67 |
| 13. Plan de entregas por historias de usuario | 68 |
| 14. Pruebas de usuario y funcionales ejecutadas sobre el módulo Tema | 81 |
| 15. Pruebas de usuario y funcionales ejecutadas sobre el módulo Plan | 82 |
| 16. Pruebas de usuario y funcionales ejecutadas sobre los módulos IFP y Defensa | 84 |
| 17. Secuencia de pasos dentro del proceso de graduación de un maestrante | 108 |
| 18. Lista de cotejo sobre el cuestionario de satisfacción aplicado al personal del DIP | 124 |
| 19. Campos de la tabla “TemasdeTesis” | 129 |
| 20. Posibles estados de un tema de proyecto de titulación de posgrados | 130 |
| 21. Campos de la tabla “MensajesTemasdeTesis” | 132 |
| 22. Tipos de autores para la tabla “MensajesTemasdeTesis” | 133 |
| 23. Tipos de mensajes que se pueden generar para la tabla “MensajesTemasdeTesis” | 133 |
| 24. Campos de la tabla “PlanesdeTesis” | 134 |
| 25. Posibles estados de un plan de titulación de posgrados..... | 138 |
| 26. Campos de la tabla “MensajesPlandeTesis” | 139 |

| | |
|--|-----|
| 27. Tipos de autores para la tabla "MensajesPlandeTesis" | 140 |
| 28. Tipos de mensajes que pueden ser generados para un plan de titulación de posgrados | 141 |
| 29. Tablas auxiliares a la tabla "PlanesdeTesis" | 142 |
| 30. Campos de la tabla TesisPosgrados | 144 |
| 31. Posibles estados de un registro de la tabla "TesisPosgrados" | 145 |
| 32. Campos de la tabla "AvancesTesis" | 146 |
| 33. Campos de la tabla "MensajesAvancesTesis" | 148 |
| 34. Campos de la tabla "CorreccionesLectoresTesis" | 149 |
| 35. Campos de la tabla "MiembrosTribunalPosgrados" | 151 |
| 36. Miembros de la clase "UsuarioGraduacionAtencion.cs" | 153 |
| 37. Métodos de la clase "General.cs" | 154 |

Lista de Figuras

| | |
|--|----|
| 1. Flujo del proceso del software..... | 14 |
| 2. Estructura del proceso del software | 16 |
| 3. Ejemplo de historia de usuario XP | 23 |
| 4. Las prácticas y principios de XP deben reforzarse entre sí | 25 |
| 5. Representación simple del hipertexto | 27 |
| 6. Diferencia básica entre HTTP y HTTPS..... | 31 |
| 7. Negociación SSL/TLS | 31 |
| 8. Estructura HTML de una página web básica | 33 |
| 9. Diagrama de funcionamiento de una petición web..... | 36 |
| 10. Flujo de información de una aplicación web con bases de datos..... | 39 |
| 11. Comparación de los ciclos de desarrollo de software entre la metodología XP y metodologías tradicionales (en cascada e iterativa)..... | 54 |
| 12. Ciclo de vida del proceso XP | 55 |
| 13. Proceso actual de envío y aprobación de temas de proyectos de titulación y directores de proyecto para maestrantes de la PUCESA..... | 58 |
| 14. Proceso actual de envío y aprobación de planes de proyectos de titulación que ejecutan los maestrantes de la PUCESA..... | 59 |
| 15. Proceso actual de envío y aprobación de proyectos de titulación de posgrados, asignación de fechas de defensa y registro de calificaciones | 60 |
| 16. Tarjeta CRC “TemadeTesis” | 71 |
| 17. Tarjeta CRC “PlandeTesis” | 72 |
| 18. Tarjeta CRC “AvancesTesis” | 72 |
| 19. Tarjeta CRC “TesisPosgrados”..... | 72 |
| 20. Tarjeta CRC “Docente” | 73 |
| 21. Tarjeta CRC “Estudiante” | 73 |
| 22. Tarjeta CRC “DirectorDIP” | 73 |

| | |
|---|-----|
| 23. Diagrama de clases..... | 77 |
| 24. Página principal de MyASP.NET..... | 88 |
| 25. Página de login de la aplicación..... | 89 |
| 26. Página principal de estudiantes | 90 |
| 27. Editor del tema de proyecto de titulación | 91 |
| 28. Confirmación del envío del tema de proyecto de titulación | 91 |
| 29. Página principal de docentes..... | 92 |
| 30. Lista de solicitudes pendientes de dirección de proyectos de titulación | 93 |
| 31. Revisión de tema por parte del docente director | 93 |
| 32. Aprobación del tema por parte del docente director | 94 |
| 33. Editor del plan de proyecto de titulación – página 1 | 97 |
| 34. Editor del plan de proyecto de titulación – página 3 | 97 |
| 35. Adición de elementos del presupuesto del proyecto..... | 98 |
| 36. Adición de elementos del cronograma..... | 98 |
| 37. Editor del plan de proyecto de titulación – página 7 | 99 |
| 38. Adición de referencias bibliográficas | 99 |
| 39. Sección de referencia bibliográficas del editor del plan..... | 100 |
| 40. Plan de proyecto de titulación en formato PDF | 101 |
| 41. Revisión del plan parte del director del proyecto..... | 101 |
| 42. Página informativa de requisitos para el maestrante | 103 |
| 43. Solicitud generada automáticamente para hoja membretada | 103 |
| 44. Envío de avances del IFP (25%)..... | 105 |
| 45. Revisión de avances del IFP por parte del director del proyecto..... | 106 |
| 46. Página principal del módulo de la dirección del DIP | 109 |
| 47. Ingreso de solicitud de aprobación de un tema | 109 |
| 48. Registro de aprobación de un tema..... | 110 |
| 49. Ingreso de solicitud de designación de docente informante..... | 110 |
| 50. Registro de designación de docente informante | 111 |

| | |
|--|-----|
| 51. Ingreso de solicitud de aprobación de plan | 111 |
| 52. Registro de aprobación del plan | 112 |
| 53. Ingreso de solicitud de designación de docente director | 112 |
| 54. Registro de designación de docente director | 113 |
| 55. Registro de matrícula para iniciar el desarrollo del proyecto | 113 |
| 56. Ingreso de solicitud de designación de docentes revisores | 114 |
| 57. Registro de designación de docentes revisores | 114 |
| 58. Registro de aprobación por docentes revisores | 115 |
| 59. Ingreso de solicitud de designación de tribunal calificador | 115 |
| 60. Registro de designación de tribunal calificador | 116 |
| 61. Registro de revisión en biblioteca para elaboración del empastado | 116 |
| 62. Ingreso de solicitud de designación de fecha y hora de defensa | 117 |
| 63. Registro de designación de fecha y hora de defensa | 117 |
| 64. Registro de designación de fecha y hora de defensa | 118 |
| 65. Registro de resultado de defensa | 118 |
| 66. Registro de resultado de defensa | 119 |
| 67. Gráfico estadístico de número de casos por estado del proceso de graduación | 119 |
| 68. Gráfico estadístico de número de casos por estado del proceso de graduación en formato PDF | 120 |

Capítulo 1

Introducción

El constante aparecimiento de nuevas tecnologías y la evolución continua de los sistemas de información hacen que cada vez sea más deseable y necesaria la automatización de procesos a todo nivel, hasta el punto en que toda organización que no considere mantenerse al día con los avances tecnológicos y las innumerables posibilidades que estos ofrecen, pierde competitividad con una rapidez impresionante.

Y dentro de este vertiginoso mundo de crecimiento tecnológico a todo nivel, toman especial relevancia la llegada constante de nuevo hardware informático cada vez más sofisticado y potente, así como las herramientas de software que permiten aprovechar al máximo esas capacidades del hardware, logrando que las tecnologías de la información se hayan convertido en un verdadero valor para empresas y organizaciones de todo tipo, llegando a considerarse hoy en día que la información es el activo máspreciado de organizaciones medianas y grandes.

Dentro de este marco de tecnologías de la información, las relacionadas a la web y al Internet cobran especial importancia, puesto que desde su aparecimiento cada vez se han ido integrando a la misma tanto organizaciones como individuos, logrando así prácticamente interconectar todo el mundo. Tal es así que, hoy en día, las aplicaciones web son mucho más apetecidas y utilizadas que las tradicionales aplicaciones de escritorio, ya que al existir en una infraestructura virtual en constante crecimiento y evolución, y con un potencial esencialmente ilimitado, se vuelven aplicaciones mucho más productivas y versátiles, a la vez que son casi totalmente independientes del hardware y de las tecnologías locales.

El presente documento justamente consiste en describir la creación de una aplicación web orientada a dinamizar una temática tan importante como contemporánea, como lo es el proceso de graduación de las escuelas de posgrados ecuatorianas. Procesos que, pese a la tecnología y recursos existentes en la actualidad, muchas veces se realizan aún de forma manual, desaprovechándose y desperdiándose valiosos recursos.

1.1. Presentación del trabajo

Analizando la temática expuesta en los párrafos anteriores, se ha decidido tomar como caso de estudio a la Unidad de Posgrados de la Pontificia Universidad Católica del Ecuador, Sede Ambato (PUCESA), prestigiosa institución de educación superior de la más alta calidad con sede en el centro del país, por lo que no solo atrae a estudiantes y profesionales de la ciudad de Ambato, sino también de otras ciudades y provincias del país.

En esta institución, y de manera particular en su Unidad de Posgrados, se logró identificar el problema de que los procesos de graduación se realizaban aún de manera manual, no contando con una herramienta informática especializada para tal fin, lo cual estaba creando problemas de ineficiencia y lentitud en los procesos respectivos. Pese a que la institución contaba con los recursos, la tecnología y la información necesarias para solventar esta necesidad no se lo había llevado a cabo aún, siendo un problema que requería atención urgente.

Por esta razón, se realizó un estudio preliminar donde se identificó de manera más detallada las tecnologías tanto de hardware como de software existentes en la institución, así como la información que ya manejaban aplicaciones desarrolladas para otras áreas académicas, llegándose a la conclusión de que muchos de esos recursos podían ser utilizados para automatizar en la mayor medida posible los procesos de graduación a los que los maestrantes debían someterse obligatoriamente como parte final de su carrera.

Así se dio inicio a este proyecto de desarrollo de aplicación web, con la finalidad de aprovechar los recursos y la información ya existe como parte de otras aplicaciones para crear una herramienta que dinamice los procesos de graduación de posgrado de la institución, aliviando la carga a su personal a la vez que mejorando tangiblemente la calidad en la atención a sus usuarios.

1.2. Descripción del documento

Una vez identificados los problemas y analizados los requerimientos de la mencionada Unidad de Posgrados, se procedió a planificar y elaborar la aplicación producto del presente proyecto, y a la par se elaboró el presente IFP (Informe Final del Proyecto), el mismo que queda constituido de la siguiente manera:

En el presente Capítulo I se procede a brindar al lector una visión general sobre la temática de avances tecnológicos en hardware y software, el avance y crecimiento en el mundo del Internet y las aplicaciones web, y el por qué es un problema que las instituciones y organizaciones de todo índole no marchen de acuerdo a la velocidad de dichos avances. Posteriormente se analiza de manera breve los antecedentes de la Unidad de Posgrados de la PUCESA y sus procesos de graduación, los mismos que se ha tomado como caso de estudio para el desarrollo del presente proyecto.

A continuación en el Capítulo II se expone al lector los resultados del análisis realizado sobre la temática de estudio, plasmándose la descripción de los problemas identificados así como el planteamiento de la solución respectiva, detallándose puntos como el estado del arte, el objetivo general y los objetivos específicos que se persigue, dando respuestas a preguntas básicas sobre cómo surge el problema a resolver, describiendo la metodología de trabajo elegida para el desarrollo del proyecto y definiendo, mediante puntos funcionales, cuáles serán las capacidades de la aplicación a desarrollarse.

En el Capítulo III, correspondiente al marco teórico, se exponen todos los conceptos, definiciones y teorías en los que se basa el presente proyecto de desarrollo de software. Cabe mencionar que se tocan todos los temas de interés y relevancia dentro de la temática del proyecto, así como también las citas bibliográficas correspondientes a las fuentes consultadas, las cuales son la base para esta investigación previa.

El Capítulo IV se adentra en más detalles en relación a la metodología aplicada para el presente desarrollo, sobre todo en las fases de la metodología ágil de desarrollo de software XP, la cual se ha empleado para el desarrollo del presente proyecto, y la misma que ha tomado un gran auge en los tiempos modernos ya que permite desarrollar aplicaciones de alta calidad, orientadas hacia la satisfacción del usuario final, en períodos de tiempo relativamente cortos y sin documentación excesiva.

El Capítulo V se titula "Resultados" y, como se podrá deducir, describe de manera detallada las características y funcionalidades de la aplicación web producto del desarrollo del presente proyecto. Se indica claramente los puntos que se ha logrado automatizar y los problemas que se ha logrado resolver, así como los resultados que se obtendría de la implementación de dicha aplicación.

El Capítulo VI expone las conclusiones y recomendaciones que se puede formular a partir del conocimiento y experiencia adquiridas del proceso de desarrollo del presente proyecto.

A continuación se incluye el Apéndice A, donde se puede conocer más detalladamente la estructura de dicha base de datos, ya que se procede a describir en detalle las tablas creadas para la misma con sus respectivos campos y, de ser el caso, incluso con una descripción de sus valores posibles.

En el Apéndice B se procede a describir las clases que se ha creado para el correcto funcionamiento de la presente aplicación, incluyendo las clases de tipo “Manager”, que permiten transaccionar con las tablas de la base de datos de manera simple y sencilla.

Finalmente, en el Apéndice C se procede a describir los Webservices habilitados por el personal del departamento de Informática de la PUCESA específicamente para el desarrollo de la presente aplicación, ya que los mismos constituyen la forma más óptima y segura de compartir la información ya existente en los servidores de la institución, para que pueda ser utilizada dentro de la nueva aplicación desarrollada.

Capítulo 2

Planteamiento de la Propuesta de Trabajo

Luego de haber determinado las múltiples falencias existentes en la manera en que los maestrantes deben iniciar, continuar y concluir su proceso de graduación, utilizando como modelo el DIP de la PUCESA, se estableció la necesidad de elaborar una aplicación web que permita agilizar y optimizar dicho proceso. El resultado del presente trabajo de desarrollo consiste, justamente, en dicha aplicación, la misma que resultará de interés para todos los actores del proceso de graduación de posgrados como son maestrantes, docentes y dirección del DIP.

Para la elaboración de la presente aplicación se ha recurrido al uso de la plataforma de desarrollo ASP.NET mediante el lenguaje de programación C#, tecnologías que se encuentran disponibles y han sido conjugadas bajo el entorno de desarrollo Visual Studio 2010 de Microsoft. Por otro lado, como motor de diseño, implementación y ejecución de bases de datos se ha empleado SQL Server 2008 R2, también de Microsoft, que es una de las herramientas de gestión de bases de datos más versátiles, potentes y utilizadas a nivel global. Las razones técnicas para haber elegido las mencionadas herramientas de desarrollo consisten principalmente en lograr la compatibilidad con las aplicaciones y los datos previamente existentes en la PUCESA, a su vez con la finalidad de hacer la posterior implementación en los servidores de dicha institución un proceso relativamente fácil, rápido y que represente la menor cantidad y gravedad posible de problemas.

Por otro lado, como metodología de desarrollo de software para el presente trabajo se ha elegido Extreme Programming (XP), ya que esta metodología ofrece múltiples ventajas y se adapta especialmente a las necesidades detectadas en el DIP de la PUCESA. Específicamente, esta metodología fomenta la participación activa de los diferentes usuarios de la aplicación a desarrollarse conjuntamente con el o los programadores, permite responder de mejor manera a las necesidades reales de los usuarios ya que se mantiene un contacto directo y frecuente con ellos, y por último permite dividir el problema a solucionarse en problemas más pequeños y concretos que se pueden resolver de manera ágil y práctica, haciendo así mismo fácil su adaptación posterior a nuevas necesidades.

2.1. Información técnica básica

Tema: “Desarrollo de una Aplicación Web para la Administración de Procesos de Graduación de una Unidad de Postgrados”

Tipo de trabajo: Proyecto de Investigación y Desarrollo

Clasificación técnica del trabajo: Desarrollo

Líneas de Investigación, Innovación y Desarrollo

Principal: Ingeniería de Software y/o Plataformas Educativas

2.2. Descripción del problema

Los procesos de graduación que normalmente deben atender los directivos de las diferentes unidades de postgrados de las universidades ecuatorianas, en su gran mayoría suelen gestionarse aún de forma manual o, en el mejor de los casos, mediante el uso de herramientas individuales no relacionadas entre sí, que de alguna manera ayudan a automatizar parcialmente ciertas tareas correspondientes a dichos procesos administrativos. Si bien existen algunas instituciones que han desarrollado sus propias aplicaciones para la gestión de procesos de graduación de postgrados, tanto a nivel nacional como internacional ([12], [34]), la inexistencia de una herramienta integral y especializada que permita manejar adecuadamente todas las etapas de estos procesos desde el punto de vista administrativo, genera problemas graves que se reflejan principalmente en inconvenientes tales como pérdidas de tiempo, duplicación innecesaria de la información, y carencia de la información adecuada en el momento en que se la requiere.

En el caso particular de la PUCESA, que no es la excepción al problema que se acaba de mencionar, se desarrolló el sistema conocido como Academics [26], con el objeto de automatizar en la mayor medida posible muchos de los procesos administrativos propios de la actividad académica y estudiantil. Sin embargo, dicho sistema no contempla la automatización ni la optimización de los procesos de graduación de la unidad de postgrados de la PUCESA, por lo que el problema expuesto en el párrafo anterior persiste.

Por lo mencionado, surge la necesidad de elaborar una aplicación adicional compatible con Academics, que provea de todas las funciones requeridas para automatizar los procesos de

graduación de postgrados en cuestión, optimizando y dinamizando las actividades de la unidad correspondiente de la PUCESA. El producto final del presente trabajo será precisamente la aplicación necesaria para atender este requerimiento. Para este fin, las autoridades correspondientes de la PUCESA han manifestado su predisposición de facilitar las herramientas e información necesarias para que la mencionada aplicación sea desarrollada, implementada y evaluada convenientemente dentro de las instalaciones de esta Institución. Además, la aplicación web producto del presente trabajo se integrará con la aplicación a desarrollarse en [32], la misma que también será compatible con Academics, ya que ambas aplicaciones están relacionadas.

2.3. Preguntas básicas

¿Qué lo origina?

La inexistencia de una aplicación que permita administrar adecuadamente los procesos de graduación de postgrados.

¿Cuándo se origina?

En el momento en que los maestrantes culminan sus programas de postgrado y desean iniciar sus trámites de graduación.

2.4. Formulación de meta

Desarrollar una aplicación que permita automatizar y optimizar los procesos de graduación de una unidad de postgrados.

2.5. Objetivos

Objetivo General.

Desarrollar una aplicación web para la administración de procesos de graduación de una unidad de postgrados.

Objetivos Específicos.

1. Diagnosticar el nivel de eficiencia de los procesos de graduación de postgrados actuales para realizar los ajustes pertinentes, de manera tal que la aplicación a desarrollarse sirva como apoyo a la gerencia – dirección del DIP en la toma de decisiones.
2. Recopilar toda la información pertinente sobre las tecnologías existentes en la actualidad que puedan ser aplicadas para optimizar dichos procesos, incluyendo lo referente a Gestión Documental ([2], [13], [14]).
3. Desarrollar la aplicación tema del presente trabajo en base a los problemas detectados y la tecnología disponible, empleando la metodología XP (eXtreme Programming).

2.6. Delimitación funcional

Pregunta 1. ¿Qué será capaz de hacer el producto final del proyecto de titulación?

- La aplicación a desarrollarse mostrará a los maestrantes un diagrama detallado del proceso de graduación optimizado y/o rediseñado, donde las diferentes etapas del proceso se irán habilitando según se vayan registrando avances en las actividades respectivas.
- Esta aplicación también permitirá a los maestrantes presentar vía online la solicitud de aprobación de su tema tesis, con su respectiva propuesta de director.
- Además, la aplicación también permitirá también a los maestrantes presentar su plan de titulación y solicitar docente informante para el mismo.
- A través de esta aplicación, los docentes informantes podrán sugerir correcciones a sus planes de tesis asignados.
- Los maestrantes podrán presentar correcciones a su plan de titulación según las sugerencias emitidas por su docente informante asignado, conjuntamente con la aprobación de su director de tesis propuesto.
- La aplicación permitirá también a los maestrantes solicitar la aprobación definitiva de su plan de titulación, una vez cumplidos con los requisitos obligatorios para tal fin.
- Además, esta aplicación emitirá un certificado cuando un maestrante haya obtenido la aprobación definitiva de su plan de titulación y haya cumplido con todos los requisitos

legales pertinentes, a fin de acreditarlo como apto para matricularse y llenar las hojas de crédito para el desarrollo de su tesis.

- El personal del DIP podrá aceptar o designar definitivamente directores de tesis, docentes informantes y miembros de tribunal para las nuevas tesis de grado presentados por los maestrantes.
- Mediante la aplicación a desarrollarse, los maestrantes podrán enviar su trabajo final de tesis de grado para su revisión por parte de los docentes informantes y/o miembros de tribunal.
- Por otro lado, esta aplicación también permitirá a los maestrantes enviar las correcciones sugeridas por los docentes informantes y/o miembros de tribunal a su trabajo final de tesis.
- Los directores de tesis podrán emitir certificados e informes sobre correcciones aplicadas a los planes de tesis o trabajos finales de tesis por parte de sus maestrantes asignados.
- La aplicación también permitirá a los docentes informantes o miembros de tribunal registrar las calificaciones de sus trabajos de tesis de grado asignados, una vez aplicadas las correcciones propuestas.
- Además, esta aplicación permitirá a los maestrantes solicitar, vía online, fecha y hora para la defensa de su trabajo de tesis, una vez que se hayan cumplido con todos los requisitos previos para tal fin.
- Los maestrantes podrán solicitar, vía online, la entrega del trabajo final de tesis de grado en la biblioteca universitaria.
- Tanto el personal del DIP como los propios maestrantes podrán realizar un seguimiento detallado de sus actividades dentro de su proceso de graduación, pudiéndose acceder a un reporte del historial de un determinado estudiante, junto con su estado actual de desarrollo y calificaciones.
- La aplicación emitirá reportes estadísticos de temas aprobados, tesis finalizadas, asignación de directores y lectores de tesis, y porcentajes de maestrantes egresados vs. graduados.
- Esta aplicación será capaz de recuperar la información referente a las maestrías existentes en la PUCESA, desde las bases de datos del sistema Academics.

- Análogamente, la aplicación también podrá recuperar la información relativa a los estudiantes que se encuentran cursando actualmente alguna de las maestrías vigentes en la institución, desde el sistema Academics.
- Además, la aplicación soportará la validación de usuarios, ya sea desde dentro o fuera de la PUCESA, estableciendo grupos de usuarios, privilegios y niveles de acceso.
- Finalmente, esta aplicación brindará un medio de comunicación ágil a todos sus usuarios mediante un sistema de generación y envío de correos electrónicos.

Pregunta 2. ¿Qué no será capaz de hacer el producto final del proyecto de titulación?

- No aplica.

Capítulo 3

Marco Teórico

3.1. Definiciones y conceptos

3.1.1. Ingeniería de software

Según Sommerville [31, Pág. 6], *“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción del software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza...”*. De aquí podemos deducir que el desarrollo de un software es en realidad un proceso de ingeniería que permite diseñar y desarrollar un producto final que proveerá una solución real a un problema determinado, sea este de naturaleza específica o general.

Es necesario recalcar que la ingeniería de software atiende de manera integral a todo el proceso de desarrollo de un software y no únicamente a sus aspectos técnicos. Es decir, que la ingeniería de software provee a los ingenieros de herramientas y técnicas útiles para la óptima elaboración de un producto de software desde sus etapas más tempranas como son las de recopilación de datos y requerimientos del usuario final y el diseño del producto de software, hasta las etapas de pruebas una vez que el producto ha sido finalizado, así como de mantenimiento luego de que el mismo ha sido puesto en funcionamiento.

Debido a la naturaleza misma del software y al vertiginoso ritmo al que cambian los requerimientos de los usuarios de los productos de software, sean estos individuos o grandes corporaciones, la ingeniería de software debe adaptarse en sí misma a dichos cambios y contemplar la necesidad de crear productos lo suficientemente flexibles que puedan adaptarse a dichos cambios con la mayor facilidad posible. Una de las más notables tendencias que se ha observado en los últimos años en empresas dedicadas al desarrollo del software es justamente la de crear productos hasta cierto punto “genéricos” para posteriormente irlos adaptando a las necesidades específicas de sus potenciales usuarios una vez que estos deciden incorporar tales productos de software a sus procesos.

3.1.1.1. El Proceso del software

En términos generales, un proceso puede considerarse como un conjunto de actividades que deben ejecutarse de manera secuencial y ordenada para obtener un producto que sea útil para resolver algún tipo de problema o cubrir algún tipo de necesidad. El proceso del software no es otra cosa que la aplicación del concepto antes mencionado a la actividad del desarrollo de software. Sin embargo, esta actividad tiene sus particularidades muy específicas, por lo que el proceso del software debe adaptarse perfectamente a cada una de ellas. Una de estas particularidades es la flexibilidad, la misma que determina que el proceso del software propiamente dicho debe considerar únicamente las actividades generales y comunes al desarrollo de todo tipo de software. Así, cada integrante del equipo de desarrollo de software debe contar con cierta libertad para decidir las acciones a tomar para cumplir cabalmente su trabajo dentro de cada proyecto, ya que cada uno es diferente del anterior. En relación a este tema, Pressman [25, Pág. 12] nos dice que *“En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán”*.

3.1.1.2. La estructura del proceso del software

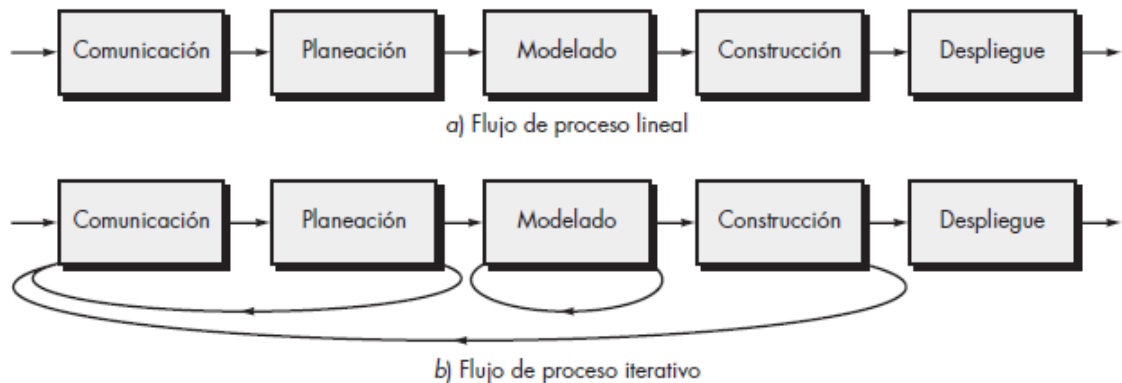
Atendiendo al principio de que el proceso del software solo debe considerar las actividades comunes a todo desarrollo de software en general, Pressman [25, Pág 12] considera que la estructura básica del proceso del software está basado en estas cinco actividades básicas:

- **Comunicación.** Debe darse de manera clara y directa entre todos los participantes de un proyecto de desarrollo de software, incluyendo el personal de la organización solicitante del mismo, y no solamente entre los técnicos desarrolladores y/o programadores.
- **Planeación.** Tiene por objetivo final la creación del llamado “plan del proyecto de software”, que a su vez define todas las actividades técnicas a realizar, riesgos, recursos requeridos y demás detalles técnicos del proyecto. En esta etapa se definen también detalles tales como lenguajes de programación, motores de bases de datos y metodología de desarrollo a utilizarse.

- **Modelado.** Al igual que en todas las ramas de la ingeniería, la ingeniería de software demanda la creación de un modelo o bosquejo preliminar del producto a desarrollarse, incluyendo diagramas de flujo, diseños de bases de datos, bosquejos de interfaces de usuario, etc. El objeto de esta actividad es que todos los miembros del equipo de desarrollo entiendan con el mayor detalle posible el problema a resolverse y la solución que se está planteando.
- **Construcción.** Consiste en la elaboración propiamente del producto final de software incluyendo la creación y población de bases de datos, generación del código, creación de webservices, diseño de reportes, pruebas y depuración del código. Para ejecutar exitosamente esta actividad, el equipo de desarrollo debe ceñirse estrictamente al bosquejo elaborado durante la etapa de modelado. Sin embargo, dicho modelo puede ir variando para adaptarse a nuevas necesidades que surjan durante el proceso o para incluir detalles que se haya omitido durante la etapa de planeación, atendiendo al principio de flexibilidad.
- **Despliegue.** Podría decirse que consiste en la entrega y puesta en marcha del producto de software a la persona o institución que haya solicitado su elaboración, la misma que deberá evaluar dicho producto y proporcionar una retroalimentación.

Según el tipo y tamaño de proyecto a desarrollarse y la metodología elegida, todo este ciclo de cinco pasos mencionado puede repetirse a lo largo de varias iteraciones, es decir que se iría construyendo el software por partes. Cada iteración permite obtener un producto de software más completo y que satisface de mejor manera las necesidades del usuario final del mismo. Es más, las cinco actividades ni siquiera deben ejecutarse secuencialmente con cada iteración, sino que por ejemplo el ciclo de comunicación – planeación puede repetirse varias veces antes de proceder con el modelado, que a su vez también puede repetirse varias veces antes de pasar a la fase de construcción. Cuando ya se llega a la fase de despliegue, las actividades anteriores pueden haberse repetido un sinnúmero de veces. Esto se puede entender más fácilmente observando el siguiente gráfico:

Figura 1: Flujo del proceso del software



Fuente: (Pressman, 2010)

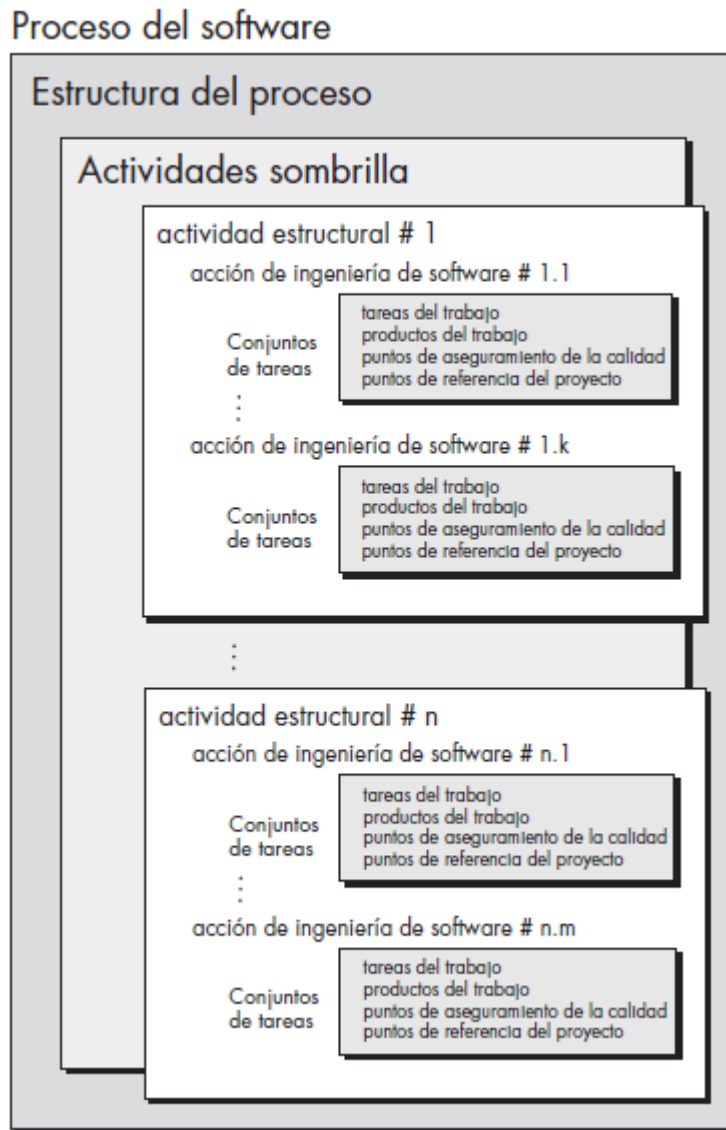
Sin embargo, además de estas cinco actividades elementales que constituyen la parte estructural del proceso del software, el mismo debe apoyarse también en otro conjunto de actividades que se conocen como “actividades tipo sombrilla”. Éstas pueden variar de un proyecto a otro pero, en términos generales, Pressman [25, pág 13] nos explica que las actividades tipo sombrilla suelen ser las siguientes:

- **Seguimiento y control del proyecto de software.** Su objetivo fundamental es el de evaluar que el producto de software elaborado esté acorde con el plan y el bosquejo obtenidos durante las etapas de planeación y modelado, y realizar las correcciones requeridas en caso de ser necesario.
- **Administración del riesgo.** Permite identificar los posibles riesgos que se pueden presentar durante todas las etapas del desarrollo del software, y tomar las acciones necesarias para evitar o neutralizar dichos riesgos.
- **Aseguramiento de la calidad del software.** Permite establecer ciertos parámetros y lineamientos para garantizar que el producto final de software es de calidad y cumple con todos los requerimientos para los que inicialmente fue creado.
- **Revisiones técnicas.** Proporciona igualmente parámetros que permiten localizar y corregir errores de manera oportuna durante cada etapa del desarrollo, evitando así que pasen de una actividad o iteración a otra.

- **Medición.** Permite evaluar numéricamente cada actividad, tarea o resultado a lo largo del proceso de desarrollo de software, a fin de identificar más fácilmente las posibles falencias que puedan presentarse en cualquier parte del proyecto y corregirlas a tiempo.
- **Administración de la configuración del software.** Permite determinar los parámetros que hay que considerar a fin de que el software sea adaptable a las nuevas necesidades.
- **Administración de la reutilización.** Ayuda a obtener un producto de software modular y basado en componentes, cuyo código se pueda reutilizar fácilmente de forma total o parcial.
- **Preparación y producción del producto de trabajo.** Establece los mecanismos necesarios para obtener de manera correcta y óptima el producto de software que se busca, respetando todos los demás lineamientos que se ha mencionado.

En el siguiente gráfico se puede observar claramente cómo las “actividades tipo sombrilla” deben estar presentes como un apoyo durante todo el ciclo de las cinco actividades estructurales mencionadas al inicio:

Figura 2: Estructura del proceso del software



Fuente: (Pressman, 2010)

3.1.2. Metodologías de desarrollo de software

Según Avison y Fitzgerald [3], *“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de*

las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo”.

De esta definición podemos deducir que todo equipo de desarrollo de software debe definir claramente la metodología que va a seguir durante la ejecución de un proyecto, una vez que se han establecido todas las características técnicas del mismo como son requerimientos del usuario, tamaño del proyecto, plazos de entrega, presupuestos y demás, ya que esta metodología será la base y apoyo fundamental dentro del subsiguiente proceso de desarrollo del software que fue analizado en el apartado anterior.

La historia de las metodologías de desarrollo de software se remontan a las década de 1960, donde en los países desarrollados, principalmente en Estados Unidos, existía gran demanda de software personalizado por parte de las grandes corporaciones a nivel nacional que deseaban agilizar sus procesos y procedimientos a todo nivel, tanto de producción como de comercialización. Este “boom” fue tan notable que las escasas empresas que en ese tiempo se dedicaban al desarrollo de software se vieron obligadas a comenzar a pensar en mecanismos para organizar y tecnificar el desarrollo del software, que en ese momento aún se realizaba de forma más bien empírica, con la finalidad de lograr mejores tiempos de producción, mayores niveles de satisfacción en sus clientes y, en general, estándares de producción más altos dentro de la naciente industria de la producción de software.

3.1.2.1. Metodologías tradicionales vs. metodologías ágiles

Una vez que comenzaron a aparecer las primeras metodologías de desarrollo de software, lo que se intentó hacer fue crear un modelo de desarrollo rígido con fases bien definidas a las que deberían amoldarse todos los proyectos de desarrollo de software, independientemente de su naturaleza, requerimientos o características específicas. De hecho se intentó planificar detalladamente todas las fases del desarrollo y sus correspondientes actividades, intentando desde un comienzo documentar exhaustivamente todo el proceso. Así nacieron las metodologías de desarrollo que hoy en día se conocen como “metodologías tradicionales”.

Desde entonces hasta la actualidad se han creado muchas metodologías de desarrollo de software, algunas de ellas orientadas a ciertos tipos específicos de proyectos, e incluso se ha llegado a

clasificarlas atendiendo a diferentes criterios, como por ejemplo su filosofía de desarrollo. Es así que hoy en día podemos hablar de metodologías ágiles de desarrollo, en contraste a las ya mencionadas metodologías tradicionales. Pero esta clasificación no se ha hecho empíricamente, sino que está basada en ciertos modelos que a su vez determinan la forma y el orden en el que se ejecutan las actividades propias del desarrollo de software.

Óscar Tinoco Gómez [16] se refiere a estos modelos como “modelos de proceso” y cita como los más comunes a los siguientes:

Modelo Secuencial. Trata de crear una herramienta de software completamente funcional de una sola vez, analizando desde el inicio todos los requerimientos y demás pormenores del problema a resolver, y ejecutando cada una de las actividades estructurales del proceso del software una sola vez por todo el sistema a construir. El ejemplo por excelencia de una metodología que se basa en este modelo secuencias es la metodología en cascada.

Desarrollo incremental. Al igual que en modelo anterior, todos los requerimientos del proceso del software son determinados desde el inicio del proyecto, sin embargo, en este modelo tales requerimientos son organizados en “incrementos”. Podríamos pensar en ellos como si fueran diferentes “capas” del proyecto a desarrollarse. La finalidad fundamental de esta “subdivisión” del proyecto es mejorar los tiempos de entrega.

Desarrollo iterativo. Tal como en el modelo anterior, se pretende hacer una división del proyecto en partes más pequeñas conocidas como “iteraciones”, para cada una de las cuales se generará un producto entregable completamente funcional. Una vez que una iteración ha sido finalizada y entregada a satisfacción del cliente, simplemente se prosigue con la siguiente, y así sucesivamente hasta finalizar el proyecto. En cierto modo, podría decirse que éste modelo se centra en dividir un proyecto grande en varios proyectos más pequeños con el objetivo de facilitar el trabajo al equipo de desarrollo y mejorar los tiempos de entrega. Este modelo se conoce también como “prototipado”.

Modelo en espiral. Podría considerarse a este modelo como una combinación de los dos modelos mencionados anteriormente: el incremental y el iterativo. Permite organizar los requerimientos del proyecto definidos inicialmente en incrementos, y a la vez, se va repitiendo el ciclo de actividades del proceso del software para cada incremento en diferentes iteraciones, con la finalidad de obtener un producto final en óptimas condiciones y en el menor tiempo posible.

Las metodologías ágiles tienen su origen en febrero de 2001 en Utah (Estados Unidos), donde se celebró un encuentro entre muchos expertos dentro del mundo del desarrollo de software, varios de ellos incluso creadores de algunas de las metodologías ya existentes hasta ese entonces. El objetivo de tal encuentro fue justamente idear nuevas técnicas de desarrollo de software que respondieran efectivamente a las nuevas exigencias de los consumidores de software, en especial empresas e instituciones de todos los tamaños, quienes requerían que sus productos de software sean entregados con mayor prontitud, y sobre todo que exista la flexibilidad de adaptarse fácilmente a los cambios que se vayan dando en los requerimientos, incluso durante el propio proceso de desarrollo de software. Ninguna de estas dos nuevas exigencias podían ser satisfechas mediante las metodologías tradicionales de desarrollo cuya rigidez, burocracia y excesiva demanda de documentación las hacían poco prácticas e inadecuadas para responder a las nuevas necesidades de un mundo que se movía mucho más rápido.

De dicho encuentro nació la denominada “Agile Alliance” (en español “Alianza Ágil”), que se constituyó como una organización sin fines de lucro cuya finalidad era la de difundir los nuevos conceptos de desarrollo ágil de software a todos los individuos y organizaciones inmersas dentro del mundo del desarrollo de software. Esta alianza a su vez creó el “Agile Manifesto” (Manifiesto Ágil) [7], cuya función era la de establecer por escrito las bases de esta nueva filosofía de desarrollo.

J. H. Canós, en su libro titulado “Metodologías Ágiles en el Desarrollo de Software” [11], nos indica que según el Manifiesto Ágil, lo que se valora en estas nuevas metodologías de desarrollo de software es lo siguiente:

Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Tradicionalmente, se establecía el entorno de trabajo y sus herramientas y se esperaba que el equipo de trabajo se adapte a las mismas, pero ahora se da prioridad a la gente, es decir, primero se conforma el equipo de trabajo y se espera que ellos mismos sean quienes definan su entorno de desarrollo.

Desarrollar software que funciona más que conseguir una buena documentación. Las metodologías ágiles de desarrollo son mucho más prácticas y por tanto se centran en la funcionalidad del producto final de software, creando únicamente la documentación que sea estrictamente necesaria. Esto en contraposición a las metodologías tradicionales donde la documentación era prioritaria.

La colaboración con el cliente más que la negociación de un contrato. El común denominador de las metodologías ágiles de desarrollo es la participación del cliente o usuario final del software en cuestión como parte del equipo de desarrollo, lo cual asegura que el proyecto marche en la dirección correcta y sea 100% satisfactorio.

Responder a los cambios más que estrictamente seguir un plan. Como se mencionó anteriormente, uno de los aspectos que más impulsó la búsqueda de nuevas metodologías de desarrollo de software fue la creciente necesidad de responder a los vertiginosos cambios en las necesidades de software de las organizaciones modernas. Por tanto, la flexibilidad debe ser pilar fundamental dentro de las metodologías ágiles de desarrollo de software. Dichos cambios pueden darse incluso durante la misma etapa de desarrollo del software, pero al haber una colaboración permanente entre el equipo de desarrollo y el cliente, y al existir flexibilidad en los proceso de desarrollo, debe ser relativamente fácil y rápido responder a estos cambios, sin tener que esperarse a finalizar el producto de software con su planificación inicial para después regresar a hacer los cambios que sean requeridos como normalmente sucedería con las metodologías tradicionales.

Atendiendo a estos valores surgieron los doce principios que quedaron plasmados en el Manifiesto Ágil, los mismos que según cita Canós [11], son los siguientes:

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.*
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*
- III. Entregar frecuentemente software que funcione desde un par de semanas a par de meses, con el menor intervalo de tiempo posible entre entregas.*
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.*
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.*
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.*
- VII. El software que funciona es la medida principal de progreso.*
- VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.*
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.*

- X. *La simplicidad es esencial.*
- XI. *Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.*
- XII. *En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.*

En la siguiente tabla, se puede observar algunas de las diferencias más notables existentes entre la filosofía de desarrollo de las metodologías tradicionales frente a la de las metodologías ágiles:

Tabla 1: Diferencias entre metodologías ágiles y tradicionales

| Metodologías Ágiles | Metodologías Tradicionales |
|--|--|
| Basadas en heurísticas provenientes de prácticas de producción de código | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |
| Especialmente preparados para cambios durante el proyecto | Cierta resistencia a los cambios |
| Impuestas internamente (por el equipo) | Impuestas externamente |
| Proceso menos controlado, con pocos principios | Proceso mucho más controlado, con numerosas políticas/normas |
| No existe contrato tradicional o al menos es bastante flexible | Existe un contrato prefijado |
| El cliente es parte del equipo de desarrollo | El cliente interactúa con el equipo de desarrollo mediante reuniones |
| Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio | Grupos grandes y posiblemente distribuidos |
| Pocos artefactos | Más artefactos |
| Pocos roles | Más roles |
| Menos énfasis en la arquitectura del software | La arquitectura del software es esencial y se expresa mediante modelos |

Fuente: (Canós, 2003)

3.1.2.2. Metodología XP (Extreme Programming)

Extreme Programming o XP es una de las más versátiles y recientes metodologías de desarrollo de software, perteneciente al grupo de las metodologías ágiles. La persona considerada su padre y creador es Kent Beck, quien lo concibió y perfeccionó a partir del año 2000, siendo uno de los integrantes de la Alianza Ágil, y quien a su respecto ha publicado múltiples libros ([4], [6] y [5]) que son de excepcional interés para cualquier desarrollador que busque adentrarse en esta metodología. Básicamente, XP fomenta la creación de un ambiente de trabajo agradable donde se da especial énfasis a las buenas relaciones interpersonales dentro del equipo de trabajo. Al ser una metodología ágil y cumplir con todos sus principios básicos que ya se ha analizado, XP prioriza la efectividad y rapidez en el desarrollo de un producto de software por sobre la rigidez y excesiva documentación. De igual forma, dentro de XP es de vital importancia la participación activa del cliente o usuario final del producto como parte del equipo de desarrollo. Al ser una metodología sumamente flexible se considera especialmente útil para proyectos donde los requerimientos iniciales no son del todo claros y/o pueden variar o aumentar en un período relativamente corto de tiempo. Además, XP se enmarca dentro del modelo de proceso iterativo, ya que su filosofía justamente consiste en el perfeccionamiento continuo de cada porción del producto de software hasta lograr la total satisfacción de la parte solicitante.

Según explica Canós, XP se basa en ciertos principios fundamentales que son:

- Realimentación continua entre el cliente y el equipo de desarrollo
- Comunicación fluida entre todos los participantes
- Simplicidad en las soluciones implementadas
- Coraje para enfrentar los cambios

3.1.2.2.1. Elementos básicos de Extreme Programming

Independientemente del tipo de proyecto a ejecutarse y sus características particulares, la metodología de desarrollo Extreme Programming se apoya en ciertos elementos comunes que son las historias de usuario, los roles y el proceso. Se da a continuación una breve explicación sobre cada uno de ellos.

Historias de usuario. Dentro de XP, consisten el elemento de mayor apoyo sobre todo en la etapa inicial de cualquier proyecto, ya que permiten recopilar los requerimientos del o de los usuarios del software de manera sencilla, precisa y práctica. Básicamente, son tarjetas de papel que se llenan mediante entrevistas directas entre los miembros del equipo y uno o más representantes del cliente. Algunos de los datos que contienen son la fecha, el tipo de actividad, el número de historia, la prioridad tanto técnica como del cliente, nivel de riesgo, y opcionalmente un número de otra historia con la que esté relacionada.

En cada historia de usuario se trata de analizar un punto específico del problema general a resolver, para el que idealmente los programadores deben crear una solución en una iteración, es decir típicamente en un tiempo de una a tres semanas.

Cada historia de usuario puede ser de diferente tipo, como nueva, corrección o mejora. Si es de tipo nueva, se refiere a un punto del problema que se está analizando por primera vez, mientras que si es de tipo corrección o mejora, significa que se está tratando de optimizar una solución ya creada en una nueva iteración, para responder a nuevos requerimientos del cliente, o a requerimientos que han cambiado recientemente, atendiendo al principio de flexibilidad.

Figura 3: Ejemplo de historia de usuario XP

| Historia de Usuario | |
|--|---|
| Número: 3 | Nombre: Tabla de Magnitudes del Balance. |
| Usuario: Especialista | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1.5 |
| Riesgo en Desarrollo: Alta | Puntos Reales: 1.2 |
| Descripción: A partir de las mediciones anuales insertadas en la rejilla de datos de la ventana de trabajo se calcularán las magnitudes del balance y se mostrarán en forma de tabla, agrupadas y promediadas por años. La tabla de magnitudes se mostrará en una nueva ventana de trabajo que presentará una barra de herramientas con las funciones de la categoría Impresión, Gráficos y Tabla de Probabilidades. | |
| Observaciones: La Tabla de Magnitudes del Balance podrá generarse a partir de las ventanas de trabajo Método de Balance e Información General de Niveles, utilizando los valores de la variable Nivel y Precipitaciones. Se podrá calcular los valores de las magnitudes del balance del Método de Balance a diferentes probabilidades. | |

Fuente: (metodología XP, 2005)

Roles. Son las funciones que desempeñan cada uno de los miembros integrantes del equipo de desarrollo de (que no es lo mismo que el equipo de programadores). Todos y cada uno de los integrantes del mismo deben estar completamente conscientes de su rol y de lo que este representa dentro del proyecto de desarrollo de software. Básicamente, el equipo de desarrollo está conformado por programadores, encargados de pruebas (o tester), encargados de seguimiento (o tracker), el entrenador (o coach), el gerente del proyecto, el cliente o uno o varios de sus representantes, y en ciertas ocasiones especiales un consultor, que es un miembro externo al equipo con conocimientos muy específicos sobre algún tema relacionado al proyecto, y que puede aportar dicho conocimiento para asistir a la solución del problema.

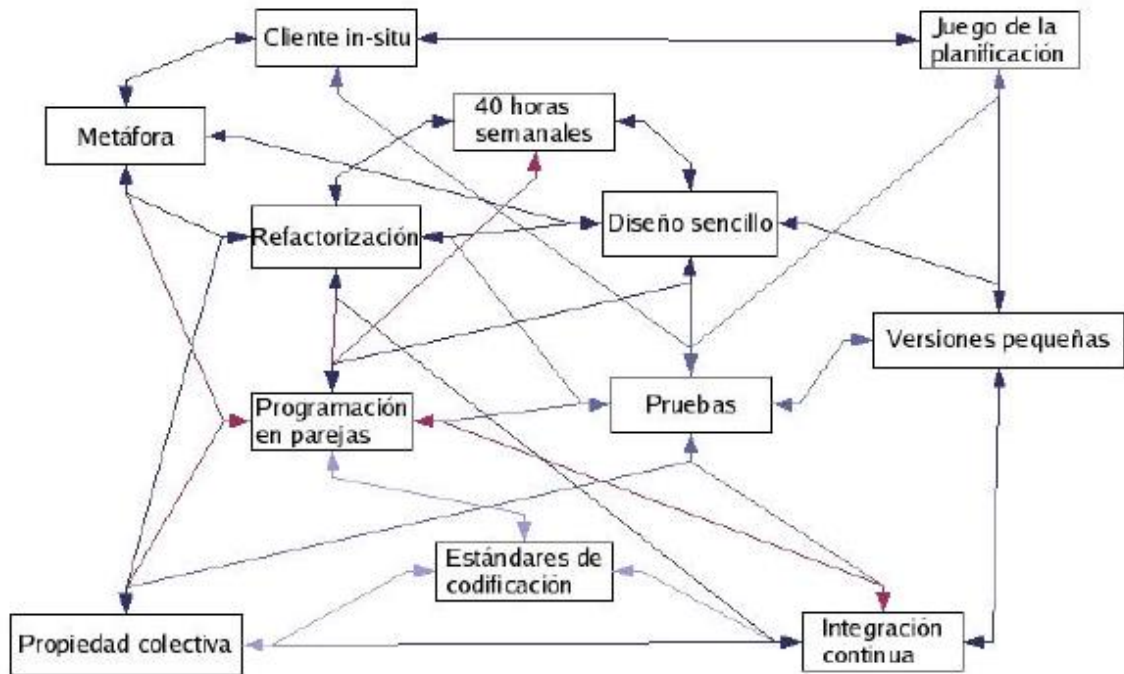
Proceso. Consiste en el ciclo de trabajo propiamente definido por XP, el mismo que consiste de cuatro pasos que se repiten durante todas las iteraciones requeridas para resolver el problema planteado. Estos pasos son:

1. Definición del valor o funcionalidad a implementar por parte del cliente,
2. Estimación de tiempos y esfuerzo necesarios por parte del programador,
3. Selección de prioridades de desarrollo por parte del cliente, quien debe tomar en cuenta todo tipo de restricciones a las que esté sometido, y
4. Desarrollo del valor o funcionalidad por parte del programador.

Además, hay que tener en cuenta que, según Canós [11], *“El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto”*. Es justamente en la fase de iteraciones donde se repiten los cuatro pasos mencionados más arriba hasta cubrir todos los puntos de interés del cliente.

En todo caso, es también muy importante entender que dentro de XP, todas las prácticas y principios deben apoyarse y estar mutuamente relacionadas entre sí, como se ilustra en la siguiente figura:

Figura 4: Las prácticas y principios de XP deben reforzarse entre sí



Fuente: (Canós, 2003)

3.1.3. Internet y las Aplicaciones WEB

3.1.3.1. Arquitectura de la World Wide Web

Internet se originó en Estados Unidos durante la guerra fría como un proyecto militar innovador de intercambio ágil de información y datos. Posteriormente, la comunidad científica y las universidades comenzaron a utilizarlo y se fue haciendo cada vez más popular, primero dentro de los Estados Unidos y luego entre comunidades científicas internacionales, como un sistema distribuido de información [1]. Tal es así que en el año de 1989, Tim Berners-Lee [8] presentaba por primera vez la propuesta de crear lo que él mismo llamó como “World-Wide Web”, que se traduciría al español como la “Telaraña Mundial o Global”. En ese entonces, Berners-Lee trabajaba como miembro de la Organización Europea para la Investigación Nuclear (CERN por sus siglas en francés), donde justamente presentó la propuesta como un medio de intercambio de información y datos entre

equipos de investigación científica que se encontraban geográficamente dispersos a lo largo de varios países.

Berners-Lee[8] manifestaba que *“En el CERN, una diversidad de datos está ya disponible: informes, datos experimentales, datos personales, listas de direcciones de correo electrónico, documentación informática, documentación experimental y muchos otros conjuntos de datos están girando continuamente en discos de ordenadores. Es sin embargo imposible ‘saltar’ de un conjunto a otro de una manera automática: una vez has encontrado que el nombre de Joe Bloggs se lista en una descripción incompleta de algún software en línea, no se encuentra directamente su dirección actual de correo electrónico. Usualmente, tendrás que utilizar un método de consulta distinto en un ordenador distinto con un interface distinto. Una vez has localizado la información, es difícil guardar sus conexiones o hacer una anotación privada que puedas después encontrar rápidamente”*. Berners-Lee fue la primera persona que logró establecer efectivamente la comunicación entre un equipo cliente y un servidor mediante el protocolo Hypertext Transfer Protocol (HTTP por sus siglas en inglés) en el año de 1989. De allí, que se considera a Tim Berners-Lee como “el padre de la web”.

Finalmente, cuando la World-Wide Web estaba completamente lista y entró en funcionamiento en el CERN en 1991, se conectó a Internet y desde entonces ha experimentado un crecimiento vertiginoso, al punto de que actualmente casi toda institución, organización, universidad o empresa de tamaño mediano y grande cuenta con su sitio web donde el público en general puede ingresar a solicitar información o documentación, y las que no lo hacen se consideran obsoletas y, sobre todo en el caso de las empresas, se considera que su competitividad se ve muy reducida.

3.1.3.1.1. Hipertexto e hipermedia

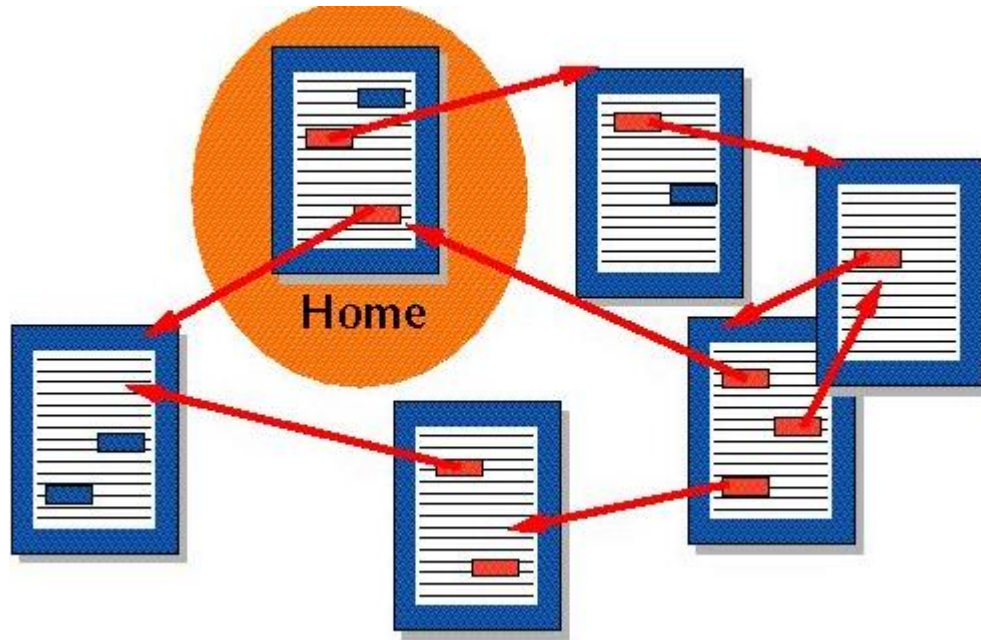
Ya en 1945, Vannevar Bush [10], como jefe del Departamento de Investigación y Desarrollo Científico de Estados Unidos, manifestaba que *“La suma de la experiencia humana se está expandiendo a un ritmo prodigioso y los medios que utilizamos para seguir el hilo a través del consiguiente laberinto de ítems momentáneamente importantes son los mismos que usábamos en los días de los barcos de vela”*. Esto en referencia la diversidad de temas en los que se producían grandes avances y descubrimientos científicos día a día, sin embargo, no era posible “saltar” de un tema a otro relacionado, y de este a otro directamente, sino que, con los rudimentarios medios de administración de la información existentes en ese momento, únicamente se podía pasar de un tema al siguiente o al anterior en una sucesión lineal. Bush consideraba que esto era un gran impedimento para la

investigación científica, ya que los temas y artículos relacionados debían ser fácilmente accesibles formando más bien una especie de “malla”, ya que de lo contrario el proceso se hacía demasiado lento y tedioso.

Sin contar en su momento con los recursos tecnológicos necesarios, Bush ya ideó lo que llamó “Memex”, un sistema que a futuro permitiría enlazar cualquier artículo, libro, revista o cualquier porción de información científica con su contenido relacionado de manera relativamente sencilla. Bush consideraba que la manera en la que se organiza la información naturalmente en la mente humana es en esta forma de malla, y no de manera lineal, de ahí lo lento y complicado de este segundo modelo.

Sin embargo, no fue hasta 1965 que Theodor Holn Nelson [22], haciendo uso de la ideas de Bush y de su “Memex”, el cual físicamente nunca se llegó a construir, introdujo por primera vez el término de “hipertexto” (hypertext en inglés). Textualmente, Nelson conceptualizó el hipertexto como *“Un cuerpo de material escrito o pictórico interconectado en una forma compleja que no puede ser representado en forma conveniente haciendo uso del papel”*. Es decir, que el hipertexto presenta la idea de que un lector puede “saltar” libremente entre los diferentes nodos de una gran malla de información, compuesta por un conjunto de libros, artículos, publicaciones o cualquier otra forma de material textual, relacionados entre sí.

Figura 5: Representación simple del hipertexto



Fuente: (metodología XP, 2005)

Finalmente, la idea evolucionó al punto de que los nodos de esta gran malla de información no necesariamente tenían que ser documentos textuales, sino que podían incluir otros tipos de elementos como gráficos, vídeos, sonidos, animaciones y otros. Todo este tipo de información “audiovisual” ya era conocida hace tiempo en el mundo de la informática como “multimedia”. De allí que, la inclusión de todo este tipo de elementos como parte de los nodos a los que se podía saltar, que hacían la experiencia del lector o investigador mucho más rica, dio origen al término “hipermedia”.

Hoy en día, la gran mayoría de la información que se encuentra presente en Internet es de carácter audiovisual, en contraposición de los documentos puramente textuales que solían conformar la red mundial en sus inicios. Las modernas herramientas de navegación en Internet como son HTML5 y CSS3, así como los navegadores web, ponen especial énfasis en la hipermedia al facilitar la publicación de elementos audiovisuales así como también la exploración de la información de un nodo a otro, de acuerdo al interés o a la necesidad de cada usuario en particular.

3.1.3.1.2. Protocolos y estándares de la World Wide Web

Según Malkin [18], un protocolo es *“Una descripción formal de los formatos de los mensajes y las reglas que deben seguir dos ordenadores para intercambiar dichos mensajes. Los protocolos pueden describir detalles de bajo nivel de los interfaces de máquina a máquina (por ejemplo, el orden en el cual deben enviarse bits y bytes a través de un cable) o intercambios de alto nivel entre programas (por ejemplo, la forma en que dos programas transfieren un fichero a través de la Internet)”*. Atendiendo a este concepto se puede ver que la implementación de un protocolo es la solución ideal para lograr comunicar equipos de diferentes tipos, marcas, modelos y capacidades, y que funcionan bajo diferentes sistema operativos. Al existir reglas claras de comunicación que todos los equipos deben seguir, independientemente de su hardware y software, es posible interconectar computadores de diferentes tipos, en diferentes países y con sistemas operativos corriendo en diferentes idiomas, prácticamente sin inconvenientes.

Al ser la World-Wide Web una red global, que a su vez interconecta redes de todo el mundo, se hace de vital importancia la implementación y el mantenimiento de protocolos de alto y bajo nivel a los que todos estos equipos deben sujetarse, de modo que la comunicación se hace no solo posible sino también fluida, con total independencia de hardware y software. A continuación se realiza un breve análisis de algunos de los protocolos y estándares de la World-Wide Web.

3.1.3.1.2.1. HTTP

Uno de los primeros protocolos que surgió para la World-Wide-Web fue el ya mencionado HTTP (Hypertext Transfer Protocol), como una respuesta a la necesidad de integrarla al ya existente Internet. Como se puede deducir de su nombre, HTTP permite la transferencia de hipertextos de un computador a otro. De allí nace también el concepto de navegación en Internet, que no es otra cosa que la transferencia de hipertextos desde un computador servidor conectado a Internet hacia un computador cliente corriendo cualquier sistema operativo, también conectado a Internet, usando HTTP. Si bien en un inicio las páginas web no eran otra cosa sino hipertextos simples que permitían saltar de una página a otra con una facilidad sin precedentes, hoy en día el protocolo HTTP se usa para transmitir no solo texto e hipertexto sino también hipermedia incluyendo elementos como son videos, gráficos, animaciones, audio y aplicaciones.

HTTP es, por tanto, el protocolo de alto nivel de la World-Wide Web por excelencia, y cuya principal función es la de permitir el intercambio de mensajes entre equipos clientes y servidores conectados Internet. Dentro de HTTP prevalecen la ligereza y la generalidad como características principales, siendo un protocolo diseñado para transmitir cualquier tipo de información a través de Internet siguiendo el estándar MIME. De igual forma, HTTP es un protocolo no diseñado para mantener la conexión entre diferentes transacciones, por lo que cada una es independiente de la anterior.

3.1.3.1.2.2. HTTPS

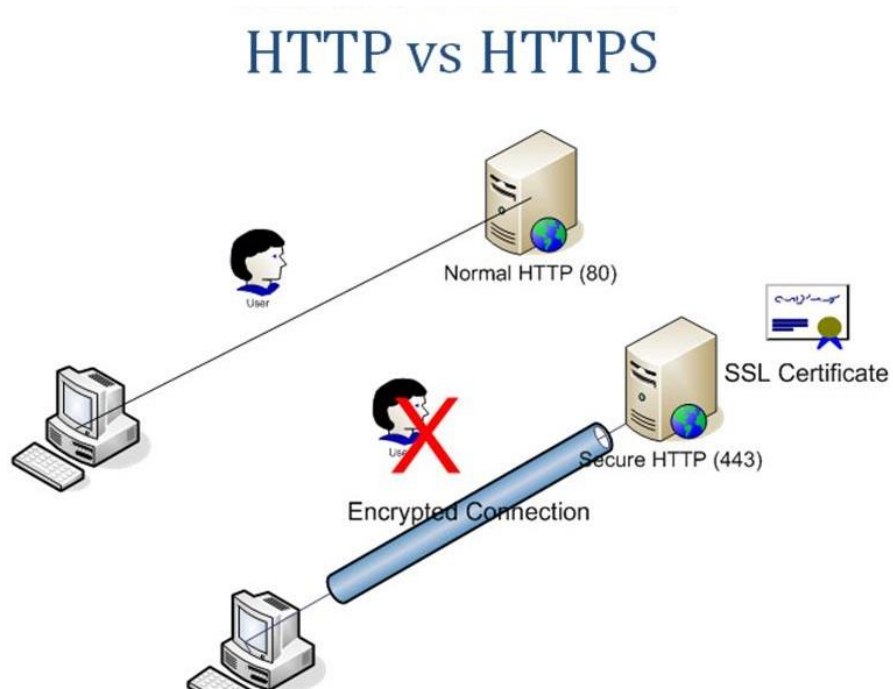
Una vez que la World-Wide Web sobre Internet se comenzó a popularizar y las páginas web se fueron haciendo más complejas, ofreciendo mayores posibilidades a sus usuarios, las grandes corporaciones tales como bancos, comercios, industrias, emisoras de tarjetas de crédito y otras de similar magnitud se fueron integrando a esta gran red mundial, tratando de aprovechar sus beneficios para ofrecer innovadores servicios a sus clientes o usuarios, como la banca en línea, y así volverse más competitivas. Sin embargo, esto a la par dio origen a los primeros delitos cibernéticos, ya que muchos de los especialistas del medio se dieron cuenta que podían utilizar las vulnerabilidades de los sistemas informáticos en su favor para robar números de cuentas bancarias y tarjetas de crédito, información personal, contraseñas y demás para sustraer dinero de las cuentas de los clientes mediante múltiples técnicas.

Utilizando HTTP, la información que viaja por Internet es en realidad información pública que circula entre diferentes puntos ubicados muchas veces en diferentes países. Cualquier persona con los conocimientos y herramientas adecuadas puede interceptar esta información y hacer una copia de la misma, para posteriormente hacer uso de ella en beneficio propio. Por esta razón surgió la necesidad de asegurar que la información que viaja por Internet lo haga de forma segura y que sea imposible de descifrar incluso si logra ser interceptada. Así surgió HTTPS (Hypertext Transfer Protocol Secure) [21], el mismo que básicamente encripta la información que se transmite por Internet, siendo prácticamente imposible de descifrar para cualquier persona que la intercepte. La información sale encriptada del punto de origen, y puede ser desencriptada únicamente en el punto de destino, una vez que se comprueba que es el receptor legítimo.

Este protocolo fue creado en 1992 por Netscape Communications y fue diseñado originalmente para transmitir información sobre SSL (Secure Sockets Layer), aunque hoy en día se utiliza su

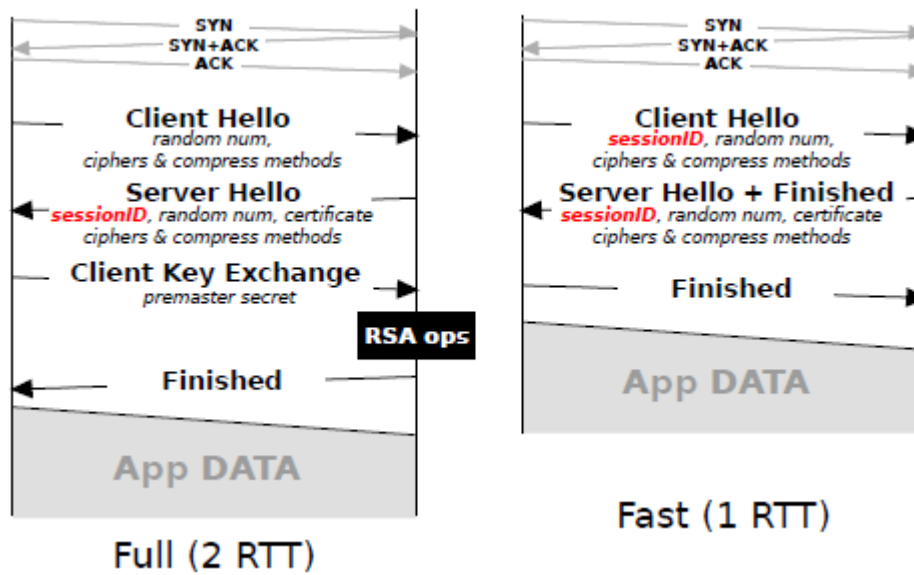
sucesor TLS (Transport Layer Security). En el siguiente gráfico se puede visualizar la diferencia básica entre HTTP y su versión segura HTTPS:

Figura 6: Diferencia básica entre HTTP y HTTPS



Fuente: (Internet Protocols, 2003)

Figura 7: Negociación SSL/TLS



Fuente: (Naylor, 2014)

3.1.3.1.2.3. HTML

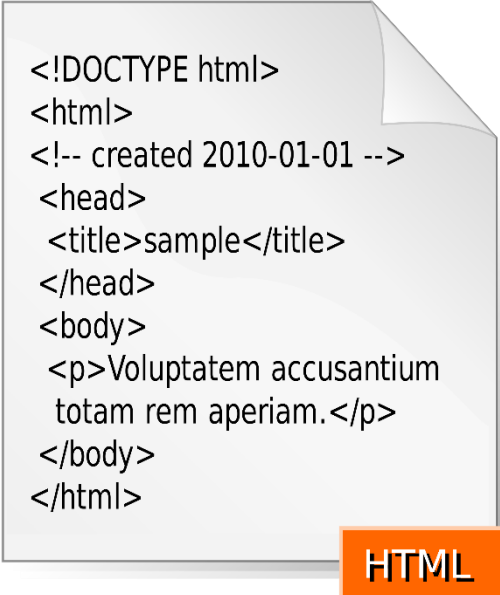
El uso de un protocolo como HTTP para la transferencia de hipertextos e hipermedia a través de Internet, sin embargo, requería de un lenguaje estándar que pudiera ser interpretado tanto por los computadores que enviaban así como por los que recibían información. En otras palabras, los mensajes que circulaban a través de Internet debían ceñirse a ciertas reglas y estándares para que pudieran ser interpretados por cualquier computador que los recibiera y fueran capaces de desplegar dicha información correctamente, ya que de otra forma la comunicación seguiría siendo imposible. Así nació el Hypertext Markup Language (HTML), que no es otra cosa que un lenguaje que permite la elaboración y el intercambio de páginas web, y cuyo mantenimiento se encuentra hoy en día a cargo de la World Wide Web Consortium (W3C), institución que se encarga no solo de HTML sino de la mayoría de estándares relacionados a la World Wide Web.

HTML fue originalmente inventado por Tim Berners-Lee en 1991 y consiste en un conjunto de “marcas” o etiquetas que permiten especificar diferentes tipos de elementos como texto, hipertexto, imágenes, vídeos, scripts y aplicaciones, si bien el conjunto original admitido por HTML era bastante modesto en relación a los estándares actuales. Sin embargo, con el pasar de los años el lenguaje fue evolucionando con la finalidad de responder a nuevas necesidades, y así en el año 1995 ya se publicaba la versión 2.0 [9]. Hoy en día, la versión vigente de HTML es la 5.0 y permite la inclusión de

muchos elementos innovadores que permiten crear páginas y aplicaciones web muy ricas tanto visual como funcionalmente. HTML es un lenguaje textual que no puede incluir directamente objetos multimedia en su contenido, solo referencias a la ubicación de dichos objetos en los servidores web que emiten las páginas. Los programas clientes, navegadores web en su gran mayoría, reciben este texto y lo interpretan, creando los efectos visuales y la estructura de las páginas web localmente en el computador cliente. Los elementos multimedia son entonces importados directamente desde el servidor y colocados en la página web a desplegarse, todo a través del protocolo HTTP.

En el siguiente gráfico se puede ver la estructura de una página web muy básica en su formato original en HTML:

Figura 8: Estructura HTML de una página web básica



```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
</body>
</html>
```

HTML

Fuente: (Internet Protocols, 2003)

Como se puede observar en el gráfico, las palabras que se encuentran entre los símbolos “<” y “>” son las etiquetas y sirven para especificar el tipo de elemento o sección de la página a la que hace referencia.

3.1.3.1.2.4. CSS

Otro de los lenguajes utilizados para el correcto despliegue y presentación de las páginas web en los computadores cliente es Cascading Style Sheets (CSS). Básicamente, surgió ante la utilidad de separar la estructura o contenido de una página web de su presentación. CSS es también mantenido por la W3C, y su origen se remonta al año 1995 donde se publicó su primera versión (CSS1), la misma que básicamente permitía definir propiedades de fuentes como tipo de letra, tamaño y color, colores de todo tipo de elementos como texto y bordes, alineación de todo tipo de elementos, espaciado y márgenes en tablas y sus celdas, y algunas propiedades de listas.

Posteriormente, en el año 1998 surgió CSS2 con la finalidad de corregir algunos errores de CSS1, aumentar su funcionalidad y eliminar algunos de sus elementos que resultaron muy poco utilizados. Esta nueva versión implementó algo más de funcionalidad visual para los elementos de las páginas web, como sombras y texto bidireccional. Al poco tiempo apareció CSS 2.1 con la misma finalidad de corregir errores e introducir mejoras con respecto a CSS2, sin embargo, no fue sino hasta junio del 2011 cuando la W3C lo publicó finalmente como recomendación oficial.

Finalmente, a la par que fue publicado CSS2 comenzó el desarrollo de CSS3, el mismo que se caracteriza por la inclusión de muchos módulos que intentan por un lado conservar la compatibilidad con las versiones anteriores de CSS, y por otro, mejorar su funcionalidad y alcance para responder a los nuevos requerimientos impuestos por las siempre cambiantes páginas y aplicaciones web. Los primeros módulos de CSS3 fueron publicados en el año 1999 y para fines del 2011 había alrededor de 50 módulos publicados. CSS3 es la versión vigente que se utiliza hasta hoy en día y algunos de los módulos que incluye son consulta de medios, selectores, color, fondos y colores, espacios de nombres y diseños multicolumna.

CSS guarda total compatibilidad con HTML y hoy en día suele utilizarle la combinación HTML5 y CSS3 para el desarrollo tanto de páginas web estáticas como páginas web dinámicas y aplicaciones web. Las especificaciones CSS para una página web pueden incluirse dentro de su mismo código HTML, como un solo bloque o como atributos individuales de sus diferentes etiquetas, o bien puede incluirse como un archivo separado. En cualquier caso, el código CSS es también 100% textual y por ende muy ligero para su transmisión mediante HTTP, por lo que no sobrecarga las páginas web y a la vez, permite introducir efectos visuales muy interesantes y funcionales en las páginas y aplicaciones web.

3.1.3.2. Servidores WEB o HTTP

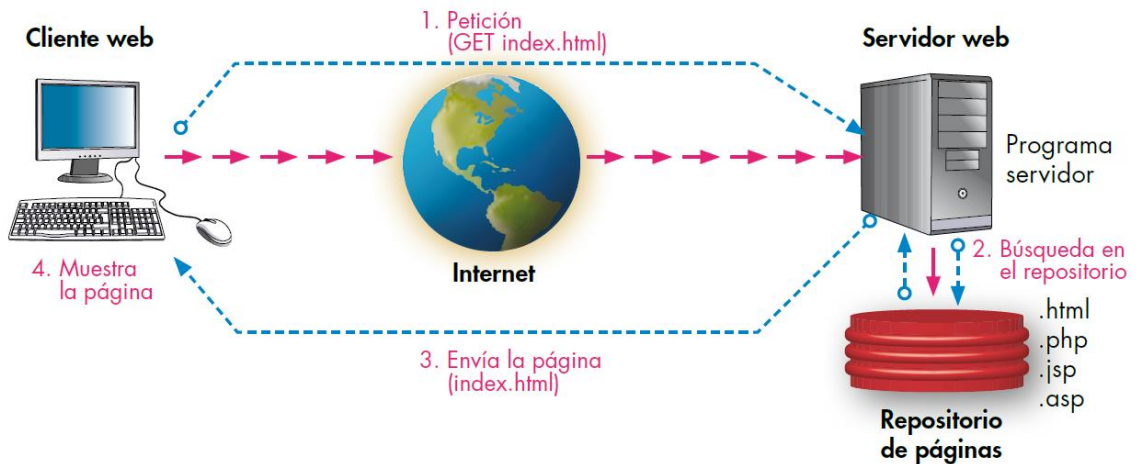
Un servidor WEB o HTTP es un programa diseñado para responder desde el lado del servidor a las peticiones web enviadas por los clientes que generalmente son navegadores web como Microsoft Internet Explorer, Microsoft Edge, Mozilla Firefox y Google Chrome. Es decir, un servidor WEB trabaja dentro del diseño de arquitectura cliente/servidor, ejecutando obviamente el lado del servidor. Los servidores web, en la gran mayoría de casos, se encargan de transmitir información de hipertexto correspondiente a páginas web, las mismas que pueden contener texto en formato enriquecido, imágenes, vídeos, scripts, enlaces a otras páginas y muchos otros tipos de elementos.

Todo este proceso, una vez más en la gran mayoría de los casos, funciona bajo el protocolo HTTP, aunque también pueden usarse otros protocolos como Hypertext Transfer Protocol Secure (HTTPS), File Transfer Protocol (FTP) y Gopher (actualmente en desuso). Básicamente, el cliente solicita a través de su navegador una página o recurso web usando la notación “http://www.host.com/page.html”, que llega hasta el servidor correspondiente al dominio “host.com”, el cual a su vez devuelve la página o recurso solicitado usando los lenguajes HTML y CSS, los mismos que son interpretables por los navegadores web, los que a su vez utilizan esta información para configurar y desplegar correctamente las páginas web, aunque dependiendo del servicio solicitado pueden usarse también otros lenguajes como Extensible Markup Language (XML), o el Extensible Hypertext Markup Language (XHTML), una versión de HTML que implementa la funcionalidad de XML. Por otro lado, el servidor web también puede conectarse con otros servidores o hosts dentro o fuera de su propia red, con el fin de poder completar las solicitudes web. Por ejemplo, podría conectarse con otro servidor dentro de su propia red que haga las veces de repositorio, de donde puede recuperar páginas, imágenes, scripts, etc., o con un servidor de base de datos que se encarga de almacenar y recuperar la información necesaria.

Dentro de este proceso, intervienen también otros protocolos que operan en diferentes capas del modelo de red OSI (HTTP opera en la capa de aplicación), como por ejemplo Domain Name System (DNS), que se encarga básicamente de convertir las direcciones tipo Universal Resource Locator (URL) como “http://www.host.com/page.html”, hasta cierto punto comprensibles por el ser humano, en direcciones IP que son las que realmente permiten localizar cualquier servidor o host existente en Internet. Otro protocolo no menos importante es Transfer Control Protocol (TCP), el cual trabaja en las capas de sesión y transporte del modelo OSI, y se encarga a su vez de garantizar que la

información se envíe y se reciba correctamente entre el cliente y el servidor, de modo que se eviten o se corrijan los errores que pueden generarse durante la transmisión de datos.

Figura 9: Diagrama de funcionamiento de una petición web



Fuente: (Internet Protocols, 2003)

Hoy en día, existen múltiples programas servidores web diseñados para trabajar en diferentes plataformas y sistemas operativos. Algunos de los más conocidos y utilizados a nivel mundial son Internet Information Server (IIS), Apache, Google Web Server, Nginx y Lighttpd, entre otros.

3.1.3.2.1. Internet Information Services

Internet Information Services, conocido también por sus siglas IIS, es un servidor web creado por Microsoft, diseñado para ejecutarse en equipos con la familia de sistemas operativos Windows, y de manera especial en sus ediciones para servidor como son Windows Server 2012 o Windows Server 2016. Permite servir páginas web tanto de forma local como de forma remota, y soporta la transmisión de información mediante los protocolos de Internet más conocidos como HTTP, HTTPS y FTP.

Inicialmente fue publicado como parte del Service Pack 3 de Windows NT 3.51, y desde allí ha ido evolucionando a lo largo de los años. Hoy en día se incluye nativamente como parte de los sistemas

operativos Windows, incluyendo Windows 10, y la versión vigente al momento de Internet Information Services es la 10.0.

Por otro lado, cabe mencionar que Internet Information Services ha sido diseñado de manera modular, por lo que puede ser fácilmente expandido, soportando lenguajes de servidor como ASP y ASP.NET de manera nativa, e incluso de otros fabricantes como PHP mediante la instalación de las herramientas adecuadas.

3.1.3.2.2. Páginas web dinámicas

Como se ha mencionado en párrafos anteriores, la finalidad de la World Wide Web fue, en un principio, posibilitar el intercambio de hipertextos entre diferentes computadores de manera relativamente rápida y sencilla, mediante el uso de múltiples protocolos y estándares, de donde surgieron las primeras páginas web. Sin embargo, quienes se encontraban en este medio pronto se fueron dando cuenta de que la World Wide Web tenía un potencial mucho mayor, y que sus usos podían expandirse hasta límites únicamente impuestos por su imaginación. Uno de los primeros avances en este sentido fue la creación de las así llamadas “páginas web dinámicas”, que básicamente se trata de páginas web cuyo contenido puede variar dependiendo de ciertos factores y de quién las esté visualizando. Un ejemplo clásico de una página web dinámica es la bandeja de entrada de un correo electrónico, ya que los mensajes desplegados dependen del usuario que haya ingresado con sus respectivos datos como son nombre de usuario y contraseña.

Sin embargo, la implementación de las páginas web dinámicas no fue tarea fácil en un inicio, en especial tomando en cuenta que los servidores web existentes en ese momento no disponían de las capacidades necesarias para tal fin. Para lograrlo, fue necesaria la creación e implementación de lenguajes del lado del servidor que fueran capaces de procesar la información enviada por los usuarios y generar como respuesta páginas web con contenido variable en respuesta a dichas solicitudes. Así surgieron lenguajes como Active Server Pages (ASP), PHP y PERL

3.1.3.2.3. Aplicaciones web

Una vez que surgieron los lenguajes del lado del servidor que permitían la creación de páginas web dinámicas, con el pasar de los años dichos lenguajes fueron evolucionando e incorporando cada vez más funciones, instrucciones y métodos, y por tanto ampliando las posibilidades para los

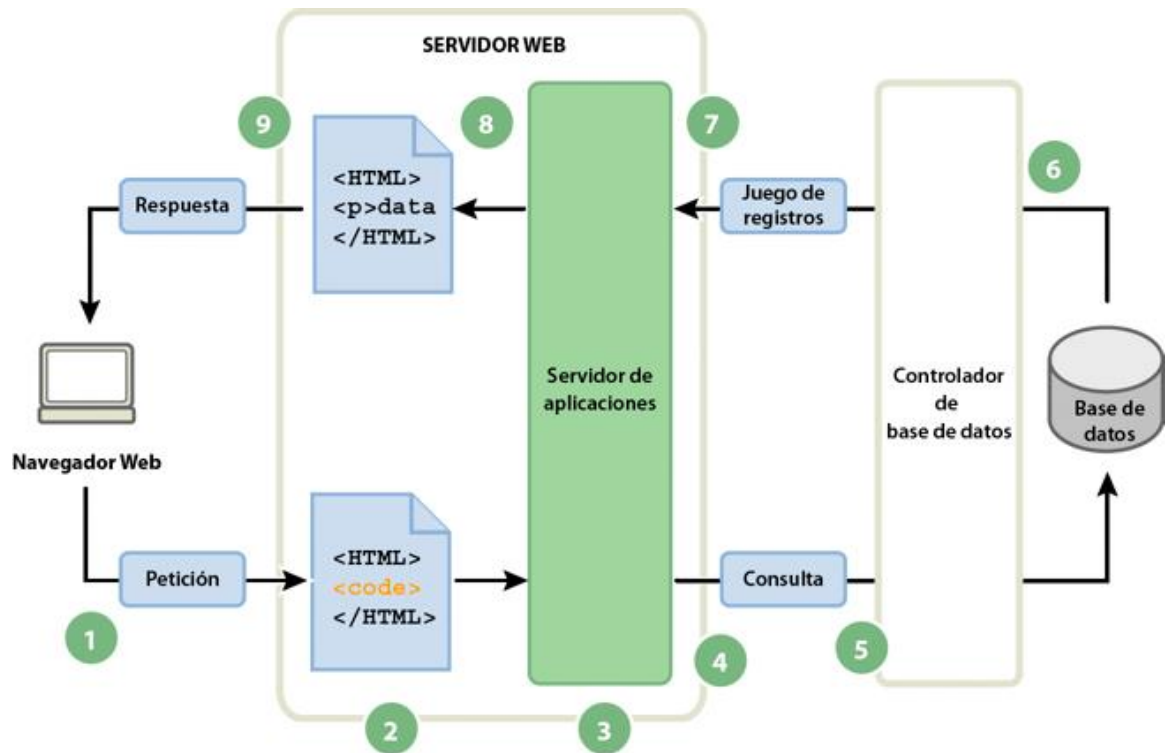
programadores web. Así, comenzaron a aparecer las primeras aplicaciones web, cuyo principal objetivo era el de procesar los datos enviados por los usuarios, para desplegar un resultado, tal como si se tratara de una aplicación tradicional de escritorio pero usando la infraestructura del World Wide Web. Esto traía múltiples ventajas tanto para los usuarios así como para los emisores de las aplicaciones web, como por ejemplo la independencia casi total del hardware y de los sistemas operativos, ya que lo único que se necesitaba era un navegador web compatible con la aplicación en cuestión.

Según Mora [19], *“Internet y la Web han influido enormemente tanto en el mundo de la informática como en la sociedad en general. Si nos concentramos en la Web, en poco menos de 10 años ha transformado los sistemas informáticos: ha roto las barreras físicas (debido a la distancia), económicas y lógicas (debido al empleo de distintos sistemas operativos, protocolos, etc.) y ha abierto todo un abanico de nuevas posibilidades. Una de las áreas que más expansión está teniendo en la Web en los últimos años son las aplicaciones web”*. De este modo, todas las grandes corporaciones, como bancos y transnacionales, se han visto en la necesidad cada vez más creciente de implementar aplicaciones web para sus clientes, sean estas empresas o individuos, ya que de lo contrario se quedan obsoletas y pierden competitividad.

Las aplicaciones web tomaron mayor fuerza y presencia en el mundo de la World Wide Web una vez que los mencionados lenguajes del lado del servidor adquirieron capacidad para conectarse a las bases de datos y realizar operaciones con ellas, básicamente guardado, recuperación y manipulación de la información. Si bien las bases de datos son un capítulo entero dentro del mundo de la informática ([30]), su adaptación a los lenguajes del lado del servidor permitió crear verdaderas herramientas de software que permitieron a su vez que una empresa o institución pueda tener clientes o usuarios en cualquier parte del mundo, tal y como si se encontraran en la misma localidad. Hoy en día, con el uso de herramientas tales como HTML5 y CSS3, las aplicaciones web han dinamizado más todavía sus posibilidades y capacidades, proceso que siguen en continua evolución y mejora que solo el tiempo logrará decir algún día hasta dónde se pueda llegar o cuál pueda ser el límite, de existir alguno.

La siguiente figura ilustra el flujo de información típico de un servidor web que sirve una aplicación web que a su vez se conecta y trabaja con una base de datos, desde la petición original que surge desde la máquina del cliente/usuario.

Figura 10: Flujo de información de una aplicación web con bases de datos



Fuente: (Aplicaciones Web, 2009)

3.1.3.2.4. ASP.NET

Uno de los lenguajes del lado del servidor que ha tenido mayor renombre y ha sido más ampliamente usado alrededor de todo el mundo es, sin duda alguna, Active Server Pages (ASP). Es un lenguaje interpretado creado por Microsoft, cuyos orígenes se remontan a la década de los 90 y fue diseñado para trabajar conjuntamente con IIS, como un módulo del mismo, sobre todo en sistemas operativos de servidor de Microsoft. Hoy en día, el lenguaje ASP tal y como se lo concibió inicialmente se lo conoce como ASP clásico, ya que las versiones modernas del lenguaje se conocen como ASP.NET, ya que funcionan dentro de la plataforma .NET Framework, un conjunto de librerías de programación desarrollado también por Microsoft que ha dinamizado inimaginablemente la programación de aplicaciones para sistemas operativos Windows y, en el caso de ASP.NET, de aplicaciones web ([33]).

ASP.NET basa su funcionamiento en dos elementos fundamentales: los “web forms” y los postbacks. Toda página que se crea bajo ASP.NET se conoce como un “web form”, ya que indefectiblemente incluirá un elemento tipo <FORM> de HTML. Esto es especialmente útil para el desarrollador bajo ASP.NET, ya que se evita la creación manual de uno o más formularios en cada página como sucede en otros lenguajes del lado del servidor como PHP. El desarrollador únicamente se concentra en añadir los elementos que requiere en su página, tales como cuadros de texto, botones, cajas de verificación, botones de opción múltiple y etiquetas, entre mucho otros tipos de controles, y éstos serán automáticamente renderizados como elementos de un formulario (con la etiqueta <FORM>) en el HTML resultante enviado al computador del cliente solicitante. De igual forma, ASP.NET no permite eliminar el formulario colocado por defecto dentro de cada página que se crea, ya que de su presencia depende el funcionamiento de muchos otros elementos y funciones propios de este lenguaje.

El otro elemento característico de ASP.NET es el postback. El funcionamiento por defecto de ASP.NET es que cada página se llame a sí misma, o se “recargue”, cada vez que el usuario presiona un botón o ejecuta alguna otra acción configurada, utilizando el formulario por defecto que ASP.NET crea en sus páginas. Esto a diferencia de otros lenguajes, como PHP, donde el comportamiento por defecto de un formulario ubicado dentro de una página es llamar a otra página, desde un formulario creado manualmente. Este “autollamado” o “recarga” de las páginas de ASP.NET es el comportamiento conocido como “postback”, y tiene varias ventajas sobre la tradicional llamada a otra página. Por ejemplo, bajo ASP.NET es muy fácil controlar si la carga de una página se la realiza por primera vez o si ya es una “recarga”, y en base a eso, variar su comportamiento o decidir lo que se devuelve como resultado.

Este comportamiento del postback está íntimamente ligado con otra característica muy poderosa propia de las páginas de ASP.NET conocida como “View State”. Ésta se refiere a que el comportamiento predeterminado de las páginas de ASP.NET es conservar automáticamente la información existente en cada uno de los controles de una página web, como los cuadros de texto. Esta característica ahorra gran cantidad de tiempo y líneas de código a los desarrolladores de aplicaciones web, así como también recursos y memoria a los servidores web, ya que los datos de una página se cargan una sola vez durante el primer despliegue de la página, mientras que en los siguientes ya no es necesario hacerlo puesto que la información se conserva entre postbacks. Esto permite, por ejemplo, reducir considerablemente la carga de transacciones a una base de datos

puesto que la información se consulta una sola vez, y de otro modo, habría que consultarla repetitivamente en cada nueva carga de la página o postback. La manera en la que ASP.NET conserva esta información es en realidad bastante sencilla: codifica la información existente en cada control de la página como una cadena de texto, y transmite esta cadena a la siguiente página como un campo oculto de HTML. Al ser texto, esta técnica no sobrecarga la página resultante, a la vez que brinda todas las ventajas ya mencionadas.

Si bien las mencionadas no son todas las características de ASP.NET, son las características fundamentales que, funcionando juntas, hacen que ASP.NET permita crear páginas sumamente versátiles, ligeras y funcionales que, a su vez, crean aplicaciones web muy potentes y listas para responder a las necesidades de hoy en día, de manera relativamente sencilla.

3.1.3.3. Web services

Hoy en día, el término web service es bastante utilizado dentro de la terminología estándar de los servidores y las aplicaciones web, aunque su significado exacto muchas veces no está del todo claro. De hecho, existe gran cantidad de definiciones del término web service, desde las más simples y genéricas hasta algunas mucho más específicas y restrictivas. Dentro de las definiciones más sencillas se puede decir que un web service es una aplicación accesible desde otras aplicaciones a través de la world wide web. Sin embargo, esta definición abarcaría un sinnúmero de applets (pequeñas aplicaciones o fragmentos de las mismas) y bloques de código que en realidad no se podrían considerar web services. Otra definición más completa, provista por el consorcio UDDI (Universal Description, Discovery and Integration) dice que los web services son *“aplicaciones de negocios modulares y auto-contenidas que contienen interfaces abiertas, basadas en estándares y orientadas a Internet”*. Esta definición obviamente habla sobre aplicaciones que ofrecen interfaces basadas en estándares de Internet, lo cual se acerca mucho más a lo que hoy en día se conoce realmente como un web service. Finalmente, se puede citar una tercera definición, mucho más completa que las dos anteriores y provista por el consorcio World Wide Web (W3C Consortium), que dice que un web service es *“Una aplicación de software identificada por un URI (Uniform Resource Identifier) cuyas interfaces y vínculos pueden ser definidos, descritos y descubiertos como objetos XML. Un web service soporta la interacción directa con otros agentes de software utilizando mensajes basados en el lenguaje XML que se intercambian basándose en protocolos de Internet”*. Esta tercera definición es la que tiene

mayor aceptación hoy en día, ya que define lo que es un web service y la manera como debería funcionar con total precisión y detalle.

Dentro del mundo del desarrollo de aplicaciones web, un web service suele ser una función o método que devuelve un conjunto específico de información en formato XML y que puede ser consumido desde otras aplicaciones (no necesariamente web). Un web service puede devolver como respuesta un conjunto de registros desde una base de datos, el resultado de un cálculo, una tabla de datos, un objeto y básicamente cualquier tipo de dato o información que pueda serializarse como un código en formato XML. Un web service puede requerir o no el envío de uno o más parámetros para su funcionamiento.

.NET Framework en general y ASP.NET en particular ofrecen múltiples librerías que permiten conectarse a web services y consumir sus servicios de manera relativamente sencilla y sin mayores complicaciones, recuperando los datos solicitados en el formato requerido, listos para ser utilizados en la aplicación en cuestión. Por otro lado, la plataforma de desarrollo de Microsoft Visual Studio y ASP.NET ofrecen gran facilidad para crear web services también de manera sencilla y ágil, los mismos que se generan listos para ser publicados y consumidos, cumpliendo con los estándares de Internet que se requiere, y pudiendo ser consumidos desde cualquier otra aplicación, sea esta creada bajo ASP.NET o bajo cualquier otro lenguaje o plataforma de desarrollo que lo soporte.

3.1.4. Gestión Documental

En toda organización, el quehacer de sus actividades diarias determina que se genera gran cantidad de información, mucha de la cual debe ser plasmada en documentos para su posterior uso y archivado. Tradicionalmente, estos documentos eran escritos en papel y de forma manual, si bien con el advenimiento de la informática se pasó a utilizar las nuevas técnicas de impresión como son matricial, inyección de tinta y láser, entre otras, con lo cual los documentos prácticamente dejaron de ser escritos a mano y pasaron a ser generados casi en su totalidad mediante el uso de impresoras. Sin embargo, dichos documentos seguían siendo creados en papel con todos los problemas que esto conlleva, tales como altos gastos en el largo plazo, susceptibilidad de pérdida o deterioro, y requerimiento de grandes espacios de almacenamiento. Todas estas actividades de generación, almacenamiento y administración de documentos es lo que en términos generales se conoce como “gestión documental”.

Sin embargo, los vertiginosos avances en el campo de la informática permitieron también la evolución de la gestión documental, mediante la creación de formatos de documentos que permitían generar documentos electrónicos y mantenerlos almacenados como archivos de computador. El surgimiento de procesadores de textos especializados permitió que las organizaciones a todo nivel comiencen a crear y mantener cada vez más archivos en formato electrónico y menos en las tradicionales formas impresas. Un ejemplo clásico de esto es el programa Microsoft Word que apareció con su primera versión por el año 1989, y siendo el primer procesador de textos que funcionaba en modo gráfico, resultó ser un éxito rotundo en ventas, superando con creces a sus antecesores tales como el popular Wordperfect.

Según Gloria Ponjuan [15], se puede considerar a la gestión documental como *“un proceso administrativo que permite analizar y controlar sistemáticamente, a lo largo de su ciclo de vida, la información registrada que se crea, recibe, mantiene y utiliza una organización en correspondencia con su misión, objetivos y operaciones”*. Ella misma también considera a la gestión documental como *“un proceso para mantener la información en un formato que permita su acceso oportuno, y por ello requiere tareas y procedimientos para cada fase y la explotación de esta información registrada, que es evidencia de las actividades y transacciones de las organizaciones y que les permite lograr una mayor eficacia”*. Por último, ella misma manifiesta que un buen sistema integral de gestión documental debe permitir:

- Acceder oportunamente a la información.
- Organizar grandes volúmenes de información.
- Mantener los flujos adecuados de información en la organización.
- Soportar la integridad y seguridad de la información.

En la actualidad, uno de los formatos que ha cobrado mayor peso dentro de la generación y archivado de documentos electrónicos es Portable Document Format (PDF), debido a la versatilidad y practicidad que ofrece para todo tipo de documentos.

3.1.4.1. PDF

El formato de documento electrónico PDF fue originalmente creado por Adobe Systems y su primera versión se publicó para el año de 1993. Originalmente se publicó como un formato propietario para ser usado únicamente por la aplicación Adobe Acrobat, pero con el pasar de los años

se ha ido volviendo más y más popular, y hoy en día es un formato de uso generalizado para la publicación e intercambio de documentos ya sea por organizaciones de carácter público, privado, grandes, medianas o pequeñas, así como por individuos.

Actualmente, un documento en formato PDF puede abrirse en prácticamente cualquier sistema operativo incluyendo Windows, Linux, Unix y MacOS, e inclusive en dispositivos móviles como tabletas y teléfonos celulares. Casi todo tipo de sistema operativo cuenta con herramientas para la lectura de archivos en formato PDF de manera gratuita, si bien las herramientas de edición o creación de archivos PDF pueden ser de pago. Hoy en día, PDF es un formato abierto cuyo estándar puede ser libremente utilizado por quien así lo requiera, sin embargo, un documento elaborado en formato PDF está pensado para ser únicamente legible y no para ser modificado, por lo que para esta segunda tarea se requiere de herramientas especiales.

Es el formato preferido para la elaboración de documentos ya que puede contener diferentes tipos de elementos tales como texto, gráficos vectoriales, gráficos de mapas de bits, elementos multimedia (vídeos, sonidos) y enlaces a otras páginas/documentos, lo que lo hace ideal para este tipo de tarea. Un documento en formato PDF puede también contener elementos de formulario, por lo que también resulta ideal para la creación de encuestas, hojas de datos personales y similares, que puedan ser llenados por los respectivos usuarios directamente en el mismo documento, quedando listo para ser impreso o guardado como archivo de disco.

Atendiendo a las nuevas necesidades que han surgido a nivel de todo tipo de organizaciones y entidades en cuanto a gestión documental se refiere, el formato PDF ha cobrado real importancia ya que, por las características ya mencionadas, se torna especialmente útil para este tipo de necesidades. Por ejemplo, los grandes sistemas de gestión documental empleados actualmente por los gobiernos de diferentes países ([13], [14]), si bien proveen funciones de gestión y seguimiento de la documentación que generan, utilizan el formato PDF para la creación y almacenamiento de los archivos correspondientes. Lo mismo sucede en universidades, bancos, industrias, empresas comerciales, y en general, en cualquier entidad que requiera cualquier nivel de generación y manejo de documentos. Este hecho ha tenido también su repercusión en los actuales lenguajes de programación y plataformas de desarrollo de software, los cuales por regla general ya incluyen librerías, controles y herramientas especializadas para la generación de documentos en formato PDF. Como un ejemplo muy claro de esto tenemos el conjunto de librerías iTextSharp, que permiten generar documentos en formato PDF desde proyectos de desarrollo de Microsoft Visual Studio bajo el

lenguaje C#, que incluyan todo tipo de elemento soportado por este formato, de manera completamente detallada y exacta, si bien existen también otras herramientas para la misma plataforma de desarrollo como lo es Crystal Reports [23], que permiten diseñar reportes de manera completamente visual y desplegarlos en los computadores clientes de los usuarios, quienes pueden guardar localmente copias de dichos reportes en formato PDF.

Finalmente, cabe mencionar que un documento en formato PDF puede hacerse muy seguro mediante el cifrado y la inclusión de firmas electrónicas. Esto ha hecho que el formato PDF se pueda utilizar incluso en entidades militares y gubernamentales que manejan información sensible, con total seguridad.

3.2. Estado del Arte

Por el año 2007, Sánchez Dromundo [29] realizó un estudio sobre el porcentaje de graduaciones en relación al porcentaje de egresados de las unidades de posgrados de algunas universidades mexicanas, y determinaba que *“La graduación y titulación de estudiantes de educación superior constituyen un problema que afecta de forma significativa a varias instituciones educativas, entre ellas la Universidad Nacional Autónoma de México (UNAM). De acuerdo con estudios realizados por diversos investigadores, el bajo índice de eficiencia terminal en el nivel de licenciatura se consolida como una tradición desde los años sesenta con porcentajes de 19% (Garza, 1986, p. 15). Pese a que durante los años ochenta y noventa se realizaron modificaciones a la legislación y los planes de estudio, la situación prevalece y en 1998 la eficiencia terminal del nivel licenciatura se ubica por debajo del 40%. El problema se extiende al posgrado que presenta índices similares de graduación de 10%, en un tiempo que puede superar los nueve años (Rojas Argüelles, Aguilar del Valle y Valle Gómez Tagle, 1992)”*.

Como se puede observar, en el año 2007 en la Universidad Nacional Autónoma de México (UNAM) existía un bajísimo porcentaje de graduados en relación a los egresados de las carreras, que en caso de los posgrados llegaba a un 10%. Esto pasó a ser una temática de estudio y se lo dejó de tomar como un asunto aislado dentro de cada institución de educación superior, y más bien se pasó a tomar como un problema social grave que llegaba a afectar a la educación como tal de todo un país. Las bajas tasa de graduados son un indicador fundamental de que los procesos de graduación de posgrados necesitan ser optimizados y agilizados, lo cual es un problema común especialmente en instituciones de educación superior latinoamericanas.

Para corroborar lo anterior, se puede también citar a Reisin [27], quien manifiesta lo siguiente: *“¿Cuáles son los desafíos más importantes para hacer una tesis de maestría en las ciencias sociales? ¿Qué recursos personales e institucionales se consideran una ayuda para poder realizarla y qué situaciones aparecen percibidas como obstáculo? Estas preguntas, que han guiado la presente investigación de tesis, se insertan en un contexto nacional e internacional de expansión del sector de posgrado, lo cual, a su vez, ha hecho emerger - en nuestro entorno tanto como en los países centrales- la preocupación por la baja tasa de graduación resultante. Recordemos que la tasa de egreso de los estudios cuaternarios, estimada para nuestro país, según Jeppesen y otros (2004) corresponde al 14,8%, constituyéndose en un problema a considerar”.*

Así, ambos estudios coinciden en que la tasa de graduados de posgrados está alrededor de un 10%, por lo que se puede decir que la cantidad de maestrantes que llegan a graduarse frente al número total de egresados es sumamente baja. Ello implica que es necesario analizar todo el proceso de graduación que sigue un maestrante desde que termina sus estudios hasta que recibe su título, para identificar los posibles puntos débiles, determinándose las causas de cualquier tipo de problema que pudiera existir, a fin de tratar de corregir dichos errores y optimizar el proceso, dando una gran ayuda tanto a maestrantes como al personal correspondiente de las instituciones de educación superior.

Por otro lado, dada la existencia de potentes herramientas, lenguajes y plataformas de desarrollo de software hoy en día, es un desperdicio inadmisibles de recursos el no aprovecharlos para la elaboración de aplicativos que permitan automatizar en la medida en que sea posible los procesos administrativos de las instituciones de educación superior en general, y de sus respectivas unidades de posgrados en particular. Sin embargo, se observa que aún hoy en día, muchas universidades no cuentan aún con aplicativos que les permitan automatizar sus procesos de graduación y posgrados, tanto a nivel local, como nacional e internacional. Así por ejemplo, Velásquez y Vargas [35], de la Escuela Politécnica Nacional (EPN) con sede en Quito, establecen que *“La automatización de los procesos inicia a través del modelado de las actividades y procesos, para lograr un mejor entendimiento del mismo y muchas veces esto presenta la oportunidad de mejorarlos. Esto asegura que los procesos se comporten siempre de la misma manera, reduciendo el margen de error y dando elementos que permitan visualizar el estado de los mismos durante cada etapa. Ésta forma de administrar los procesos permite asegurar que los mismos se ejecuten eficientemente, cumpliendo con estándares de calidad previamente establecidos, y ayudando a la obtención de información que luego puede ser usada para*

mejorarlos. Es a través de la información que se obtiene la ejecución diaria de los procesos, que se puede identificar posibles ineficiencias o fallas en los mismos, y actuar sobre ellos para optimizarlos". Así mismo, Jame y Maza [17] de la Universidad de las Fuerzas Armadas (ESPE), también con sede en Quito, manifiestan en el año 2015 que *"Los programas de posgrado del Departamento de Ciencias de la Computación, requieren prestar un mejor servicio a los estudiantes de los programas proveyéndoles atención rápida y personalizada, que permita cumplir los trámites burocráticos que exigen las actividades del procedimiento para la obtención del título de posgrado; para este efecto se ha desarrollado la norma de procedimiento de titulación de programas de posgrado, la que al ser automatizada brindará el servicio que esperan los estudiantes de los programas que se encuentran vigentes"*. De ambos autores podemos deducir que es una tendencia general a nivel de las grandes universidades ecuatorianas el buscar crear herramientas adecuadas de software para facilitar y optimizar tanto como fuere posible los procesos de graduación de posgrados que tienen que ejecutar sus maestrantes, por todas las razones ya expuestas.

Incluso a nivel local, en la ciudad de Ambato, Vaca [28] exponía en el año 2014 que *"Actualmente los procesos del manejo de información son realizados de forma manual y en hojas electrónicas, los cuales no abastecen a la gran cantidad de información que se maneja, generando pérdida de la misma, de tiempo, datos duplicados, demora de trámites, desperdicio de recursos humanos y materiales. Por esta razón, cada vez se hace más imprescindible la necesidad de tomar la decisión de desarrollar una aplicación para manejar y almacenar la información que se genera en la Dirección de Posgrado de la UTA, con la cual podremos tener un mejor control, mejorar la gestión, optimizando el tiempo y los recursos económicos"*. Por ende, se puede deducir que hoy en día, toda universidad e institución de educación superior a nivel nacional, que no lo haya conseguido aún, se encuentra en el proceso de buscar aprovechar los recursos técnicos y tecnológicos existentes en la actualidad para automatizar sus procesos administrativos en general y de sus unidades de posgrados en particular, ya que las instituciones que no lo hicieren así, se estarían condenando a sí mismas a permanecer en la obsolescencia.

Finalmente, en cuanto a la gestión documental, como ya se ha expuesto existen herramientas tecnológicas especializadas encargadas de facilitar las tareas de creación, almacenamiento y seguimiento de documentos, generalmente en el ya analizado formato PDF, debido a su gran versatilidad y uso generalizado. Por otro lado, los maestrantes de las unidades de posgrados de las universidades ecuatorianas, en sus trámites de graduación, generan una cantidad muy considerable

de documentos, sean estos solicitudes, informes, certificados, temas y planes de proyectos de titulación, etc. Por tanto, como parte de la automatización de dichos proyectos de las unidades de posgrados, debe inevitablemente considerarse la generación y el manejo de todos aquellos documentos mencionados, tarea que hoy en día es posible realizarla de manera eficiente gracias a las poderosas herramientas tecnológicas que ya se ha descrito en párrafos anteriores.

A nivel general se puede también observar que muchas empresas nacionales se encuentran en proceso de implementación de aplicaciones de gestión documental para optimizar y asegurar dicho proceso en sus instalaciones, como es el caso de la textil Vicunha, muchas veces con ayuda de profesionales o estudiantes universitarios expertos en el tema. En este caso, Morillo [20] ha explicado que: *“En toda empresa se requiere mantener una organización de sus documentos, por lo que a un nivel tecnológico nos ayudamos del concepto DMS, los cuales están revolucionando la manera de almacenamiento de documentos. En una empresa que se encuentra en sus inicios tal vez no sienta la necesidad de mantener una estructura organizativa para sus documentos, a diferencia que en empresas que tienen una larga trayectoria surge la necesidad de tener un conocimiento adicional de cómo almacenar sus documentos, con la ayuda de aplicaciones que facilitan la gestión de ello”*.

Por tanto, se puede decir con total certeza que la gestión documental es una parte muy importante para cualquier aplicativo o sistema que se esté desarrollando para una empresa o institución en general. Por lo tanto, debe incluirse un módulo específico para esta tarea, ya sea un software para una empresa, institución gubernamental o universidad. Si bien los sistemas de gestión documental existentes en la actualidad pueden ofrecer diversas y variadas funciones de mantenimiento de documentos en formato digital, el sistema más sencillo, dependiendo de las necesidades, debe al menos ser capaz de generar, almacenar y desplegar documentos en el ya renombrado formato PDF. Más aún, tratándose de aplicaciones web, se puede aprovechar la gran compatibilidad que suele existir entre los diferentes navegadores web con el formato de documento PDF,

Capítulo 4

Metodología

El presente trabajo consiste básicamente en el desarrollo de un aplicativo web cuyo objetivo es el de facilitar y agilizar los procesos de graduación de una unidad de posgrados. Como caso de estudio se ha considerado los procesos que actualmente deben seguir los maestrantes de las diferentes carreras de posgrados de la PUCESA, por lo que ha sido sumamente útil y necesario el acceder a la información ya existente en las bases de datos de la institución, las mismas que corresponden al previamente existente sistema conocido como “Academics” [26], ya que de otro modo surgirían múltiples problemas como desperdicio de recursos, duplicación de la información, generación de información inconsistente, sobrecarga a los servidores de la institución y otros.

Así, una vez que se identificó el problema existente por la falta de automatización de los procesos de graduación de la PUCESA, así como también se logró determinar que éste es un problema común presente en muchas de las instituciones de educación superior del país, se decidió que es necesario elaborar una aplicación que permitiera a todos los usuarios del DIP de la PUCESA, tanto internos como externos, esto es maestrantes, docentes y personal administrativo del DIP, poder ejecutar sus diversas funciones a través de la mencionada aplicación.

Por otro lado, se decidió que la mejor alternativa era desarrollar una aplicación web, ya que de este modo los diferentes actores del proceso de graduación de posgrados de la PUCESA podrían hacer uso de dicha aplicación desde cualquier computador con acceso a Internet, independientemente del lugar donde se encuentren. Además, para guardar compatibilidad con el mencionado sistema Academics, la presente aplicación se desarrolló considerando el uso de las mismas herramientas que se utilizaron para crear Academics, como plataforma de desarrollo (Microsoft Visual Studio 2010) y motor de bases de datos (Microsoft SQL Server 2008 R2). Incluso se procedió a utilizar librerías de

controles complementarios para Visual Studio 2010, a fin de lograr una aplicación visualmente atractiva y compatible con Academics incluso en este sentido.

Por último, para la presente aplicación web se ha decidido emplear la metodología ágil de desarrollo de software XP, ya que la misma fomenta la participación activa de todos los actores involucrados en el proceso de elaboración de la aplicación, incluyendo en este caso al personal del DIP de la PUCESA, y al ser una metodología ágil permite obtener un producto de software de gran calidad en un tiempo relativamente corto, lo cual es particularmente interesante en el caso de desarrollo de aplicaciones web.

A continuación se procede a describir cada una de las etapas correspondientes a la metodología XP en relación al desarrollo de la presente aplicación web.

4.1. Diagnóstico

El problema fundamental que se ha identificado y que ha originado el desarrollo del presente trabajo de titulación, es la falta de automatización en los procesos de graduación de posgrados de la PUCESA, lo cual genera lentitud en los procesos tal como se dan actualmente. Esto se ha determinado en base a diferentes factores como la experiencia personal y entrevistas realizadas tanto con maestrantes de la carrera de Maestría en Gerencia Informática así como con el personal administrativo del DIP de la PUCESA.

Al ser todos los procesos presenciales, los maestrantes deben, en primer lugar, extraer tiempo de sus actividades diarias para acercarse a la PUCESA a realizar los diferentes trámites propios del proceso, esto de por sí ya puede significar una pérdida de tiempo considerable para los maestrantes, ya que muchas veces pueden pasar días hasta que un maestrante pueda acercarse a la institución a realizar un trámite, por simple que éste sea.

Por otro lado, dentro del mismo DIP de la PUCESA se generan grandes pérdidas de tiempo, ya que los trámites que se llevan a cabo en dicha dependencia suelen ser lentos, y muchas veces un maestrante debe esperar considerablemente para ser atendido. Si por algún motivo el maestrante olvidó uno de los requisitos necesarios para realizar un trámite, todo el proceso se repite nuevamente, y como se mencionó, ese simple hecho puede causarle la pérdida de varios días.

Otro de los puntos débiles dentro del proceso de graduación de posgrados de la PUCESA que se observa actualmente se relaciona con los docentes. Los maestrantes tienen que comunicarse frecuentemente con los docentes, ya sean tutores, directores, informantes o revisores de sus proyectos. Sin embargo este tipo de entrevistas suelen ser de carácter personal y dentro de los horarios dispuestos por la PUCESA para tal fin.

Esto significa otra pérdida de tiempo sustancial para los maestrantes, ya que muchas veces deben esperar tiempos considerables para poder entrevistarse con sus directores de proyecto, o docentes revisores. Generalmente, recibirán solicitudes de aplicar correcciones a sus respectivos documentos, lo cual nuevamente inicia el proceso con las ya mencionadas pérdidas de tiempo.

Fuera de todo lo mencionado, otro problema identificado es el uso desmedido de recursos de todo tipo, como por ejemplo, papel. Para mencionar un ejemplo, un maestrante debe imprimir su plan de proyecto de titulación para presentarlo a su director de proyecto, y si este le envía a hacer correcciones, el maestrante deberá aplicarlas e imprimir nuevamente para volverlo a presentar. Posteriormente, una vez que el director aprueba el plan, éste pasará a manos del docente informante, quien nuevamente puede solicitar se realicen correcciones, proceso que puede generarse varias veces. Considerando que un plan de proyecto de titulación puede ser de 8 o 10 páginas en promedio, esto ya representa un desperdicio sustancial de recursos no renovables.

Como se puede observar, existe una cantidad considerable de problemas de diferente índole que en su conjunto causa retrasos y pérdidas de tiempo considerables a todos los actores dentro del proceso de graduación de posgrados de los maestrantes de la PUCESA. Todos estos problemas han sido identificados principalmente mediante la observación directa y la experiencia propia, especialmente durante el proceso de desarrollo del presente proyecto.

Igualmente, todos estos problemas ya habían sido identificados previamente por el personal administrativo del DIP de la PUCESA, quienes en su momento debían enfrentar diariamente acumulación de documentos y trámites pendientes que, por su naturaleza, tomaban mucho tiempo procesarlos, al igual que largas colas de maestrantes en espera de ser atendidos, causando una molesta aglomeración de personas en un espacio relativamente reducido.

En el caso de los trámites de graduación de posgrados, los maestrantes al ser atendidos podían únicamente realizar consultas breves o entregar su documentación pendiente, debiendo esperar en

la gran mayoría de los casos varios días antes de recibir una respuesta, autorización o aprobación que les permita continuar con sus trámites.

Atendiendo a todos estos antecedentes, el mismo personal administrativo del DIP determinó su propia necesidad urgente de contar con un sistema automatizado que les permita aliviar toda esa carga innecesaria de trámites burocráticos, agilizando los procesos correspondientes para todas las partes implicadas, y a la vez optimizando y rediseñando todo el proceso de graduación de los maestrantes, para dejarlo tan liviano como fuera posible.

4.2. Método(s) aplicado(s)

Una vez que se identificó la necesidad inequívoca del DIP de la PUCESA de contar con un sistema que le permita agilizar sus procesos, automatizando todas las tareas que fuera posible así como también eliminando pasos redundantes que causaba pérdidas innecesarias de tiempo a sus usuarios, se decidió que la solución más óptima para este problema sería la elaboración de un producto de software que hiciera posible lograr dicha optimización.

Para lograrlo, se comenzó realizando entrevistas periódicas en el personal administrativo del DIP y en especial con su director, en su momento el PhD. Juan Mayorga Zambrano, quien ya contaba con una noción bastante clara de los procesos clave que debían ser optimizados, así como del tipo de herramienta que se requería para tal fin.

Conjuntamente se decidió que la mejor opción de solución consistiría en una aplicación web, ya que esta ofrece varias ventajas para sus usuarios en relación a una aplicación tradicional de escritorio, como por ejemplo la posibilidad de acceso remoto y la independencia del hardware y sistema operativo de los usuarios.

Por otro lado, se logró identificar que la mayor parte de los datos que se requeriría para la elaboración de la mencionada aplicación ya se encontraban disponibles en los servidores de la PUCESA, en las bases de datos correspondientes a su ya existente sistema conocido como "Academics". Específicamente, ya existía la información correspondiente a carreras de posgrados, docentes, estudiantes de posgrados, periodos académicos y pénsums de estudios. Por tanto, se decidió que lo más óptimo en este caso era enlazar la aplicación web a desarrollarse con las bases de datos de Academics, a fin de aprovechar la información ya existente, evitando así la sobrecarga a los servidores de la institución, la duplicidad de la información y la inconsistencia en los datos.

Sin embargo, se identificó también la necesidad de guardar nueva información que no estaba aún presente en las bases de datos de Academics, como la relativa a temas de proyectos de titulación, planes de proyectos, objetivos, líneas de investigación, avances del proyecto, correcciones solicitadas y enviadas, etc. Para solventar esto, se ha creado las tablas y bases de datos necesarias que vendrán a complementar a las ya existentes en Academics. Por tanto, la aplicación a desarrollarse funcionará tanto en base a datos propios almacenados y administrados en sus propias bases de datos, así como en base a los datos ya existentes previamente en la PUCESA.

Una vez analizados los requerimientos y los datos ya existentes en la institución, se procedió a realizar una entrevista inicial con el personal del departamento de Informática de la PUCESA, en su momento el Ing. César Guevara y, de manera especial, con el Ing. Mg. José Enríquez, quienes luego de escuchar los requerimientos y analizar el tipo de proyecto que se preveía realizar, acordaron compartir toda la información que fuera útil o necesaria para el desarrollo de la presente aplicación mediante la habilitación de webservices. Básicamente un webservice es una función habilitada en un servidor accesible desde Internet, que devuelve un conjunto de datos necesario para realizar una tarea específica.

Por último, después de analizar exhaustivamente toda la información recopilada inicialmente así como los requerimientos del DIP claramente definidos, se decidió conjuntamente con mi director de proyecto de titulación, el Ing. Mg. José Marcelo Balseca Manzano, utilizar Extreme Programming (XP) como metodología de desarrollo de software, debido al sinnúmero de ventajas que ofrece y a que su filosofía es el desarrollo de software rápido pero de calidad, mediante la participación del cliente o usuario final en el proceso, lo cual garantiza que el producto del proyecto quede a su entera satisfacción y acorde a sus necesidades.

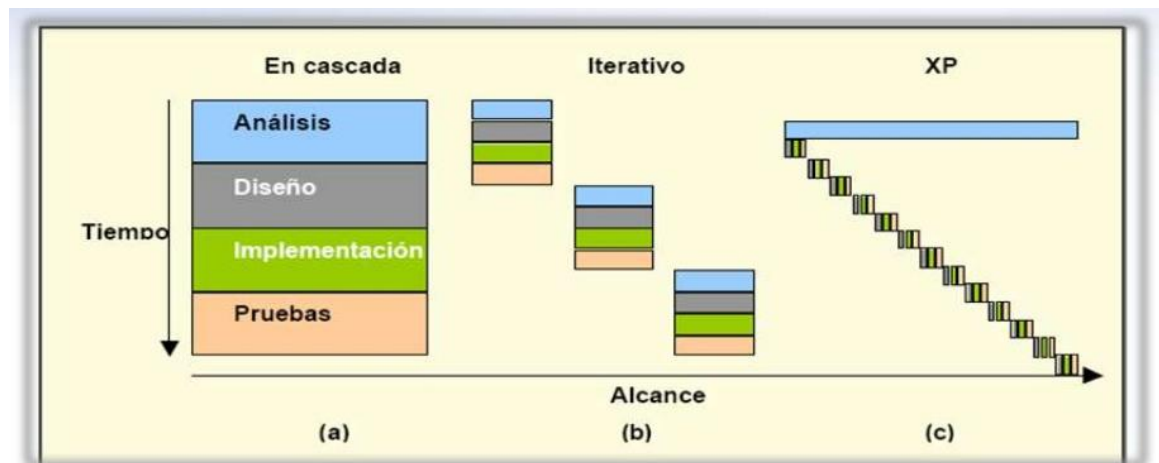
4.2.1. Desarrollo de software mediante la metodología Extreme Programming (XP)

A partir de la década del 2000, comenzaron a surgir las así denominadas metodologías ágiles de desarrollo de software, las cuales se basaron en un enfoque mucho más práctico, menos burocrático y centrado en responder a las necesidades reales de los futuros usuarios de un producto de software, por lo que hoy en día son indudablemente las que más se utilizan.

Entre éstas metodologías ágiles se encuentra la así llamada Extreme Programming (o XP), la misma que ofrece un enfoque sumamente interesante y práctico de desarrollo de software que

permite obtener en relativamente poco tiempo un producto de software de alta calidad, garantizando la satisfacción del usuario final a la vez que manteniéndose abierto a los cambios repentinos en las necesidades del mismo.

Figura 11: Comparación de los ciclos de desarrollo de software entre la metodología XP y metodologías tradicionales (en cascada e iterativa)



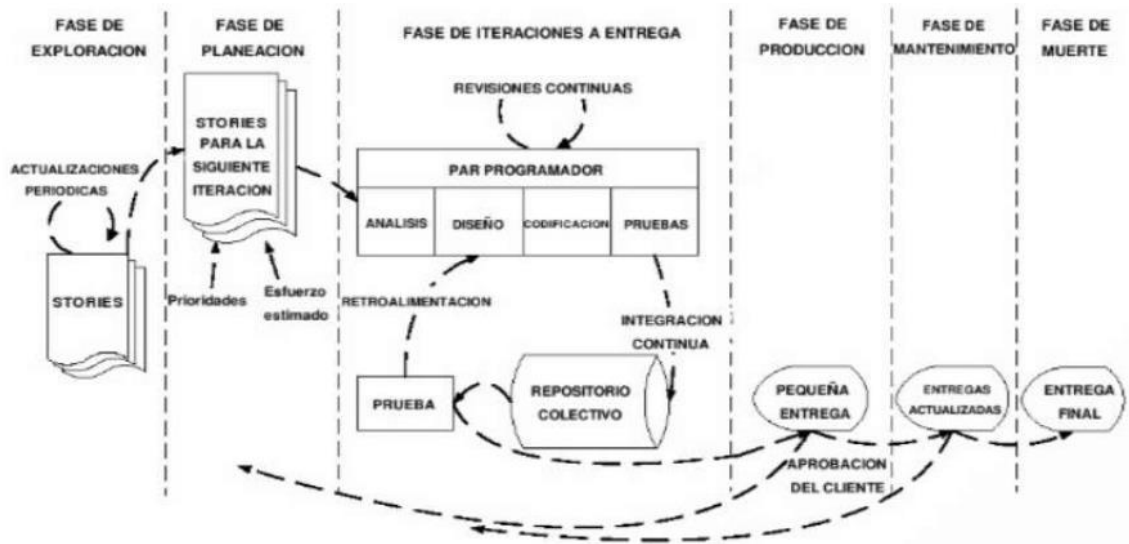
Fuente: (Metodologías de Desarrollo, 2012)

Algunas de las principales prácticas propias de la metodología de desarrollo XP ([4], [5]) son las siguientes:

- Elaboración de historias de usuario
- Desarrollo guiado por pruebas (TDD, por su siglas en inglés)
- Pruebas unitarias
- Diseño simple
- Refactorización de código
- Integración y pruebas continuas
- Programación en parejas

La siguiente figura muestra el ciclo de vida del proceso de Extreme Programming, tal como lo concibió su creador, Kent Beck originalmente:

Figura 12: Ciclo de vida del proceso XP



Fuente: Kent Beck

Básicamente, para utilizar la metodología XP se debe formar un equipo de trabajo donde debe existir al menos un representante del cliente o usuario final de la aplicación a desarrollarse, los programadores o desarrolladores de código, un gerente del proyecto y un agente de pruebas. En el caso de proyectos pequeños, una misma persona puede cumplir más de un rol siempre que esto no sobrecargue de trabajo a nadie y se pueda mantener la calidad en el producto de software a desarrollarse y la satisfacción en el usuario final del mismo. De igual forma, puede haber más de una persona que realice las funciones correspondientes a un determinado rol, como por ejemplo, puede haber dos o más representantes del cliente como parte del equipo de trabajo, según los requerimientos que éste tenga.

En el caso de desarrollo de la presente aplicación, el equipo de trabajo quedó conformado de la siguiente manera:

Tabla 2: Equipo de trabajo para desarrollo de la presente aplicación

| Nombre | Roles |
|-----------------------------------|---|
| PhD. Juan Mayorga Zambrano | <ul style="list-style-type: none">• Co-gerente del proyecto |
| Lcda. Myriam Mayorga | <ul style="list-style-type: none">• Representante del cliente / usuario final (DIP de la PUCESA) |
| Ing. Gabriela De la Torre Sánchez | <ul style="list-style-type: none">• Gerente del proyecto• Programador/Desarrollador |
| Ing. Mg. José Enríquez | <ul style="list-style-type: none">• Agente de soporte desde el Departamento de Informática de la PUCESA |
| Ing. Mg. José Balseca Manzano | <ul style="list-style-type: none">• Agente de pruebas (Tester) |

Fuente: Elaboración propia

Una vez conformado el equipo de trabajo, se procedió a programar reuniones diarias en primer lugar y posteriormente semanales, con la finalidad de analizar el proceso de graduación de posgrados tal y como se llevaba a cabo en su momento, identificando los procedimientos más burocráticos y engorrosos, y por ende que llevaban más tiempo tanto a los maestrantes como al personal del DIP.

Dentro de la metodología XP, el siguiente paso es la elaboración de las historias de usuario, donde se pueden identificar claramente los requerimientos del cliente y priorizarlos, para posteriormente iniciar con el diseño y la codificación del aplicativo a desarrollarse. Estas tarjetas se las obtiene a partir de entrevistas personales con el cliente o su representante, quien como ya se mencionó debe formar parte del equipo de desarrollo en la metodología XP, y en la mayoría de metodologías ágiles.

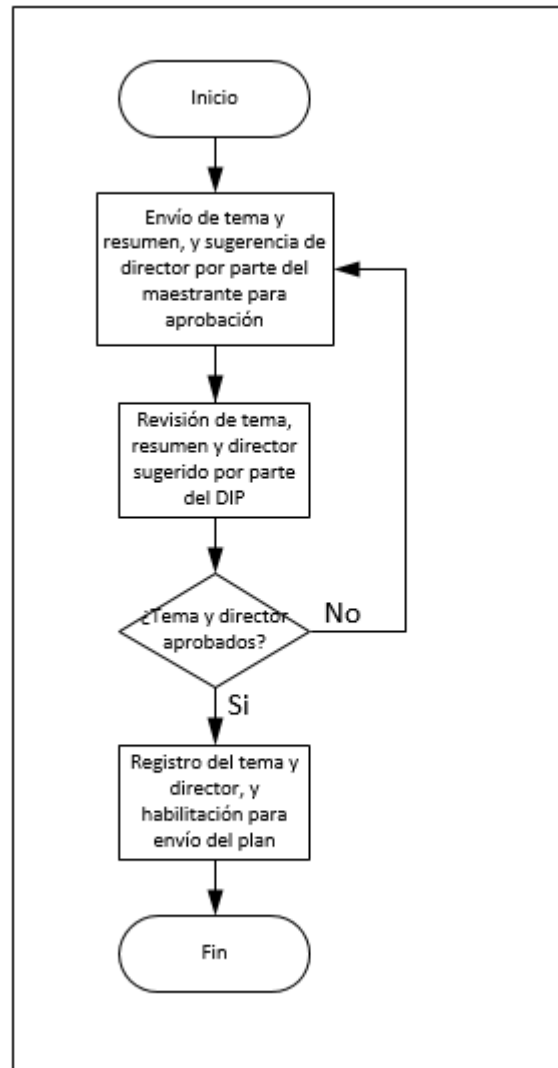
Las historias de usuario suelen contener un conjunto definido de información sobre cada aspecto del proceso o los procesos a automatizar, expresados por el propio cliente, si bien este conjunto de información puede variar dependiendo de las características del proyecto a desarrollar o del problema a resolver. Como mínimo, las historias de usuario deberían contener un número secuencial,

un título o nombre, estimaciones de los niveles de prioridad en el negocio y riesgo en el desarrollo, una estimación de puntos (equivalentes a semanas dentro del proceso de desarrollo), un número de iteración, el cual está relacionado con la prioridad técnica de la historia, y la descripción de la misma, la cual suele proveerla el mismo cliente, y suele llenarse a mano, aunque también puede hacerse en formato digital.

Para el diseño de la presente solución informática, se obtuvo un total de 10 historias de usuario, donde la representante del cliente, en este caso del DIP de la PUCESA, Lcda. Myriam Mayorga, describió detalladamente todos los pasos del proceso de graduación de los maestrantes de la institución, desde el momento cuando presentan su tema de proyecto de titulación para su respectiva aprobación, hasta el momento de la defensa de su proyecto, donde se deben registrar las calificaciones a todos los aspectos establecidos por la PUCESA por parte de los miembros correspondientes del tribunal calificador, esto es del director del proyecto y de los dos docentes lectores.

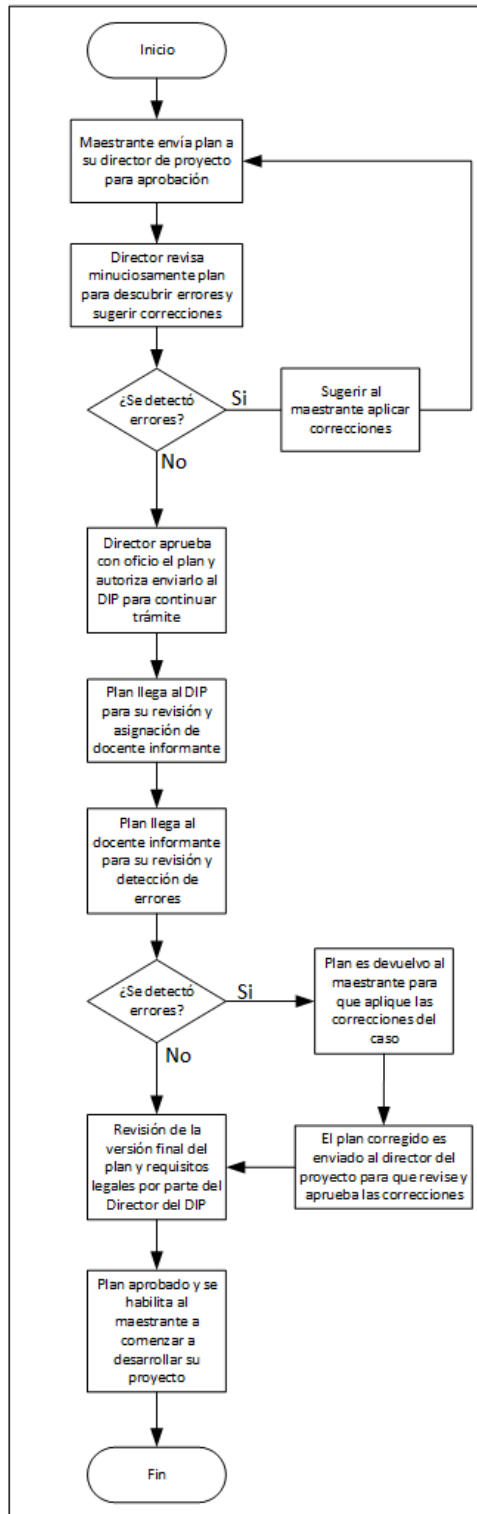
De la información contenida en estas tarjetas se pudo determinar los procesos a los que actualmente se someten los maestrantes de la PUCESA para realizar su proceso de graduación, lo mismos que han sido representados a través de flujogramas de la siguiente manera:

Figura 13: Proceso actual de envío y aprobación de temas de proyectos de titulación y directores de proyecto para maestrantes de la PUCESA



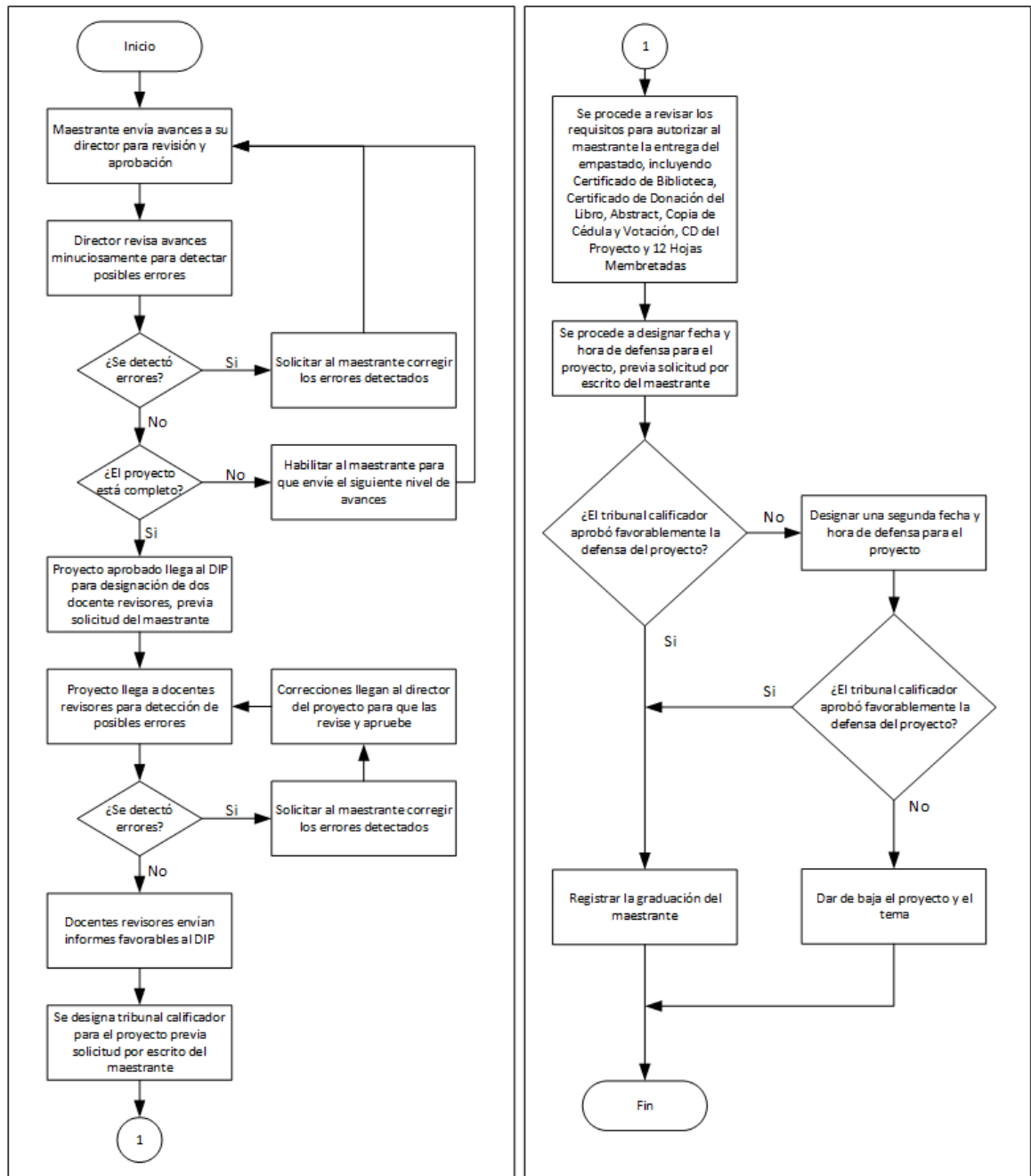
Fuente: elaboración propia

Figura 14: Proceso actual de envío y aprobación de planes de proyectos de titulación que ejecutan los maestrantes de la PUCESA



Fuente: elaboración propia

Figura 15: Proceso actual de envío y aprobación de proyectos de titulación de posgrados, asignación de fechas de defensa y registro de calificaciones



Fuente: elaboración propia

Una vez que se tuvo claro la secuencia de procesos y el flujo de la información tal como se llevaba en su momento dentro del proceso global de graduación de los maestrantes de la PUCESA, se procedió a realizar un análisis exhaustivo de las bases de datos ya existentes en la institución como parte de su sistema Academics, con la finalidad de determinar los datos ya existentes que se pudiera reutilizar en el presente proyecto, y los datos que al no existir deberían ser añadidos, con la finalidad primordial de evitar la duplicidad de los datos, la sobrecarga de los servidores y la inconsistencia en la información.

Lo expuesto se realizó mediante varias reuniones que se llevaron a cabo con el personal del departamento de Informática de la PUCESA, en especial con el Ing. Mg. José Enríquez, durante varias semanas, quien en su calidad de administrador de la base de datos institucional se encargó de proporcionar acceso a los datos ya existentes en la misma mediante la elaboración de múltiples webservices, cuya función es la de extraer dicha información de la base de datos y enviar una copia a la aplicación solicitante, generalmente en formato de tablas de datos.

Del análisis anterior se llegó a determinar que la información ya existente que se podía reutilizar de las bases de datos de la PUCESA era principalmente la relacionada a:

- Carreras de posgrados
- Docentes
- Estudiantes (incluyendo maestrantes)
- Información parcial sobre títulos de los docentes
- Datos de validación de acceso al Academics

Al disponer de toda la información requerida y sobre todo de los webservices requeridos para la elaboración de la presente aplicación, se dio inicio al desarrollo de la presente aplicación web, atendiendo a los principios de desarrollo incremental e iterativo, que son la base de la metodología XP.

4.2.1.1. Fase de planeación del proyecto

Como todo proyecto de desarrollo de software, se inicia por la fase de planeación, donde el principal objetivo es registrar los requerimientos iniciales del cliente, tal y como él los concibe al momento de dar inicio al proyecto. Sin embargo, es importante considerar que dichos

requerimientos pueden ir variando con el pasar del tiempo, por lo que la metodología de desarrollo de software XP se vuelve ideal para el presente proyecto.

4.2.1.1.1. Historias de usuario

La primera etapa dentro de la fase de planificación del proyecto consiste en la elaboración de las historias de usuario, a fin de que el equipo de desarrollo tenga claros los requerimientos del cliente desde su punto de vista de usuario. Dentro del desarrollo del presente proyecto, la principal colaboradora en este sentido fue la Lcda. Miriam Mayorga, quien como secretaria del DIP de la PUCESA proporcionó la información necesaria sobre los procesos de graduación que tienen lugar actualmente en el departamento, obteniéndose un total de 10 tarjetas que se presenta a continuación.

Tabla 3: Historia de usuario 1, solicitud de aprobación del tema

| Historia de Usuario | | | |
|-------------------------------|---|------------------------------|-----------------|
| Número: | 1 | Usuario: | Maestrante, DIP |
| Nombre de la Historia: | Solicitud de aprobación del tema | | |
| Prioridad en Negocio: | Alta | Riesgo en desarrollo: | Alto |
| Puntos estimados: | 4 | Iteración asignada: | 1 |
| Descripción: | El maestrante solicita por escrito la aprobación de su tema de proyecto de investigación y desarrollo de grado y conjuntamente propone un docente tutor. Esta solicitud es procesada por el DIP, cuyo director procede a aprobar el tema, y generalmente acepta el docente tutor propuesto, aunque de ser el caso se puede designar a otro docente. | | |
| Observaciones: | | | |
| | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 4: Historia de usuario 2, elaboración del plan

| Historia de Usuario | | | |
|-------------------------------|---|------------------------------|---------------------|
| Número: | 2 | Usuario: | Maestrante, Docente |
| Nombre de la Historia: | Elaboración del plan | | |
| Prioridad en Negocio: | Alta | Riesgo en desarrollo: | Alto |
| Puntos estimados: | 8 | Iteración asignada: | 2 |
| Descripción: | El maestrante procede a elaborar su plan de proyecto de investigación y desarrollo de grado, bajo la guía y dirección de su docente tutor designado, quien a su vez deberá revisar y aprobar dicho plan. El docente deberá entonces emitir un oficio dirigido al DIP indicando que ha revisado y aprueba el plan en cuestión. | | |
| Observaciones: | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 5: Historia de usuario 3, solicitud de docente informante

| Historia de Usuario | | | |
|-------------------------------|---|------------------------------|-----------------|
| Número: | 3 | Usuario: | Maestrante, DIP |
| Nombre de la Historia: | Solicitud de docente informante | | |
| Prioridad en Negocio: | Media | Riesgo en desarrollo: | Medio |
| Puntos estimados: | 4 | Iteración asignada: | 2 |
| Descripción: | El maestrante solicita la designación de docente informante, para lo cual debe presentar su plan y el oficio de revisión y aprobación emitido por su docente tutor. La dirección del DIP procede entonces a designar un docente informante quien también deberá revisar el plan y sugerir las correcciones del caso. Una vez corregido, el maestrante debe presentar su solicitud de aprobación definitiva del plan, lo cual es gestionado internamente por el DIP. | | |
| Observaciones: | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 6: Historia de usuario 4, solicitud de docente director

| Historia de Usuario | | | |
|-------------------------------|--|------------------------------|-----------------|
| Número: | 4 | Usuario: | Maestrante, DIP |
| Nombre de la Historia: | Solicitud de docente director | | |
| Prioridad en Negocio: | Baja | Riesgo en desarrollo: | Bajo |
| Puntos estimados: | 2 | Iteración asignada: | 3 |
| Descripción: | Una vez que el plan ha sido definitivamente aprobado por la dirección del DIP, el maestrante debe presentar su solicitud de designación de docente director. La dirección del DIP procede a designar un docente, quien generalmente es el mismo que venía desempeñándose como tutor en las fases de tema y plan, aunque por diversos motivos se puede designar a otro docente. | | |
| Observaciones: | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 7: Historia de usuario 5, matrícula para desarrollo del proyecto

| Historia de Usuario | | | |
|-------------------------------|---|------------------------------|-------|
| Número: | 5 | Usuario: | Todos |
| Nombre de la Historia: | Matrícula para desarrollo del proyecto | | |
| Prioridad en Negocio: | Baja | Riesgo en desarrollo: | Bajo |
| Puntos estimados: | 2 | Iteración asignada: | 4 |
| Descripción: | El maestrante debe presentarse en el DIP para registrar su matrícula para iniciar con el desarrollo del proyecto. Para esto deberá cancelar los aranceles y llenar las hojas de crédito correspondientes. Hecho esto, el maestrante puede iniciar con el desarrollo del proyecto bajo la guía de su docente director, enviándole avances periódicos al mismo quien a su vez, de ser el caso, deberá emitir un certificado sobre el porcentaje total de avance que presenta el maestrante. | | |
| Observaciones: | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 8: Historia de usuario 6, solicitud de docentes revisores

| Historia de Usuario | | | |
|-------------------------------|--|------------------------------|-------|
| Número: | 6 | Usuario: | Todos |
| Nombre de la Historia: | Solicitud de docentes revisores | | |
| Prioridad en Negocio: | Alta | Riesgo en desarrollo: | Alto |
| Puntos estimados: | 8 | Iteración asignada: | 5 |
| Descripción: | Una vez concluido el proyecto, el maestrante debe solicitar la designación de docentes revisores, quienes serán designados por el DIP. Son dos docentes que deberán revisar el IFP y sugerir las correcciones del caso, cuantas veces sea necesario. Una vez que el IFP se encuentre a satisfacción de ambos docentes revisores, éstos procederán a emitir sus respectivos informes favorables, que deberán ser entregados en la secretaría del DIP. | | |
| Observaciones: | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 9: Historia de usuario 7, solicitud de tribunal calificador

| Historia de Usuario | | | |
|-------------------------------|--|------------------------------|-----------------|
| Número: | 7 | Usuario: | Maestrante, DIP |
| Nombre de la Historia: | Solicitud de tribunal calificador | | |
| Prioridad en Negocio: | Baja | Riesgo en desarrollo: | Bajo |
| Puntos estimados: | 2 | Iteración asignada: | 6 |
| Descripción: | El maestrante debe acercarse a la secretaría del DIP para solicitar designación de tribunal calificador para la defensa de su proyecto de investigación y desarrollo de grado. El único requisito para esto es la solicitud en hoja membretada. La dirección del DIP designará entonces tres docentes para conformar el tribunal calificador, los mismos que podrán ser cambiados más adelante de requerirse por cualquier motivo. | | |
| Observaciones: | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 10: Historia de usuario 8, revisión en biblioteca para empastado

| Historia de Usuario | | | |
|-------------------------------|---|------------------------------|-----------------|
| Número: | 8 | Usuario: | Maestrante, DIP |
| Nombre de la Historia: | Revisión en biblioteca para empastado | | |
| Prioridad en Negocio: | Media | Riesgo en desarrollo: | Bajo |
| Puntos estimados: | 2 | Iteración asignada: | 7 |
| Descripción: | El maestrante debe realizar varias actividades como obtener su certificado de biblioteca, certificado de donación del libro y aprobación del abstract. Debe presentarse con estos requisitos además del CD del proyecto y 12 hojas membretadas en la dirección del DIP para registrar dicha revisión. | | |
| Observaciones: | | | |
| | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 11: Historia de usuario 9, solicitud de fecha y hora de defensa

| Historia de Usuario | | | |
|-------------------------------|--|------------------------------|-----------------|
| Número: | 9 | Usuario: | Maestrante, DIP |
| Nombre de la Historia: | Solcitud de fecha y hora de defensa | | |
| Prioridad en Negocio: | Media | Riesgo en desarrollo: | Medio |
| Puntos estimados: | 2 | Iteración asignada: | 8 |
| Descripción: | El maestrante debe presentarse en la secretaría del DIP con su solicitud por escrito, en hoja membretada, de designación de fecha y hora de defensa para su proyecto. Esta fecha y hora serán designadas por el DIP, pudiendo cambiarse más adelante si por cualquier motivo esto fuera requerido. | | |
| Observaciones: | | | |
| | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

Tabla 12: Historia de usuario 10, defensa del proyecto

| Historia de Usuario | | | |
|-------------------------------|---|------------------------------|-------|
| Número: | 10 | Usuario: | Todos |
| Nombre de la Historia: | Defensa del proyecto | | |
| Prioridad en Negocio: | Alta | Riesgo en desarrollo: | Medio |
| Puntos estimados: | 4 | Iteración asignada: | 9 |
| Descripción: | El maestrante debe presentarse a la fecha y hora designadas para la defensa de su proyecto de investigación y desarrollo de grado. Luego de esto, se generarán las actas de grado que serán el requisito para registrar la graduación del maestrante. Si por cualquier motivo fallara la defensa, el maestrante podrá solicitar una segunda fecha y hora, y si en esta segunda oportunidad volviera a fallar, se dará de baja el tema y el proyecto de grado. | | |
| Observaciones: | | | |
| | | | |
| | | | |
| | | | |

Fuente: Elaboración propia

4.2.1.1.2. Plan de entregas

Se procede a analizar detenidamente, ordenar y clasificar las historias de usuario considerándose los módulos que integrarían la aplicación a desarrollarse, a fin de definir prioridades y tiempos de entrega, estableciéndose las iteraciones iniciales y posteriores incrementos que se llevará a cabo, atendiendo a los principios de la metodología de desarrollo de software XP. Por tanto, a cada historia de usuario se ha asignado un nivel de prioridad en el negocio así como uno de riesgo para el desarrollo, y también un número de puntos que representa una estimación inicial del tiempo que puede tomar la solución del requerimiento en cuestión, expresado en semanas.

Se considera también que para el desarrollo del presente proyecto se cuenta con un equipo relativamente pequeño y se cuenta con una sola programadora, por lo que los tiempos inicialmente estimados podrían extenderse, en especial si se tiene en cuenta que los requerimientos del DIP de la PUCESA tienden a cambiar con cierta frecuencia a lo largo del tiempo, lo que es un fenómeno previsto dentro de la metodología de desarrollo de software XP.

Adicionalmente a esto se establece que, tanto debido a la disponibilidad de tiempo del personal del DIP de la PUCESA, así como de la programadora, el personal del Departamento de Informática de

la PUCESA, y el resto de integrantes del equipo de trabajo, se destinará al desarrollo del presente proyecto una ventana de trabajo de 3 horas diarias durante los 5 días laborables de la semana, y durante 4 semanas al mes. Esta consideración de tiempo incluye reuniones del equipo de trabajo cuando sea pertinente, trabajo autónomo de desarrollo de código y pruebas, movilización hacia las instalaciones de la PUCESA cuando se requiera, y cualquier actividad adicional relacionada con cualquiera de las fases de la metodología de desarrollo de software XP.

Atendiendo a estas consideraciones, se procede a elaborar el plan de entregas para el presente proyecto, el mismo que queda expresado en la siguiente tabla:

Tabla 13: Plan de entregas por historias de usuario

| Módulo | Historia de usuario | Tiempo estimado | | | Iteración | | | | | | | | | |
|---|---|-----------------|----|-----|-----------|---|---|---|---|---|---|---|---|--|
| | | S. | D. | H. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| Tema | 1. Solicitud de aprobación del tema | 4 | 20 | 60 | X | | | | | | | | | |
| Plan | 2. Elaboración del plan | 8 | 40 | 120 | | X | | | | | | | | |
| | 3. Solicitud de docente informante | 4 | 20 | 60 | | X | | | | | | | | |
| Informe Final del Proyecto (IFP) | 4. Solicitud de docente director | 2 | 10 | 30 | | | X | | | | | | | |
| | 5. Matrícula para desarrollo del proyecto | 2 | 10 | 30 | | | | X | | | | | | |
| | 6. Solicitud de docentes revisores | 8 | 40 | 120 | | | | | X | | | | | |
| Defensa | 7. Solicitud de tribunal calificador | 2 | 10 | 30 | | | | | | X | | | | |
| | 8. Revisión en biblioteca para empastado | 2 | 10 | 30 | | | | | | | X | | | |
| | 9. Solicitud de fecha y hora de defensa | 2 | 15 | 45 | | | | | | | | X | | |
| | 10. Defensa del proyecto | 4 | 20 | 60 | | | | | | | | | X | |

Fuente: Elaboración propia

4.2.1.2. Fase de diseño del proyecto

Una vez recolectados todos los datos que reflejan las necesidades iniciales del usuario, la siguiente etapa considerada dentro de la metodología XP es la fase de diseño, cuyo objetivo es crear las bases que van a servir de guía y soporte para dar inicio al desarrollo de la codificación.

4.2.1.2.1. Metáfora de la aplicación

Dentro de la metodología de desarrollo de software XP, el diseño del proyecto implica la elaboración de la metáfora de la aplicación. Para el presente caso, la aplicación se ha dividido en diferentes módulos, de acuerdo a lo detallado en la tabla anterior. Se elabora una metáfora para cada uno de ellos, ya que resulta imposible elaborar una sola para todo el proyecto, puesto que abarca demasiadas secciones.

Para el módulo Tema, la aplicación se encarga de permitirle al maestrante el envío de su tema de proyecto de titulación, conjuntamente con su resumen inicial y su propuesta de docente director del proyecto. Como docente, el director propuesto puede aceptar o rechazar la dirección del tema en cuestión, y en caso de aceptarla, puede solicitarle al maestrante corregir el tema o resumen, o puede aprobarlos. Luego, el director del DIP debe también revisar el tema y puede enviar a hacer correcciones o aprobar el tema, pudiendo también solicitar el cambio de director para el mismo.

El módulo Plan permite al estudiante enviar su plan de proyecto de titulación conjuntamente con todas sus secciones. Como docente, el director del proyecto puede revisar el mismo y enviar a hacer correcciones al maestrante las veces que sea necesario. Una vez que esté de acuerdo, aprueba el plan y lo envía al DIP. El director del DIP puede entonces proceder a nombrar un docente informante para dicho plan. Como informante, el docente también puede revisar completamente el plan y sugerir correcciones para el mismo por una sola vez. El maestrante tiene nuevamente acceso a la edición de su plan para aplicar las correcciones de la manera en que considere conveniente conjuntamente con su director del proyecto, quien debe aprobar también las correcciones. Finalmente, el director del DIP puede proceder a aprobar el plan de proyecto de titulación.

El siguiente módulo corresponde a la elaboración del informe final del proyecto. El maestrante puede enviar avances del proyecto a su docente director, quien puede revisar y solicitar correcciones las veces que considere necesarias para los avances de 25%, 50%, 75% y 100%. Una vez que el director aprueba el 100% del proyecto, el maestrante queda habilitado para poder entregar su

trabajo en el DIP, cumpliendo con todos los requisitos del caso, y solicitar la designación de dos docentes revisores. Como revisores, ambos docentes pueden revisar el trabajo enviado por el maestrante y sugerir la aplicación de correcciones, las veces que sea necesario. El maestrante puede entonces enviar el informe corregido a cualquiera de los dos docentes lectores que lo haya solicitado, las veces que sea necesario. Una vez que los revisores consideran que el informe se encuentra correcto, pueden proceder a aprobarlo, con lo que el maestrante queda habilitado para continuar con el siguiente paso del proyecto, que sería la solicitud de tribunal calificador, constituido a su vez por tres miembros, generalmente el mismo director del proyecto y sus dos docentes revisores.

El siguiente módulo está dirigido a la defensa del proyecto de titulación. El director del DIP o su persona designada pueden ingresar a la aplicación para designar el mencionado tribunal calificador, registrar el cumplimiento de los requisitos de biblioteca para la elaboración del empastado, registrar la solicitud y designación de fecha y hora para la defensa y registrar el resultado de la defensa, sea este en el sentido de aprobación o reprobación. El personal del DIP también tiene la posibilidad de dar de baja un determinado tema de proyecto de titulación.

El último módulo de la aplicación se encarga de la generación de un gráfico estadístico que refleja en tiempo real el número de maestrantes que se encuentra en cada punto detallado del proceso de graduación. Este gráfico puede ser filtrado por cada una de las diferentes carreras de posgrados o ser aplicado sobre la totalidad de maestrantes indistintamente de sus carreras. El personal del DIP tiene también la posibilidad de exportar este gráfico a formato PDF, a fin de que el mismo pueda ser guardado como archivo y fácilmente impreso. Este gráfico estadístico es una herramienta clave que indiscutiblemente sirve como herramienta de apoyo en la toma de decisiones para la gerencia – administración del DIP.

Adicionalmente, la aplicación implementa un servicio de mensajería interna entre sus usuarios, que permite que tanto maestrantes como docentes emitan su opinión, comentarios o instrucciones para las demás personas que guardan relación con un determinado proyecto de titulación. Estos mensajes a su vez son desplegados en orden cronológico para que todos puedan conocer cómo se ha ido desarrollando el proyecto. Existen también mensajes que son automáticamente generados por la aplicación, los que son especialmente útiles para las operaciones realizadas por el personal del DIP.

4.2.1.2.2. Tarjetas Clase – Responsabilidad – Colaboración (CRC)

Al analizar las historias de usuario se logra identificar a priori las clases primarias que van a formar parte de la presente aplicación, pudiéndose citar las siguientes:

- Temas de proyectos
- Planes de proyectos
- Avances de informes finales
- Proyectos de graduación
- Docentes
- Estudiantes
- Dirección del DIP

Con estas clases base se procede a analizar las responsabilidades de cada una de ella y la forma como deben estar relacionadas entre sí, para la respectiva elaboración de las tarjetas CRC, que se listan a continuación:

Figura 16: Tarjeta CRC “TemadeTesis”

| TemadeTesis | |
|--------------------------------|------------|
| Guardar información sobre | Docente |
| temas de proyectos incluyendo | Estudiante |
| resumen, director, estudiante, | |
| estado y fecha de aprobación | |
| | |
| | |

Fuente: Elaboración propia

Figura 17: Tarjeta CRC “PlandeTesis”

| PlandeTesis | |
|---|--------------------------------------|
| Guardar información sobre planes de proyectos incluyendo fecha de aprobación y estado | Docente Estudiante TemadeTesis |
| | |
| | |
| | |
| | |

Fuente: Elaboración propia

Figura 18: Tarjeta CRC “AvancesTesis”

| AvancesTesis | |
|--|----------------|
| Guardar avances del IFP incluyendo porcentajes, fechas de aprobación y estados | TesisPosgrados |
| | |
| | |
| | |
| | |

Fuente: Elaboración propia

Figura 19: Tarjeta CRC “TesisPosgrados”

| TesisPosgrados | |
|--|--------------|
| Guardar información sobre proyectos de titulación incluyendo fecha de aprobación, calificaciones, fecha y hora de defensa y estado | AvancesTesis |
| | |
| | |
| | |
| | |

Fuente: Elaboración propia

Figura 20: Tarjeta CRC “Docente”

| Docente | |
|-----------------------------|-----------------------|
| Aceptar dirección | TemadeTesis |
| Aprobar temas | SolicitudAprobacionTT |
| Aprobar planes | DirectorDIP |
| Aprobar avances | AvancesTesis |
| Revisar planes como D. inf. | PlandeTesis |
| Revisar IFP como D. lec. | |
| Calificar IFP como M. trib. | |

Fuente: Elaboración propia

Figura 21: Tarjeta CRC “Estudiante”

| Estudiante | |
|---|----------------------------------|
| Solicitar aprobación y dirección del tema | SolicitudAprobaciónTT Docente |
| Solicitar aprobación del plan | PlandeTesis |
| Enviar avances del IFP | AvancesTesis |
| | DirectorDIP |
| | |
| | |

Fuente: Elaboración propia

Figura 22: Tarjeta CRC “DirectorDIP”

| DirectorDIP | |
|---------------------------|-----------------------|
| Aprobar temas | TemadeTesis |
| Aprobar directores | SolicitudAprobacionTT |
| Aprobar planes | PlandeTesis |
| Asignar D. informantes | Docente |
| Asignar D. lectores | TesisPosgrados |
| Asignar fechas de defensa | |
| Asignar miembros trib. | |

Fuente: Elaboración propia

Cabe mencionar que estas tarjetas identifican únicamente las clases primarias que se ha identificado del análisis de las historias de usuario, sin embargo, durante el desarrollo del proyecto pueden identificarse clases secundarias que sean necesarias para apoyar a las primarias, y se implementan según se vayan requiriendo. De igual forma, el conglomerado total de las clases identificadas sirve de base para la elaboración de las bases de datos y sus respectivas tablas, relaciones y demás elementos que darán soporte a la aplicación desarrollada.

4.2.1.3. Fase de codificación y desarrollo del proyecto

Una vez identificadas las clases primarias que intervienen dentro de los procesos de graduación del DIP de la PUCESA, las responsabilidades de cada clase y las relaciones existentes entre ellas, se da inicio a la codificación propiamente dicha de la aplicación web destinada a optimizar dichos procesos.

Se inicia por el módulo denominado Tema, siguiendo el orden de iteraciones ya establecido en la planificación de entregas, procurando satisfacer el requerimiento central de cada iteración, a la vez que mantener la sencillez y claridad, tanto en las interfaces de usuario como en el código. Una vez logrado un código que funcione y cumpla el requerimiento, se procede a validar el mismo conjuntamente con el cliente de la aplicación, en este caso el personal del DIP de la PUCESA. De requerirse, se vuelve a modificar el código para lograr nueva funcionalidad o mejorar la existente, en una nueva iteración. Más adelante se procede a detallar las pruebas realizadas con sus respectivos resultados para todos los módulos de la aplicación web. Una vez que se termina de desarrollar y probar satisfactoriamente un módulo, luego de las iteraciones que esto demande, se procede con el siguiente módulo iniciando nuevamente desde una primera iteración, atendiendo al principio del desarrollo incremental que demanda la metodología de desarrollo de software XP.

4.2.1.3.1. Refactorización y clarificación del código

Una vez que se termina de codificar cada iteración, y se realizan las respectivas pruebas funcionales, se procede a revisar el código en busca de errores de forma o de bloques repetidos que se pueda eliminar, atendiendo a los principios de refactorización y reaprovechamiento del código. Esto a su vez permite generar archivos más ligeros y eficientes, reduciendo así la carga del servidor y mejorando notablemente la usabilidad del sistema.

Por otro lado, se adopta la buena práctica de introducir comentarios explicativos para las funciones, métodos, miembros de clases e inclusive algunas instrucciones individuales, con la finalidad de crear un código lo más entendible posible ya que, al tratarse de un código que será de propiedad de la PUCESA, es muy importante que cualquier programador que posteriormente sea asignado por la institución para revisar y modificar el código entienda lo que se ha codificado y para qué sirve cada función y método, pues de no ser por dichos comentarios suele ser muy difícil entender la lógica y flujo de información empleados por otro programador.

Es muy importante además mencionar que, si bien la metodología de desarrollo de software XP fomenta la creación de un repositorio de código, de preferencia con un sistema de control de versiones, aunque este proceso no es obligatorio, a fin de que otros programadores integrantes del grupo de desarrollo puedan entender fácilmente lo que se ha codificado, en el caso del presente proyecto no se ha elaborado dicho repositorio, puesto que en el equipo de trabajo se cuenta únicamente con una programadora (Ing. Gabriela De la Torre Sánchez), por lo que carece de sentido también el control de versiones. Se decide por tanto dejar la correcta explicación del código a través de los mencionados comentarios como medio de ayuda para futuros programadores que deseen analizar el mismo.

4.2.1.3.2. Diseño e implementación de bases de datos y clases

A partir de las tarjetas CRC que se elaboró en una etapa anterior se procedió a identificar las clases primarias que forman parte del esquema de desarrollo del presente proyecto. Dichas clases sirven de base para el diseño tanto de bases de datos como de clases auxiliares que son necesarias para el correcto funcionamiento de la presente aplicación web. En el Apéndice A se puede leer la descripción detallada de las bases de datos y tablas creadas para esta aplicación, con su correspondiente descripción de campos.

De igual forma, en el Apéndice B se detallan las clases que se ha creado para la presente aplicación, en especial la clase `UsuarioGraduacionAtencion.cs`, que sirve como apoyo para el proceso de login de los usuarios a la aplicación, así como las clases tipo manager que son aquellas que permiten guardar y recuperar la información de las bases de datos. El listado de dichas clases también se puede observar en el Apéndice B.

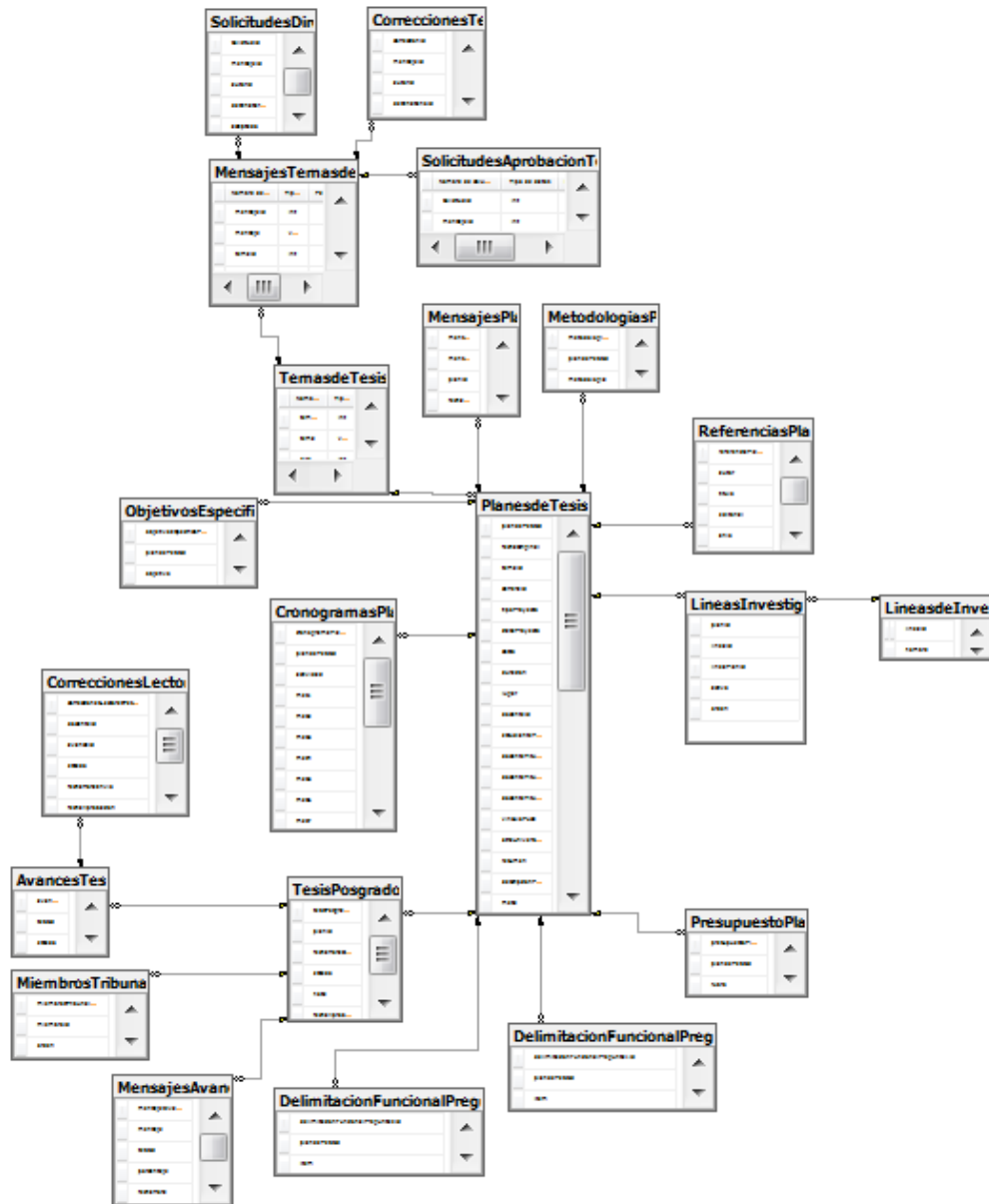
Además es muy importante recalcar que, al tratarse de una aplicación que toma como caso de estudio el proceso de graduación de posgrados de la PUCESA, y por tanto utiliza mucha de la información ya existente en la institución, como la relacionada a docentes y maestrantes, se ha establecido conjuntamente con el personal del departamento de informática de la institución que la mejor forma de acceder a dichos datos es mediante la habilitación de webservices, tarea a cargo del mencionado personal. Al igual que en los casos anteriores, la descripción de dichos webservices puede observarse en el Apéndice C del presente documento.

Por último, cabe mencionar que al tratarse de un proyecto desarrollado usando la plataforma de desarrollo ASP.NET y el IDE de Visual Studio 2010, se ha procedido a emplear la técnica de programación por capas, creándose tres proyectos dentro de la misma solución. El primero se denomina Logic y se encarga de almacenar las clases auxiliares y tipo managers que apoyan a la aplicación, mientras que el segundo proyecto recibe el nombre de Model y su función es la de proveer la conexión a las bases de datos creadas para la aplicación. Finalmente, el tercer proyecto se denomina Ui.Web, y se encarga de almacenar las páginas web creadas para la aplicación, así como la estructura de directorios que formará parte del sitio web una vez que sea publicado. Por supuesto, los tres proyectos se encuentran interconectados e interrelacionados, pudiendo cualquiera de ellos hacer uso de la funcionalidad que provee otro.

4.2.1.3.3. Diagrama de clases

Como resultado del análisis antes expuesto se obtiene el diagrama de clases, que no es otra cosa que una representación gráfica de las clases primarias y secundarias identificadas durante el desarrollo de la aplicación, así como las relaciones existentes entre sí. Desde el punto de vista de la base de datos, este diagrama representa también el modelo entidad – relación, que indica las tablas y campos que se incluye en la base de datos principal y de igual forma sus relaciones.

Figura 23: Diagrama de clases



Fuente: elaboración propia

4.2.1.3.4. Ejemplos de la codificación

El siguiente bloque de código corresponde a un extracto del archivo AvancesdeTesisManager.cs, el mismo que representa a una de las clases tipo manager que a su vez se encarga del envío y de la

extracción de información de la tabla AvancesdeTesis, de la base de datos Graduacion. Se incluyen 5 de sus métodos como ejemplo.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Graduacion.Model;
using Graduacion.Logic.wspucesa;

namespace Graduacion.Logic
{
    public class AvancesdeTesisManager
    {
        public static AvancesTesis GetAvancedeTesisbyId(int avancedeTesisId)
        {
            try
            {
                using (var db = new GraduacionEntities())
                {
                    return db.AvancesTesis.Where(p => p.avanceTesisId ==
avancedeTesisId).FirstOrDefault();
                }
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }
        }

        public static List<AvancesTesis> GetAvancesdeTesisbyTesisId(int tesisId)
        {
            try
            {
                using (var db = new GraduacionEntities())
                {
                    List<AvancesTesis> lista = db.AvancesTesis.Where(p => p.tesisId ==
tesisId).OrderByDescending(q => q.porcentaje).ToList();
                    return lista;
                }
            }
            catch
            {
                return null;
            }
        }
    }
}
```

```

public static AvancesTesis CreateAvancebyTesisPosgradosId(int tesisId, int porcentaje)
{
    try
    {
        AvancesTesis avance = new AvancesTesis();
        avance.tesisId = tesisId;
        avance.estado = 0;
        avance.porcentaje = porcentaje;
        avance.iteracion = 0;
        using (var db = new GraduacionEntities())
        {
            db.AvancesTesis.AddObject(avance);
            db.SaveChanges();
        }
        return avance;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

```

```

public static int CreateAvancebyObject(AvancesTesis avance)
{
    try
    {
        using (var db = new GraduacionEntities())
        {
            db.AvancesTesis.AddObject(avance);
            db.SaveChanges();
            return avance.avanceTesisId;
        }
    }
    catch
    {
        return 0;
    }
}

```

```

public static bool Update(AvancesTesis avance)
{
    try
    {
        using (var db = new GraduacionEntities())
        {
            AvancesTesis _avance = GetAvancedeTesisbyId(avance.avanceTesisId);
            db.AttachTo("AvancesTesis", _avance);
            db.AvancesTesis.ApplyCurrentValues(avance);
            db.SaveChanges();
            return true;
        }
    }
}

```

```

    }
    catch
    {
        return false;
    }
}

```

```

public static String GetCarreraNombrebyAvanceId(int avanceId)
{
    try
    {
        using (wsSeguimientoTesisSoapClient ws = new wsSeguimientoTesisSoapClient())
        {
            var db = new GraduacionEntities();
            AvancesTesis avance = db.AvancesTesis.Where(p => p.avanceTesisId ==
avanceId).FirstOrDefault();
            int carrerald = (int)avance.TesisPosgrados.PlanesdeTesis.carrerald;
            return PlandeTesisManager.GetCarreraNombrebyId(carrerald);
        }
    }
    catch
    {
        return "";
    }
}

```

```

public static int GetInformanteldbyAvanceId(int avanceId)
{
    try
    {
        using (var db = new GraduacionEntities())
        {
            AvancesTesis avance = db.AvancesTesis.Where(p => p.avanceTesisId ==
avanceId).FirstOrDefault();
            return (int)avance.TesisPosgrados.PlanesdeTesis.docenteLectorId;
        }
    }
    catch
    {
        return 0;
    }
}
}

```

4.2.1.4. Pruebas funcionales

Las siguientes tablas describen las pruebas de usuario y funcionales que se han ido realizando durante el desarrollo del presente proyecto al final de cada iteración, con los respectivos resultados obtenidos:

Tabla 14: Pruebas de usuario y funcionales ejecutadas sobre el módulo Tema

| PRUEBA | RESULTADO |
|---|-----------|
| Tema del proyecto se guarda correctamente en la base de datos | Sí |
| Tema del proyecto se modifica correctamente en la base de datos | Sí |
| Resumen del tema del proyecto se guarda correctamente en la base de datos | Sí |
| Resumen del tema del proyecto se modifica correctamente en la base de datos | Sí |
| Lista de docentes para la dirección del proyecto se despliega correctamente con el número de proyectos que se encuentra dirigiendo cada uno | Sí |
| Mensajes del maestrante se guardan correctamente en la base de datos | Sí |
| Mensajes de los docentes se guardan correctamente en la base de datos | Sí |
| Mensajes del director del DIP se guardan correctamente en la base de datos | Sí |
| Lista de mensajes guardados se despliega correctamente en orden cronológico | Sí |
| Docente obtiene correctamente la lista de solicitudes de dirección de proyecto | Sí |
| Docente puede aceptar o rechazar correctamente la dirección del proyecto | Sí |
| Docente puede solicitar correcciones al tema o resumen correctamente | Sí |
| Docente puede aprobar correctamente el tema | Sí |
| Director del DIP puede solicitar correcciones al tema o resumen correctamente | Sí |
| Director del DIP puede solicitar correctamente cambio de director de proyecto | Sí |
| Director del DIP puede aprobar definitivamente el tema en forma correcta | Sí |
| Todos los documentos correspondientes se generan correctamente | Sí |
| Todos los usuarios que tienen acceso a esta sección pueden descargar y visualizar correctamente los documentos generados | Sí |

Fuente: elaboración propia

Tabla 15: Pruebas de usuario y funcionales ejecutadas sobre el módulo Plan

| PRUEBA | RESULTADO |
|--|------------------|
| Plan del proyecto se guarda correctamente en la base de datos | Sí |
| Plan del proyecto se modifica correctamente en la base de datos | Sí |
| Líneas de investigación se guardan correctamente en la base de datos | Sí |
| Líneas de investigación se eliminan correctamente de la base de datos | Sí |
| Objetivos específicos se guardan correctamente en la base de datos | Sí |
| Objetivos específicos se eliminan correctamente de la base de datos | Sí |
| Metodologías específicas se guardan correctamente en la base de datos | Sí |
| Metodologías específicas se eliminan correctamente de la base de datos | Sí |
| Delimitaciones funcionales se guardan correctamente en la base de datos | Sí |
| Delimitaciones funcionales se eliminan correctamente de la base de datos | Sí |
| Referencias bibliográficas se guardan correctamente en la base de datos | Sí |
| Referencias bibliográficas se eliminan correctamente de la base de datos | Sí |
| Rubros del presupuesto se guardan correctamente en la base de datos | Sí |
| Rubros del presupuesto se eliminan correctamente de la base de datos | Sí |
| Actividades del cronograma se guardan correctamente en la base de datos | Sí |
| Actividades del cronograma se eliminan correctamente de la base de datos | Sí |
| La información preexistente se despliega correctamente en el editor del plan | Sí |
| Se listan correctamente las líneas de investigación para su adición al plan | Sí |
| Los títulos de tercer y cuarto nivel para docentes y maestrantes se listan correctamente | Sí |
| Mensajes del maestrante se guardan correctamente en la base de datos | Sí |
| Mensajes de los docentes se guardan correctamente en la base de datos | Sí |
| Mensajes del director del DIP se guardan correctamente en la base de datos | Sí |
| Lista de mensajes guardados se despliega correctamente en orden cronológico | Sí |
| Docente obtiene correctamente la lista de planes pendientes de revisar | Sí |
| Docente puede solicitar correcciones al plan correctamente | Sí |
| Docente puede aprobar correctamente el plan | Sí |
| Director del DIP puede solicitar correcciones al plan correctamente | Sí |

| | |
|--|----|
| Director del DIP puede aprobar definitivamente el plan en forma correcta | Sí |
| Todos los documentos correspondientes se generan correctamente | Sí |
| Todos los usuarios que tienen acceso a esta sección pueden descargar y visualizar correctamente los documentos generados | Sí |

Fuente: elaboración propia

Tabla 16: Pruebas de usuario y funcionales ejecutadas sobre los módulos IFP y Defensa

| PRUEBA | RESULTADO |
|--|-----------|
| Los archivos enviados por el maestrante se guardan correctamente en el servidor | Sí |
| El maestrante puede enviar múltiples archivos como parte de un avance | Sí |
| El maestrante puede eliminar archivos cargados antes de enviarlos | Sí |
| Resumen del tema del proyecto se modifica correctamente en la base de datos | Sí |
| El nivel de avances se despliega correctamente en las páginas | Sí |
| Mensajes del maestrante se guardan correctamente en la base de datos | Sí |
| Mensajes de los docentes se guardan correctamente en la base de datos | Sí |
| Mensajes del director del DIP se guardan correctamente en la base de datos | Sí |
| Lista de mensajes guardados se despliega correctamente en orden cronológico | Sí |
| Docente (director o lector) obtiene correctamente la lista de avances pendiente de revisión | Sí |
| Docente (director o lector) puede solicitar correcciones a avances correctamente | Sí |
| Docente puede descargar archivos enviados en iteraciones anteriores para realizar comparaciones | Sí |
| Docente puede aprobar correctamente el avance correspondiente | Sí |
| Cuando el docente aprueba el avance del 100%, los archivos correspondientes son enviados al DIP para revisión | Sí |
| Director del DIP puede solicitar correcciones a avances correctamente | Sí |
| Director del DIP puede asignar correctamente docentes lectores a avances | Sí |
| Director del DIP puede asignar fecha y hora de defensa correctamente | Sí |
| Docentes pueden registrar notas de la defensa del proyecto correctamente, según disposiciones de la PUCESA | Sí |
| Todos los documentos correspondientes se generan correctamente | Sí |
| Todos los usuarios que tienen acceso a esta sección pueden descargar y visualizar correctamente los documentos generados | Sí |

Fuente: elaboración propia

4.3. Materiales y herramientas

Como se ha venido mencionando a lo largo del presente documento, el objetivo principal del aplicativo a desarrollarse en este proyecto es complementar o incrementar la funcionalidad del ya existente sistema conocido como Academics, por lo que ha sido necesario poner especial atención en que cada archivo, página web, tabla de base de datos y demás elementos a generarse sea elaborado utilizando las mismas herramientas y bajo los mismos estándares que Academics, lo cual implica realizar el desarrollo utilizando las mismas herramientas, versiones y lenguajes utilizados originalmente para el desarrollo de Academics y bajo las cuales se ejecuta dicha aplicación actualmente en los servidores de la PUCESA.

Como motor de bases de datos se ha utilizado Microsoft SQL Server 2008 R2 [24], que es un potente motor de bases de datos que ha ido evolucionando favorablemente desde sus primeras versiones lanzadas al mercado allá a finales de la década de los 80. Este motor de bases de datos utiliza el lenguaje Transact-SQL (TSQL) tanto para la creación y mantenimiento de tablas y otras estructuras de datos, así como para la manipulación de los datos. Sin embargo, incluye también una aplicación gráfica conocida como Management Studio, que permite realizar las mencionadas operaciones en un entorno visual mucho más atractivo e intuitivo, que permite la generación de las estructuras de datos requeridas sin necesidad de dominar el lenguaje TSQL.

Microsoft SQL Server 2008 R2 también permite la creación y el manejo de relaciones entre tablas de una base de datos de manera simple e intuitiva, lo cual resulta muy conveniente para el desarrollo de la presente aplicación.

Por otro lado, como entorno de desarrollo de software se ha optado por Microsoft Visual Studio 2010, ya que es la plataforma bajo la cual fue desarrollado Academics, y además la que se encuentra disponible en la PUCESA. Es un entorno de desarrollo sumamente completo, intuitivo y potente, también desarrollado por Microsoft, que permite crear diferentes tipos de aplicaciones orientadas a objetos, entre ellas aplicaciones web, y se integra perfectamente con bases de datos Microsoft SQL Server 2008 R2. Este entorno de desarrollo permite crear soluciones que a su vez pueden contener varios proyectos, cada uno de los cuales se centra en diferentes áreas del desarrollo de aplicaciones. Por ejemplo, un proyecto puede encargarse de la conexión a bases de datos, otro puede almacenar clases que se encargan de manipular los datos en las bases de datos, otro puede contener las páginas web o formularios visuales del proyecto y así sucesivamente.

En cuanto al lenguaje de programación, de entre los que soporta Microsoft Visual Studio 2010 se ha optado por C#, ya que es un lenguaje muy potente y versátil que ofrece todas las facilidades necesarias para trabajar con clases y objetos sin sobrecargar de código los archivos fuentes. Desde luego cabe también mencionar que Academics ha sido elaborado utilizando el lenguaje C#.

Cabe mencionar que en relación a las licencias de software, especialmente en lo referente a Microsoft SQL Server 2008 R2 y a Microsoft Visual Studio 2010 se ha trabajado bajo la modalidad de licencia institucional, la misma que faculta a la institución adquiriente de la licencia, en este caso a la PUCESA, a otorgar a sus estudiantes y/o personal copias legales del paquete en cuestión, siempre que el producto que se desarrolle utilizando dichas herramientas, en este caso la aplicación web objeto del presente proyecto, quede bajo propiedad de la institución que extiende la licencia.

Finalmente, en cuanto al hardware, todo el desarrollo del presente proyecto, tanto la parte práctica como la parte documental, se ha llevado a cabo utilizando un ordenador portátil marca Toshiba con procesador Intel Core I5, 4 GB de memoria RAM y 500 GB de disco duro, propiedad de la Ing. Gabriela Fabiola De la Torre Sánchez.

Capítulo 5

Resultados

5.1. Producto final del proyecto de titulación

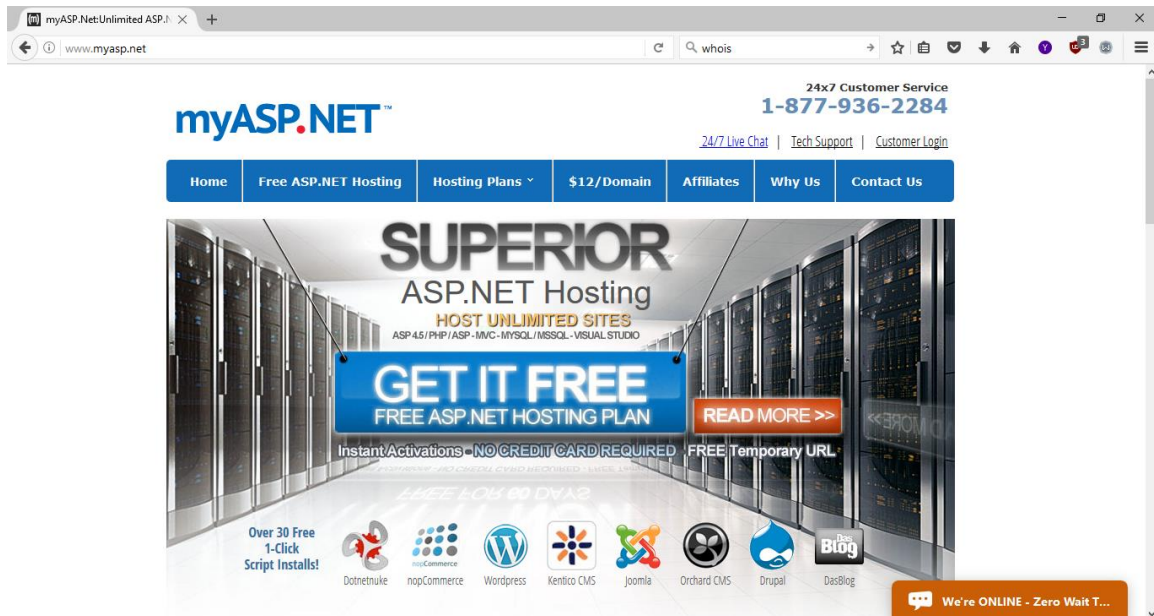
Como resultado del presente proyecto de investigación y desarrollo de grado se elabora una aplicación web que permite gestionar los procesos de graduación a los que se someten los maestrantes de la PUCESA. Como se ha mencionado, esta aplicación está diseñada para integrarse totalmente con la ya existente plataforma de software llamada “Academics”, puesto que utiliza gran parte de la información almacenada en sus bases de datos, y además se ha procurado guardar compatibilidad visual con dicha aplicación en la mayor medida posible. De igual forma, al utilizar las mismas herramientas y versiones de plataformas de desarrollo y de bases de datos que se utilizaron para la elaboración de Academics, se minimizan los problemas de compatibilidad con el mismo.

En vista de que el proceso de implementación de la aplicación desarrollada en los servidores de la PUCESA es un proceso complejo que demanda múltiples pasos tales como verificaciones de seguridad, autorizaciones por parte de autoridades, prueba internas del departamento de informática, etc., es un proceso que toma bastante tiempo, llegándose a estimar que de varios meses, por lo que se procede a publicar temporalmente la aplicación usando un servicio de hosting provisto por MyASP.NET (<http://www.myasp.net/>), el mismo que ofrece un plan gratuito para pruebas con limitada funcionalidad que sin embargo incluye soporte para las tecnologías que se requiere para la exitosa ejecución de la presente aplicación, es decir ASP.NET 4.0, base de datos Microsoft SQL Server 2012 (que es 100% compatible con las bases de datos de la presente aplicación creadas usando Microsoft SQL Server 2008 R2), así como el registro de un dominio de segundo nivel.

Por tanto, para fines de prueba, la aplicación con sus correspondientes bases de datos ha sido instalada y configurada en la siguiente dirección web:

<http://www.graduacionposgrados.com/>

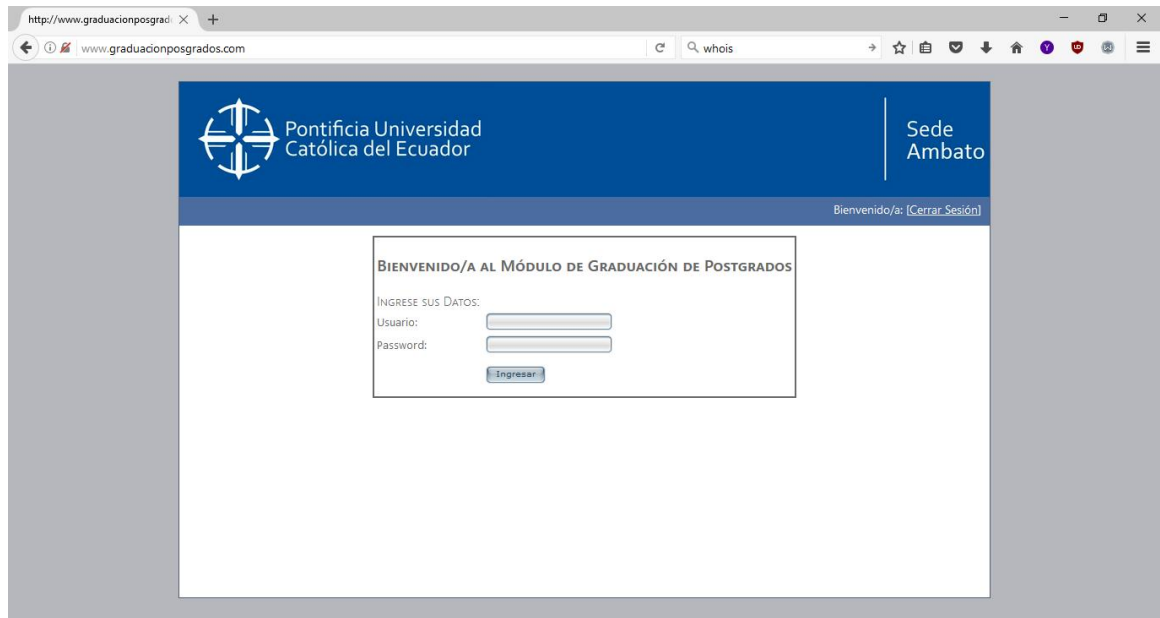
Figura 24: Página principal de MyASP.NET



Fuente: elaboración propia

La aplicación producto del presente proyecto de desarrollo consiste básicamente en 3 secciones, orientadas a los diferentes tipos de usuarios que pueden acceder a la misma. Mediante una página web de login, los usuarios pueden ingresar sus datos como son su usuario y contraseña, y el sistema automáticamente reconoce el tipo de usuario que ha ingresado, sea este maestrante, docente o dirección del DIP. Cabe mencionar que, al ser una aplicación que comparte los datos de la plataforma Academics y cuyo proceso de validación e ingreso de usuarios también está ligado a dicha plataforma, los datos de login son los mismos que cualquiera de estos tres tipos de usuarios puede utilizar para ingresar al Academics. Según el tipo de usuario reconocido, el sistema automáticamente lo redirigirá a su correspondiente página principal, donde podrá acceder a todas las opciones que tiene disponibles dentro de la aplicación mediante un menú ubicado en la parte izquierda de dicha página principal.

Figura 25: Página de login de la aplicación

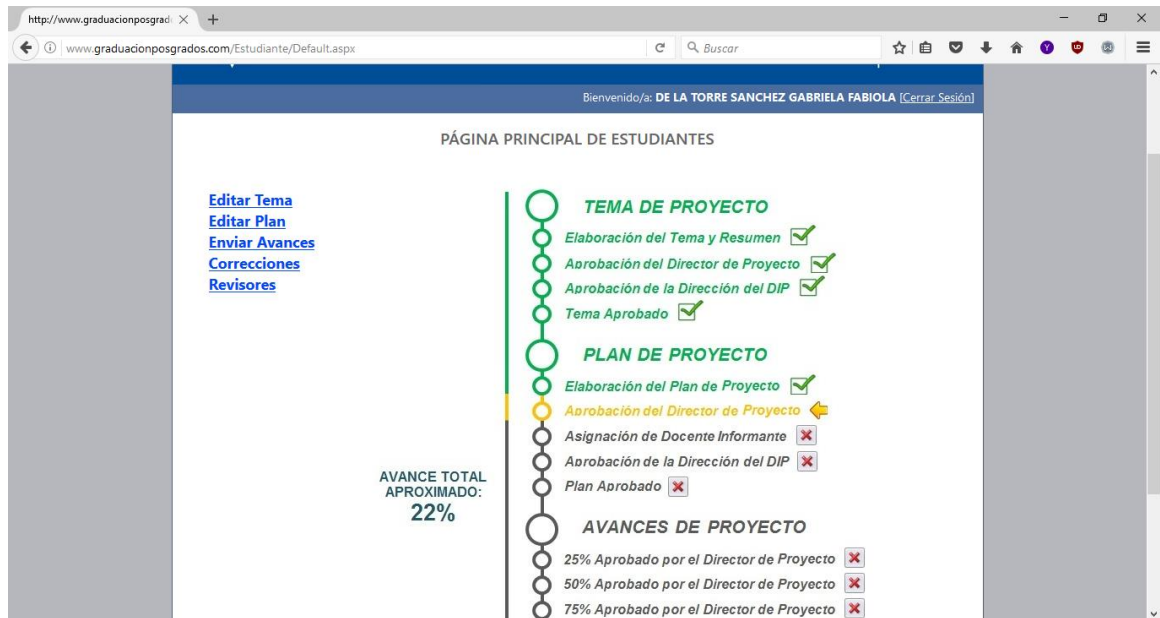


Fuente: Elaboración propia

En cuanto a los estudiantes o maestrantes que accedan al sistema, tendrán las opciones de enviar sus temas de proyectos de titulación, solicitando paralelamente la dirección del proyecto a un determinado docente, quien podrá aceptar o no la dirección del tema presentado, y en caso de aceptarlo podrá también revisar el tema y aprobarlo o sugerirle modificarlo al maestrante en cuestión. El maestrante podrá corregir su tema y volverlo a enviar a su posible director cuantas veces sea necesario. Una vez aprobado el tema, el mismo llegará al DIP donde su personal podrá aprobarlo definitivamente o sugerirle realizar más correcciones al maestrante en cuestión, pudiendo también aceptar definitivamente o rechazar el docente director propuesto, incluso si éste por su lado ya hubiese aceptado la dirección del tema. En este caso, el tema regresará al maestrante para que vuelva a corregirlo de ser el caso, y/o seleccione otro docente director.

La página principal de estudiantes muestra un diagrama que indica en qué parte del proceso de graduación se encuentra actualmente el maestrante, así como un porcentaje estimado del avance del proceso, tal como se muestra en la siguiente figura. El maestrante puede dar click en cualquier parte del gráfico, y será automáticamente redirigido a la página web correspondiente a su estado actual de avance dentro del proyecto.

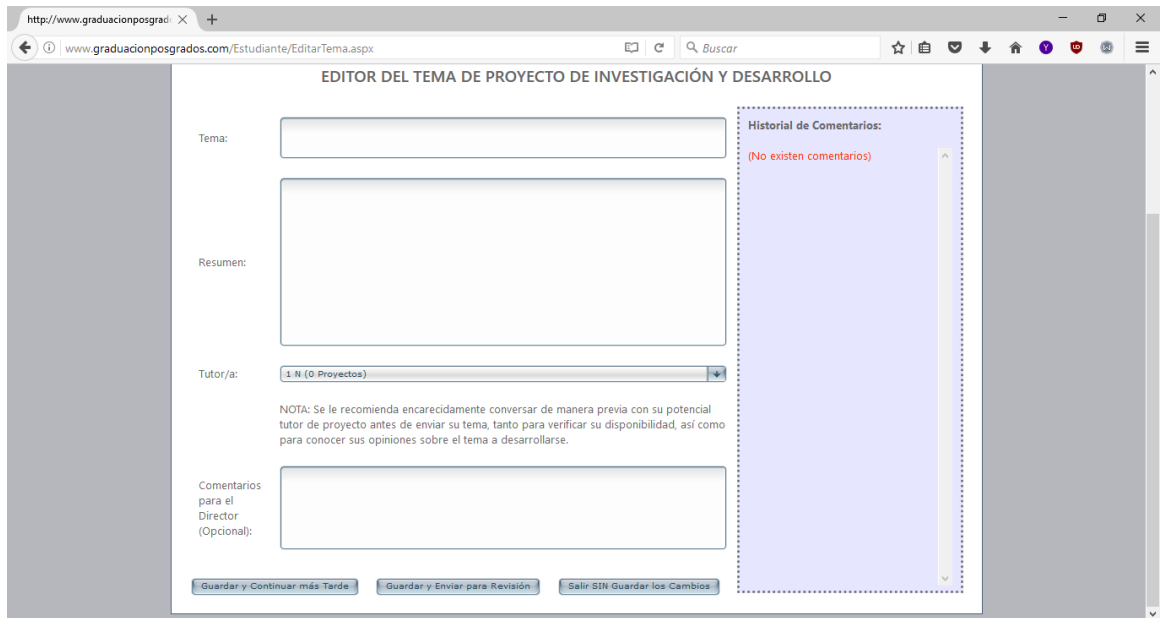
Figura 26: Página principal de estudiantes



Fuente: Elaboración propia

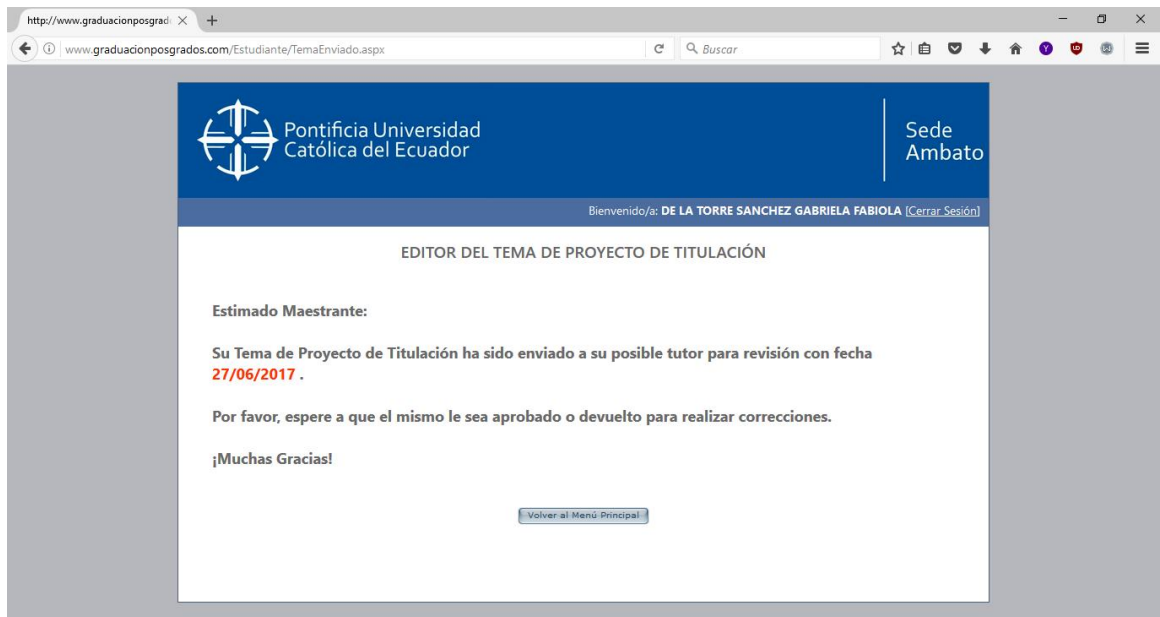
Para el proceso de envío del tema, se ha diseñado un editor donde el maestrante puede digitar su tema y su resumen de proyecto de titulación, y proponer su docente tutor seleccionándolo desde una lista. Esto evita que el mismo tenga que imprimir el documento y acudir a entregarlo personalmente en el DIP, ahorrándole tiempo y demás recursos.

Figura 27: Editor del tema de proyecto de titulación



Fuente: Elaboración propia

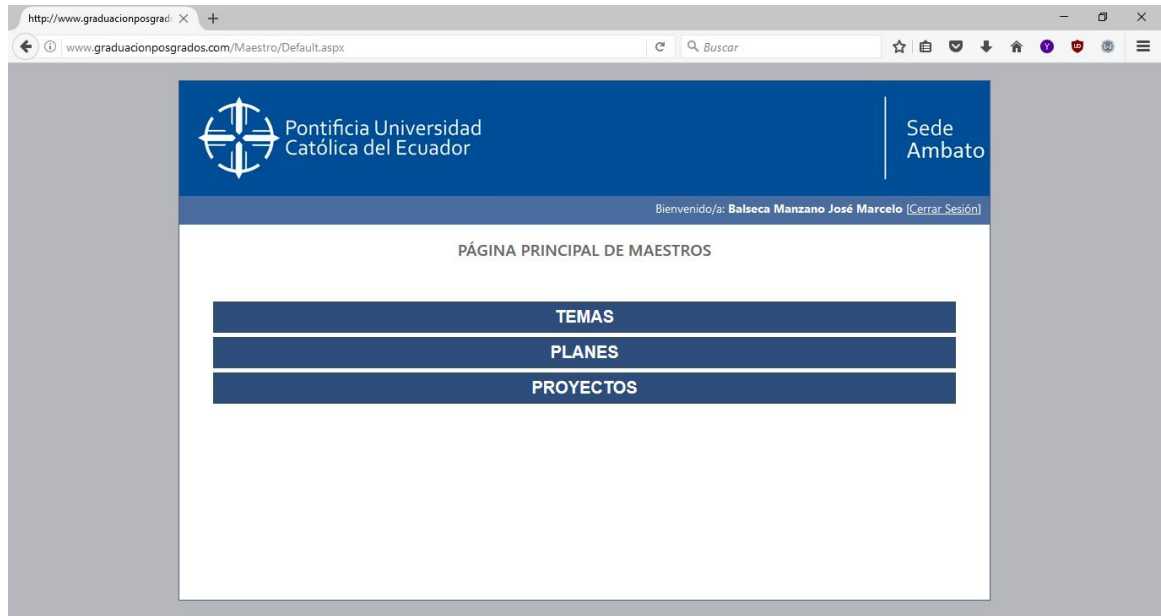
Figura 28: Confirmación del envío del tema de proyecto de titulación



Fuente: Elaboración propia

El docente propuesto como tutor del proyecto tiene ahora la potestad de aceptar o rechazar la dirección del tema, y en caso de aceptarlo puede aprobar el tema y resumen, o solicitar correcciones al maestrante.

Figura 29: Página principal de docentes



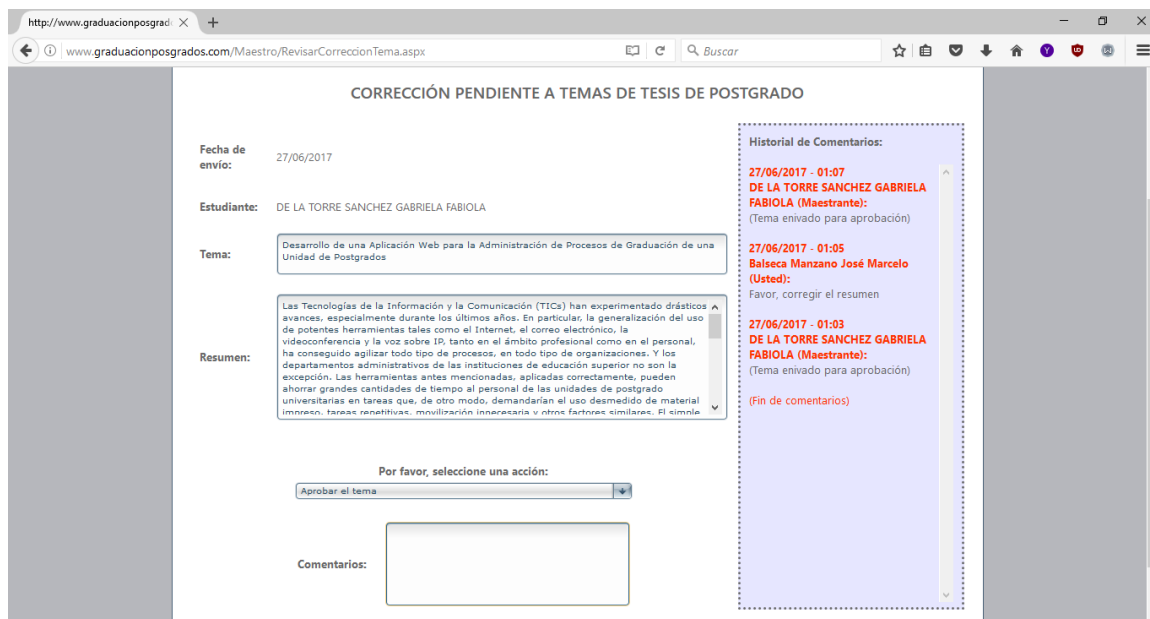
Fuente: Elaboración propia

Figura 30: Lista de solicitudes pendientes de dirección de proyectos de titulación



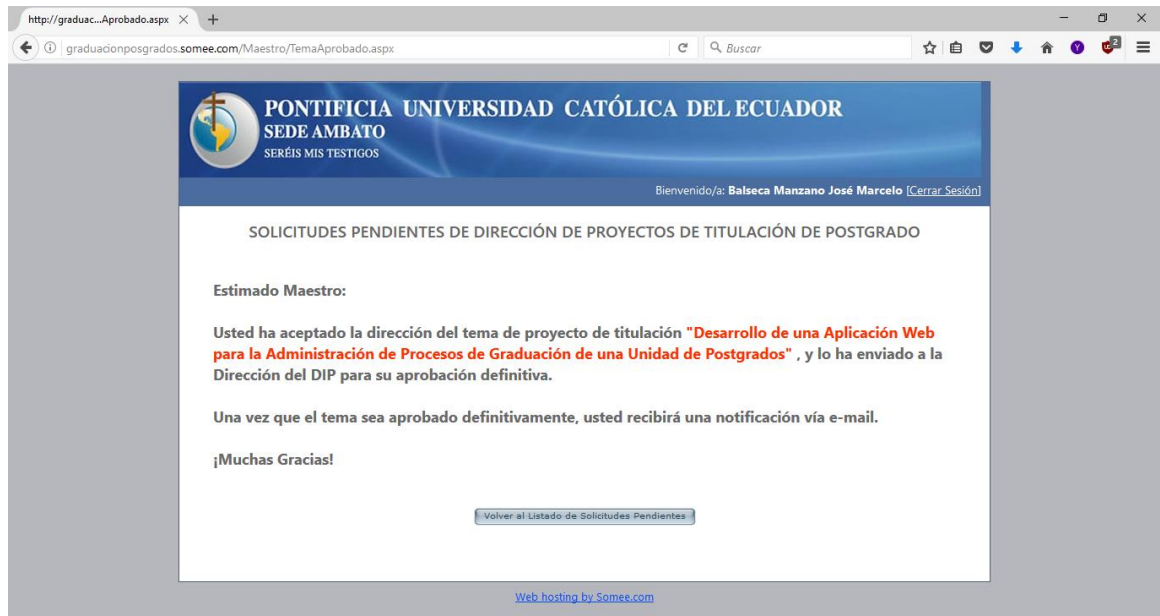
Fuente: Elaboración propia

Figura 31: Revisión de tema por parte del docente director



Fuente: Elaboración propia

Figura 32: Aprobación del tema por parte del docente director



Fuente: Elaboración propia

Una vez que un docente acepta la dirección de un proyecto de titulación y aprueba su tema, el maestrante puede acercarse al DIP portando los requisitos correspondientes para iniciar formalmente su proceso de graduación, cuyo primer paso es la solicitud de aprobación del tema.

Se debe mencionar que el módulo de la aplicación correspondiente al DIP funciona de manera totalmente independiente de los módulos de docentes y maestrantes, los mismos que únicamente sirven como herramientas de apoyo para sus respectivos usuarios, de manera especial para los maestrantes, a quienes se procura ahorrarles tiempo y movilización innecesarios e informarles del estado en el que se encuentra su proceso de graduación y, de ser el caso, los requisitos con los que deben acercarse al DIP para dar continuidad a dicho proceso. A pesar de esto, los mencionados módulos no están diseñados para ser utilizados obligatoriamente por sus respectivos usuarios, quienes puede omitir el uso de esta aplicación y enviar sus temas, planes y proyectos de titulación de la manera tradicional, sin que esto afecte en nada al funcionamiento del módulo del DIP.

Por lo expuesto, se procede a describir las diferentes secciones del módulo correspondiente al DIP más adelante, en un solo bloque.

En todas las secciones de la aplicación, se ha implementado un sistema de mensajes que pueden ser creados por cualquiera de los usuarios de la aplicación. Dichos mensajes son visibles para todos ellos y se despliegan en un panel lateral ubicado en la parte derecha de la sección principal de cada página, ordenados cronológicamente. Esto permite a cada tipo de usuario justificar el motivo por el que realiza una acción determinada, y al ser visible por todos los usuarios, estos pueden saber, por ejemplo, por qué razón se ha devuelto un tema y qué correcciones se debe aplicar, o por qué razón no se ha aceptado a un determinado docente como director de un proyecto de titulación.

Es muy importante también mencionar que, en el caso de los maestrantes, las diferentes secciones a las que tienen acceso a través de su menú principal, únicamente se van habilitando una vez que la sección o secciones anteriores correspondientes han sido completadas. Esto significa que, por ejemplo, un maestrante no podrá enviar su plan de titulación mientras su tema y director no han sido previamente designados por el personal administrativo del DIP.

Una vez que el tema de un maestrante y su respectivo director han sido aprobados por la dirección del DIP utilizando su módulo, el maestrante puede proceder con el envío de su plan del proyecto. Para el efecto, se ha creado igualmente un editor muy completo donde el maestrante puede ingresar su información correspondiente como lo son líneas de investigación, resumen, descripción del problema, respuestas a preguntas básicas, formulación de meta, hipótesis de trabajo o pregunta de estudio, variables, objetivo general y objetivos específicos, estado del arte, metodología general y métodos específicos, descripción de población y muestra, delimitación funcional, evaluación preliminar, referencias bibliográficas, rubros del presupuesto y actividades del cronograma de desarrollo del proyecto.

El mencionado editor despliega automáticamente toda la información que ya debería estar allí, como por ejemplo tema del proyecto, costo total (que se calcula de los rubros ingresados individualmente), duración del proyecto (por defecto 6 meses), lugar de ejecución (por defecto "PUCE Sede Ambato"), etc. Además, este editor está organizado en siete pestañas que agrupan la información del plan de modo tal que no se sobrecargue toda la información en un solo bloque, lo que podría resultar fatigoso y confuso para el maestrante.

Dentro del mencionado editor, todas las secciones del plan que conllevan el ingreso de varios ítems permiten el ingreso de dichos ítems uno a la vez, los mismos que se van listando y pueden ser eliminados si se han ingresado por equivocación o si es necesario corregirlos. En cuanto a las líneas

de investigación, que se pueden incluir varias en un plan de proyecto de titulación, las mismas se eligen de un listado donde aparecen las líneas que se encuentran actualmente vigentes, y se pueden igualmente eliminar en caso de error.

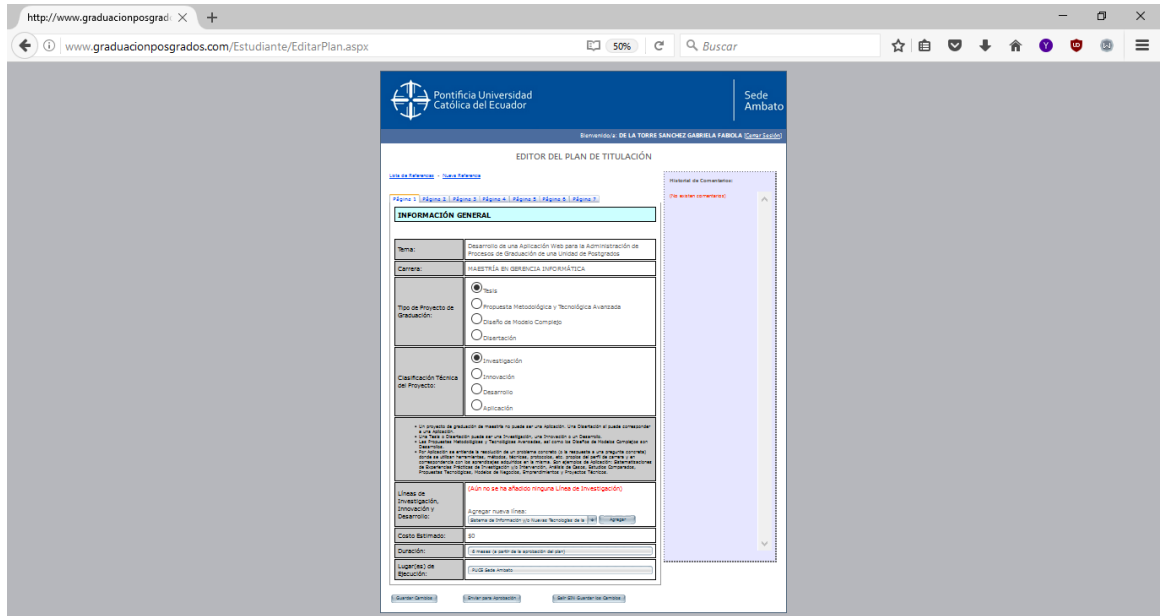
Un punto muy importante a tomar en cuenta dentro del editor en cuestión es el que se refiere a las referencias bibliográficas. El maestrante puede incluir todas las referencias bibliográficas que requiera su plan de proyecto de titulación, las cuales se van ordenando alfabéticamente de manera automática, según lo exigen las normas APA. La aplicación automáticamente se encarga de listar las referencias en cuestión respetando dichas normas APA, y realiza automáticamente los cambios requeridos en las citas a dichas referencias a lo largo de todo el plan, por lo que el maestrante no debe preocuparse de hacerlo manualmente, ahorrándole un considerable tiempo y evitándole errores en este sentido.

El ingreso de la información de referencias bibliográficas, rubros de presupuesto y actividades de cronograma se realizan a través de ventanas modales especialmente diseñadas para tales fines. El maestrante únicamente se encarga de ingresar la información correspondiente a cada una de ellas, y la aplicación automáticamente se encarga de ordenar la información, y desplegarla dentro del formato correspondiente y según los estándares establecidos por la PUCESA para la presentación de planes de proyectos de titulación.

En vista de que la elaboración de un plan puede tomar un tiempo considerable, se le da al maestrante la opción de guardar la información que va añadiendo al mismo, dentro de todos los puntos mencionados en párrafos anteriores, se trate de elementos listados o no. Esto permite que el maestrante guarde su trabajo y pueda posteriormente volver a ingresar al editor y continuar desde el punto donde lo dejó y aplicar modificaciones a cualquier parte del mismo.

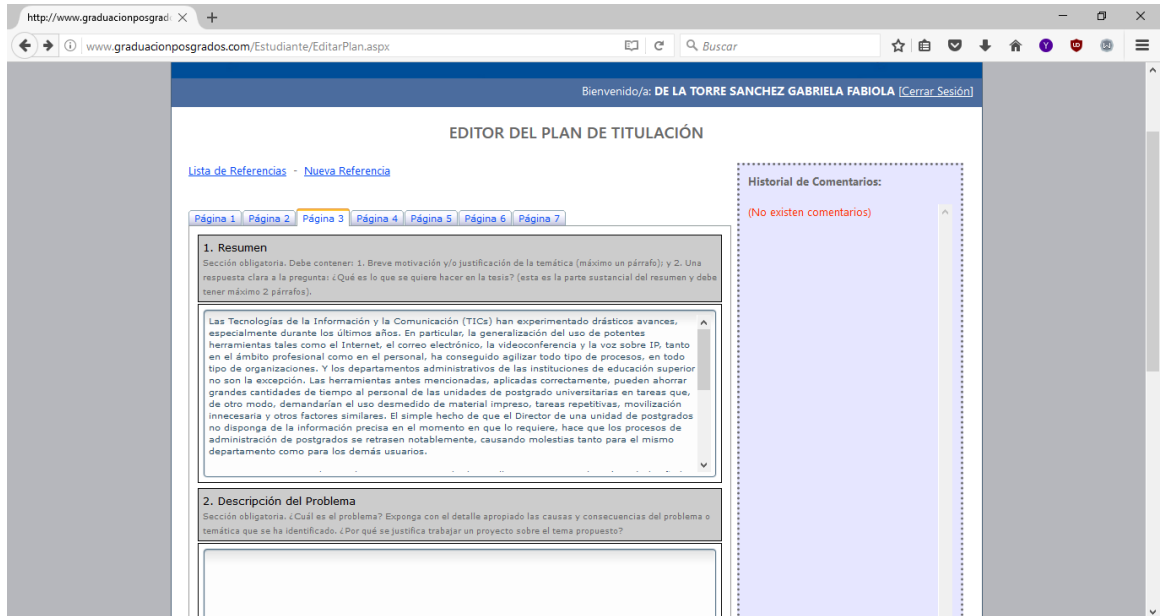
De igual modo, una vez que el maestrante ha ingresado toda la información correspondiente a su plan de proyecto de titulación y está listo para enviarlo, puede ingresar un mensaje o comentario correspondiente para su director o para el personal del DIP, quienes desde sus respectivas interfaces podrán también incluir dichos comentarios, los mismos que igualmente serán desplegados en orden cronológico dentro del panel especial destinado para esta función que se mencionó anteriormente.

Figura 33: Editor del plan de proyecto de titulación – página 1



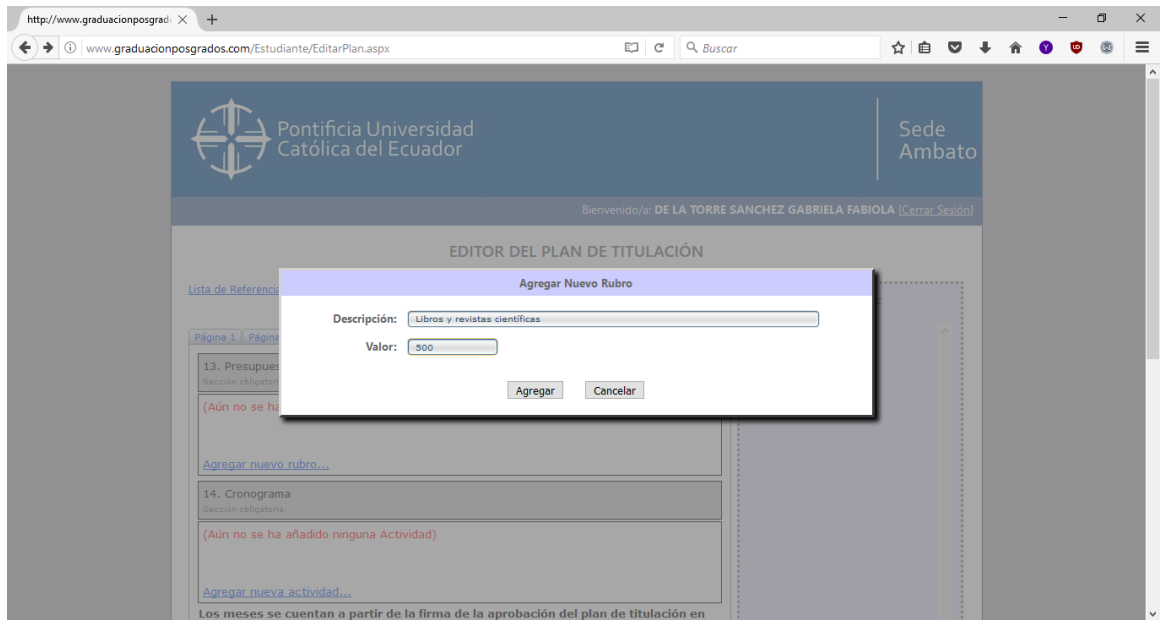
Fuente: Elaboración propia

Figura 34: Editor del plan de proyecto de titulación – página 3



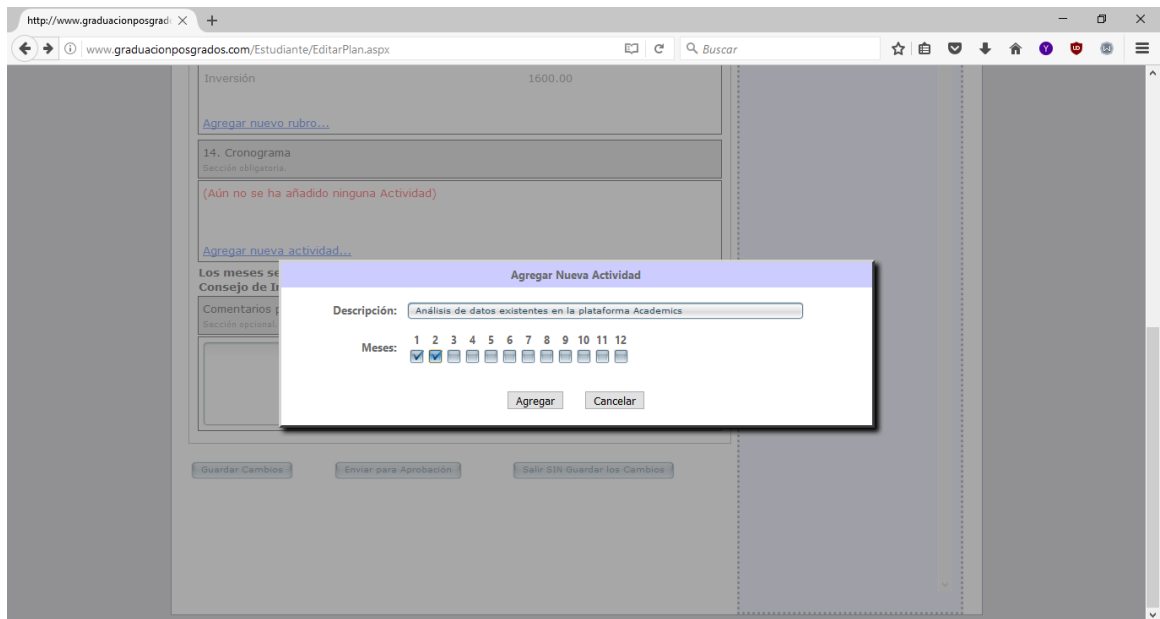
Fuente: Elaboración propia

Figura 35: Adición de elementos del presupuesto del proyecto



Fuente: Elaboración propia

Figura 36: Adición de elementos del cronograma



Fuente: Elaboración propia

Figura 37: Editor del plan de proyecto de titulación – página 7

13. Presupuesto
Sección obligatoria. El presupuesto debe ser resumido e incluir únicamente los ítems relevantes al proyecto.

| Rubro | Valor | |
|-------------------------------|---------|---|
| Libros y revistas científicas | 500.00 | X |
| Licencias de Software | 700.00 | X |
| Papelaría | 100.00 | X |
| Transporte | 100.00 | X |
| Suministros varios | 200.00 | X |
| Inversión | 1600.00 | |

[Agregar nuevo rubro...](#)

14. Cronograma
Sección obligatoria.

| ACTIVIDAD | MESES | | | | | | | | | | | |
|---|-------|---|---|---|---|---|---|---|---|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Análisis de requerimientos | X | | | | | | | | | | | X |
| Análisis de la información existente en la plataforma Academics | X | | | | | | | | | | | X |
| Diseño y desarrollo de la aplicación | X | X | | | | | | | | | | X |
| Implementación y pruebas | | X | | | | | | | | | | X |
| Detección y corrección de errores | | X | X | | | | | | | | | X |
| Implementación de la aplicación | | | | X | | | | | | | | X |
| Elaboración del IFP | X | X | X | X | | | | | | | | X |

[Agregar nueva actividad...](#)

(No existen comentarios)

Fuente: Elaboración propia

Figura 38: Adición de referencias bibliográficas

11. Evaluación Preliminar
Sección obligatoria en caso de Desarrollo, Innovación o Aplicación. Indicar la(s) fuente(s) de los datos que se procesarán.

12. Referencias
Indicar al menos una referencia normativa en el ítem.

Referencias académicas (Aún no se ha añadido ninguna Referencia)

[Agregar nueva referencia...](#)

Agregar Nueva Referencia

Autor(es):

Título:

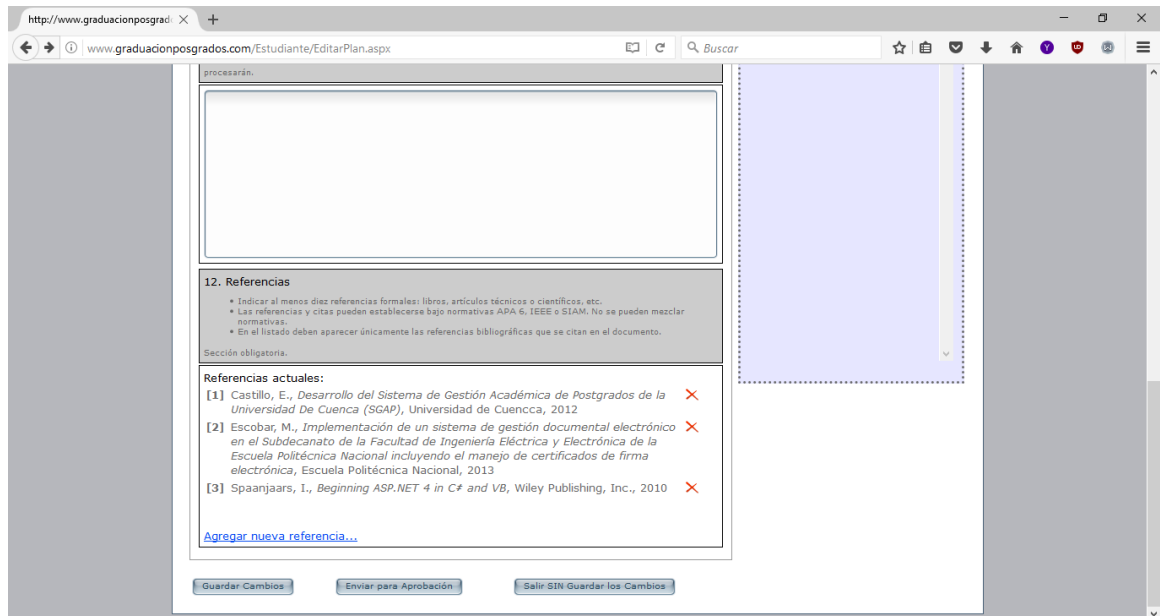
Editorial/Fuente:

Año:

URL:

Fuente: Elaboración propia

Figura 39: Sección de referencia bibliográficas del editor del plan



Fuente: Elaboración propia

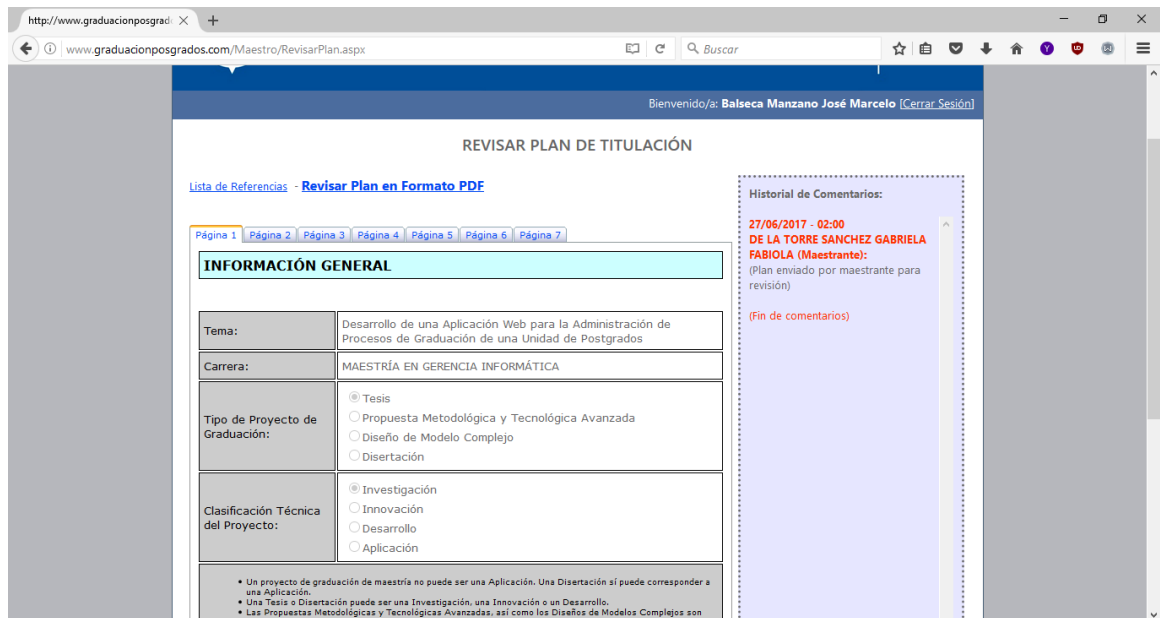
Una vez que el maestrante concluye su trabajo y envía su plan de proyecto de titulación, la aplicación automáticamente genera el plan el formato PDF, siguiendo el formato establecido por la PUCESA. El plan entonces se dirige en primer lugar hacia su director de proyecto, quien puede revisarlo tanto utilizando el mencionado editor del plan, aunque no puede aplicar correcciones a nombre de su maestrante, o puede descargar el archivo PDF generado para imprimirlo o revisarlo en pantalla. Una vez revisado, el director de proyecto podrá sugerir correcciones o aprobar el plan. En el primer caso, al maestrante se le habilitará nuevamente el acceso al editor para que pueda aplicar las correcciones del caso y volver a enviar el plan; en el segundo caso, el maestrante será notificado de que puede acercarse al DIP portando los requisitos del caso para el siguiente paso en su proceso de graduación, que en este caso sería la solicitud de designación de docente informante.

Figura 40: Plan de proyecto de titulación en formato PDF



Fuente: Elaboración propia

Figura 41: Revisión del plan parte del director del proyecto

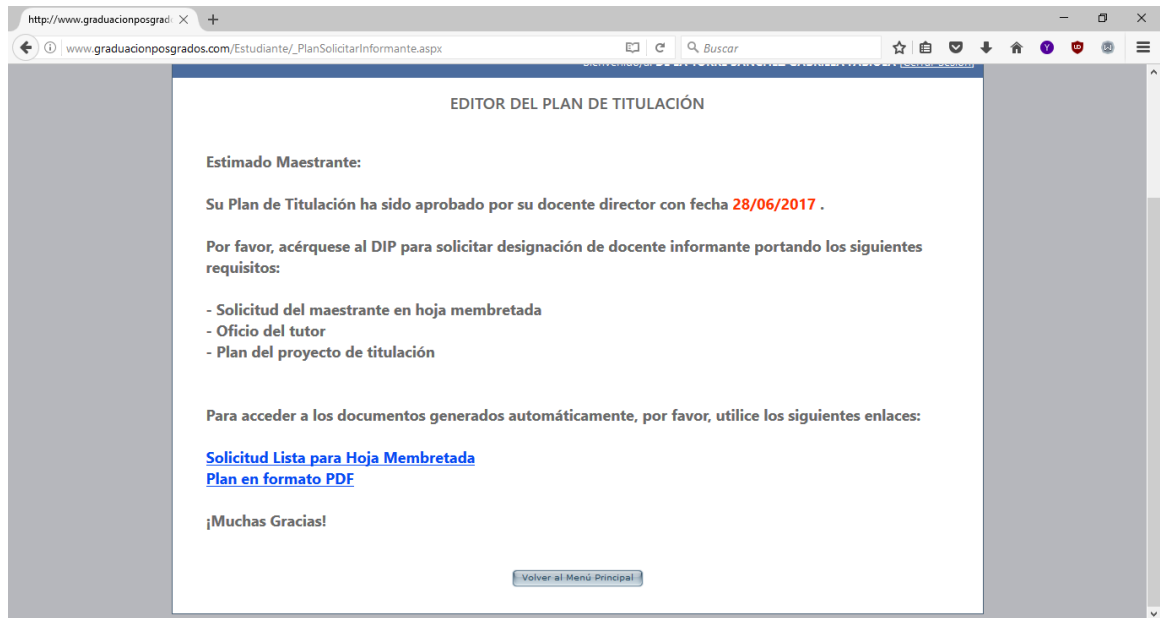


Fuente: Elaboración propia

Es muy importante mencionar que, una vez que un maestrante envía su tema o plan de proyecto de titulación, no tiene acceso a sus respectivos editores hasta que uno u otro le hayan sido devueltos para aplicar correcciones, por parte de cualquiera de los docentes relacionados con su proceso de graduación. Es decir que, una vez que el maestrante envía su tema o plan, tiene que esperar a que le sean devueltos para poder corregirlos, ya que de lo contrario no tiene acceso a los mismos. De igual forma, en el caso de que cualquiera de los dos requerimientos le hayan sido aprobado, el maestrante ya no podrá realizar cambios a los mismos.

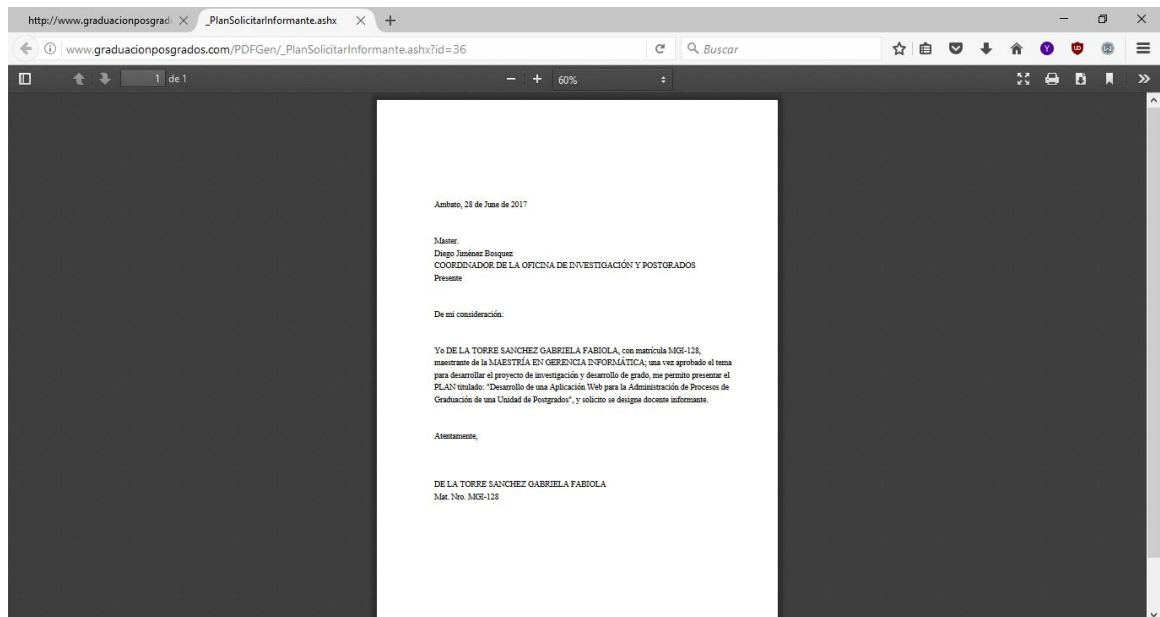
Toda vez que un plan de proyecto de titulación ha sido aprobado por el respectivo director de proyecto, el maestrante debe continuar con el proceso acercándose personalmente al DIP. Una vez que ingrese su solicitud de designación de docente informante y el mismo haya sido designado por parte del DIP, el mencionado docente deberá proceder a revisar el plan y sugerir las correcciones del caso para lo cual podrá también utilizar la presente aplicación como herramienta de apoyo. Finalmente, una vez que el plan se encuentre a satisfacción del docente informante, el maestrante podrá continuar una vez más con el proceso de manera presencial en el DIP, portando los requisitos necesarios para el siguiente paso que sería la solicitud de aprobación del plan. Como se mencionó, el módulo de maestrantes se encarga también de detectar cuál es el siguiente paso que debe ejecutar el maestrante dentro de su proceso de graduación y mostrarle los requisitos necesarios para el mismo, lo que le ahorraría al maestrante la necesidad de acercarse al DIP únicamente para averiguar los requisitos. Además de eso, la aplicación se encarga también de generar automáticamente en formato PDF los documentos que sea posible como apoyo al maestrante dentro de cada paso de su proceso de graduación, en particular las solicitudes que debe presentar en el DIP de acuerdo a los formatos actuales que se maneja en dicho departamento, listos para imprimirse en hojas membretadas de la PUCESA.

Figura 42: Página informativa de requisitos para el maestrante



Fuente: Elaboración propia

Figura 43: Solicitud generada automáticamente para hoja membretada



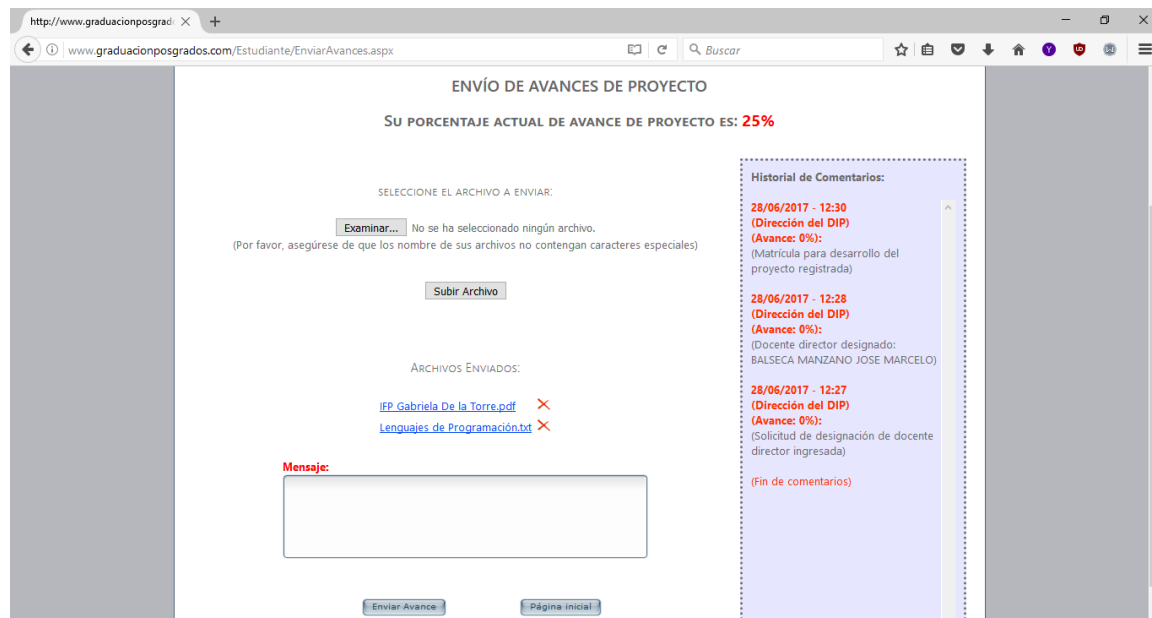
Fuente: Elaboración propia

Una vez que el plan es definitivamente aprobado por la dirección del DIP, el siguiente paso en el proceso es la solicitud de docente director que el maestrante debe presentar en el DIP de manera presencial, y para cuyo registro el DIP utilizará su módulo de la aplicación. Posteriormente a que el docente director, que generalmente es el mismo docente que venía siendo el tutor durante las fases de tema y plan del proceso de graduación pero si es del caso el DIP lo puede cambiar, haya sido designado, el maestrante deberá acudir una vez más al DIP, esta vez para registrar su proceso de matriculación que le habilitará para iniciar el desarrollo del proyecto, para lo cual debe también llenar las respectivas hojas de crédito.

Cumplidos estos pasos, se habilita finalmente al maestrante a que envíe avances del proyecto de titulación, que normalmente deberían contener los borradores del Informe Final del Proyecto (IFP), así como cualquier archivo adicional de soporte, como archivos con datos adjuntos, códigos fuente, ejecutables, etc., dependiendo del proyecto en desarrollo por el maestrante.

Básicamente, la aplicación ha sido diseñada para soportar el envío de avances en cuatro niveles: hasta el 25%, hasta el 50%, hasta el 75% y hasta el 100%. El sistema arranca automáticamente en el nivel de avances de 25%, permitiendo al maestrante enviar los archivos que considere necesarios, los mismos que llegarán a su director quien podrá descargarlos para revisión. El director tendrá entonces la potestad de aprobar el avance o solicitar al maestrante aplicar correcciones o enviar archivos adicionales. En ese caso, el maestrante tendrá nuevamente acceso a la interfaz de envío de archivos para poder reemplazar los archivos enviados y/o enviar nuevos archivos. Dentro del contexto de la presente aplicación, se ha denominado a cada uno de estos envíos una "iteración". Una vez que el maestrante haya enviado sus nuevos archivos, su director tendrá nuevamente acceso a los archivos enviados y podrá descargarlos para su revisión. En este punto, es muy importante tener en cuenta que la aplicación desarrollada guarda todas las versiones de los archivos enviados por un maestrante como avances de su proyecto. Por tanto, en la interfaz del docente director del proyecto, éste tendrá acceso a todas las iteraciones de envíos de archivos de sus maestrantes por cada nivel de avances, lo que lo habilitará a descargar archivos similares de diferentes iteraciones para verificar si el maestrante ha realizado correctamente las correcciones sugeridas, o ha resuelto convenientemente los problemas planteados.

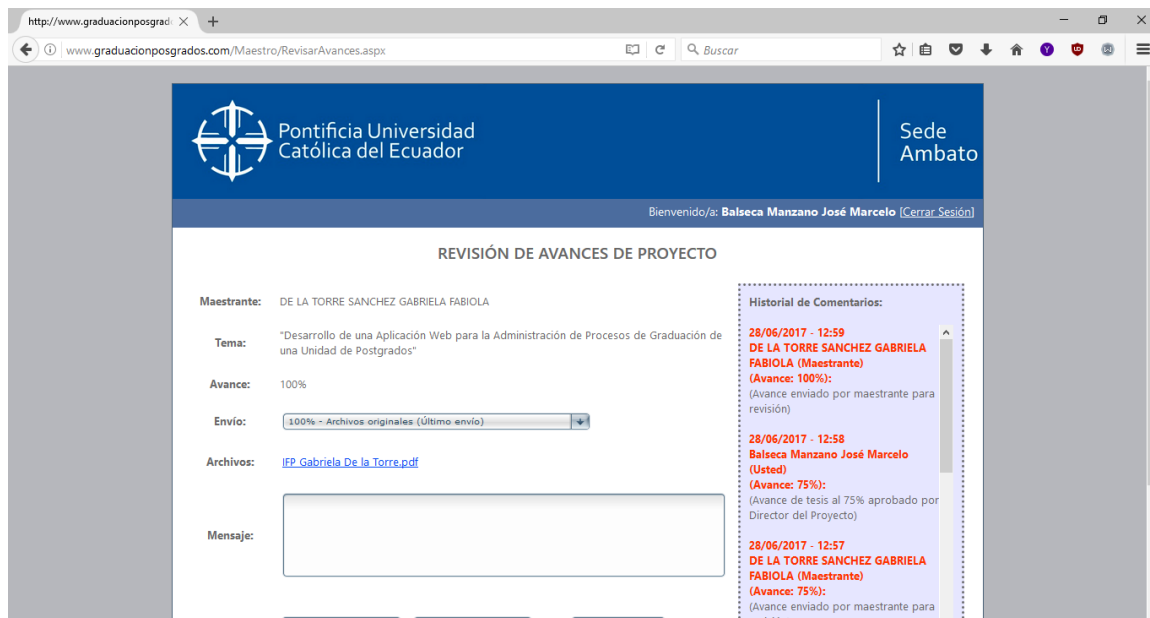
Figura 44: Envío de avances del IFP (25%)



Fuente: Elaboración propia

Una vez que los avances enviados se encuentren a satisfacción del director, éste podrá aprobarlas habilitando al maestrante a continuar enviando avances para el siguiente nivel, es decir, si estaba en un 50% y el docente director aprueba los avances, el maestrante automáticamente quedará habilitado para el envío de archivos del 75%. Cuando ya se hubiese alcanzado el 100% de avances, lo cual significa que el maestrante ya ha enviado el IFP completo además de todos los archivos de soporte adicionales requeridos, y el director decide aprobar este nivel, significa que éste ya habrá aprobado el trabajo en su totalidad, en cuyo caso el maestrante deberá nuevamente acercarse al DIP portando múltiples requisitos, entre ellos dos anillados elaborados a partir de este IFP aprobado, para proceder con la solicitud de designación de docentes revisores.

Figura 45: Revisión de avances del IFP por parte del director del proyecto



Fuente: Elaboración propia

En este punto, los docentes designados como revisores para el proyecto en cuestión tienen la potestad de elegir si continúan utilizando su módulo correspondiente dentro de la presente aplicación como herramienta de apoyo para la revisión y solicitud de correcciones al maestrante. De hacerlo así, el maestrante podrá también enviar sus correcciones de manera similar a como envió sus avances, y éstas deberán también ser revisadas y aprobadas por su docente director.

Sea que la utilicen o no, esta es la última página a la que pueden acceder los docentes dentro de su correspondiente módulo dentro de la aplicación, siempre que hayan sido designados docentes revisores para un determinado proyecto de investigación y desarrollo de grado. Una vez que el proyecto designado se encuentre a satisfacción de ambos docentes revisores, los mismos deberán enviar su informe favorable al DIP para su respectivo registro. El maestrante deberá entonces continuar su proceso de graduación con la solicitud de designación de tribunal calificador y posteriormente con el registro de la revisión de sus requisitos en biblioteca para la elaboración del empastado. Finalmente, el maestrante podrá solicitar fecha y hora de defensa, que es el último paso que debe realizar en el DIP previa la conclusión de su proceso de graduación.

Todos los pasos mencionados en el párrafo anterior con respecto al maestrante, deberá realizarlos de manera presencial en el DIP portando los respectivos requisitos, sin embargo, el módulo de maestrantes de la presente aplicación continúa siendo de gran ayuda para el maestrante ya que le permite conocer cuando hay avances dentro de su proceso de graduación (por ejemplo cuando se designan los docentes revisores), así como también le permiten saber cuál es el siguiente paso dentro de dicho proceso, si debe acercarse al DIP a gestionarlo y cuáles son los requisitos que debe llevar. El módulo para la dirección del DIP de la aplicación, por otro lado, permite que la secretaria de dicho departamento continúe registrando todos los pasos mencionados, junto con la verificación de cumplimiento de requisitos en cada uno de ellos,

Finalmente, toda esta información registrada por la secretaría o dirección del DIP permitirá generar un gráfico estadístico de suma utilidad para el departamento, que permite conocer de manera instantánea el número de casos que se encuentran en cada fase detallada del proceso de graduación de posgrados, tanto a nivel general como por cada Maestría. Este gráfico constituye una verdadera herramienta de apoyo a la toma de decisiones para la dirección del DIP, ya que al momento, si se requiere saber cuántos casos se encuentran en un determinado estado dentro del proceso de graduación se realiza un conteo manual, lo cual es un proceso muy demorado y susceptible de errores.

Como se menciona en un párrafo anterior, el módulo para la secretaría/dirección del DIP funciona independientemente de los otros dos módulos (de maestrantes y de docentes), ya que el uso de los mismos es opcional. En el módulo del DIP la secretaria del departamento debe registrar la información correspondiente para cada paso del proceso de graduación de cualquier maestrante que así lo requiriere, poniendo especial atención al cumplimiento de los requisitos que cada paso exige. Por otro lado, este módulo permite generar automáticamente correos electrónicos informativos a las partes implicadas (maestrantes y docentes) cuando así se lo requiriere, pudiendo la secretaria del DIP habilitar o deshabilitar el envío de dichos correos en cada caso. Por último, hay que mencionar que todos los pasos existentes dentro de este módulo deben ser ejecutados secuencialmente y en orden para cada maestrante, según el detalle que se observa en la siguiente tabla:

Tabla 17: Secuencia de pasos dentro del proceso de graduación de un maestrante

| ORDEN | PASO |
|--------------|--|
| 1 | Ingreso de solicitud de aprobación de tema |
| 2 | Registro de aprobación de tema |
| 3 | Ingreso de solicitud de designación de docente informante |
| 4 | Registro de designación de docente informante |
| 5 | Ingreso de solicitud de aprobación del plan |
| 6 | Registro de aprobación del plan |
| 7 | Ingreso de solicitud de designación de docente director |
| 8 | Registro de designación de docente director |
| 9 | Registro de matrícula para desarrollo del proyecto |
| 10 | Ingreso de solicitud de designación de docentes revisores |
| 11 | Registro de designación de docentes revisores |
| 12 | Registro de aprobación de docentes revisores |
| 13 | Ingreso de solicitud de designación de tribunal calificador |
| 14 | Registro de designación de tribunal calificador |
| 15 | Registro de revisión en biblioteca para empastado |
| 16 | Ingreso de solicitud de designación de fecha y hora de defensa |
| 17 | Registro de designación de fecha y hora de defensa |
| 18 | Registro de graduación del maestrante |
| 19 | Registro de defensa fallida |
| 20 | Ingreso de solicitud de nueva fecha y hora de defensa |
| 21 | Baja de tema de proyecto de investigación y desarrollo |

Fuente: elaboración propia

A continuación se muestran las diferentes páginas que forman parte del módulo para la Dirección del DIP de la presente aplicación. Los mensajes que se pueden ver en el panel titulado como “Historial

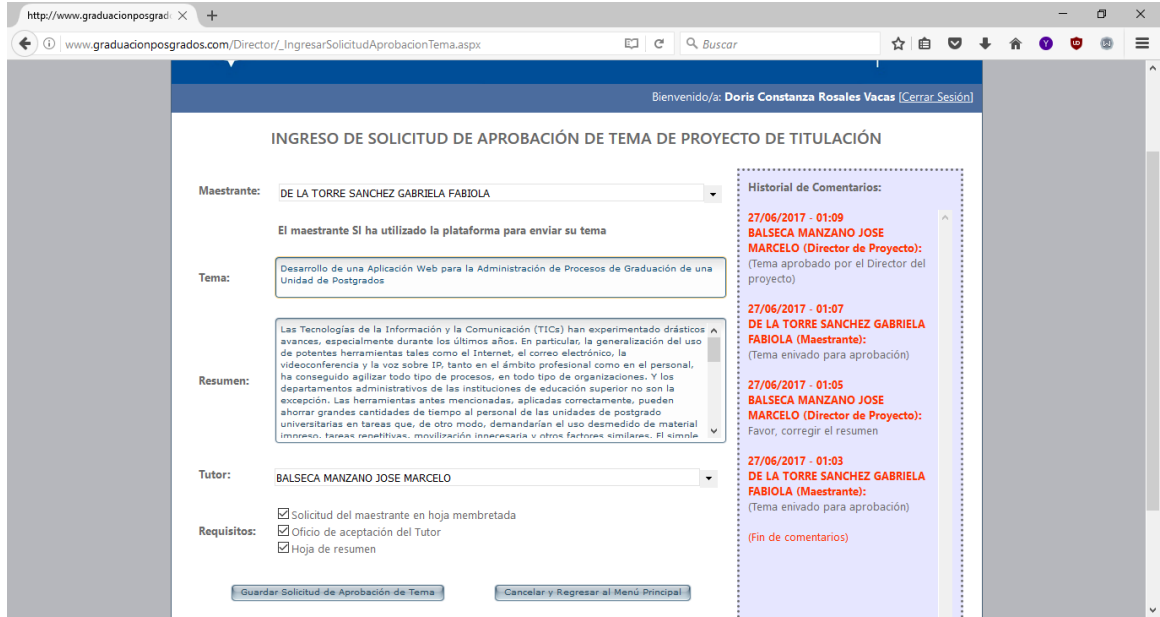
de Comentarios” corresponden a un maestrante y sus docentes designados que han utilizado sus respectivos módulos de la aplicación.

Figura 46: Página principal del módulo de la dirección del DIP



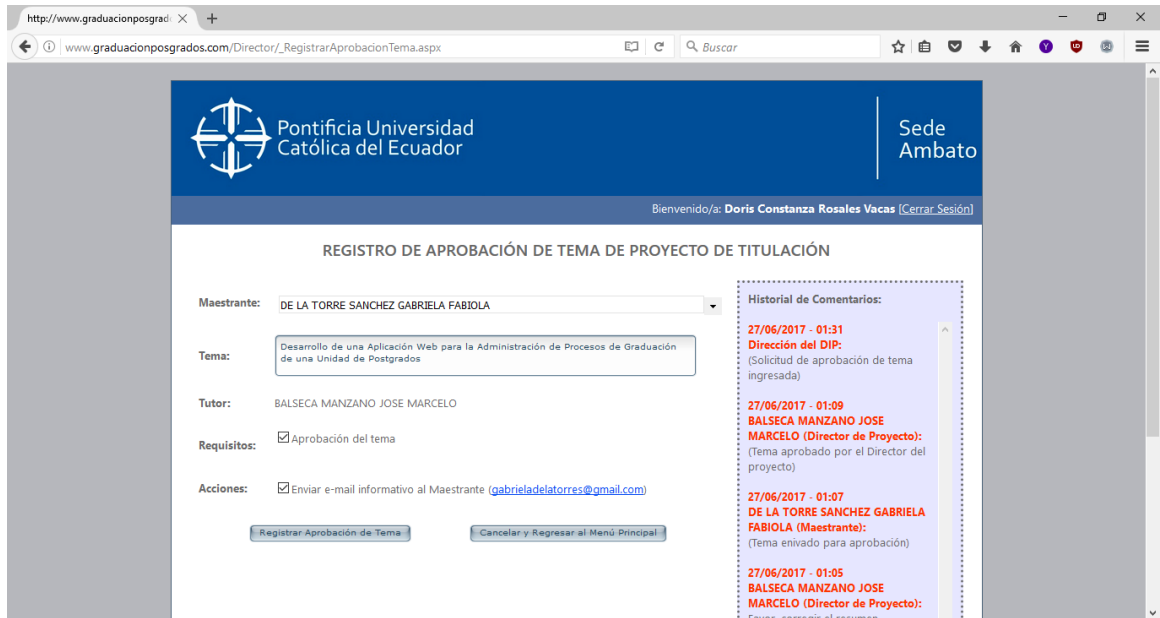
Fuente: Elaboración propia

Figura 47: Ingreso de solicitud de aprobación de un tema



Fuente: Elaboración propia

Figura 48: Registro de aprobación de un tema



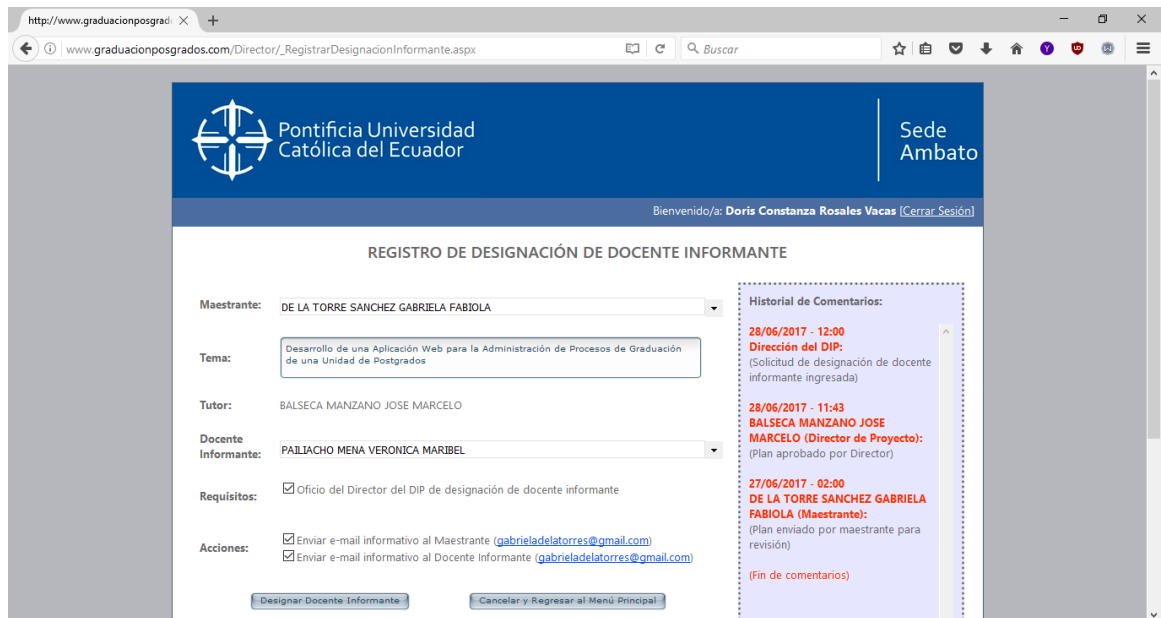
Fuente: Elaboración propia

Figura 49: Ingreso de solicitud de designación de docente informante



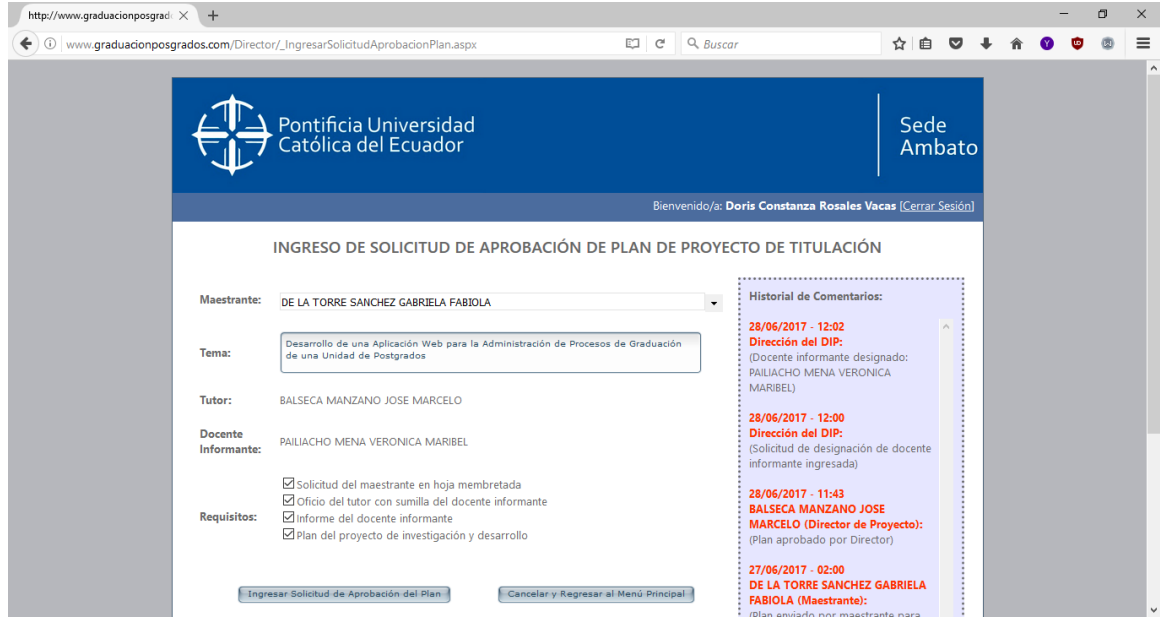
Fuente: Elaboración propia

Figura 50: Registro de designación de docente informante



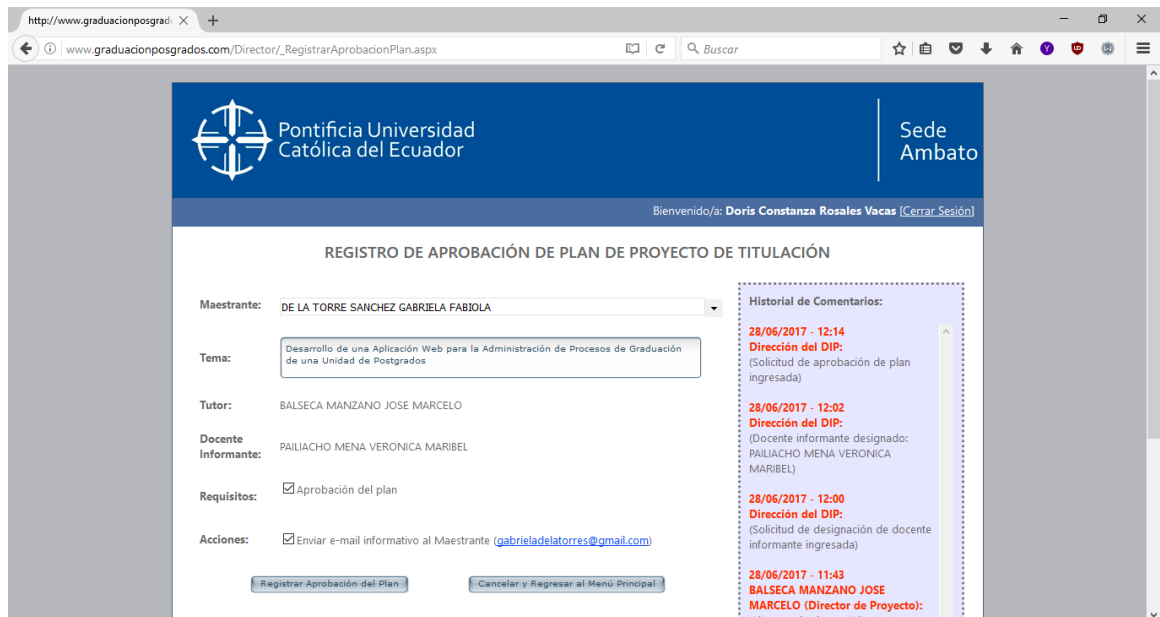
Fuente: Elaboración propia

Figura 51: Ingreso de solicitud de aprobación de plan



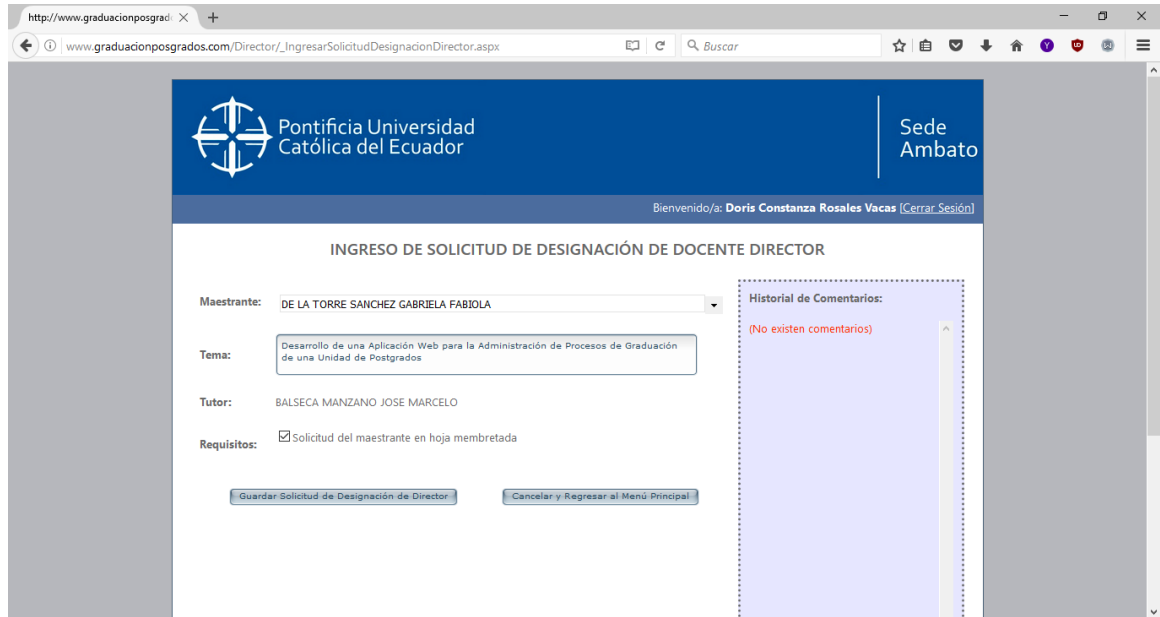
Fuente: Elaboración propia

Figura 52: Registro de aprobación del plan



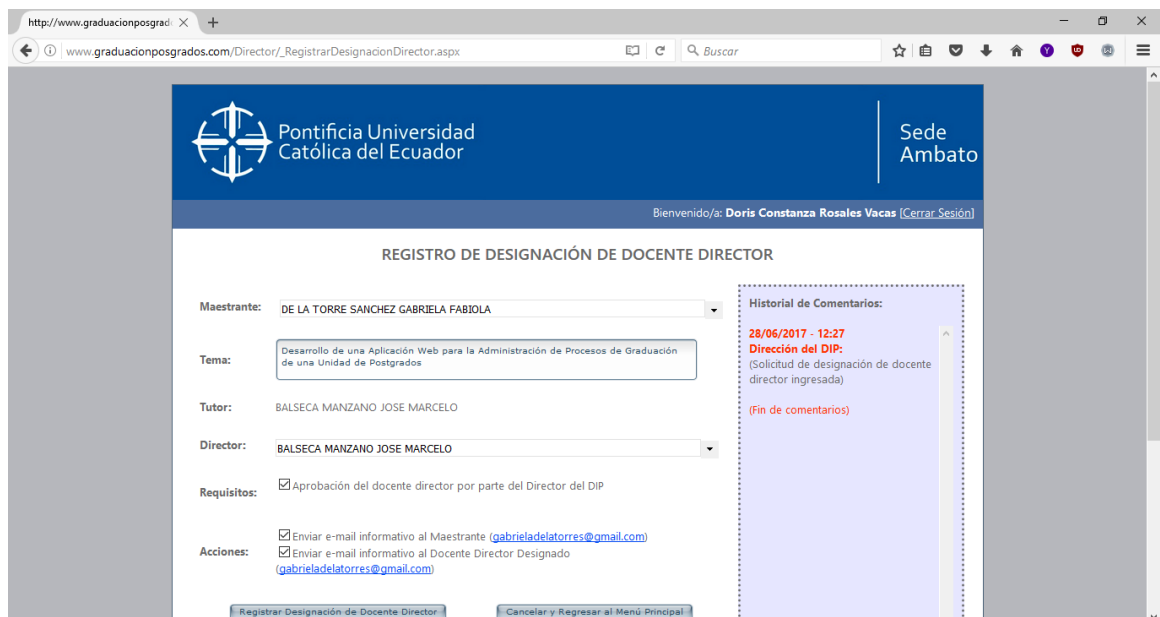
Fuente: Elaboración propia

Figura 53: Ingreso de solicitud de designación de docente director



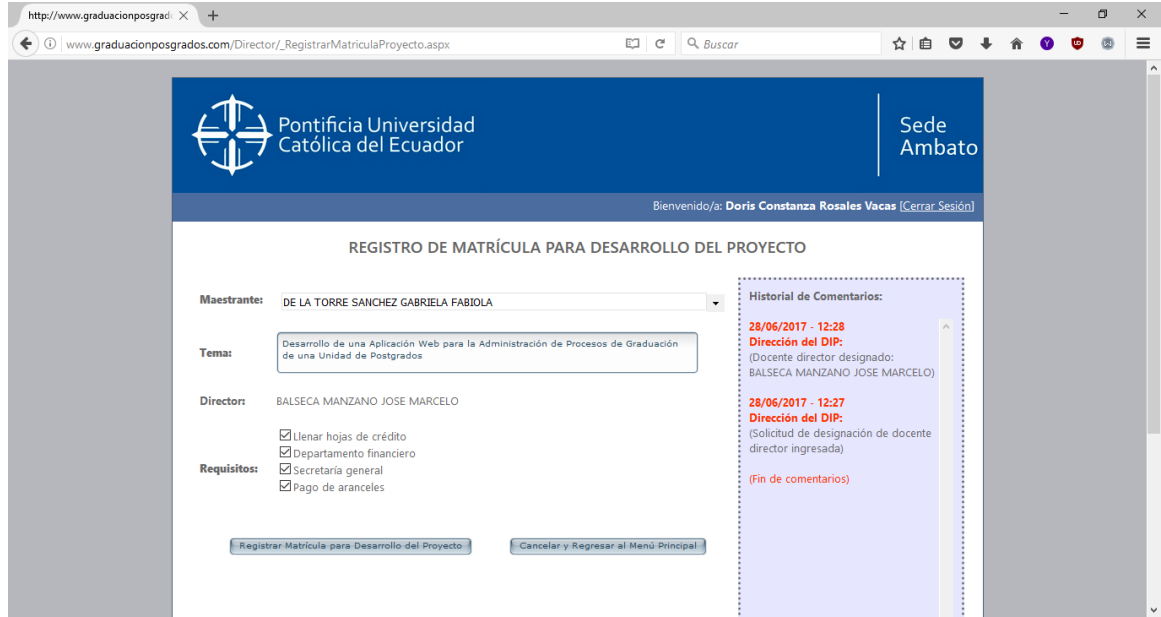
Fuente: Elaboración propia

Figura 54: Registro de designación de docente director



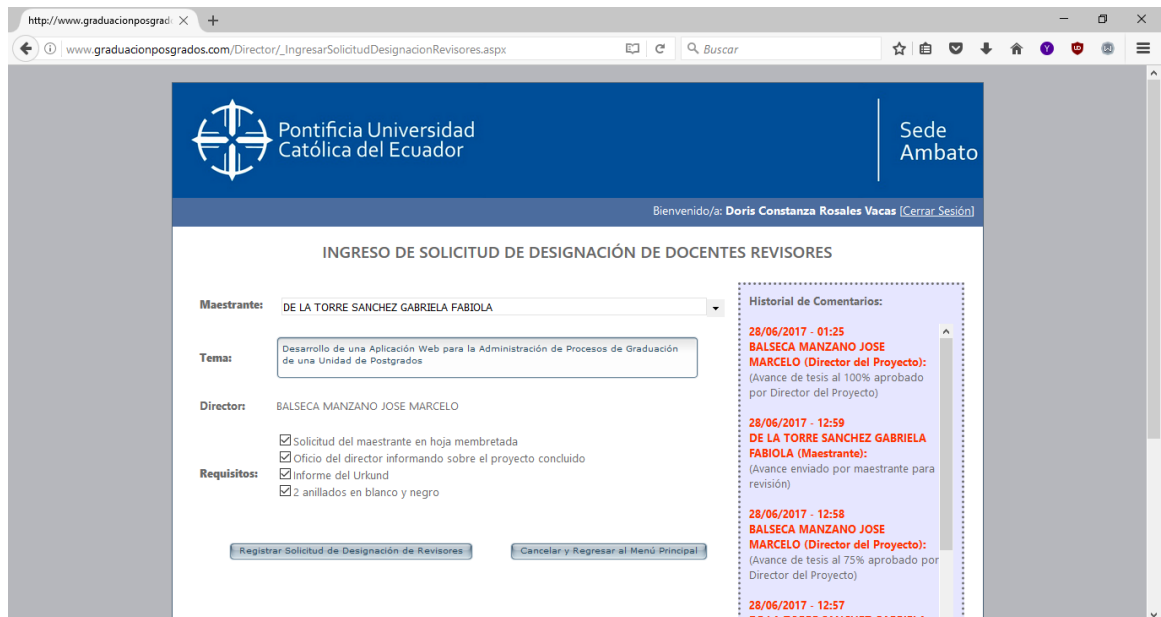
Fuente: Elaboración propia

Figura 55: Registro de matrícula para iniciar el desarrollo del proyecto



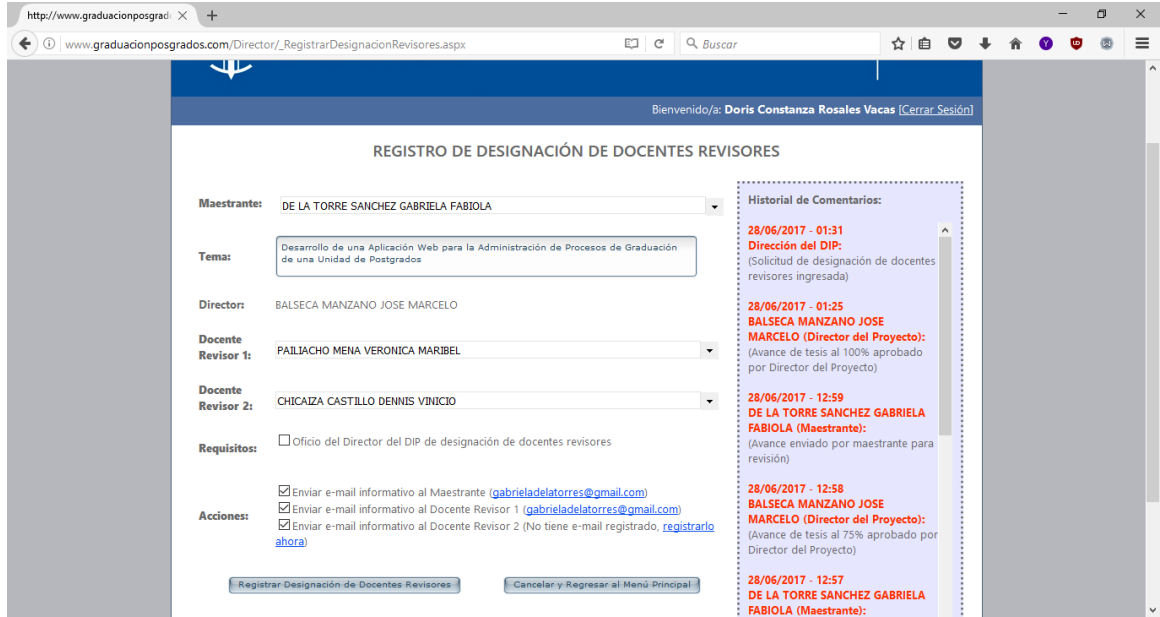
Fuente: Elaboración propia

Figura 56: Ingreso de solicitud de designación de docentes revisores



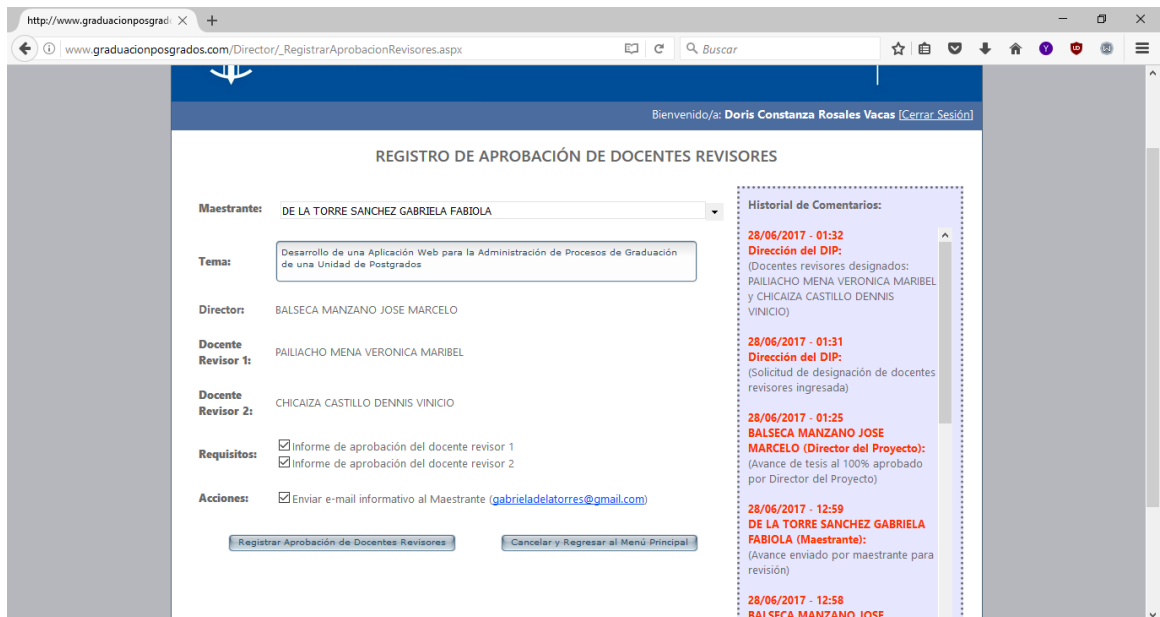
Fuente: Elaboración propia

Figura 57: Registro de designación de docentes revisores



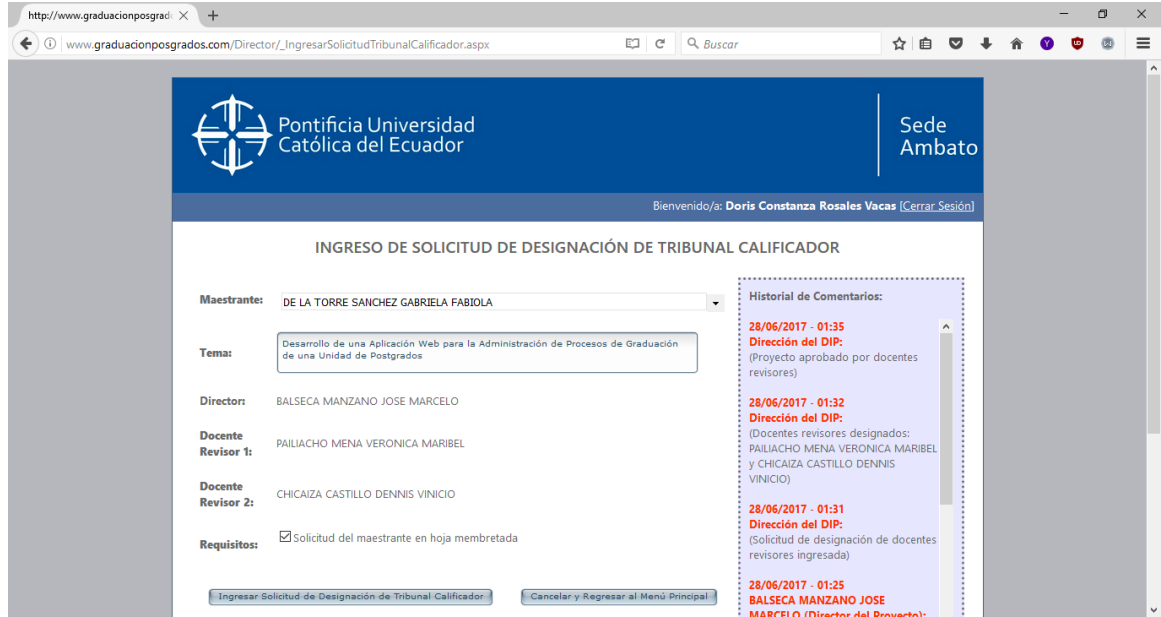
Fuente: Elaboración propia

Figura 58: Registro de aprobación por docentes revisores



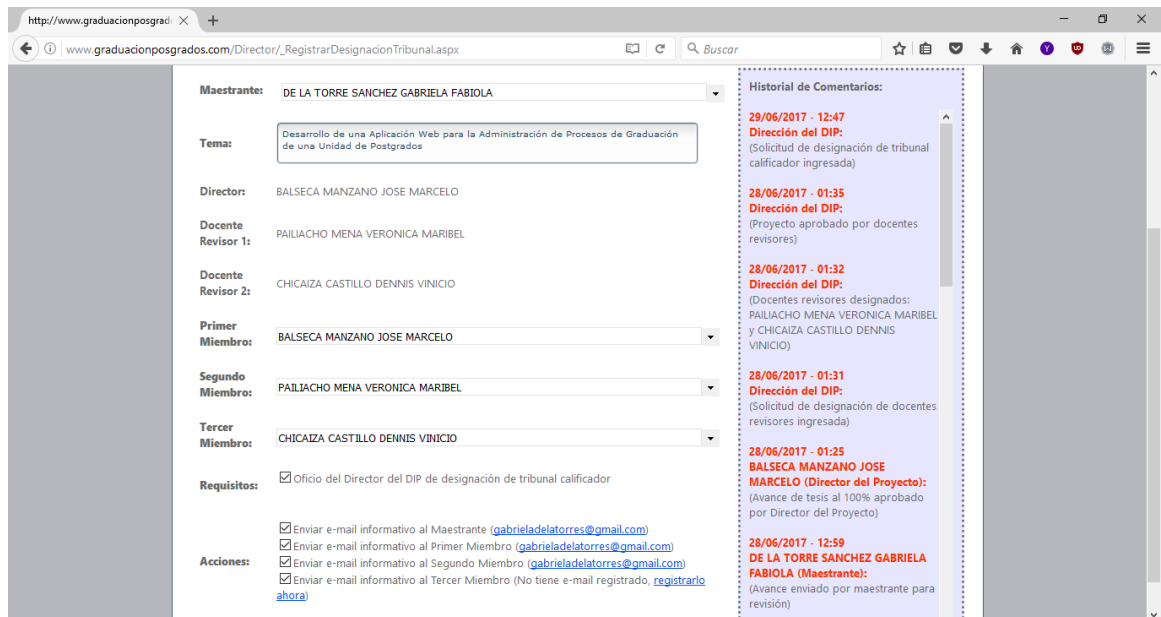
Fuente: Elaboración propia

Figura 59: Ingreso de solicitud de designación de tribunal calificador



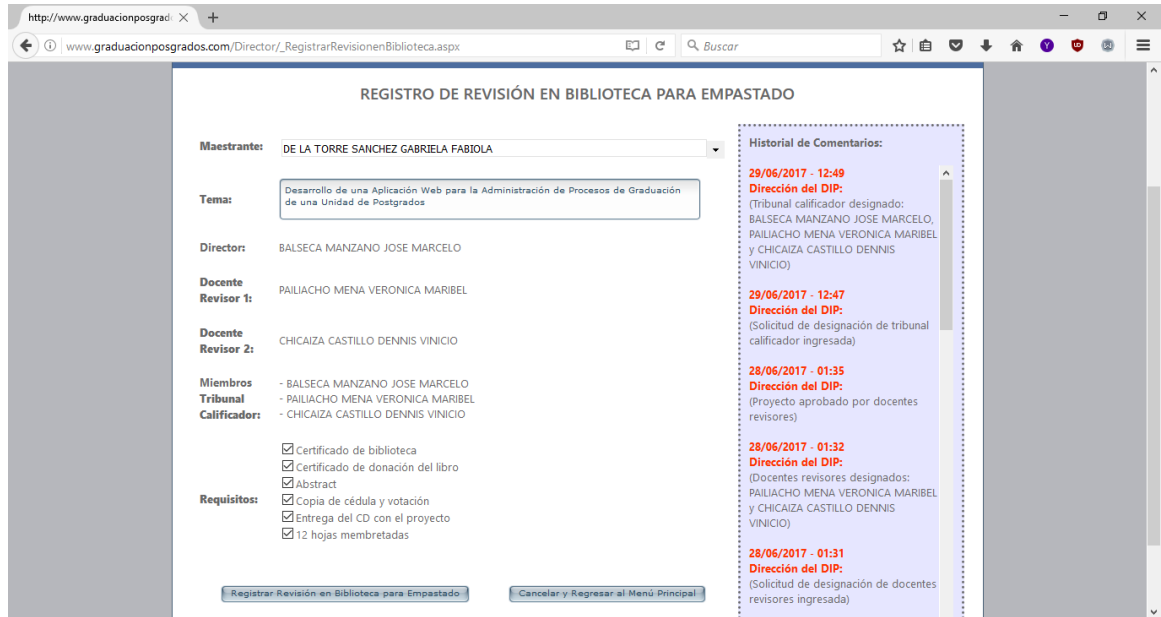
Fuente: Elaboración propia

Figura 60: Registro de designación de tribunal calificador



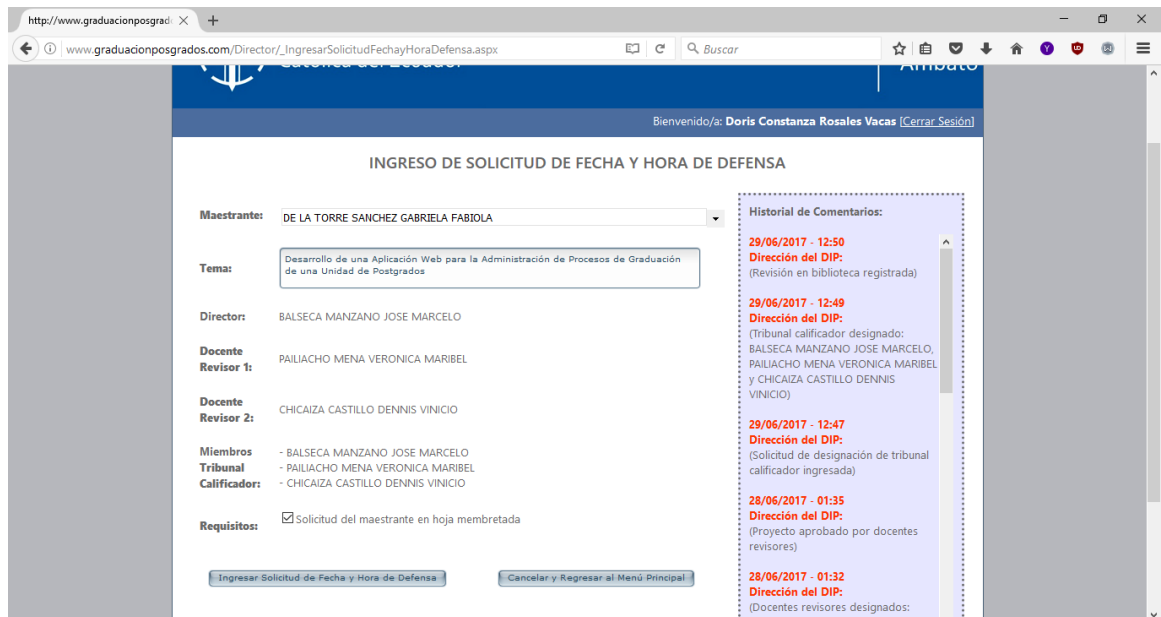
Fuente: Elaboración propia

Figura 61: Registro de revisión en biblioteca para elaboración del empastado



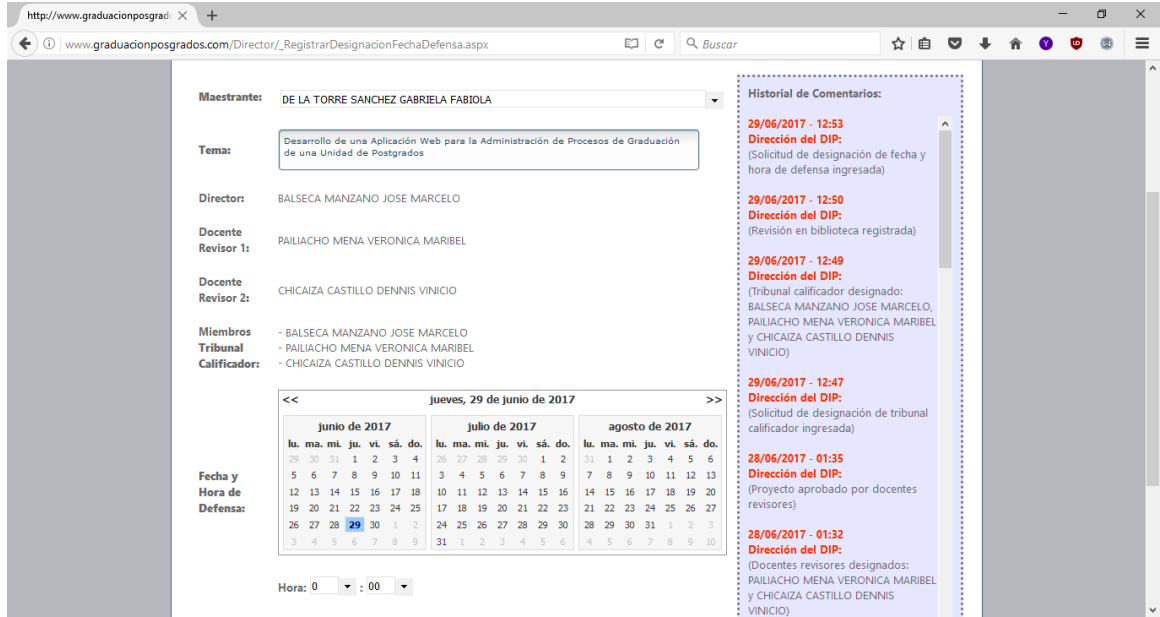
Fuente: Elaboración propia

Figura 62: Ingreso de solicitud de designación de fecha y hora de defensa



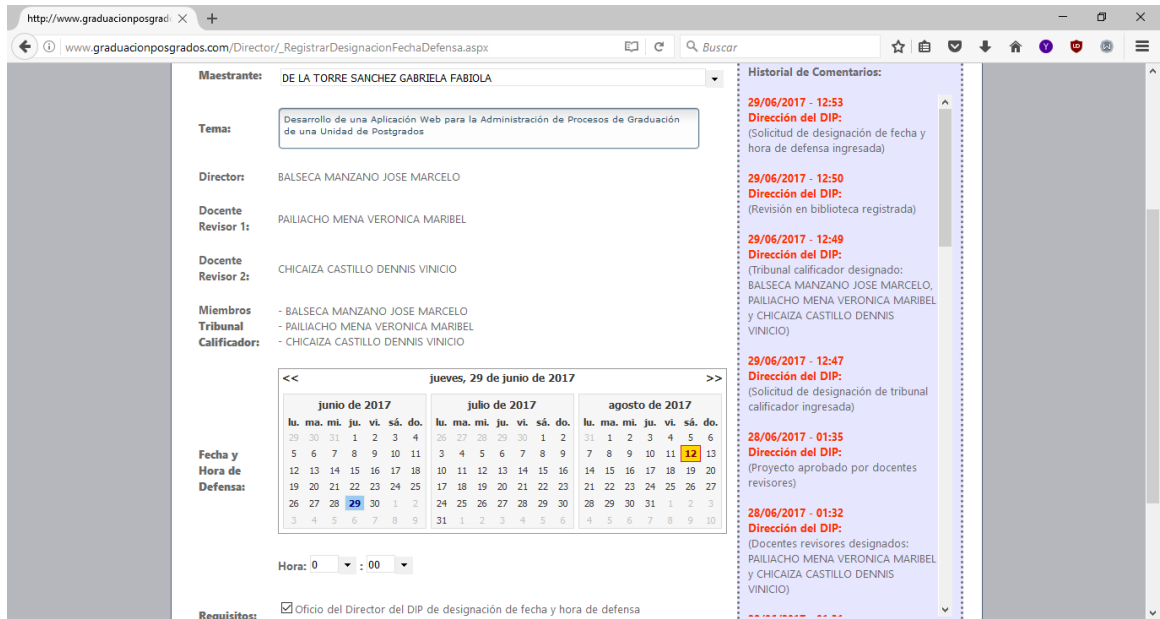
Fuente: Elaboración propia

Figura 63: Registro de designación de fecha y hora de defensa



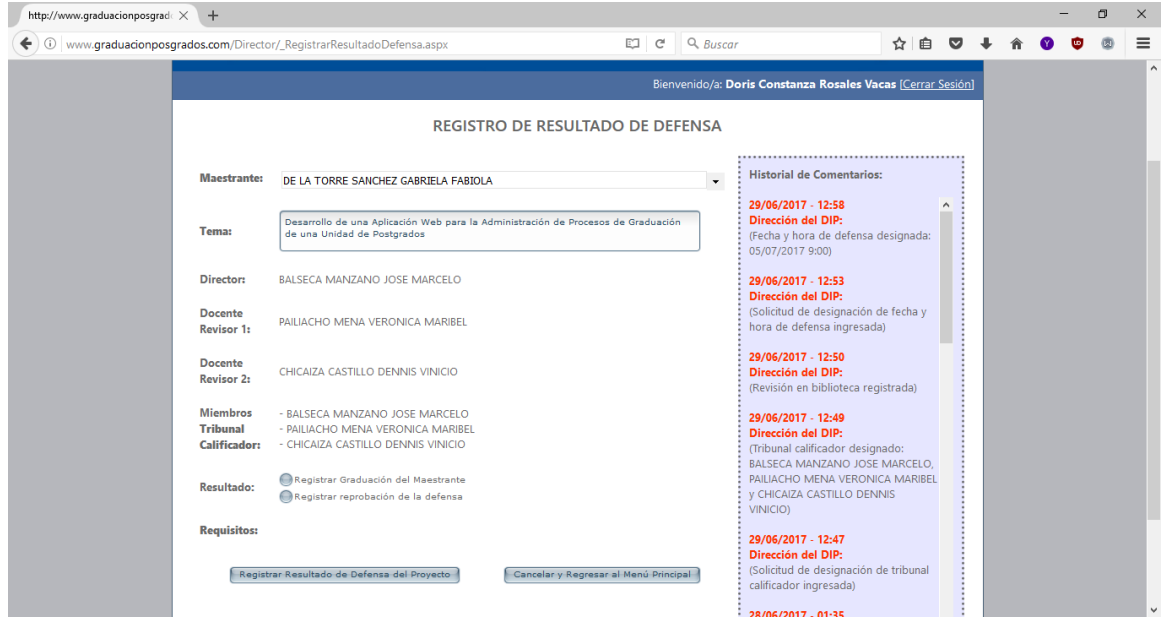
Fuente: Elaboración propia

Figura 64: Registro de designación de fecha y hora de defensa



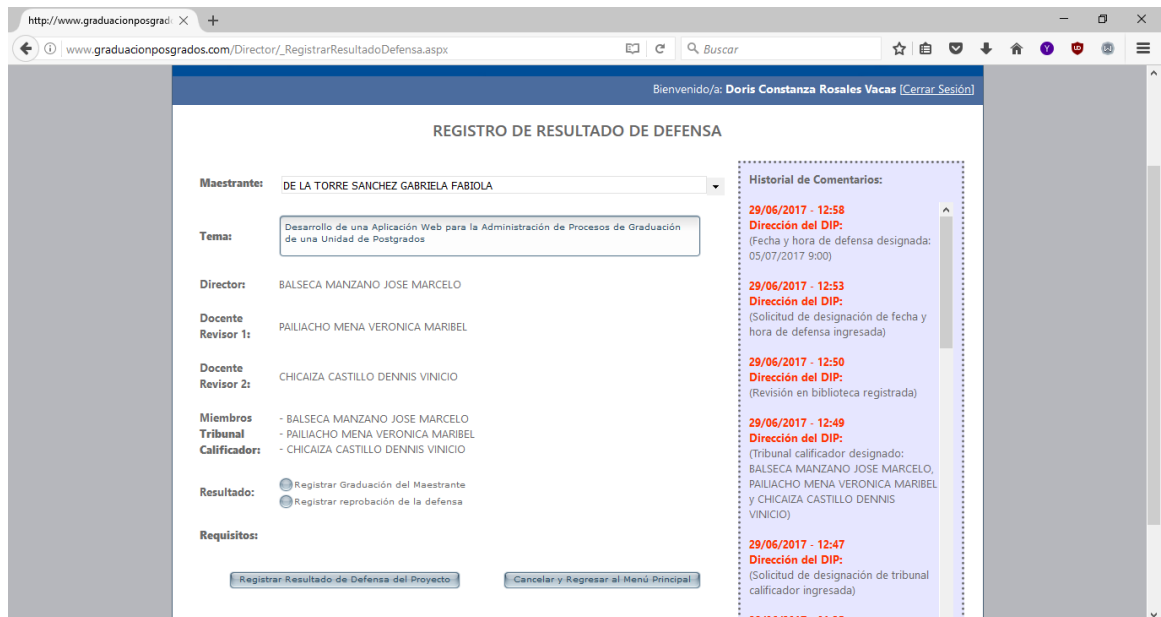
Fuente: Elaboración propia

Figura 65: Registro de resultado de defensa



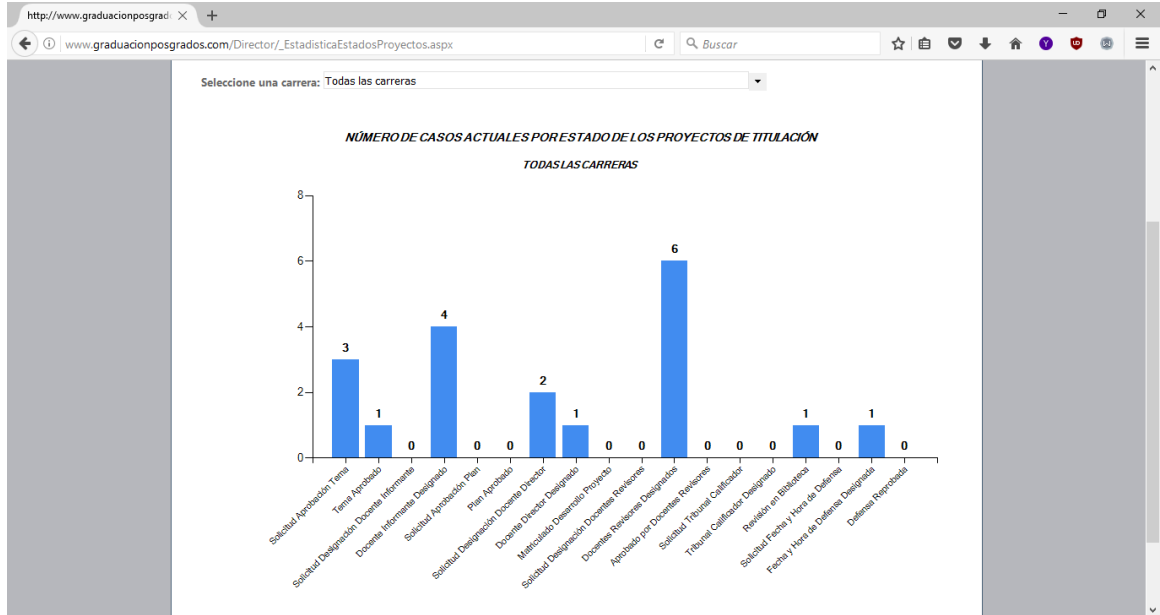
Fuente: Elaboración propia

Figura 66: Registro de resultado de defensa



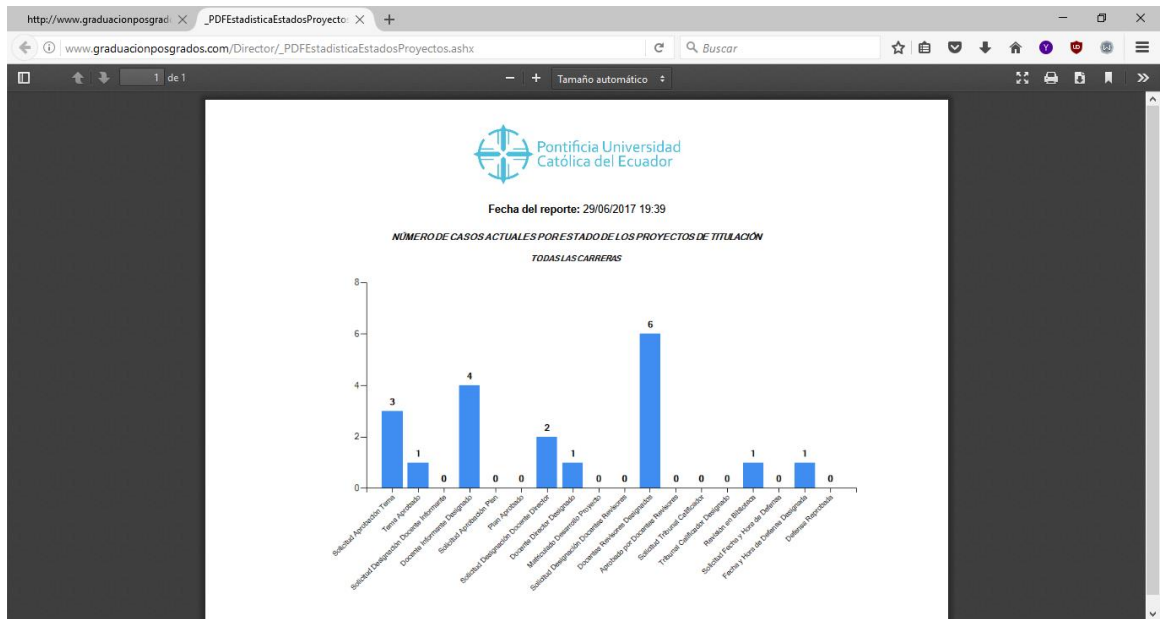
Fuente: Elaboración propia

Figura 67: Gráfico estadístico de número de casos por estado del proceso de graduación



Fuente: Elaboración propia

Figura 68: Gráfico estadístico de número de casos por estado del proceso de graduación en formato PDF



Fuente: Elaboración propia

5.2. Evaluación preliminar

La aplicación web resultante del presente proyecto de titulación ha sido suficientemente probada y evaluada en todas y cada una de sus secciones, tanto durante el desarrollo de las mismas como desde un punto de vista global una vez que la aplicación ha sido finalizada. Esto se puede evidenciar tanto en las figuras incluidas en el presente capítulo así como mediante las pruebas funcionales y de usuario realizadas cuyos resultados se detallan en el capítulo anterior.

5.3. Análisis de resultados

De manera general, se puede decir que los resultados obtenidos a partir de las pruebas realizadas que se especifican en las tablas anteriores, son muy satisfactorios ya que se ha verificado el correcto funcionamiento de la aplicación en relación a todos los aspectos evaluados. Atendiendo a los lineamientos establecidos por la metodología de desarrollo de software XP, en especial las relativas al desarrollo iterativo e incremental, la presente aplicación ha sido desarrollada atendiendo a los mismos, ya que todo módulo se inició mediante un desarrollo simple que se fue corrigiendo paulatinamente hasta que todas las pruebas funcionales fueran superadas satisfactoriamente, y luego se fue aumentando su funcionalidad para seguir cumpliendo con las tareas y requerimientos identificados en fase de análisis hasta donde fuera pertinente, todo lo cual se logró a través de múltiples iteraciones en el desarrollo.

En relación a los objetivos establecidos inicialmente para el presente proyecto, se puede decir que todos se han cumplido satisfactoriamente. La aplicación web desarrollada permite gestionar y administrar los procesos de graduación de postgrados de la PUCESA, automatizando convenientemente todo el proceso. Por otro lado, se logró identificar los principales factores que ocasionan pérdidas de tiempo debidos a la forma tradicional de gestión de dichos procesos de graduación, y la presente aplicación se desarrolló en el sentido de eliminar dichos problemas y optimizar todo el proceso en general en la medida que fuera posible. Además de eso, el presente aplicativo web ofrece cierta información estadística clave en forma de gráficos de pastel y de barras que se convierten en un apoyo para el director o directora del DIP al momento de tomar decisiones.

Para lograrlo, se ha realizado un estudio exhaustivo de las tecnologías y métodos aplicados en otras instituciones de educación superior tanto a nivel nacional como internacional ([12], [34], [35], [17]), donde ya se han llevado a cabo procesos de automatización similares, pero siempre atendiendo

a la tecnología y herramientas disponibles en la PUCESA, sobre todo para lograr la compatibilidad requerida con Academics y sus bases de datos.

Por último, como ya se ha mencionado, la presente aplicación se ha desarrollado atendiendo a los principios y fundamentos de la metodología de desarrollo de software XP, puesto que los mismos permiten obtener un producto de software de alta calidad en un tiempo relativamente corto, atendiendo sobre todo al fiel cumplimiento de requerimientos y a la total satisfacción del cliente o usuario final.

5.3.1. Validación técnica y funcional de la aplicación

Una vez que finaliza el desarrollo de la aplicación, la misma procede a ser validada desde los puntos de vista técnico y funcional, por parte del personal correspondiente de la PUCESA. Desde el punto de vista técnico, primeramente se realiza una reunión con el personal del departamento de TI de la institución, en concreto con el Ing. Gabriel Altamirano y el Ing. José Enríquez, con la finalidad de conversar sobre los pormenores y detalles del proyecto y de su desarrollo. A continuación, el Ing. José Enríquez procede a realizar un análisis detenido sobre el código y los lineamientos de desarrollo que se han seguido para la presente aplicación, verificando que la misma cumple satisfactoriamente con los principios de mantenibilidad y legibilidad del código así como del desarrollo en capas, uso de las librerías .NET 4.0, .NET Entity Framework, componentes Obout, uso del entorno de desarrollo Microsoft Visual Studio 2010 y base de datos Microsoft SQL Server 2008 R2, y demás principios que desde un comienzo se tomaron en cuenta con estricto apego con la finalidad de lograr una aplicación totalmente compatible con Academics [26], lo cual a su vez es de vital importancia para el exitoso funcionamiento del proyecto.

Una vez que el Ing. José Enríquez emite un informe sobre los aspectos mencionados, el Ing. Gabriel Altamirano procede a autorizar la instalación de una versión de prueba de la aplicación en los servidores de la PUCESA. Esto se lo realiza de manera presencial en la institución con la asistencia del Ing. José Enríquez, ya que es él la persona autorizada a acceder a dichos servidores, instalándose y poniéndose en ejecución exitosamente la aplicación web.

Inmediatamente se habilita un enlace y se informa al personal del DIP para que el mismo realice pruebas de uso sobre la aplicación, lo cual correspondería a la validación funcional de la aplicación, y también se realiza exitosamente utilizando información real de casos actuales de maestrantes de

diferentes maestrías. Como prueba de ello, se observa que tanto el Mtr. Diego Jiménez como la Lcda. Doris Rosales, las personal que manejan el DIP, responden favorablemente a unos cuestionarios de satisfacción que se les aplica para tal fin, los mismos que incluyen firmas de responsabilidad. Se observa copias fieles de las páginas de dichos cuestionarios en el apéndice D, y a continuación se presenta una lista de cotejo donde se resume las preguntas de evaluación aplicadas y el número de respuestas positivas y negativas obtenidas.

Tabla 18: Lista de cotejo sobre el cuestionario de satisfacción aplicado al personal del DIP

| Aspecto a evaluar | Número de respuestas | |
|--|----------------------|----------|
| | Sí | No |
| ¿Permite la aplicación registrar detalladamente todos los pasos del proceso de graduación a los que se someten los maestrantes de la PUCESA? | 2 | 0 |
| ¿Permite la aplicación verificar el cumplimiento de los diferentes requisitos correspondientes a cada paso del proceso de graduación de posgrados de la PUCESA? | 2 | 0 |
| ¿Detecta la aplicación cuando el usuario ingresa datos erróneos o realiza operaciones inconsistentes, y despliega mensajes de error adecuados y comprensibles para el usuario? | 2 | 0 |
| ¿Refleja la aplicación correctamente la información correspondiente a maestrantes y docentes existentes en las bases de datos de la PUCESA? | 2 | 0 |
| ¿Refleja la aplicación correctamente la secuencia de pasos correspondientes al proceso de graduación de posgrados de la PUCESA? | 2 | 0 |
| ¿Despliega la aplicación un historial de comentarios con fecha y horas exactas sobre los diferentes pasos que tienen lugar dentro de cada proceso de graduación de posgrados de la PUCESA? | 2 | 0 |
| ¿La aplicación es intuitiva, amigable y fácil de usar? | 2 | 0 |
| ¿Genera la aplicación la información estadística requerida por el DIP de manera clara y entendible? | 2 | 0 |
| ¿Permite la aplicación exportar la información estadística generada de modo que sea fácil portarla e imprimirla? | 2 | 0 |
| ¿Permite la aplicación generar la información estadística requerida tanto a nivel general como por cada una de las diferentes carreras de posgrado de la PUCESA? | 2 | 0 |
| En general, ¿considera usted que la aplicación cumple satisfactoriamente los objetivos para los que fue creada y será de la utilidad prevista para el DIP? | 2 | 0 |
| TOTAL DE RESPUESTAS OBTENIDAS: | 22 | 0 |

Fuente: Elaboración propia

De la lista de cotejo anterior se puede deducir que, una vez que el personal del DIP y futuros usuarios de la aplicación la han evaluado de manera exhaustiva y total, usando información real y actualizada del departamento, están de acuerdo en que todos los aspectos evaluados sobre la aplicación y su uso funcionan de manera correcta y satisfactoria, por lo que todas las respuestas obtenidas son positivas y no hay ninguna negativa (ver el apéndice D). Paralelamente a los cuestionarios aplicados, el Mtr. Diego Jiménez, Coordinador del departamento, expresa por escrito al Ing. Gabriel Altamirano su conformidad y satisfacción con la aplicación evaluada, por lo que el mismo procede a autorizar su puesta en producción, lo cual obviamente estará a cargo del personal del departamento de TI de la institución, ya que son las únicas personas autorizadas a realizar este tipo de tareas en los servidores de la PUCESA.

Capítulo 6

Conclusiones y Recomendaciones

6.1. Conclusiones

- Los procesos tradicionales de graduación a los que deben someterse los maestrantes de las diferentes carreras de posgrados de la PUCESA son, en su gran mayoría, procesos que deben realizarse manualmente y presencialmente por parte de los maestrantes, debiendo estos además imprimir la mayoría de los documentos requeridos, lo cual genera un notable desperdicio de tiempo y otros recursos a todas las partes implicadas.
- Existe una gran cantidad de tecnologías y herramientas que pueden ser aplicadas en la actualidad para el desarrollo de aplicaciones web de alto rendimiento, en particular de aquellas destinadas a la automatización de procesos universitarios. Algunas instituciones ecuatorianas han hecho uso de dichas tecnologías de manera exitosa, sin embargo, la mayoría de ellas incluyendo la PUCESA no hay automatizado ni optimizado aún muchos de sus procesos, pese a contar con las herramientas informáticas adecuadas, desaprovechando los recursos existentes.
- Se desarrolla una nueva aplicación web que se integra y es compatible con las herramientas, datos, aplicaciones y tecnologías existentes en la PUCESA, orientada a la optimización de los procesos de graduación del DIP, en especial en lo que se refiere a ahorro de tiempo. Dicha aplicación resulta igualmente útil a todas las partes implicadas en el mencionado proceso, esto es maestrantes, docentes y personal administrativo del DIP, permitiéndoles dar solución a sus respectivos requerimientos de manera remota y sin restricciones de horarios, ya que toda la actividad pasa a desarrollarse a través de Internet.

6.2. Recomendaciones

- Adoptar la metodología de desarrollo ágil XP para la creación de las aplicaciones requeridas para las diferentes áreas y departamentos de la PUCESA, ya que permite obtener software de calidad en un tiempo relativamente corto, orientándose hacia el cumplimiento de requerimientos y la satisfacción del cliente.
- Adquirir o desarrollar un sistema de gestión documental general para la PUCESA, que permita la generación, envío y seguimiento de documentos electrónicos entre todas las áreas, departamentos y secciones de la institución, a fin de agilizar los procesos en general y evitar el desperdicio de recursos no renovables.
- Analizar periódicamente los procesos a los que se someten los usuarios de los diferentes departamentos de la PUCESA a fin de identificar factores que generen pérdidas de tiempo y otros recursos, para darles solución oportuna aplicando un plan de reingeniería constante de procesos.

Apéndice A

Descripción de tablas y campos

A.1. Bases de datos creadas para la aplicación

Para la presente aplicación, se ha creado la base de datos “Graduacion” bajo el motor de bases de datos “SQL Server 2008 R2”. Esta base de datos permitirá almacenar toda la información que no se encuentre previamente almacenada en las bases de datos del “Academics”, y que es requerida para el correcto funcionamiento de la presente aplicación. La razón para haber elegido el motor de base de datos “SQL Server 2008 R2” es básicamente crear una base de datos totalmente compatible con las del mencionado sistema “Academics”, ya que posteriormente deberán integrarse y coexistir dentro del mismo servidor e intercambiar información.

Adicionalmente, la presente aplicación también extrae información de algunas de las tablas pertenecientes a la base de datos “AtencionUsuario”, la misma que corresponde a la aplicación a desarrollarse en [32]. Dicha base de datos también está desarrollada dentro de SQL Server 2008 R2, ya que también debe guardar compatibilidad con Academics.

A.2. Tablas creadas para la base de datos “Graduación”

A.2.1. Tabla “TemasdeTesis”

Esta tabla tiene como objeto el almacenar la información correspondiente a los temas de tesis que los estudiantes de postgrado envían como etapa inicial de su proceso de graduación.

A.2.1.1. Campos de la tabla “TemasdeTesis”

Tabla 19: Campos de la tabla “TemasdeTesis”

| Nombre del Campo | Tipo de dato | Comentarios |
|------------------|--------------|--|
| temald | Entero | Clave principal |
| tema | Varchar(MAX) | Es el tema propiamente dicho. Puede ser un texto largo |
| directorId | Entero | Identificador del maestro director de tesis |
| resumen | Varchar(MAX) | Es el resumen correspondiente al tema de proyecto de titulación. Puede ser un texto largo |
| directorAprobado | Bit | Especifica si el director de tesis está aprobado para el presente tema |
| activo | Bit | Especifica si el tema de proyecto de titulación se encuentra activo |
| estado | Entero | Guarda un valor que indica el estado en el que se encuentra el presente tema de proyecto de titulación. A continuación se incluye una tabla explicativa sobre los diferentes valores que puede contener este campo |
| estudianteld | Entero | Identificador del Estudiante al que corresponde el tema |
| fechaAprobacion | Fecha/Hora | Fecha y hora en las que ha sido aprobado el tema en cuestión por parte del director del DIP |
| fechaUltimoEnvio | Fecha/Hora | Fecha y hora en la que el tema ha sido enviado para revisión / aprobación por última vez |

Fuente: elaboración propia

A.2.1.2. Posibles estados de un tema de proyecto de titulación de posgrados

La siguiente tabla especifica los diferentes valores que puede contener el campo “estado” de la tabla “TemasdeTesis”:

Tabla 20: Posibles estados de un tema de proyecto de titulación de posgrados

| Valor | Significado |
|--------------|--|
| 0 | <p>El estudiante ha guardado datos en los demás campos de la tabla “TemasdeTesis”, pero el tema no ha sido enviado aún para revisión o aprobación. O bien, el tema ha sido devuelto por el Director de Tesis o por el Director del DIP para corrección.</p> <p>En este estado, el tema puede ser editado por el estudiante y enviado para revisión y aprobación.</p> |
| 1 | <p>El estudiante ha enviado el tema de proyecto de titulación para su revisión y aprobación. Implícitamente, se ha creado o bien una solicitud de dirección de tesis enviada al profesor elegido por el estudiante, o bien una notificación de aplicación de correcciones en caso de que ya existiere un Director de tesis confirmado, y por tanto de que se estuvieren aplicando correcciones al tema.</p> <p>En este estado, el tema no puede ser editado por el estudiante y éste deberá esperar a que el tema le sea definitivamente aprobado, definitivamente desaprobado, o se le envíen a hacer correcciones.</p> |
| 2 | <p>El Profesor Director de tesis ha aprobado el tema y resumen enviados por el estudiante, y ha enviado dicho tema al Director del DIP para su aprobación definitiva.</p> <p>En este estado, el tema no puede ser editado por el estudiante y éste deberá esperar a que el tema le sea definitivamente aprobado, definitivamente desaprobado, o se le envíen a hacer correcciones.</p> |
| 3 | <p>El Director del DIP ha aprobado definitivamente el tema de proyecto de titulación, y por tanto el estudiante ya puede proceder a elaborar y enviar el plan de titulación.</p> |

| | |
|---|--|
| | <p>En este estado, el tema de proyecto de titulación ya no puede sufrir modificación alguna, excepto la anulación del mismo por parte del Director del DIP, ya sea por solicitud expresa del Estudiante, por caducidad del tema o por cualquier otro motivo considerado por los reglamentos internos del DIP.</p> |
| 4 | <p>El Director del DIP ha desaprobado definitivamente el tema de proyecto de titulación, y por tanto el estudiante debe proceder a elaborar y enviar un nuevo tema. Implícitamente, el campo “activo” correspondiente al tema de proyecto de titulación en la tabla “TemasdeTesis” pasará a contener el valor “falso”.</p> <p>En este estado, el tema de proyecto de titulación ya no puede sufrir modificación alguna, pero se quedará registrado en la base de datos de la aplicación, principalmente para fines informativos.</p> |
| 5 | <p>El tema ha sido dado de baja por el Director del DIP, pese a haber sido previamente aprobado por el mismo, ya sea por solicitud expresa del Estudiante, por caducidad del tema o por cualquier otro motivo considerado por los reglamentos internos del DIP.</p> <p>En este estado, el tema de proyecto de titulación ya no puede sufrir modificación alguna, pero se quedará registrado en la base de datos de la aplicación, principalmente para fines informativos.</p> |

Fuente: elaboración propia

A.2.2. Tabla “MensajesTemasdeTesis”

Esta tabla tiene por objeto guardar los mensajes que se van originando para un tema de proyecto de titulación específico, ya sea por parte del estudiante, de su Director de proyecto o del Director del DIP. Cuando se dan devoluciones de un tema por parte del Director de proyecto o del Director del DIP, éstos pueden adjuntar mensajes al estudiante para que el mismo sepa qué debe corregir o por

qué razón se le ha devuelto el tema. La presente tabla por tanto, guarda el historial de estos mensajes.

A.2.2.1. Campos de la tabla “MensajesTemasdeTesis”

Tabla 21: Campos de la tabla “MensajesTemasdeTesis”

| Nombre del Campo | Tipo de dato | Comentarios |
|------------------|----------------|--|
| mensajeId | Entero | Clave principal |
| mensaje | Varchar(MAX) | Es el mensaje propiamente dicho. Puede ser un texto largo |
| temaId | Entero | Identificador del tema de proyecto de titulación al que corresponde el mensaje |
| fechaHora | Fecha/Hora | Guarda la fecha y hora en la que se generó el mensaje |
| autorId | Entero | Identificador del autor del mensaje, sea este el estudiante, el Director de tesis o el Director del DIP |
| tipoAutor | Entero pequeño | Indica el tipo de autor del mensaje. A continuación se detalla los valores que puede contener este campo y su significado |
| tipoMensaje | Entero pequeño | Indica el tipo de mensaje emitido. La información contenida en este campo puede servir principalmente para la generación de reportes con propósitos estadísticos (gerenciales). A continuación se detalla los valores que puede contener este campo y su significado |

Fuente: elaboración propia

A.2.2.2. Tipos de autores que pueden generar mensajes para un tema de proyecto de titulación de Posgrados

En la siguiente tabla se detalla los valores que puede contener el campo “tipoAutor” de la tabla “MensajesTemasdeTesis”.

Tabla 22: Tipos de autores para la tabla “MensajesTemasdeTesis”

| Valor | Significado |
|-------|--|
| 0 | El mensaje ha sido generado por el estudiante |
| 1 | El mensaje ha sido generado por el docente Director de tesis |
| 2 | El mensaje ha sido generado por el Director del DIP |

Fuente: elaboración propia

A.2.2.3. Tipos de mensajes que pueden ser generados para un tema de proyecto de titulación de posgrados

En la siguiente tabla se detalla los valores que puede contener el campo “tipoMensaje” de la tabla “MensajesTemasdeTesis”.

Tabla 23: Tipos de mensajes que se pueden generar para la tabla “MensajesTemasdeTesis”

| Valor | Significado |
|-------|--|
| 0 | El mensaje corresponde a una solicitud de dirección de tesis, o bien de revisión de corrección del tema enviada por el estudiante |
| 1 | El mensaje corresponde a una devolución por parte del maestro Director de tesis para que se realicen correcciones por parte del estudiante |
| 2 | El mensaje corresponde a la aprobación del tema por parte del maestro Director de tesis |
| 3 | El mensaje corresponde al rechazo de dirección del tema por parte del maestro propuesto como director |
| 4 | El mensaje corresponde a una devolución por parte del Director del DIP para que se realicen correcciones por parte del estudiante |
| 5 | El mensaje corresponde a la aprobación definitiva del tema por parte del Director del DIP |

| | |
|---|--|
| 6 | El mensaje corresponde a la desaprobación definitiva del tema, o bien a su levantamiento o anulación, por parte del Director del DIP |
|---|--|

Fuente: elaboración propia

A.2.3. Tabla “PlanesdeTesis”

Esta tabla tiene por objeto guardar la información correspondiente a los planes de proyectos de titulación de posgrados enviados por los estudiantes en proceso de grado. Hay que mencionar que, debido a la naturaleza de la información contenida en los planes de titulación de posgrados, la tabla “PlanesdeTesis” por sí sola no basta para almacenar toda la información que puede contener un plan, sino que se necesitan varias tablas adicionales para almacenar toda la información completa. Dichas tablas serán listadas y descritas más adelante.

A.2.3.1. Campos de la tabla “PlanesdeTesis”

La tabla “Planes de Tesis” es sin duda la tabla más grande y compleja dentro del presente trabajo de desarrollo, ya que consta de 41 campos de diferentes tipos y longitudes. A continuación se describen dichos campos.

Tabla 24: Campos de la tabla “PlanesdeTesis”

| Nombre del campo | Tipo de datos | Descripción |
|------------------|---------------|---|
| plandeTesisId | Entero | Identificador del plan de titulación, que sirve como clave principal |
| TemaId | Entero | Clave foránea que indica con qué registro se relaciona en la tabla “TemasdeTesis” |
| CarreraId | Entero | Contiene un identificador que permite saber a qué carrera pertenece el plan en cuestión. La información sobre dichas carreras |

| | | |
|----------------------------|--------------|---|
| | | se encuentra almacenada en las bases de datos preexistentes de Academics |
| TipoProyecto | Entero | Permite identificar mediante un número el tipo de proyecto al que se refiere el plan en cuestión |
| ClaseProyecto | Entero | Permite identificar mediante un número la clase de proyecto a la que se refiere el plan en cuestión |
| Costo | Decimal | Permite almacenar el valor monetario estimado que costará el desarrollo del proyecto en cuestión |
| Duración | Varchar(50) | Almacena la duración aproximada que se estima tomará el desarrollo del proyecto en cuestión |
| Lugar | Varchar(200) | Almacena el lugar donde principalmente se desarrollará el proyecto en cuestión |
| DocenteId | Entero | Almacena un valor numérico que identifica al docente designado como director del proyecto. La información sobre los docentes se encuentra almacenada en las bases de datos de Academics |
| DocenteTituloPregradoId | Entero | Almacena un valor numérico que identifica el título de pregrado del docente director de proyecto que más se ajusta al proyecto en cuestión |
| DocenteTituloPosgradoId | Entero | Almacena un valor numérico que identifica el título de posgrado del docente director de proyecto que más se ajusta al proyecto en cuestión |
| DocenteTituloPosdoctoralId | Entero | Permite identificar el título de formación posdoctoral del docente director del proyecto que más se ajusta a la realidad del mismo |

| | | |
|----------------------------|--------------|--|
| EstudianteTituloPregradoId | Entero | Permite identificar el título de pregrado del maestrante que más se ajusta a la realidad de la carrera y proyecto |
| VinculoPuce | Bit | Indica si el docente director del proyecto tiene vínculo con la Universidad Católica |
| FechaOriginal | Fecha/Hora | Almacena la fecha en la que fue enviado el plan por primera vez por parte del maestrante |
| OtraUniversidad | Varchar(100) | Almacena la información de otra Universidad de vínculo con el director del proyecto, en caso de no ser la PUCE |
| Resumen | Varchar(MAX) | Almacena el resumen propio del plan de titulación, que puede ser igual al del tema o puede haber variado |
| DescripcionProblema | Varchar(MAX) | Guarda información sobre la descripción detallada del problema que se pretende solucionar mediante el desarrollo del presente proyecto |
| Meta | Varchar(MAX) | Permite guardar la información correspondiente a la meta que se persigue |
| VariableDependiente | Varchar(MAX) | Guarda información correspondiente a la variable independiente, útil sobre todo en el caso de Investigaciones |
| ObjetivoGeneral | Varchar(MAX) | Almacena el objetivo general del presente proyecto |
| EstadodelArte | Varchar(MAX) | Almacena el estado del arte para el presente plan de titulación |
| Metodología | Varchar(MAX) | Permite guardar información sobre la metodología general que se prevé aplicar para el desarrollo del presente proyecto |
| PoblacionyMuestra | Varchar(MAX) | Guarda información sobre poblaciones y muestras, útil sobre todo en el caso de investigaciones |

| | | |
|-----------------------|--------------|---|
| MetodoGeneral | Varchar(MAX) | Almacena información sobre el método general a aplicarse para el desarrollo del proyecto en cuestión |
| EvaluacionPreliminar | Varchar(MAX) | Almacena información sobre las técnicas de evaluación preliminares que se aplicará para el presente proyecto |
| HipotesisdeTrabajo | Varchar(MAX) | Almacena información útil sobre todo en el caso de investigaciones. |
| VariableIndependiente | Varchar(MAX) | Permite guardar información muy útil en el caso de las investigaciones |
| PreguntadeEstudio | Varchar(MAX) | Guarda la pregunta principal que da origen al desarrollo del proyecto en cuestión |
| MetodoEspecifico | Varchar(MAX) | Almacena detalles sobre la metodología a aplicarse en el proyecto en cuestión |
| DocenteLectorId | Entero | Permite identificar al docente investigador designado por el DIP para la revisión de un determinado plan de titulación |
| Estado | Entero | Es un campo clave dentro del proceso de graduación del estudiante de posgrados, ya que permite conocer en qué estado se encuentra el plan de titulación elaborado y/o enviado por el estudiante. A continuación se detalla los diferentes estados en los que puede encontrarse un plan de titulación de posgrados |
| Activo | Bit | Permite identificar si el plan en cuestión se considera activo o no |
| FechaUltimoEnvio | Date/Time | Almacena la fecha en la que por última vez el maestrante envió su plan de titulación |
| FechaAprobacion | Date/Time | Almacena la fecha en la que un plan de titulación es definitivamente aprobado por la dirección del DIP |
| ComoAparece | Varchar(MAX) | Almacena las respuestas a dichas preguntas |

| | | |
|-----------------|--------------|--|
| PorqueseOrigina | Varchar(MAX) | en relación del tema de proyecto de titulación a desarrollarse |
| QueloOrigina | Varchar(MAX) | |
| CuandoseOrigina | Varchar(MAX) | |
| DondeseOrigina | Varchar(MAX) | |
| DondeseDetecta | Varchar(MAX) | |

Fuente: elaboración propia

A.2.3.2. Posibles estados de un plan de titulación de posgrados

Tabla 25: Posibles estados de un plan de titulación de posgrados

| Estado | Descripción |
|--------|--|
| 0 | El plan de titulación es editable por parte del estudiante. Este estado se da cuando el plan recién ha sido creado dentro de la base de datos, o bien cuando ha sido devuelto por el director de tesis, el docente informante asignado o la dirección del DIP para aplicar correcciones |
| 1 | El plan ha sido enviado por el estudiante a su director de tesis para que el mismo lo apruebe o se lo devuelva para aplicar correcciones. En este estado, el plan no es editable ni visible por el estudiante, docente informante o director del DIP, sino únicamente por el director de tesis |
| 2 | El plan ha sido aprobado por el director de tesis y requiere asignación de docente informante. En este estado, el plan es visible únicamente por la dirección del DIP quien debe proceder a asignar un docente informante |
| 3 | La dirección del DIP ha asignado un docente informante, y el plan está en espera de que dicho docente informante lo revise y apruebe o envíe a aplicar correcciones adicionales. En este estado, el plan es visible únicamente por el docente informante asignado |
| 4 | El plan está listo para su revisión final por parte de la dirección del DIP, sea que el docente informante haya enviado a aplicar correcciones o lo haya aprobado directamente. En este estado, el plan es visible únicamente por la dirección del DIP |

| | |
|---|--|
| 5 | El plan ha sido definitivamente aprobado por la dirección del DIP. En este estado, el plan ya no volverá a ser editable por el estudiante, y a su vez éste podrá ahora acceder a la interfaz para comenzar a enviar avances del desarrollo mismo de su trabajo de tesis de grado |
|---|--|

Fuente: elaboración propia

A.2.4. Tabla “MensajesPlandeTesis”

Esta tabla tiene por objeto guardar todos los mensajes que se generen para un plan de titulación, durante su proceso de envío por parte del estudiante, aprobación o devolución por parte de su docente director, asignación de docente informante, aprobación o devolución por parte del docente informante, y aprobación definitiva o devolución por parte del director del DIP. Estos mensajes serán desplegados en orden cronológico en el editor del plan al que tiene acceso el estudiante, o la página de revisión a la que tienen acceso los docentes y el director del DIP.

A.2.4.1. Campos de la tabla “MensajesPlandeTesis”

Tabla 26: Campos de la tabla “MensajesPlandeTesis”

| Nombre del Campo | Tipo de dato | Comentarios |
|------------------|--------------|--|
| mensajeId | Entero | Clave principal |
| mensaje | Varchar(MAX) | Es el texto del mensaje propiamente dicho. Puede ser un texto largo |
| planId | Entero | Identificador del plan de titulación al que corresponde el mensaje |
| fechaHora | Fecha/Hora | Guarda la fecha y hora en la que se generó el mensaje |
| autorId | Entero | Identificador del autor del mensaje, sea este el estudiante, el Director de tesis, el docente informante o el Director del DIP |

| | | |
|-------------|----------------|--|
| tipoAutor | Entero pequeño | Indica el tipo de autor del mensaje. A continuación se detalla los valores que puede contener este campo y su significado |
| tipoMensaje | Entero pequeño | Indica el tipo de mensaje emitido. La información contenida en este campo puede servir principalmente para la generación de reportes con propósitos estadísticos (gerenciales). A continuación se detalla los valores que puede contener este campo y su significado |

Fuente: elaboración propia

A.2.4.2. Tipos de autores que pueden generar mensajes para un plan de titulación de Posgrados

En la siguiente tabla se detalla los valores que puede contener el campo “tipoAutor” de la tabla “MensajesPlandeTesis”.

Tabla 27: Tipos de autores para la tabla “MensajesPlandeTesis”

| Valor | Significado |
|-------|--|
| 0 | El mensaje ha sido generado por el estudiante |
| 1 | El mensaje ha sido generado por el docente Director de tesis |
| 2 | El mensaje ha sido generado por el docente informante asignado |
| 3 | El mensaje ha sido generado por el Director del DIP |

Fuente: elaboración propia

A.2.4.3. Tipos de mensajes que pueden ser generados para un plan de titulación de posgrados

En la siguiente tabla se detalla los valores que puede contener el campo “tipoMensaje” de la tabla “MensajesPlandeTesis”.

Tabla 28: Tipos de mensajes que pueden ser generados para un plan de titulación de posgrados

| Valor | Significado |
|-------|---|
| 0 | El mensaje corresponde al envío del plan de titulación por parte del estudiante, para que el mismo sea revisado por su director quien a su vez podrá aprobar el plan o devolverlo al estudiante |
| 1 | Devolución por parte del director de tesis al estudiante para que aplique correcciones al plan previamente enviado |
| 2 | Aprobación del plan de titulación por parte del director de tesis |
| 3 | Mensaje de asignación del docente informante por parte de la dirección del DIP. Se manejará internamente |
| 4 | Devolución por parte del docente informante asignado para aplicar correcciones al plan de titulación enviado por el estudiante y aprobado por su director |
| 5 | Aprobación por parte del docente informante asignado sin enviar a aplicar correcciones al plan enviado por el estudiante |
| 6 | Devolución del plan de titulación por parte de la dirección del DIP al estudiante para aplicación de correcciones |
| 7 | Aprobación definitiva del plan de titulación por parte del DIP, en cuyo caso cambiará también el estado del plan a "Aprobado definitivamente" (campo estado con valor 5) y el estudiante podrá proceder a enviar sus avances correspondientes al desarrollo mismo de la tesis |

Fuente: elaboración propia

A.2.5. Tablas auxiliares a la tabla "PlanesdeTesis"

Como se mencionó antes, dada la complejidad y cantidad de información que manejan los planes de titulación de posgrados, se requiere de varias tablas para almacenar dicha información de manera completa y consistente. A continuación se realiza una breve descripción de cada una de ellas.

Tabla 29: Tablas auxiliares a la tabla “PlanesdeTesis”

| Nombre de la Tabla | Descripción | Lista de campos |
|---------------------------------|---|---|
| ObjetivosEspecificosPlandeTesis | Almacena los objetivos específicos de las planes de titulación de posgrados | <ul style="list-style-type: none"> • Id • PlandeTesisId • Objetivo • Orden • Activo |
| MetodologiasPlandeTesis | Permite almacenar una lista de ítems metodológicos perteneciente a un plan en particular | <ul style="list-style-type: none"> • Id • PlandeTesisId • Metodología • Orden • Activo |
| ReferenciasPlandeTesis | Almacena las referencias bibliográficas propias de un plan de titulación | <ul style="list-style-type: none"> • Id • Autor • Título • Editorial • Anio • Url • Activo • PlandeTesisId • Orden |
| PresupuestosPlandeTesis | Almacena los ítems correspondientes a rubros del presupuesto de un plan de titulación | <ul style="list-style-type: none"> • Id • PlandeTesisId • Rubro • Valor • Activo |
| CronogramasPlandeTesis | Permite guardar la información correspondiente a las actividades que se planifica para el desarrollo de un plan de titulación | <ul style="list-style-type: none"> • Id • PlandeTesisId • Actividad • Mes1 |

| | | |
|------------------------------------|---|--|
| | | <ul style="list-style-type: none"> • Mes2 • Mes3 • Mes4 • Mes5 • Mes6 • Mes7 • Mes8 • Mes9 • Mes10 • Mes11 • Mes12 • Orden • Activo |
| DelimitacionFuncionalPregu nta1 | Guarda información sobre las características que se espera cumpla el producto final del proyecto en cuestión | <ul style="list-style-type: none"> • Id • PlandeTesisId • Item • Orden • Activo |
| DelimitaciónFuncionalPregu nta2 | Permite definir algunas de las características que por diversos motivos no estarán presente en el producto final del proyecto de titulación | <ul style="list-style-type: none"> • Id • PlandeTesisId • Item • Orden • Activo |

Fuente: elaboración propia

A.2.6. Tabla TesisPosgrados

Esta tabla contiene información general sobre los proyectos de titulación de los maestrantes de las diferentes carreras de posgrado de la PUCESA, una vez que su plan de titulación ha sido aprobado y se encuentran habilitados para enviar avances de su proyecto.

A.2.6.1. Campos de la tabla TesisPosgrados

Tabla 30: Campos de la tabla TesisPosgrados

| Nombre | Tipo de dato | Descripción |
|--------------------|--------------|---|
| TesisPosgradosId | Entero | Clave principal |
| PlanId | Entero | Permite identificar el plan de titulación al que se encuentra vinculado el presente registro |
| FechaHoraDefensa | Date/Time | Almacena la fecha y hora para la defensa del proyecto en cuestión, definidas por el DPI |
| Estado | Entero | Indica el estado actual en el que se encuentra el trámite de aprobación de un proyecto de titulación dado |
| FechaAprobacionDip | Date/Time | Registra la fecha en la que el proyecto fue definitivamente aprobado por la dirección del DIP |

Fuente: elaboración propia

A.2.6.2. Posibles estados de un registro de la tabla “TesisPosgrados”

Es de vital importancia saber el estado exacto en que se encuentra un determinado proyecto de titulación, ya que de ello dependen muchos factores del comportamiento de varias de las páginas dentro de la aplicación. A continuación, se detallan dichos estados.

Tabla 31: Posibles estados de un registro de la tabla “TesisPosgrados”

| Estado | Descripción |
|--------|---|
| 0 | Estado inicial que indica que el registro apenas ha sido creado automáticamente por la aplicación |
| 1 | El docente director del proyecto ha aprobado el trabajo final enviado por el maestrante |
| 2 | La dirección del DIP ha aprobado el trabajo final enviado por el maestrante y previamente aprobado por su director de proyecto |
| 3 | El DIP ha asignado docentes lectores para el proyecto en cuestión |
| 4 | El proyecto final enviado por el maestrante ha sido definitivamente aprobado por ambos docentes lectores |
| 5 | La dirección del DIP ha asignado fecha de defensa para el proyecto en cuestión |
| 6 | La dirección del DIP ha asignado definitivamente los miembros para el tribunal calificador de un determinado proyecto de titulación |
| 7 | El proyecto final ha sido definitivamente aprobado y calificado por los miembros del tribunal |

Fuente: elaboración propia

A.2.7. Tabla AvancesTesis

Esta tabla permite guardar información sobre los diferentes avances que los maestrantes envían a sus proyectos de titulación, ya sean envíos originales o correcciones solicitadas por el docente director del proyecto o por docentes lectores.

A.2.7.1. Campos de la tabla AvancesTesis

Tabla 32: Campos de la tabla “AvancesTesis”

| Nombre | Tipo de dato | Descripción |
|-----------------|---------------------|---|
| AvanceTesisId | Entero | Clave principal |
| TesisId | Entero | Permite identificar el registro de la tabla TesisPosgrados con el que se relaciona el presente registro |
| Estado | Entero | Permite identificar el estado actual en el que se encuentra un avance de proyecto de titulación. Puede contener 3 valores: 0, que significa que el maestrante apenas ha enviado el trabajo, 1 que significa que está en revisión por parte de su docente director, y 2 que significa que el avance ha sido aprobado por el mismo |
| Porcentaje | Entero | Almacena el porcentaje al cual corresponde el presente avance de proyecto |
| Iteración | Entero | Permite identificar el número de veces que el maestrante ha enviado correcciones a un determinado nivel de avances de su proyecto |
| FechaEnvio | Date/Time | Almacena la fecha en la que el maestrante envió el respectivo avance |
| FechaAprobacion | Date/Time | Almacena la fecha en la que el |

| | | |
|--|--|-------------------------------------|
| | | avance fue definitivamente aprobado |
|--|--|-------------------------------------|

Fuente: elaboración propia

A.2.7.2. Notas sobre la tabla AvancesTesis

La tabla AvancesTesis guarda un solo registro por cada porcentaje de avance permitido para que el estudiante pueda enviar sus avances de tesis como archivos adjuntos. Para el caso de que al estudiante se le envíen a realizar correcciones y tenga que enviar nuevos archivos, existe el campo iteración, que va registrando el número de veces que el estudiante ha enviado archivos (a su director) para el porcentaje en cuestión. Todos los archivos de todas las iteraciones de cada porcentaje enviados por un estudiante se guardan en el servidor en una carpeta con ruta:

~/Avances/TesisId/AvanceId/Iteracion

El campo estado de esta tabla puede contener 3 posibles valores cuyos significados son: 0 significa que se está esperando que el estudiante cargue archivos y envíe el avance, 1 significa que el avance ya ha sido enviado y está a la espera de ser revisado por el docente director del proyecto, y 2 significa que el avance para ese porcentaje ha sido finalmente aprobado por el director del proyecto. En caso de que el director solicite hacer correcciones, el campo estado vuelve a un valor de 0, y el campo iteracion aumenta en 1. Finalmente, el campo fechaEnvio corresponde a la última fecha en que el estudiante ha enviado avances para ese porcentaje, mientras que el campo fechaAprobacion indica la fecha en la que finalmente fue aprobado el avance para el porcentaje correspondiente por parte del director del proyecto.

A.2.8. Tabla “MensajesAvancesTesis”

La función de esta tabla, al igual que en los casos del tema y del plan de titulación, es almacenar los mensajes que se generan por cualquiera de los usuarios de la aplicación a lo largo de todo el proceso de revisión y aprobación del proyecto de titulación de un maestrante, desde el momento en que su plan es aprobado y comienza a enviar avances hasta cuando culmina con la calificación de la defensa de su proyecto por parte de los miembros del tribunal.

A.2.8.1. Campos de la tabla “MensajesAvancesTesis”

A continuación se describe los cambios de la mencionada tabla.

Tabla 33: Campos de la tabla “MensajesAvancesTesis”

| Nombre | Tipo de dato | Descripción |
|-----------------------|--------------|---|
| MensajeAvancesTesisId | Entero | Clave principal |
| Mensaje | Varchar(MAX) | Contiene el texto propiamente del mensaje emitido por cualquiera de los usuarios. |
| TesisId | Entero | Permite identificar el registro de la tabla TesisPosgrados con el que se relaciona el de esta tabla |
| Porcentaje | Entero | Almacena el porcentaje al cual corresponde el presente mensaje. Esta información es necesaria al momento de desplegar el historial de mensajes |
| FechaHora | Date/Time | Almacena la fecha y hora exactas en que el mensaje fue emitido |
| AutorId | Entero | Permite identificar al usuario que ha emitido el mensaje |
| TipoAutor | Entero | Indica el tipo de usuario que ha enviado el mensaje. Este campo puede contener cinco posibles valores: 0 que significa que el autor es el maestrante, 1 que corresponde |

| | | |
|--|--|--|
| | | al docente director, 2 al director del DIP, 3 a un docente lector y 4 a un miembro del tribunal calificador. |
|--|--|--|

Fuente: elaboración propia

A.2.9. Tabla CorreccionesLectoresTesis

Esta tabla se encarga de guardar la información pertinente a las correcciones solicitadas por los docentes lectores asignados por la dirección del DIP a una tesis en particular.

A.2.9.1. Campos de la tabla CorreccionesLectoresTesis

A continuación se describe los campos correspondientes a esta tabla.

Tabla 34: Campos de la tabla “CorreccionesLectoresTesis”

| Nombre | Tipo de dato | Descripción |
|-----------------------------|--------------|--|
| CorreccionesLectoresTesisId | Entero | Clave principal |
| DocenteID | Entero | Permite identificar el docente lector al que corresponde el registro en cuestión de esta tabla |
| AvanceId | Entero | Permite identificar el registro de la tabla “AvancesTesis” con el que se relaciona el registro de esta tabla |
| Estado | Entero | Almacena un valor que |

| | | |
|-----------------|-----------|---|
| | | <p>permite identificar el estado en el que se encuentra este registro.</p> <p>Los posibles valores son 0, que indica que la corrección aún no ha sido enviada por el maestrante, 1 que significa que ya está a cargo de su docente director de proyecto, y 2 significa que la corrección ha sido definitivamente aprobado por un docente lector</p> |
| FechaHoraEnvio | Date/Time | Almacena la fecha y hora exactas en la que se creó el registro en cuestión |
| FechaAprobacion | Date/Time | Fecha en la que la corrección fue definitivamente aprobada por un docente lector |
| Nota | Decimal | Registra la calificación con la que el docente lector aprueba una proyecto de titulación |

Fuente: elaboración propia

A.2.9.2. Notas sobre la tabla CorreccionesLectoresTesis

Esta tabla guarda dos registros por cada tesis existente, uno por cada docente lector asignado a la misma. El campo estado de la tabla puede tener 3 posibles valores: 0 que indica que se está esperando que el estudiante envíe las correcciones solicitadas por el docente lector correspondiente, 1 que indica que el docente lector correspondiente debe revisar los archivos enviados previamente por el estudiante para bien aprobar el proyecto o bien enviar a hacer correcciones, y 2 que indica que el docente lector correspondiente ha aprobado finalmente el proyecto.

Por otro lado, el campo avanceId indica el avance (registro de la tabla AvancesTesis) al que corresponde la corrección en cuestión, y va cambiando según le van enviando a hacer nuevas correcciones al estudiante, esto debido a que cada corrección que envíe el estudiante a docentes lectores generará un nuevo registro en la tabla AvancesTesis con un valor de 100 en el campo porcentaje. El campo fechaHoraEnvio guarda la última fecha y hora en la que el estudiante ha enviado nuevos archivos, y finalmente el campo fechaAprobacion indica la fecha en la que el docente lector correspondiente ha aprobado finalmente el Proyecto.

Cabe mencionar que, el estudiante puede enviar archivos a cada docente lector por separado, pero si tiene correcciones pendientes para ambos lectores, el estudiante podrá enviar los mismos archivos a ambos lectores de una sola vez.

A.2.10. Tabla “MiembrosTribunalPosgrados”

Esta tabla contiene información que le permite al director del DIP asignar definitivamente miembros del tribunal calificador para un proyecto de titulación. Cuando el director del DIP lo hace, en esta tabla se generan 3 registros por proyecto, uno por cada miembro del tribunal designado. Una vez culminada la defensa del proyecto, en esta tabla se almacenan también las diferentes notas que dichos miembros de tribunal asignan a la defensa del proyecto.

A.2.10.1. Campos de la tabla “MiembrosTribunalPosgrados”

Tabla 35: Campos de la tabla “MiembrosTribunalPosgrados”

| Nombre | Tipo de dato | Descripción |
|-----------------------------|---------------------|--|
| MiembrosTribunalPosgradosId | Entero | Clave principal |
| MiembroId | Entero | Permite identificar el docente miembro del tribunal al que corresponde el registro en cuestión de esta tabla |
| Orden | Entero | Indica qué orden de miembro de tribunal es cada uno de ellos |

| | | |
|---|---------|---|
| TesisId | Entero | Permite identificar el registro de la tabla "TesisPosgrados" con el que se relaciona el registro de esta tabla |
| Nota11, Nota12, Nota13, Nota21, Nota22, Nota31, Nota32, Nota33, Nota41, Nota42 | Decimal | Estos 10 campos permiten a cada miembro del tribunal calificador, registrar las10 calificaciones que deben emitir cada uno de ellos, correspondientes a los diferentes aspectos de una defensa de proyecto de titulación de posgrados |

Fuente: elaboración propia

Apéndice B

Descripción de Clases

En este apéndice se procede a describir las clases que se ha creado dentro de la aplicación, necesarias para su correcto funcionamiento.

B.1. Clase “UsuarioGraduacionAtencion.cs”

El objetivo de esta clase es guardar los datos del usuario que ingresa a la aplicación mediante su usuario y contraseña, guardándolo en una variable de sesión, sea este maestrante, docente (director o informante de tesis) o el(la) director(a) del DIP.

B.1.1. Miembros de la Clase “UsuarioGraduacionAtencion.cs”

Tabla 36: Miembros de la clase “UsuarioGraduacionAtencion.cs”

| Miembro | Tipo de Dato | Significado | Comentarios |
|-----------|--------------|----------------------------------|--|
| id | Entero | Id del usuario | |
| apellidos | Cadena | Apellidos del usuario | |
| nombres | Cadena | Nombres del usuario | |
| cedula | Cadena | Cédula del usuario | |
| tipo | UInt16 | Tipo de usuario que ha ingresado | 0 indica que es un maestrante, 1 que es un docente y 2 que el el(la) director(a) del DIP |

Fuente: elaboración propia

B.2. Clase “General.cs”

Se trata de una clase abstracta que proporciona varios métodos útiles que satisfacen ciertas necesidades que se presentan a lo largo de toda la aplicación. Todos ellos son métodos estáticos, lo que significa que se pueden usar directamente, sin necesidad previa de instanciar objetos.

A continuación se describe los métodos de la clase General.cs:

Tabla 37: Métodos de la clase “General.cs”

| Método | Descripción | Parámetros |
|-----------------------|--|--|
| CerrarSesion | Permite cerrar la sesión de la aplicación desde cualquier página que se la esté usando, sin importar el tipo de usuario. Es necesario enviar la página como parámetro | Pagina(tipo Page) |
| VerificarCerrarSesion | Permite verificar si se ha hecho click en el enlace “Cerrar Sesión”, disponible en cualquier página de la aplicación, de cualquier tipo de usuario. También es necesario enviar la página en cuestión como parámetro | Pagina(tipo Page) |
| EnviarEmail | Permite enviar un email a cualquier dirección de email del mundo. Usa una cuenta de correo electrónico de Gmail creada específicamente para este fin. Dentro de la aplicación se usa para enviar notificaciones sobre las acciones que tienen lugar, tanto a maestrantes como a docentes y | Email (tipo cadena), asunto (tipo cadena) y cuerpo (tipo cadena) |

| | | |
|-------------------|---|-----------------------------|
| | director del DIP | |
| CrearFirmaDigital | Permite generar una cadena de texto que corresponde a una firma digital, a partir de un texto original. Se usa principalmente para la firma de documentos generados por la aplicación | TextoOriginal (tipo cadena) |

Fuente: elaboración propia

B.3. Clases de tipo “Managers”

Las clases de tipo “Managers” se crean con la finalidad de realizar las operaciones básicas de transacción de datos (CRUD, Create, Read, Update, Delete) con las tablas de la base de datos de la aplicación, lo cual provee un medio simple y práctico de manipular la información almacenada, pudiendo recuperarla, modificarla o eliminarla a través de simples llamadas a métodos y mediante el envío de parámetros.

De manera general, se crea una clase de tipo “Manager” por cada tabla de la base de datos, dentro de la cual se codifica los métodos que se requiere para manipular la información. Sin embargo, este tipo de clases pueden también servir para recuperar información de otras fuentes distintas a las tablas de bases de datos, como por ejemplo de webservices. Dentro de la presente aplicación, se ha procedido a crear las siguientes clases de tipo “Manager”:

- AvancesdeTesisManager.cs
- CarrerasPosgradosManager.cs
- CorreccionesLectoresTesisManager.cs
- CorreccionesTemasdeTesisManager.cs
- CronogramasPlandeTesisManager.cs
- DelimitacionFuncionalPregunta1PlandeTesisManager.cs
- DelimitacionFuncionalPregunta2PlandeTesisManager.cs
- DocentesManager.cs

- EstudiantesManger.cs
- LineasdeInvestigacionManager.cs
- LineasdeInvestigacionPlanesManager.cs
- MensajesAvancesTesisManager.cs
- MensajesPlandeTesisManager.cs
- MansajesTemasdeTesisManager.cs
- MetodologiasPlandeTesisManager.cs
- MiembrosTribunalPosgradosManager.cs
- ObjetivosEspecificosPlandeTesisManager.cs
- PlandeTesisManager.cs
- PresupuestosPlandeTesisManager.cs
- ReferenciasPlandeTesisManager.cs
- SolicitudesAprobacionTemasdeTesisManager.cs
- SolicitudesDireccionTesisManager.cs
- TemadeTesisManager.cs
- TesisPosgradosManager.cs

Como se observa, es recomendable que los nombres de este tipo de clases sean lo más descriptivos posible, a fin de que sean fáciles de identificar y utilizar.

Apéndice C

Descripción de Webservices

Como se ha mencionado a lo largo de todo este informe, pese a que la presente aplicación ha sido desarrollada bajo una óptica general, se ha tomado como caso de estudio el DIP de la PUCESA, y se decidió utilizar la información con la que dicha institución ya cuenta, haciendo esta aplicación compatible con “Academics”. Para lograrlo fue necesaria la activación de múltiples webservices, que son métodos que proporcionan diferentes grupos de información según sea necesaria.

En este apéndice se describe los webservices activados por el personal del departamento de Informática de la PUCESA, con la finalidad de posibilitar el desarrollo de la presente aplicación.

C.1. GetCarrerasPosgrados

Este webservice devuelve información referente a las carreras de posgrados vigentes actualmente en la PUCESA. No requiere el envío de ningún parámetro para su funcionamiento. Los datos que devuelve son:

- CarraraId
- Descripción
- Siglas
- EscuelaId
- TipoCarrera
- Activa

C.2. GetDatosDocentes

Devuelve información sobre los docentes de la PUCESA, de cualquier carrera que sean, tanto activos como inactivos. Tampoco requiere el envío de parámetros para su funcionamiento. Los datos que devuelve son:

- DocenteId
- Identificacion
- Nombres
- Apellidos
- Activo

C.3. GetDatosEstudiantes

Devuelve un conjunto bastante amplio de información sobre un estudiante específico de la institución. Requiere el Id interno del estudiante como parámetro para poder trabajar. La información que devuelve es la siguiente:

- EstudianteId
- Identificacion
- TipoIdentificacion
- Nombres
- Apellidos
- Genero
- EstadoCivil
- FechaNacimiento
- Direccion
- Telefono
- Celular
- FechaCreacion
- BeneficiarioSeguro
- Nacionalidad
- Colegioid
- Tituloid
- Parroquiald
- EtniaId
- CreditoIECE
- EmailInstitucional

- ParleloId

C.4. GetEstudiantesdePosgrado

Este webservice devuelve un conjunto básico de información sobre los estudiantes de posgrados registrados en la institución. No requiere el envío de ningún parámetro, y los datos que devuelve son:

- CarreraId
- EstudianteId
- Cedula
- Estudiante
- Matricula

C.5. GetTodoslosEstudiantes

Devuelve un set bastante completo de información sobre todos los estudiantes que cursan o han cursado alguna vez en la institución. No requiere el envío de parámetros, pero devuelve un conjunto de información bastante pesado, por lo que hay que manejarlo con cuidado. Los datos que devuelve incluyen:

- EstudianteId
- Identificacion
- Tipoidentificacion
- Nombres
- Apellidos
- Genero
- EstadoCivil
- FechadeNacimiento
- Direccion
- Telefono
- Celular
- FechadeCreacion

C.6. EspecificarPerfilUsuario

Este webservice ayuda a validar el acceso a los usuarios, basándose en la información preexistente en Academics. Requiere el envío de 2 parámetros, el usuario y la contraseña, y lo único que devuelve como resultado es una cadena de texto que especifica el perfil del usuario siempre que haya proporcionado los datos correctamente. Los resultados pueden ser “ESTUDIANTE”, “DOCENTE”, “DIRECTOR” o “USUARIO NO ENCONTRADO”.

Apéndice D

Encuestas de satisfacción aplicadas al personal del DIP

D.1. Encuestas reales respondidas personalmente por el personal administrativo del DIP

En las siguientes páginas se observa las encuestas de satisfacción aplicadas al personal administrativo del DIP, tanto a la Lcda. Doris Constanza Rosales Vacas como al Mtr. Diego Armando Jiménez Bósquez, llenadas manual y personalmente por los mismos luego de haber probado detalladamente la funcionalidad de la aplicación web producto del presente trabajo de investigación, la misma que ha sido instalada y configurada en los equipos informáticos de la PUCESA garantizándose así su total compatibilidad.

**CUESTIONARIO DE SATISFACCIÓN PARA LA EVALUACIÓN DE LA APLICACIÓN WEB
"GRADUACIONPOSGRADOS", CORRESPONDIENTE AL TEMA DE PROYECTO DE
INVESTIGACIÓN Y DESARROLLO DE GRADO TITULADO "DESARROLLO DE UNA
APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE PROCESOS DE GRADUACIÓN DE
UNA UNIDAD DE POSGRADOS"**

Aplicado a: Lcda. Doris Constanza Rosales Vacas (usuaria de la aplicación)

Preguntas:

1. ¿Permite la aplicación registrar detalladamente todos los pasos del proceso de graduación a los que se someten los maestrantes de la PUCESA?

Sí NO

2. ¿Permite la aplicación verificar el cumplimiento de los diferentes requisitos correspondientes a cada paso del proceso de graduación de posgrados de la PUCESA?

Sí NO

3. ¿Detecta la aplicación cuando el usuario ingresa datos erróneos o realiza operaciones inconsistentes, y despliega mensajes de error adecuados y comprensibles para el usuario?

Sí NO

4. ¿Refleja la aplicación correctamente la información correspondiente a maestrantes y docentes existente en las bases de datos de la PUCESA?

Sí NO

5. ¿Refleja la aplicación correctamente la secuencia de pasos correspondiente al proceso de graduación de posgrados de la PUCESA?

Sí NO

6. ¿Despliega la aplicación un historial de comentarios con fechas y horas exactas sobre los diferentes pasos que tienen lugar dentro de cada proceso de graduación de posgrados de la PUCESA?

Sí NO

7. ¿La aplicación es intuitiva, amigable y fácil de usar?

Sí NO

8. ¿Genera la aplicación la información estadística requerida por la OIP de manera clara y entendible?

Sí NO

9. ¿Permite la aplicación exportar la información estadística generada de modo que sea fácil portarla e imprimirla?

Sí NO

10. ¿Permite la aplicación generar la información estadística requerida tanto a nivel general como por cada una de las diferentes carreras de posgrado de la PUCESA?

Sí NO

11. En general, ¿Considera usted que la aplicación cumple satisfactoriamente los objetivos para los que fue creada y será de la utilidad prevista para la OIP?

Sí NO


Lcda. Doris Constanza Rosales Vacas

Secretaria de la OIP / Usuaría de la Aplicación

**CUESTIONARIO DE SATISFACCIÓN PARA LA EVALUACIÓN DE LA APLICACIÓN WEB
"GRADUACIONPOSGRADOS", CORRESPONDIENTE AL TEMA DE PROYECTO DE
INVESTIGACIÓN Y DESARROLLO DE GRADO TITULADO "DESARROLLO DE UNA
APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE PROCESOS DE GRADUACIÓN DE
UNA UNIDAD DE POSGRADOS"**

Aplicado a: Prof. Diego Jiménez Bósquez (usuaria de la aplicación)

Preguntas:

1. ¿Permite la aplicación registrar detalladamente todos los pasos del proceso de graduación a los que se someten los maestrantes de la PUCESA?

Sí NO

2. ¿Permite la aplicación verificar el cumplimiento de los diferentes requisitos correspondientes a cada paso del proceso de graduación de posgrados de la PUCESA?

Sí NO

3. ¿Detecta la aplicación cuando el usuario ingresa datos erróneos o realiza operaciones inconsistentes, y despliega mensajes de error adecuados y comprensibles para el usuario?

Sí NO

4. ¿Refleja la aplicación correctamente la información correspondiente a maestrantes y docentes existente en las bases de datos de la PUCESA?

Sí NO

5. ¿Refleja la aplicación correctamente la secuencia de pasos correspondiente al proceso de graduación de posgrados de la PUCESA?

Sí NO

6. ¿Despliega la aplicación un historial de comentarios con fechas y horas exactas sobre los diferentes pasos que tienen lugar dentro de cada proceso de graduación de posgrados de la PUCESA?

Sí NO

7. ¿La aplicación es intuitiva, amigable y fácil de usar?

Sí NO

8. ¿Genera la aplicación la información estadística requerida por el DIP de manera clara y entendible?

Sí NO

9. ¿Permite la aplicación exportar la información estadística generada de modo que sea fácil portarla e imprimirla?


Sí NO

10. ¿Permite la aplicación generar la información estadística requerida tanto a nivel general como por cada una de las diferentes carreras de posgrado de la PUCESA?

Sí NO

11. En general, ¿Considera usted que la aplicación cumple satisfactoriamente los objetivos para los que fue creada y será de la utilidad prevista para el DIP?

Sí NO



Master Diego Jiménez Bosquez

Coordinador del DIP

Referencias

- [1] Adell, J., & Bellver, C. (2010). *La internet como telaraña: el World-Wide Web*. Métodos de información, 2(3), 25-32.
- [2] Athento, *Gestión Documental Inteligente*, <http://www.athento.com/gestion-documental-inteligente/>
- [3] Avison, D., & Fitzgerald, G. (2003). *Information systems development: methodologies, techniques and tools*. McGraw Hill.
- [4] Beck, K. (2000). *Extreme programming explained: embrace change*. addison-wesley professional.
- [5] Beck, K., & Fowler, M. (2001). *Planning extreme programming*. Addison-Wesley Professional.
- [6] Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.
- [7] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). *Manifesto for agile software development*.
- [8] Berners-Lee, T., Fischetti, M., & Foreword By-Dertouzos, M. L. (2000). *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. HarperInformation.
- [9] Berners-Lee, T., & Connolly, D. (1995). *Hypertext markup language-2.0* (No. RFC 1866).
- [10] Bush, V. (1945). *Science: The endless frontier*. Transactions of the Kansas Academy of Science (1903-), 48(3), 231-264.
- [11] Canós, J. H., Letelier, P., & Penadés, M. C. (2003). *Metodologías ágiles en el desarrollo de software*. Metodologías Ágiles en el Desarrollo de Software, 1.
- [12] Castillo, Fernández, Rea & Tapia, E., *Desarrollo del Sistema de Gestión Académica de Postgrados de la Universidad De Cuenca (SGAP)*, 2012
- [13] Correlibre.org, *Herramienta de gestión documental y de procesos OrfeoGPL*, 2013, <http://orfeogpl.org/ata/>
- [14] Doknos, *Quipux para su Gestión Documental*, 2010, <http://www.doknos.com/en/node/70>
- [15] Fernández Valdés, M. D. L. M., & Ponjuán Dante, G. (2008). *Análisis conceptual de las principales interacciones entre la gestión de información, la gestión documental y la gestión del conocimiento*. Acimed, 18(1), 0-0.
- [16] Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2010). *Criterios de selección de metodologías de desarrollo de software*. Industrial Data, 13(2), 070-074.
- [17] Jame, F., David, J., Maza, P., & Patricio, R. (2015). *Análisis diseño e implementación de un sistema Web que permita automatizar el procedimiento de titulación en posgrado de la Universidad de las Fuerzas Armadas ESPE. Mediante la utilización de la plataforma Java enterprise Edition JEE6 Web aplicando la metodología SCRUM* (Doctoral dissertation, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática.).
- [18] Malkin, G. S. (1996). *Internet Users' Glossary*.

- [19] Mora, S. L. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario.
- [20] Morillo Cadena, D. I. (2015). *Implementación de un sistema de digitalización y gestión documental DMS para la Empresa Textil Vicunha SA-Ecuador*.
- [21] Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., ... & Steenkiste, P. (2014, December). *The cost of the s in https*. In Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies (pp. 133-140). ACM.
- [22] Nelson, T. H. (1965, August). *Complex information processing: a file structure for the complex, the changing and the indeterminate*. In Proceedings of the 1965 20th national conference (pp. 84-100). ACM.
- [23] Peck, G. K. (2004). *Crystal Reports 10: The Complete Reference*. McGraw-Hill, Inc..
- [24] Pérez, M. (2011). *SQL Server 2008 R2: motor de base de datos y administración*. RC Libros.
- [25] Pressman, R. S., & Troya, J. M. (2010). *Ingeniería del software (No. 001.64 P74s.)*. McGraw Hill.
- [26] PUCESA, *Manual de usuario del Sistema Academics*, 2012
- [27] Reisin, S. (2009). *Percepción de factores que favorecen u obstaculizan la graduación en posgrados de Ciencias Sociales. Un estudio de caso* (Doctoral dissertation, Tesis de Maestría en Educación, Universidad de San Andrés, Buenos Aires, Argentina).
- [28] Reyes Vaca, J. G. (2014). *Sistema Web para la Gestión de los Procesos Críticos de la Dirección de Posgrado de la UTA*.
- [29] Sánchez Dromundo, R. A. (2007). *La teoría de los campos de Bourdieu, como esquema teórico de análisis del proceso de graduación en posgrado*. Revista electrónica de investigación educativa, 9(1), 1-21.
- [30] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Fundamentos de bases de datos* (No. 04; QA76. 9. D3, S5y 2002.). McGraw-Hill.
- [31] Sommerville, I., *Ingeniería del software, séptima edición*, PEARSON Educación S.A., Madrid, 2005
- [32] Soria, R., *Desarrollo de una Aplicación Web para la Optimización de Procesos de Atención al Usuario de una Unidad de Postgrados*, PUCESA, 2014
- [33] Thai, T., & Lam, H. (2003). *NET framework essentials*. " O'Reilly Media, Inc."
- [34] Universidad Central de Venezuela, *Manual de usuario del Sistema de Gestión de los Procesos Administrativos de los Trabajos Finales de Postgrado de la UCV*, 2009
- [35] Velásquez Altamirano, G. A., Vargas, Q., & Maribel, V. (2012). *Propuesta de automatización del proceso de graduación de la Facultad de Ciencias Administrativas de la Escuela Politécnica Nacional utilizando la plataforma Auraportal* (Doctoral dissertation, Quito, 2012.).