

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CARACAS – VENEZUELA

**Desarrollo de Aplicación Web para la gestión
de actividades de Seminarios y de
Trabajos Especiales de Grado
de la Escuela de Computación**



Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres

Ignacio de Jesús Ortiz Diago
C.I. 16.004.622

Humberto David De Sousa Ramírez
C.I. 16.343.811

Para optar al título de Licenciado en Computación

Tutoras
Prof. Jossie Zambrano
Prof. Eleonora Acosta

Caracas, Octubre de 2013

Acta del Veredicto

Quienes suscriben, miembros del jurado designado por la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado de los bachilleres **IGNACIO DE JESÚS ORTIZ DIAGO** y **HUMBERTO DAVID DE SOUSA RAMÍREZ**, titulares de las cédulas de identidad No. 16.004.622 y 16.343.811 respectivamente, con el título: **“Desarrollo de Aplicación Web para la gestión de actividades de Seminarios y de Trabajos Especiales de Grado de la Escuela de Computación”**, a los fines de cumplir con el requisito legal para optar al grado de Licenciado en Computación, dejan constancia de lo siguiente:

Una vez leído el presente trabajo por cada uno de los miembros del jurado, se fijó el día 21 de Octubre de 2013, como fecha para la defensa pública mediante una presentación oral. Dicha presentación fue realizada en la fecha acordada en la Sala Uno de la Escuela de Computación, luego de lo cual, los bachilleres respondieron satisfactoriamente a las preguntas que le fueron formuladas por el jurado, todo ello conforme dispuesto a la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela, por lo cual el jurado decidió APROBARLO.

En fe de lo cual se levanta la presente acta: _____.

Prof. Jossie Zambrano (Tutor)

Prof. Eleonora Acosta (Tutor)

Prof. Eugenio Scalise (Jurado)

Prof. Sergio Rivas (Jurado)

Resumen

El Plan de Estudio de la Licenciatura de Computación de la Universidad Central de Venezuela incluye, entre otros, un componente de práctica profesional que está constituido por las asignaturas de Seminario y Trabajo especial de Grado. Este Trabajo Especial de Grado tiene como propósito desarrollar una aplicación Web, que permita gestionar las actividades que se llevan a cabo en las asignaturas Seminario y Trabajo Especial de Grado, fomentando la comunicación, simplicidad y retroalimentación entre estudiantes y tutores. Para lograr este propósito se llevó a cabo un análisis detallado de las tareas y actividades que se realizan al cursar estas asignaturas, tomando en cuenta la perspectiva tanto del profesor tutor como del estudiante. Adicionalmente, se realizó un análisis de los métodos de desarrollo de software más utilizados en este tipo de proyectos para construir la aplicación de manera que se adapte en la medida de lo posible a los mismos. En el desarrollo de esta investigación se implementó una aplicación utilizando el método de desarrollo ágil XP, el cual es iterativo y basado en un diseño de aplicación emergente que permite la construcción progresiva de software así como su escalabilidad a medida de que son detectados nuevos requerimientos. Como aporte de esta investigación se obtiene una herramienta automatizada para la gestión de las actividades y tareas de los Seminarios y Trabajos Especiales de Grado. Con esta nueva herramienta se espera que los profesores tutores puedan brindar una asesoría más oportuna y específica a cada estudiante a lo largo del desarrollo de sus proyectos. Asimismo, se espera una mayor apertura de oferta de temas por parte de los profesores. Por último, se llevó a cabo un conjunto de pruebas funcionales para validar el buen funcionamiento de la aplicación en conjunto con algunos profesores y estudiantes de la Escuela de Computación.

Palabras Clave: seminarios, tesis, modelos de desarrollo ágiles, aplicación Web.

Tabla de Contenidos

ACTA DEL VEREDICTO	2
RESUMEN	3
INTRODUCCIÓN	8
DESCRIPCIÓN DEL PROBLEMA	9
SITUACIÓN ACTUAL	9
SOLUCIÓN PROPUESTA.....	11
OBJETIVOS.....	12
ESTRUCTURA DEL DOCUMENTO	13
CAPÍTULO I. TECNOLOGÍAS WEB	14
1.1 HTML5	14
1.2 HAML.....	16
1.3 WIREFRAMING	18
1.4 CSS3.....	19
1.5 SASS: SCSS	19
1.6 JQUERY	21
1.7 JQUERY-UI	22
1.8 COFFEESCRIPT	23
1.9 RUBY	24
1.10 RUBY ON RAILS.....	26
1.11 PATRÓN DE DISEÑO ARQUITECTÓNICO MODELO VISTA CONTROLADOR (MVC)	28
1.12 MYSQL	29
1.13 GIT	33
1.14 PLATAFORMA	36
CAPÍTULO II. MÉTODOS DE DESARROLLO DE SOFTWARE	37
2.1 MÉTODOS DE DESARROLLO ÁGILES.....	37
2.1.1 <i>Valores de los Métodos Ágiles</i>	38
2.1.2 <i>Método por Prototipos</i>	40
2.1.3 <i>Método RAD</i>	40
2.1.4 <i>Método de Programación Extrema (XP)</i>	41
2.1.5 <i>Método SCRUM</i>	46
2.2 VENTAJAS Y DESVENTAJAS DE LOS MÉTODOS ÁGILES.....	51
CAPÍTULO III. DESARROLLO DE LA APLICACIÓN WEB	53
3.1 ANÁLISIS DE MÉTODOS ÁGILES DE DESARROLLO.....	53
3.2 PRÁCTICAS.....	54
3.3 PLANIFICACIÓN DE CICLO TRIMESTRAL.....	56
3.3.1 <i>Cartelera de Ofertas de Temas</i>	57
3.3.2 <i>Trabajo de Seminario</i>	57
3.3.3 <i>Trabajo Especial de Grado</i>	58
3.4 ITERACIONES	60
3.4.1 <i>Iteración 1 - (16-01-2013 al 31-01-2013)</i>	60

3.4.2 Iteración 2 - (01-02-2013 al 15-02-2013)	65
3.4.3 Iteración 3 - (15-02-2013 al 28-02-2013)	70
3.4.4 Iteración 4 - (01-03-2013 al 15-03-2013)	77
3.4.5 Iteración 5 - (16-03-2013 al 30-03-2013)	82
3.4.6 Iteración 6 - (01-04-2013 al 10-04-2013)	90
3.5 PRUEBAS DE ACEPTACIÓN	97
CONCLUSIONES	98
RECOMENDACIONES	100
REFERENCIAS BIBLIOGRÁFICAS	101

Tabla de Ilustraciones

FIGURA 1.1 – LA SEMÁNTICA EN HTML5 FACILITA SU LECTURA. (BRITO, 2009)	16
FIGURA 1.2 – EJEMPLO DE HAML VS. HTML (FISHER, 2009)	18
FIGURA 1.3 – COMPARACIÓN ENTRE SCSS Y CSS EN IDE VISUAL STUDIO 2010. (HANSELMAN, 2011)	21
FIGURA 1.4 – COMPARACIÓN DE COFFEESCRIPT Y JAVASCRIPT EN IDE VISUAL STUDIO 2010. (HANSELMAN, 2011)	24
FIGURA 1.5 – REPRESENTACIÓN ESQUEMÁTICA DE LA ARQUITECTURA MVC (HARTL, 2011).....	29
FIGURA 1.6 – VISTA LÓGICA DE LA ARQUITECTURA DE MYSQL (DUBOUIS., 2009).....	33
FIGURA 1.7 – DIAGRAMA DE PLATAFORMA.	36
FIGURA 2.1 – REPRESENTACIÓN DE MÉTODO POR PROTOTIPOS (OSC, 2011)	40
FIGURA 2.2 – REPRESENTACIÓN DEL MÉTODO RAD	41
FIGURA 2.3 – DISTRIBUCIÓN DEL ESPACIO DE TRABAJO EN XP	43
FIGURA 2.4 – MODELO DE ESPACIO INFORMATIVO EN XP (KENT BECK, 2005)	43
FIGURA 2.5 – EJEMPLO DE HISTORIA DE USUARIO (KENT BECK, 2005)	44
TABLA 2.1 – VENTAJAS DE LOS MÉTODOS DE DESARROLLO ÁGILES (ASIF IRSHAD KHAN, 2011)	51
TABLA 2.2 – DESVENTAJAS DE LOS MÉTODOS DE DESARROLLO ÁGILES (ASIF IRSHAD KHAN, 2011)	52
FIGURA 3.1 – GRÁFICO COMPARATIVO DEL USO DE LOS TIPOS DE MÉTODOS DE DESARROLLO EN LA ESCUELA DE COMPUTACIÓN.	54
TABLA 3.1 – FORMATO DE HISTORIA DE USUARIO UTILIZADO EN CADA ITERACIÓN DEL CICLO DE DESARROLLO.	55
FIGURA 3.2 – DIAGRAMA DE MÓDULOS IDENTIFICADOS EN PLANIFICACIÓN TRIMESTRAL.	57
FIGURA 3.3 – MAPA CONCEPTUAL DE LA VISIÓN GLOBAL DEL SISTEMA CONCEBIDA PARA LA PLANIFICACIÓN DEL CICLO TRIMESTRAL.....	59
TABLA 3.2 – PLANIFICACIÓN DE ITERACIÓN 1.....	60
FIGURA 3.4 – WIREFRAME DE AUTENTICACIÓN.....	61
FIGURA 3.5 – WIREFRAME DE HOME – PROFESOR AUTENTICADO.	62
FIGURA 3.6 – WIREFRAME DE LISTADO DE ESTUDIANTES – PROFESOR AUTENTICADO.....	62
FIGURA 3.7 – WIREFRAME DE ITERACIÓN DE DOCUMENTOS – PROFESOR AUTENTICADO.	63
FIGURA 3.8 – WIREFRAME DE PROGRESOS DE DESARROLLO – PROFESOR AUTENTICADO.	63
FIGURA 3.9 – WIREFRAME DE ITERACIÓN DE PRESENTACIONES – PROFESOR AUTENTICADO.	64
TABLA 3.3 – PLANIFICACIÓN DE ITERACIÓN 2.....	66
FIGURA 3.12 – PÁGINA DE INICIO DE ESTUDIANTES.	67
FIGURA 3.13 – PÁGINA DE INICIO DE PROFESORES.....	67
FIGURA 3.14 – FORMULARIO DE CREACIÓN DE PROYECTOS.....	68
FIGURA 3.15 – PANTALLA DE CARTELERA DISPONIBLE PARA ESTUDIANTES Y PROFESORES.	68
TABLA 3.4 – PLANIFICACIÓN DE ITERACIÓN 3.....	71
FIGURA 3.16 – PANTALLA DE SOLICITUD DE TEMA. A TRAVÉS DE ELLA LOS ESTUDIANTES SE POSTULAN PARA UN TEMA.....	72
FIGURA 3.17 – PANTALLA DE ASIGNACIÓN DE TEMAS.	73
FIGURA 3.18 – PANTALLA CORRESPONDIENTE AL LISTADO DE PROYECTOS DE UN PROFESOR.	74
FIGURA 3.19 – DETALLE DE PROYECTO CONSULTADO POR UN PROFESOR.....	74
FIGURA 3.20 – CONFIRMACIÓN DE QUE SE HA INICIADO UN PROYECTO.	75
TABLA 3.5 – PLANIFICACIÓN DE ITERACIÓN 4.....	77
FIGURA 3.21 – PANTALLA DE PROGRESOS DE PROYECTO.	79
FIGURA 3.22 – PANTALLA DE DEFINICIÓN DE ESTRUCTURA DE CONTENIDOS PARA PRESENTACIÓN O DOCUMENTO.	79
FIGURA 3.23 – PANTALLA DE ITERACIONES DE PRESENTACIÓN O DOCUMENTO.	80
TABLA 3.6 – PLANIFICACIÓN DE ITERACIÓN 5.....	82
FIGURA 3.24 – PANTALLA DE NUEVOS COMENTARIOS EN ITERACIONES DE DOCUMENTO O PRESENTACIÓN.	84
FIGURA 3.25 – PANTALLA DE PROGRESOS, EN PARA EXPORTAR UN SEMINARIO Y CONTINUARLO COMO TESIS.	85
FIGURA 3.26 – PANTALLA DE GESTIÓN DE TAREAS DE DESARROLLO	86

FIGURA 3.27 – FORMULARIO PARA AGREGAR TAREAS DE DESARROLLO, ASÍ COMO ARCHIVOS Y COMENTARIOS RELACIONADOS.....	86
TABLA 3.7 – PLANIFICACIÓN DE ITERACIÓN 6.....	90
FIGURA 3.28 – PANTALLA PARA CONVOCAR A REUNIÓN. DISPONIBLE PARA ESTUDIANTES Y PROFESORES.	92
FIGURA 3.29 – REPORTE DE SOLICITUD DE NOMBRAMIENTO DE JURADO DE TEG.....	93
FIGURA 3.30 – REPORTE RESUMEN DEL PROYECTO.	94
FIGURA 3.31 – REPORTE DE OFERTAS DE PROYECTOS DE SEMINARIOS Y TEG.	95

Introducción

El Plan de Estudio de la Licenciatura de Computación de la Universidad Central de Venezuela finaliza con un componente denominado Práctica Profesional constituido por las asignaturas de Seminario y Trabajo Especial de Grado (TEG).

La elaboración de proyectos de Seminario y TEG constituyen un requisito de cierta dificultad para los estudiantes de la Escuela de Computación; además de la carga que estos acarrearán, por los procesos involucrados en su realización, tales como la búsqueda de tutor, aceptación de tema, reuniones con tutor, solicitud de jurados, entre otros.

En la actualidad, los estudiantes que cumplen con los requisitos para inscribir Seminario o TEG, frecuentemente se encuentran ante el problema de conseguir un tutor que les pueda ofrecer un tema y que éste a su vez, sea de su agrado y especialidad. Posteriormente, durante la elaboración de los proyectos los estudiantes se enfrentan con diversas situaciones para obtener de sus tutores la asesoría necesaria para culminar estos proyectos exitosamente y en los tiempos pautados.

Entre los obstáculos que se presentan con mayor frecuencia durante el desarrollo de estos proyectos, están la poca disponibilidad de tiempo con que cuentan los tutores para dar la asesoría adecuada y los estudiantes que se desempeñan profesionalmente, para recibirla.

De igual manera, los profesores se ven abrumados por solicitudes de temas de parte de los estudiantes así como por requerimientos de asesoría para los proyectos que dirigen. Estos requerimientos se efectúan no solo en reuniones presenciales sino muchas veces de manera remota a través de correos electrónicos. Por esta razón, resulta complicado para los profesores tutores dar un seguimiento adecuado a las solicitudes y al progreso individual de los estudiantes, prolongando los tiempos de respuesta.

Una vez identificada la problemática actual en la elaboración de estos proyectos, resulta necesario desarrollar una herramienta que mejore sustancialmente la forma actual de interacción entre tutores y estudiantes. Para ello se toman en cuenta las necesidades tanto de profesores como estudiantes al momento de ejecutar los proyectos en conjunto, para facilitar los procesos y alcanzar los objetivos de los mismos.

Descripción del Problema

A continuación se describe la situación actual de la gestión de los procesos de desarrollo de Seminarios y TEG de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela considerando aquellas cuya producción de software se lleve a cabo con métodos ágiles de desarrollo, así como la solución propuesta. Se presentan los objetivos del trabajo así como su importancia y justificación.

Situación Actual

El Plan de Estudios de la Licenciatura en Computación define como requerimientos finales, la elaboración de un trabajo de Seminario y un posterior TEG, ambos trabajos pautados para ser desarrollados en un plazo de un semestre cada uno.

El Seminario consiste habitualmente, en un trabajo de investigación que sienta las bases teóricas y metodológicas para el TEG. Este, debe ser aprobado a fin de inscribir luego la asignatura de TEG. El Seminario, suele estar conformado por un Marco Conceptual, que describe todos los conceptos referenciados; un Marco Metodológico, que describe los métodos de desarrollo a seguir; una Propuesta de TEG que a su vez contiene una descripción de la situación actual, una propuesta de solución, objetivos generales y específicos, y por último el alcance del sistema a desarrollar.

El TEG, generalmente, es el desarrollo del ya aprobado Seminario. Durante el mismo, se ponen en práctica los métodos, herramientas y técnicas descritas durante el Seminario para alcanzar los objetivos planteados en el mismo, obtener un producto final y concluir sobre los resultados obtenidos. Los TEG, a menudo involucran el desarrollo de aplicaciones. En el contexto de la Escuela de Computación es común la utilización de métodos ágiles de desarrollo de software.

Para poder obtener un tema de TEG, los estudiantes deben consultar personalmente con cada tutor sobre su disponibilidad; cada tutor a su vez, debe llevar una bitácora personal con todos los estudiantes tutorados así como temas disponibles para un momento dado.

Una vez se inicia un TEG que utiliza métodos ágiles, los estudiantes y tutores deben establecer una forma de trabajo. Usualmente estos métodos, basan el éxito de los desarrollos entre otros factores, en una comunicación efectiva; por esta razón, una de las prácticas

habituales entre este tipo de métodos consiste en la planificación de reuniones constantes y periódicas entre todas las partes que están involucradas en el desarrollo. Esto implica que gran parte del éxito que puedan tener estos TEG, dependerá de que los tutores y estudiantes, logren coordinar reuniones de manera periódica para elaborar en conjunto los instrumentos a utilizar.

Actualmente, este tipo de reuniones se llevan a cabo de forma irregular por cuestiones de disponibilidad de tiempo de todas las partes; la mayor parte de las comunicaciones entre tutores y estudiantes se realizan a través de correo electrónico. En el caso de un estudiante, pueden resultar frustrantes los largos tiempos de respuesta de estos correos electrónicos; del mismo modo que para un tutor, dar seguimiento a un TEG resulta una tarea compleja, debido a que tiene numerosos estudiantes a su cargo en distintos temas.

Si bien la realización de estas reuniones de coordinación no son esenciales para el desarrollo del TEG, muchas veces la consecuencia de su omisión, es que grandes cambios son realizados precipitadamente ante el advenimiento de las fechas límites de entrega. Otras veces la descoordinación en la forma de trabajo es tal, que termina siendo causa del fracaso y abandono del TEG.

Además de las reuniones, otra de las prácticas más habituales entre los métodos ágiles de desarrollo es el uso de iteraciones del proceso de desarrollo. Estas iteraciones se planifican entre todo el equipo de trabajo, sin embargo, para el caso específico de los TEG, a menudo se planifican de manera unilateral por parte del estudiante, sin que el tutor pueda conocer el estado real del desarrollo, la iteración o los avances alcanzados.

Una vez finalizado el desarrollo del TEG, los tutores deben realizar las últimas correcciones sobre el documento, la presentación y el producto de software final, para luego acordar con el jurado una posible fecha y hora de presentación. Este proceso de últimos refinamientos del documento y presentación puede demorar más de lo necesario, ya que para fechas de entrega la demanda de trabajo para los tutores de TEG aumenta considerablemente.

Adicionalmente, los procesos de solicitud de asignación de jurados y asignación de fecha de estas presentaciones, ocurren con mayor frecuencia durante un pequeño período de tiempo en cada semestre para todos los profesores y estudiantes de la Escuela, resultando en un proceso lento y engorroso.

Como consecuencia del problema planteado, muchos tutores pierden la noción de cuántos estudiantes dirigen, cuántos presentan su TEG en un semestre, cuál es el progreso de cada proyecto y/o cuál es su disponibilidad para ofertar nuevos temas. Por otra parte, para los estudiantes cada vez es más complicado obtener un tema para un TEG, no solo por la reducida oferta de los mismos sino por la dificultad que acarrea encontrar un tema que sea de su agrado o por no disponer del tiempo necesario para coordinar reuniones, entre otras razones, que terminan ralentizando la finalización de sus estudios.

Solución Propuesta

De acuerdo a lo expuesto anteriormente, surge la iniciativa de crear una aplicación Web que permita gestionar las actividades de Seminarios y Trabajos Especiales de Grado de la Escuela de Computación de la Facultad de Ciencias, que utilicen métodos ágiles de desarrollo, fomentando la comunicación, simplicidad y retroalimentación entre los estudiantes y tutores, como valores fundamentales dentro de los proyectos.

Esta aplicación permitirá a los tutores conocer en todo momento el número de estudiantes dirigidos que posee en proyectos de Seminario y TEG. Asimismo, podrá conocer en detalle los avances realizados por los mismos en cada uno de sus proyectos; asignar listas de tareas y actividades, ver y modificar documentos relacionados con el TEG; medir avances generales en cada tema; conocer el rendimiento global de sus estudiantes; ofertar nuevos temas para Seminarios y TEG; coordinar asignación de jurados, fechas y horas de presentación; asignar tareas específicas a estudiantes. Del mismo modo, los estudiantes podrán obtener una retroalimentación activa y específica de sus consultas; medir sus avances globales dentro del Seminario y TEG, bien sea el caso; evaluar sus actividades realizadas, actividades pendientes y actividades futuras; y planificar adecuadamente las actividades de desarrollo.

Según una investigación realizada por la Cámara Venezolana de Comercio Electrónico sobre las Tendencias Digitales sobre las Estadísticas de Internet en Venezuela 2012, más del 40% de la población del país tiene acceso a Internet, por lo que se puede tomar provecho de ello para construir un sistema que soporte a múltiples usuarios y pueda ser accedido desde cualquier ubicación.

Para lograr el objetivo de este proyecto, se desarrolla una aplicación Web de gestión de las actividades de Seminario y TEG, basada en una arquitectura MVC, que proporcione

funcionalidades para controlar, auditar y disponer de información sobre el estado de los proyectos.

Objetivos

A continuación se presentan los objetivos generales y específicos de este TEG.

Objetivo General

Desarrollar una aplicación Web que permita gestionar las actividades de Seminarios y TEG de la Escuela de Computación de la Facultad de Ciencias, que utilicen métodos ágiles de desarrollo, fomentando la comunicación, simplicidad y retroalimentación entre los estudiantes y tutores, como valores fundamentales dentro de los proyectos.

Objetivos Específicos

Con la finalidad de alcanzar el objetivo general se plantean los siguientes objetivos específicos.

- Comparar los métodos ágiles más utilizados en el desarrollo de software en la Escuela de Computación de la Facultad de Ciencias, para establecer las prácticas e instrumentos comunes que permitan elaborar una herramienta única para la gestión de Seminarios y TEG.
- Determinar el flujo de las actividades académicas vinculadas al desarrollo de Seminarios y TEG para incorporar la gestión de las mismas a la aplicación, en la medida de lo posible.
- Modelar, Diseñar e Implementar una estructura de la base de datos, que permita almacenar los datos concernientes a la gestión de Seminarios y TEG
- Diseñar la IU para la aplicación Web de gestión de proyectos de Seminarios y TEG tomando en cuenta los criterios de usabilidad y los elementos de diseño de experiencias de usuario, así como las recomendaciones de posibles usuarios del sistema.
- Desarrollar una aplicación Web basada en una arquitectura MVC, que permita gestionar los proyectos de Seminario y TEG.

Estructura del Documento

Una vez descrito el problema de investigación, el objetivo general y los objetivos específicos que determinaron el curso de la presente investigación, se definieron cuatro capítulos en este TEG.

En el Capítulo 1 se expone de manera general todo lo relacionado a las tecnologías Web que fueron utilizadas en el desarrollo de esta aplicación; desde la definición de técnicas de elaboración de prototipos, pasando por tecnologías de cliente y servidor, así como bases de datos y manejadores de versiones.

En el Capítulo 2 se profundiza en los métodos de desarrollo de software, para conocer no sólo los beneficios de las prácticas que serán empleadas en este TEG sino también las posibilidades de unificar criterios para permitir que la aplicación que será desarrollada permita el uso de la mayor cantidad de métodos de desarrollo posibles.

En el Capítulo 3 se presentan los resultados obtenidos en la experiencia del desarrollo que dan respuesta a los objetivos planteados. Seguidamente se presenta una serie de planteamientos conclusivos y recomendaciones.

Por último, se presentan las referencias bibliográficas que soportan esta investigación.

Capítulo I. Tecnologías Web

En este capítulo se establecen los conceptos fundamentales referenciados en el desarrollo de este trabajo de investigación, profundizando enfáticamente en aquellos que permitan la comprensión global del proyecto. Dado que el objetivo general de este proyecto se centra sobre el desarrollo de una aplicación Web, para la codificación e implementación de la misma se utilizan las siguientes tecnologías: HTML5, HAML, Wireframing, CSS3, SASS: SCSS, JQuery, JQuery-UI, CoffeeScript, Ruby, Rails, MVC, MySQL, GIT; éstos serán descritos brevemente a continuación.

1.1 HTML5

La mayoría de las páginas Web son realizadas utilizando lenguaje de marcado de hipertextos también conocido como HTML. Los documentos HTML son archivos de texto que contienen etiquetas, utilizadas para marcar los elementos propios del lenguaje. Estos documentos son almacenados generalmente con la extensión .html, aunque menos frecuente pero también usada es la extensión .htm.

Las etiquetas HTML son las que utilizan los navegadores para desplegar las páginas, generalmente siguiendo los estándares definidos por el World Wide Consortium (W3C; organización encargada de desarrollar los lineamientos y especificaciones para muchas tecnologías Web). Estas etiquetas se identifican dentro de un documento HTML por estar delimitadas con los símbolos de “<” y “>”, por ejemplo la etiqueta “<p>” para definir un párrafo. De hecho es una buena práctica al escribir HTML, cerrar los elementos una vez hayan finalizado, a pesar de que muchos navegadores son suficientemente inteligentes hoy en día para interpretar correctamente estos errores. Las etiquetas de cierre son idénticas a las de apertura exceptuando que después del símbolo “<” se coloca el símbolo “/” dejando el resto de la misma manera; por ejemplo, “</p>”.

La sintaxis HTML5 es la última versión disponible de este lenguaje de marcado. Su sintaxis está diseñada para ser más simple, más flexible y más amigable con el programador; igualmente posee una mayor compatibilidad con sus versiones anteriores que sus predecesoras HTML4 y XHTML. En esta versión se incluyen algunas nuevas funcionalidades tales como animaciones, capacidades offline, audio, gráficos avanzados, tipografías, transiciones y más.

Este tipo de mejoras generaron una gran ampliación en los estándares Web así como también eliminar la necesidad de usar algunas tecnologías propietarias que venían siendo utilizadas en la Web hasta ahora, tales como Flash y otras plataformas móviles nativas.

Esta nueva y mejorada versión de HTML, es compatible con la mayoría de los últimos navegadores disponibles en el mercado. Los cambios incluidos se enfocan más en la semántica del código, por lo que fueron incluidas nuevas etiquetas como <header>, <footer>, <section>, y <article> entre otras, para facilitar su lectura y su comprensión. La meta para la que fue concebida esta nueva versión es lograr un marcado semántico, asegurando que los elementos así como los estilos utilizados, definan el significado del contenido de la mejor manera posible.

Las especificaciones de HTML5 lo definen como un lenguaje abstracto para describir documentos y aplicaciones; del mismo modo definen algunos API (Application Programming Interface) para interactuar con lo que se conoce como DOM (Document Object Model). Existen varias sintaxis concretas para este lenguaje, dos de ellas son HTML y XHTML. (Craig Grannell, 2012).

En la Figura 1.1 se puede observar las diferencias entre la estructura de un documento HTML5 con respecto a sus versiones anteriores; en los documentos HTML5 es posible saber parte de la semántica del documento con solo ver su estructura, mientras que en las anteriores versiones es imposible conocer la semántica a partir de la estructura, siendo necesario evaluar individualmente todos los elementos HTML.

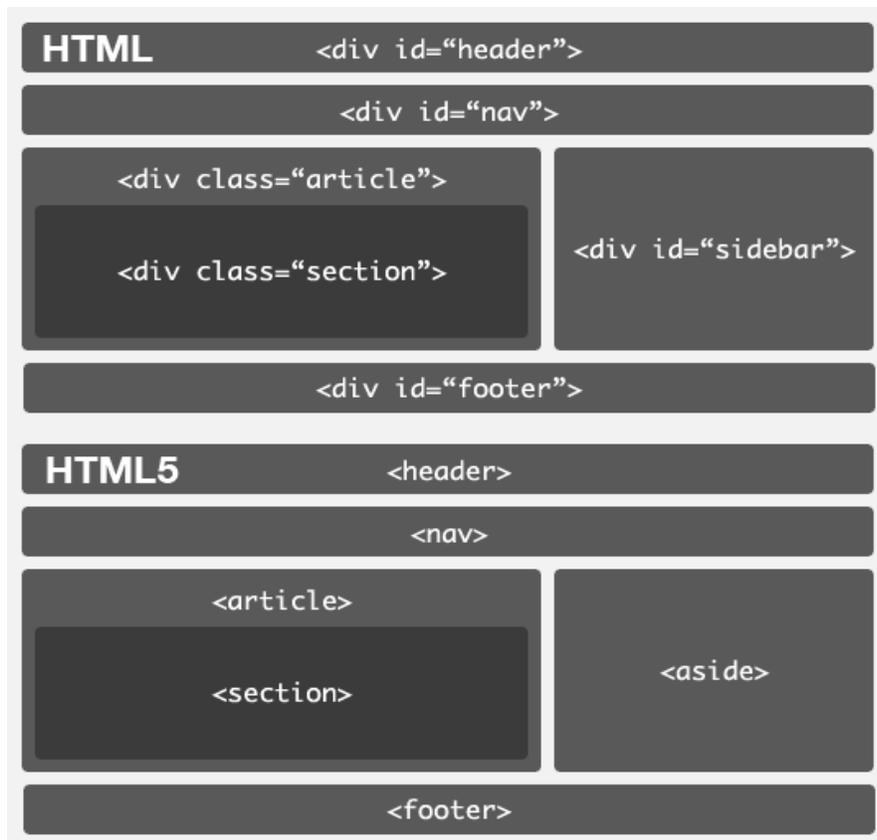


Figura 1.1 - La semántica en HTML5 facilita su lectura. (Brito, 2009)

1.2 HAML

HAML es un lenguaje de marcado que se utiliza para simplificar el código HTML de cualquier documento Web al eliminar el uso de código en línea. HAML funciona como un reemplazo para los sistemas de plantillas tales como PHP, ASP y ERB, siendo Ruby el lenguaje en el que suele encontrarse con mayor frecuencia. (HAML, 2012)

HAML elimina la necesidad de escribir HTML dentro de la plantilla, ya que por sí mismo es una descripción de HTML, con algo más de código para generar contenido dinámico.

Se puede definir como un motor de plantillas para HTML. Está diseñado para facilitar y hacer más cómoda la tarea de escribir documentos HTML, al eliminar la redundancia, reflejando la estructura subyacente que el documento representa a través de una sintaxis elegante que es tan poderosa como fácil de entender. (HAML, 2012)

HAML fue creado por Hampton Catlin; sin embargo, Nathan Weizenbaum fue por muchos años el desarrollador primario y arquitecto de las implementaciones modernas en Ruby de HAML. Fue el esfuerzo y el trabajo de Weizenbaum lo que mantuvo el proyecto con vida, tras el abandono de Catlin. Hoy en día, Norman Clarke, es el nuevo encargado de mantener el proyecto desde abril de 2012. (HAML, 2012)

HAML fue creado para evitar tener que escribir a mano plantillas constantemente. Si bien la iniciativa surge a partir de un experimento, su utilidad fue rápidamente demostrada siendo utilizado en numerosos sitios Web en producción.

Suele ser muy frecuente ver el uso de HAML en proyectos desarrollados en Rails. Principalmente, puede ser instalado simplemente como una gema y ser usado de inmediato sin mayores complicaciones.

El desarrollo de HAML se basa en cuatro principios fundamentales:

- El lenguaje de marcado debe ser legible. No basta con que el navegador lo muestre correctamente. A fin de cuentas, el lenguaje de marcado puede ser modificado por otras personas; por lo que debe ser fácil de leer, de entender y de modificar.
- El lenguaje de marcado no debe repetirse. Sin embargo HTML involucra una repetición constante del código. Cada elemento es nombrado hasta dos veces y adicionalmente cada sistema agrega otros caracteres al marcado. HAML evita todo esto valiéndose en la tabulación y no en el texto para determinar donde comienzan y terminan los elementos y bloques de código. Como resultado no sólo se obtienen plantillas más pequeñas sino un código mucho más sencillo y legible.
- El lenguaje de marcado debe estar bien tabulado. En el caso de los archivos ERB en Rails por ejemplo, esta tarea resulta no solo desafiante sino imposible lo que conlleva a un HTML ilegible. HAML automáticamente corrige los formatos de todas las etiquetas para que sea así
- La estructura HTML debe ser clara. Los formatos XML y HTML son formatos construidos sobre la idea de un documento estructurado. Dicha estructura se refleja en su marcado, y debe por tanto ser reflejada en meta-marcados tales como HAML. Dado que la lógica de HAML está basada en la tabulación de los elementos, esta estructura se conserva y hace el documento mucho más fácil de leer. (HAML, 2012)

En la Figura 1.2 es posible diferenciar un segmento de código escrito tanto en HAML como en HTML para entender sus diferencias. Se puede observar que la versión de HAML es más corta y simple que su versión equivalente en HTML.



Figura 1.2 - Ejemplo de Haml vs. Html (Fisher, 2009)

1.3 Wireframing

Wireframing en un castellano anglosajón, es la acción de realizar wireframes. Un wireframe es una manera de diseñar un sitio Web en un nivel estructural analizando las necesidades del usuario, así como la información del sitio. El objetivo de un wireframe es proporcionar una comprensión visual de una página de inicio de un proyecto Web para obtener la aprobación de los interesados y del equipo del proyecto, antes de que la fase creativa se ponga en marcha.

Un wireframe también se puede utilizar para crear la navegación primaria y secundaria garantizando que la estructura cumple con las expectativas del interesado, así como los usuarios también pueden utilizarlo como un mecanismo de retroalimentación en las primeras pruebas de usabilidad del prototipo. (maquetando.com)

Los wireframes pueden ser simplemente dibujados a mano, pero a menudo se forman por medio de software para proporcionar una entrega de la pantalla. Sin embargo, si los wireframes se van a utilizar para un test de usabilidad de prototipo, lo mejor es crearlos en HTML básico.

En el desarrollo de aplicaciones Web puede resultar útil construir prototipos de interfaz en una fase temprana ya que permite tener una idea visual de la interfaz y del sistema de navegación, así como identificar requerimientos funcionales de la aplicación.

1.4 CSS3

CSS u hojas de estilo, denominadas así a partir de su acrónimo en inglés (Cascading Style Sheets, CSS), fueron creadas para separar la capa de presentación de la capa lógica en una aplicación. Su propósito siempre ha sido proporcionar a los usuarios un lenguaje simple que defina los aspectos de estilo de sus páginas Web así como su apariencia.

Un hoja de estilo CSS declara una serie de propiedades para el contenido, tales como la familia de fuentes, el tamaño de la fuente, el color, entre otras. El World Wide Web Consortium (W3C) ha publicado diferentes versiones de CSS con el transcurrir de los años que suelen añadir nuevas funcionalidades.

Sin embargo, en lo últimos tres años, se ha notado una aceleración significativa en el crecimiento de nuevas especificaciones. De hecho, para CSS3 la W3C, intenta crear un enfoque clasificado por módulos, según sus objetivos funcionales. Este nuevo enfoque incluye propiedades, técnicas y métodos que son finalmente refinadas para satisfacer las necesidades reales de los diseñadores y programadores, que a fin de cuentas son quienes elaboran los sitios Web. Algunas especificaciones clasificadas por módulos, son selectores, modelos de caja, fondos y bordes, efectos de textos, transformaciones 2D/3D, animaciones, diagramación de múltiples columnas, interfaz de usuario, entre otros.

CSS3 es el último estándar para CSS, su sintaxis permite controlar el estilo y la diagramación de las páginas Web. Es totalmente compatible con sus predecesores, por lo que no requiere una actualización en diseños existentes.

Como se mencionó anteriormente, las especificaciones CSS3, siguen bajo desarrollo por el W3C, sin embargo, muchas de las nuevas propiedades que incluye, han sido ya incorporadas a los navegadores modernos y están disponibles para experimentar al día de hoy. (Marco Casario, 2012).

1.5 SASS: SCSS

Sass es una extensión de CSS que incrementa el poder y elegancia del lenguaje básico. Permite el uso de variables, reglas anidadas, combinaciones, importar a nivel de línea, y muchas otras funcionalidades absolutamente compatibles con la sintaxis CSS. Sass permite

mantener las hojas de estilo de cierto tamaño, mejor organizadas, obteniendo en su lugar hojas de estilo más pequeñas mucho más sencillas de codificar.

Entre sus funcionalidades incluye:

- Absolutamente compatible con CSS3.
- Extensiones del lenguaje, tales como variables, anidamientos y combinaciones.
- Un amplio conjunto de funciones para la manipulación de colores y otros valores.
- Funcionalidades avanzadas tales como directivas de control para librerías.
- Salidas correctamente formateadas y personalizables.
- Integración con Firebug.

Existen dos sintaxis disponibles para el uso de Sass. La primera es SCSS (Sassy CSS), es una extensión de CSS3. Es decir, que cualquier hoja de estilos CSS3 es por sí misma un archivo SCSS válido para el mismo propósito. Adicionalmente, SCSS comprende la mayoría de los trucos y sintaxis específica existente (como los filtros para IE). Los archivos que utilizan SCSS tienen la extensión `.scss`.

La segunda y más antigua de las dos sintaxis existentes es conocida como la sintaxis tabulada, proporciona una forma más concisa de escribir CSS. Utiliza tabulaciones en lugar de llaves para indicar los anidamientos de selectores, y saltos de línea en lugar de punto y coma para separar propiedades. Algunas personas opinan que es más fácil de leer y de escribir que SCSS. La sintaxis tabulada tiene las mismas funcionalidades, aunque algunas de ellas poseen sintaxis diferentes. Los archivos que utilizan esta sintaxis tienen la extensión `.sass`.

Cualquiera de las dos sintaxis puede importar archivos escrita en la otra y pueden ser convertidos automáticamente entre ellos a través de la instrucción `sass-convert`.

SASS fue originalmente creado por Hampton Catlin. Luego de la versión 2.0 continuó el desarrollo en conjunto con Nathan Weizenbaum y Chris Eppstein. El proyecto se encuentra descargable de forma libre bajo la licencia MIT (Sass-lang, 2012).

En la Figura 1.3 se ilustra de forma comparativa, una hoja de estilos escrita utilizando SCSS y CSS. Se puede observar que SCSS permite declarar los estilos de los elementos de forma anidada para evitar repetición de código y facilitar su lectura.

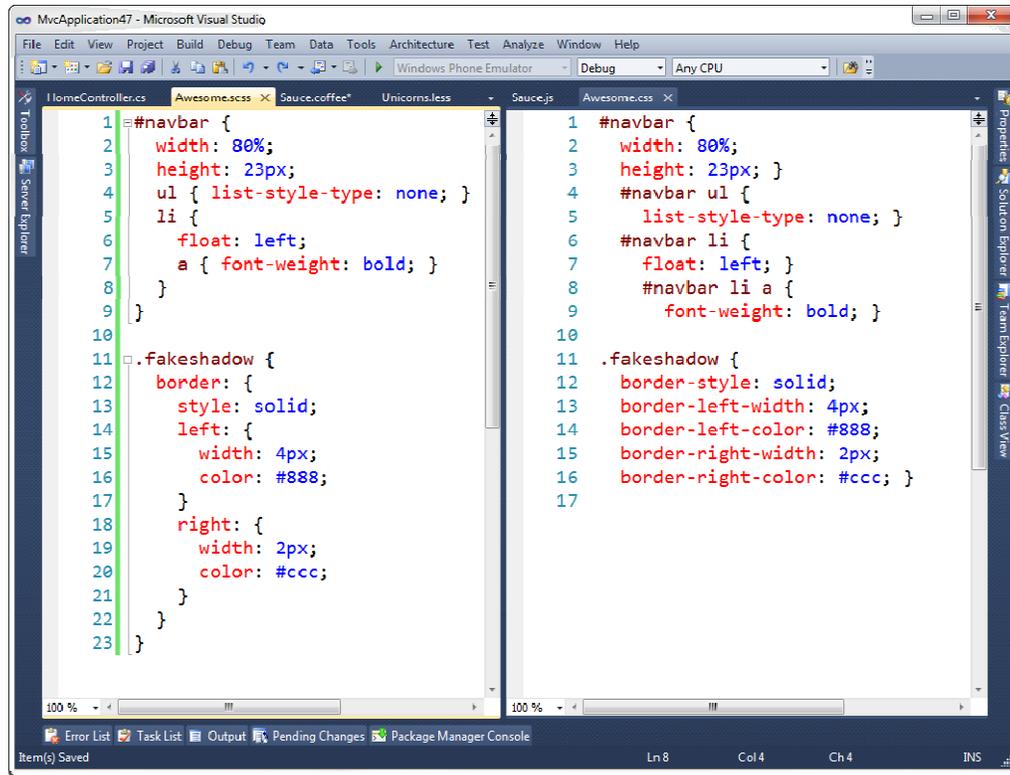


Figura 1.3 - Comparación entre SCSS y CSS en IDE Visual Studio 2010. (Hanselman, 2011)

1.6 JQuery

Hoy en día la Web es un entorno altamente dinámico, y sus usuarios poseen un estándar alto para estilos y funcionalidades dentro de los sitios Web (Jonathan Chaffer, 2009). Esto ha ocasionado que JavaScript haya recuperado su prestigio en los últimos años como resultado de un renovado interés en la alta interactividad, en las aplicaciones de internet de última generación y las tecnologías de AJAX. Sin embargo, el lenguaje ha sido forzado a crecer rápidamente optando por el uso conveniente de librerías con múltiples funcionalidades que resuelvan los problemas de interoperabilidad de los navegadores, así como también proporcionen nuevos y mejorados patrones para desarrollo Web. (Bear Bibeault, 2010)

Una de estas librerías muy popular es JQuery. La razón es por su habilidad de asistir en un amplio número de tareas. Se basa en un conjunto de conceptos tomados de la estructura de

HTML y CSS para ofrecer una colección inmensa de funcionalidades. Su diseño permite un rápido comienzo a los diseñadores con poca experiencia en programación, ya que muchos desarrolladores Web están más familiarizados con estos dos lenguajes que con el mismo JavaScript. Del mismo modo los programadores avanzados se benefician de su uso gracias a su consistencia conceptual. (Jonathan Chaffer, 2009)

JQuery es una vía fácil para que los programadores y diseñadores puedan crear efectos sofisticados con el menor código posible. Dado que JQuery resulta tan sencillo de implementar, su popularidad creció de forma exponencial en los últimos años. Actualmente existen más de 4 millones de extensiones de JQuery en la Web. Facebook y Twitter, por ejemplo utilizan un buen número de efectos de JQuery.

Al comparar JQuery con otras librerías similares que se enfocan de forma pesada en mejorar las técnicas de JavaScript, el objetivo central de JQuery es cambiar la forma en que los desarrolladores Web piensan al crear funcionalidades del lado del cliente en sus páginas. La idea es que en lugar de pasar tiempo resolviendo complejidades de grado avanzado en JavaScript, los programadores puedan hacer uso de sus conocimientos en CSS, HTML, XHTML y JavaScript en manipular los elementos de las páginas directamente, haciendo el desarrollo rápido una realidad. (Beighley, 2010)

1.7 JQuery-UI

La librería de JQuery tiene entre sus virtudes que resulta muy fácil de extender; no obstante existen numerosas extensiones de la misma en la Web. Una de estas extensiones es JQuery-UI, que resulta ser de las más populares entre programadores y diseñadores pues proporciona una cantidad de herramientas para la creación de interfaces de usuario de alto nivel.

Incluso se reconoce a JQuery-UI como la extensión oficial de la librería JQuery dirigida a proveer nuevas funcionalidades en el desarrollo de interfaces de usuario.

Las herramientas que otorga JQuery para usar dentro del ambiente del navegador si bien facilitan el trabajo, no son la solución completa pues lograr interacciones complejas entre los elementos puede ser una tarea extenuante.

JQuery permite crear interacciones y controles propios a través de sus métodos. Sin embargo, la extensión de JQuery-UI proporciona un número cuantioso de los controles generalmente usados dentro de las aplicaciones Web, para evitar tener que volverlos a crear.

Por ejemplo, una barra de progreso puede ser implementada a través de JQuery, pero con el uso de la librería JQuery-UI, esta barra de progreso ya está incluida y solo a través de unas pocas líneas de código de configuración puede ser utilizada.

Es importante destacar que JQuery-UI normalmente trabaja sobre un tema de CSS para los elementos visibles. Es posible crear temas propios a través de su página para descargar el más conveniente según el patrón de colores utilizado en un sitio Web. (Bear Bibeault, 2010)

JQuery-UI es sin lugar a dudas una de las extensiones más importantes y más populares de JQuery y su utilización puede ahorrar numerosas líneas de código así como tiempo que puede invertirse en tareas complejas que lo ameriten.

1.8 CoffeeScript

CoffeeScript es un lenguaje que compila bajo JavaScript. Fue desarrollado para mantener similitudes con otros lenguajes tales como Ruby y Python con la intención de ayudar a los desarrolladores a escribir JavaScript de manera más eficiente. Para lograrlo, elimina todas las puntuaciones innecesarias tales como llaves, puntos y comas, entre otros, y usando en su lugar espacios en blanco para reemplazarlas.

CoffeeScript también le proporciona al programador una serie de atajos para escribir código en lugar de secciones complicadas de JavaScript con una sintaxis mucho más resumida.

Además de su sintaxis mejorada, ayuda a escribir un mejor código JavaScript asistiendo en tareas tales como el enfoque de las variables y las clases apropiadamente, asegurando el uso de los operadores correctos, y muchas otras.

Es muy frecuente escuchar mencionar CoffeeScript, Ruby y Python juntos frecuentemente. CoffeeScript fue directamente modelado para mantener lo simple de las sintaxis que estos lenguajes de programación ofrecen. Por esta razón, CoffeeScript tiene una apariencia mucho más moderna que JavaScript, que fue modelado tomando como referencia lenguajes como Java y C++. No obstante a pesar de haberse concebido premeditadamente tomando Ruby y Python como referencias, no importa el lenguaje de programación utilizado, CoffeeScript

funcionará bien ya que funciona bajo JavaScript. Al compilar bajo JavaScript se puede usar cualquier o todas las librerías JavaScript más frecuentes. Es decir, no existe problema alguno en usar CoffeeScript en conjunto con JQuery, Zepto, Backbone, Jasmine o cualquier otra librería de JavaScript.

Como parte negativa, su uso implica un nuevo lenguaje que debe ser aprendido por los programadores de un equipo de desarrollo, sin embargo, su sintaxis hace que sea fácil de aprender, por lo que a largo plazo no supone un problema. (Bates, 2012)

En la Figura 1.4 se ilustra de forma comparativa, un archivo JavaScript, escrito utilizando CoffeeScript y JavaScript tradicional. Se puede observar que el archivo de la izquierda correspondiente a la versión en CoffeeScript resulta más corto y legible que su contraparte en JavaScript tradicional.

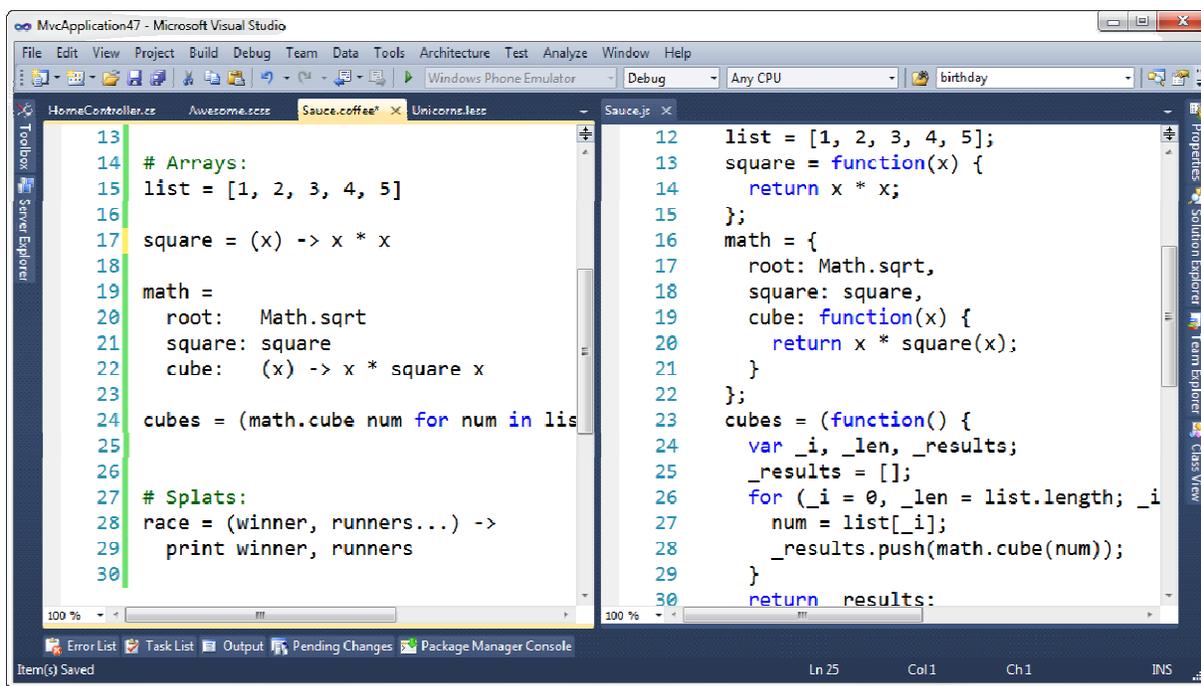


Figura 1.4 - Comparación de CoffeeScript y JavaScript en IDE Visual Studio 2010. (Hanselman, 2011)

1.9 Ruby

En 1995, Yukihiro Matsumoto publicó la versión de Ruby como un nuevo lenguaje de programación a disponibilidad de los desarrolladores. Fue creado con la intención de optimizar

el proceso de desarrollo de software por lo que es válido afirmar que es un lenguaje de programación que persigue la filosofía de un diseño eficiente. (Ediger, 2008)

El autor Michael Fitzgerald define a Ruby como un lenguaje de programación orientado a objetos aceptado mundialmente como un lenguaje fácil de aprender, poderoso e interpretado. De acuerdo a Fitzgerald, el creador de Ruby tomó partes de sus lenguajes favoritos para construir este nuevo lenguaje, que involucrara tanto la programación funcional como la programación imperativa. De aquí que Ruby comparte similitudes sintácticas con otros lenguajes como Python, Perl y SmallTalk. Su implementación oficial es distribuida bajo una licencia de software libre. (Fitzgerald, 2007)

Las principales características ofrecidas por este nuevo lenguaje fueron:

- **Interpretado:** Ruby es un lenguaje interpretado. Por tanto, cuando se hace un cambio al código fuente, no hay necesidad de compilar el código y volverlo a correr para ver los efectos del cambio. Como resultado de esta funcionalidad, el ciclo programar-compilar-ejecutar se transforma en programar-ejecutar.
- **Orientado a Objetos:** Ruby es puramente orientado a objetos. Es decir que todo dentro de Ruby es un objeto. Esto incluye a tipos de datos primitivos y números. Adicionalmente, soporta las funcionalidades requeridas por un lenguaje orientado a objetos.
- **Funcional:** Ruby soporta la programación funcional utilizando bloques.
- **Tipología Dinámica:** Ruby decide el tipo de datos de la variable mientras el programa se está ejecutando, evaluando el valor contenido en la variable en ese instante.
- **Manejo de Memoria Automático:** en otros lenguajes también se conoce como recolección de basura. Al igual que la mayoría de los lenguajes de muy alto nivel (VHLL – Very High Level Language) Ruby proporciona métodos de recolección de basura automáticos por lo que no hay que preocuparse de este aspecto.
- **Hilos:** la versión actual de Ruby proporciona parcialmente hilos independientes de plataforma. Parcialmente, dado que los hilos de Ruby se ejecutan sobre una máquina virtual en lugar de ejecutarse como hilos del lenguaje operativo nativo.

- **Reflexión:** Ruby proporciona un programa con la habilidad de mirarse a sí mismo mientras se ejecuta. Esta habilidad, conocida con distintos nombres, como introspección, reflexión u otros, permite modificar ciertos aspectos de sí mismo durante la ejecución, o crear un objeto completamente nuevo en tiempo de ejecución basado en los requerimientos de un momento dado.

La versión 1.8.6 se convirtió en el lenguaje con más rápido crecimiento desde su publicación. Ruby comienza a ganar popularidad alrededor del año 2001 con la publicación de RubyGems. RubyGems es un gestor de paquetes que permitía de una manera muy sencilla instalar paquetes y distribuir aplicaciones y librerías. Esto hizo que se crearan en Ruby numerosas aplicaciones y documentación, que hasta entonces no existían, lo que era una debilidad del lenguaje. (Ediger, 2008).

Algunas de las ventajas y beneficios de su implementación son:

- **Flexibilidad:** permite modificar funcionalidades en sus clases base y métodos.
- **Multiplataforma:** se integra perfectamente en gran cantidad de arquitecturas.
- **Dinamismo:** no se necesitan declarar variables y las mismas no tienen tipo, por lo cual se adaptan al contexto en el que estén presentes.
- **Extensibilidad:** además de sus respectivas librerías, se puede ampliar utilizando otros lenguajes.
- **Versatilidad:** puede ser usado tanto para el desarrollo de aplicaciones Web como para la simulación de ambientes complejos.
- **Productividad:** por cómo está constituido, promueve las mejores prácticas de programación sin perder usabilidad.

1.10 Ruby on Rails

Ruby on Rails, también conocido como Rails, es un framework para construir aplicaciones Web. Es por muchos considerado el mejor por ser de código abierto y multiplataforma. Proporciona una capa de abstracción de la base de datos bastante poderosa denominada Active Record, la cual trabaja con todos los sistemas manejadores de bases de datos más

populares. Por encima de todo Rails tiene su propia filosofía de arte de desarrollo Web que se centra alrededor de la productividad y la eficiencia.

Rails fue originalmente creado por David Heinemeier Hansson, no obstante, la primera versión del framework fue extraída de una aplicación funcional denominada Basecamp. Rails es un lenguaje práctico y no requiere funcionalidades extras. Su objetivo como framework es resolver el 80% de los problemas ocurridos durante un proceso de desarrollo Web, asumiendo que el 20% restante son los problemas únicos del dominio de la aplicación. (Jeffrey Allan Hardy, 2007)

Rails sigue unas ideas rígidas y específicas sobre la estructura de directorios, nombramientos de archivos, estructuras de datos, argumentos y prácticamente de todo lo involucrado en la infraestructura de una aplicación Web. Por tanto, cuando se escribe una aplicación Rails se espera que se sigan las convenciones que ya han sido establecidas, en lugar de implementar todas nuevamente. (Dave Thomas, 2005)

Rails fue diseñado tomando en cuenta los conceptos ágiles y toma cada principio del manifiesto ágil a pie de la letra. Con Rails, se puede responder a las necesidades de un proyecto de forma rápida y sencilla, y trabajar bien en un ambiente colaborativo. Rails consigue todo esto adhiriéndose a su propio conjunto de principios que favorecen la programación ágil:

- **Don't Repeat Yourself (DRY):** Al traducir, "No te repitas", es un principio que se enfoca en reducir la duplicidad de información manteniendo una parte del conocimiento en un solo lugar en el sistema. Es decir, este principio se refiere a que las definiciones deben realizarse una única vez. Dado que Ruby on Rails es un framework de pila completa, los componentes están integrados de tal manera que no hace falta establecer puentes entre ellos. Por ejemplo, en Active Record, las definiciones de las clases no necesitan especificar los nombres de las columnas; Ruby puede averiguarlos a partir de la propia base de datos, de forma que definirlos tanto en el código como en el programa será redundante.
- **Convention over Configuration (COC):** es un principio de diseño de programación que favorece un conjunto de convenciones en vez de configurar un framework de aplicación. Es decir, el programador solo necesita definir aquellas configuraciones que no sean convencionales. Ciertas configuraciones usadas comúnmente (por convención y no por reglas) están preestablecidas y el framework trabaja siguiendo las mismas. Por ejemplo, si hay una clase Usuario en el modelo, la tabla correspondiente de la base de datos es

usuarios, pero si la tabla no sigue la convención debe ser especificada manualmente. Así, cuando se diseña una aplicación desde cero y sin una base de datos preexistente, el hecho de seguir las convenciones de Rails puede traducirse como menos cantidad de código.

Rails emplea un patrón arquitectónico consagrado y consolidado que separa la lógica de la aplicación y el trabajo en tres distintas categorías: el modelo, la vista y controlador. (Jeffrey Allan Hardy, 2007)

1.11 Patrón de Diseño Arquitectónico Modelo Vista Controlador (MVC)

Es una arquitectura para el desarrollo de aplicaciones en la que el diseño está dividido en tres tipos de componentes: Modelos, Vistas y Controladores (MVC). Es la estructura estándar para las aplicaciones Rails. Esta arquitectura garantiza la separación entre el dominio lógico (también llamada lógica de negocio), las entradas y la presentación lógica asociada con una interfaz gráfica de usuario. En el caso de las aplicaciones Web, el dominio lógico generalmente consiste en modelos de datos como por ejemplo usuarios, artículos, productos, y la interfaz gráfica de usuario no es más que una página Web en un navegador.

Cuando se interactúa con una aplicación MVC, un navegador envía una petición la cual es recibida por un servidor Web y pasada al controlador, el cual está a cargo de dirigir qué se debe hacer a continuación. En algunos casos, el controlador inmediatamente desplegará una vista, la cual es una plantilla que se convierte a HTML y se envía de regreso al navegador que hizo la petición. En otros casos más comunes para sitios dinámicos, el controlador interactuará con el modelo, el cual es un objeto que representa un elemento del sitio y está a cargo de comunicarse con la base de datos. Después de invocar al modelo, el controlador despliega la vista y retorna la página Web completa al navegador como HTML. (Hartl, 2011)

- **Modelos:** los modelos son los responsables de mantener el estado de la aplicación. Algunas veces este estado trasciende y dura solo un par de interacciones con el usuario. Algunas veces el estado es permanente y será almacenado afuera de la aplicación o en una base de datos. Los modelos se encargan de aplicar las reglas de negocios. Las definiciones de las clases también detallan las relaciones entre clases con sentencias de mapeo objeto relacional.
- **Vistas:** Las vistas son responsables de generar la interfaz de usuario, normalmente basadas en los modelos.

- **Controladores:** los controladores son los encargados de dirigir la aplicación. Reciben eventos externos que interactúan con los modelos para luego desplegar información a través de las vistas. Un controlador define las acciones necesarias que pueden ser invocados desde la Web, y a su vez, presenta los datos usando el archivo de plantilla correspondiente a no ser que el método redirija a algún otro lugar de la aplicación.

A continuación, en la Figura 1.5 se presenta una representación esquemática de la arquitectura modelo-vista-controlador (MVC). En ella se observa cómo las peticiones son enviadas y recibidas por el controlador, que a su vez interactúa con los modelos obteniendo los datos necesarios para construir la vista que en última instancia es retornada al cliente.

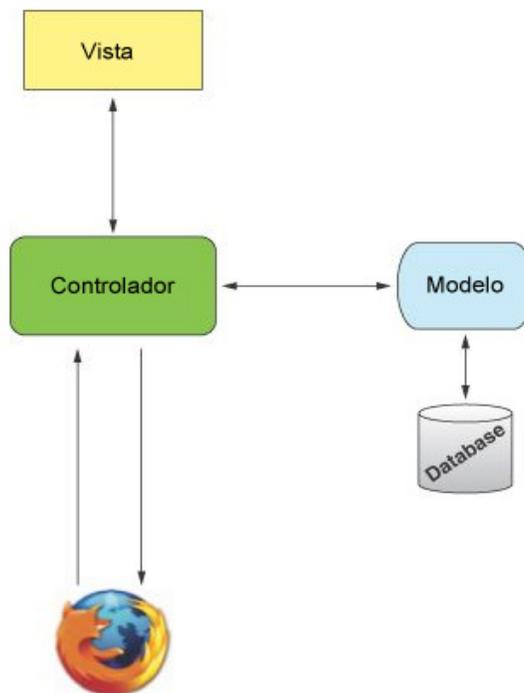


Figura 1.5 - Representación Esquemática de la Arquitectura MVC (Hartl, 2011)

1.12 MySQL

La elección adecuada de un sistema de gestión de bases de datos relacionales es esencial en diversos ambientes de desarrollo. Con el auge del movimiento de software libre se han generado muchos productos importantes de software, entre ellos, diversos sistemas de bases de datos. Uno de estos sistemas es MySQL, un sistema manejador de bases de datos relacionales de origen escandinavo. MySQL incluye un servidor SQL, clientes para acceder al

servidor, herramientas administrativas y una interfaz de programación para escribir programas propios.

Inicialmente la popularidad de MySQL se dio dada su velocidad y simplicidad. No obstante, carecía de funcionalidades importantes tales como las transacciones y soporte para claves foráneas. MySQL continuó creciendo a lo largo de su historia agregando muchas otras funcionalidades que lo impulsaron a competir duramente con otros sistemas de bases de datos propietarios. MySQL es portable y funciona en sistemas operativos libres y comerciales. Más allá, su rendimiento compite con cualquier sistema manejador de bases de datos, logrando manejar eficaz y eficientemente bases de datos grandes y tablas de billones de filas.

El uso por tanto de MySQL también ha continuado creciendo a medida de que las distintas organizaciones descubren su potencial y el importante ahorro que significa su uso para la misma.

Al comparar MySQL con otros sistemas de bases de datos, se consideran una serie de aspectos que suelen ser importantes como lo son, rendimiento, soporte, funcionalidades, licencia, restricciones y precio. MySQL aporta una serie de beneficios que lo convierten en uno de los manejadores de bases de datos más utilizados hoy en día.

- **Velocidad:** MySQL es rápido. Los desarrolladores de MySQL afirman que se trata del sistema de bases de datos más rápido que se pueda tener.
- **Facilidad de uso:** MySQL es un sistema de bases de datos de alto rendimiento y aun así es relativamente simple y mucho menos complejo de configurar y administrar que sistemas más grandes.
- **Soporte SQL:** es totalmente compatible con SQL, el lenguaje estándar utilizado por la mayoría de los sistemas de bases de datos modernos.
- **Capacidad:** el servidor de MySQL es multiproceso, por lo que puede conectar los clientes necesarios al mismo tiempo. Cada cliente puede utilizar a su vez múltiples bases de datos simultáneamente. Se puede acceder a MySQL interactivamente a través de distintas interfaces que permiten introducir consultas y ver los resultados: clientes de línea de comando, navegadores Web o clientes con GUI. Adicionalmente, las interfaces de programación están disponibles para muchos lenguajes tales como C, Perl, Java, PHP,

Python y Ruby. También se puede acceder MySQL utilizando aplicaciones que soporten ODBC y .NET.

- **Conectividad y Seguridad:** MySQL funciona totalmente en red por lo que las bases de datos pueden ser accedidas desde cualquier lugar en Internet. Adicionalmente, MySQL proporciona control de acceso para restringir el acceso a los datos según sea necesario. Para garantizar la seguridad MySQL soporta conexiones encriptadas utilizando el protocolo SSL.
- **Portabilidad:** MySQL es un sistema altamente portable; funciona perfectamente bajo distintas versiones de Unix y Linux, así como también otros sistemas operativos tales como Windows y NetWare. Por otro lado, MySQL funcionará en servidores de alto rendimiento así como también en pequeñas computadoras personales.
- **Tamaño pequeño:** MySQL tiene un tamaño de distribución modesto, especialmente al compararlo con la gran cantidad de espacio de disco utilizado por otros sistemas de bases de datos.
- **Disponibilidad y costo: MySQL** es un proyecto Open Source disponible bajo términos de múltiple licenciamiento. Primeramente, está disponible bajo los términos de la Licencia GNU General Public License (GPL), es decir, está disponible sin costo para la mayoría de los usos. No obstante, también existen licencias comerciales disponibles para organizaciones que prefieran o requieran arreglos formales o que no desean ser sometidos a las condiciones de la GPL.
- **Distribución y Código Fuente Abiertos:** MySQL es fácil de obtener; solo se requiere utilizar el navegador Web. Si no se entiende bien cómo funciona, o se desea más información sobre un algoritmo, o simplemente se desea realizar una auditoría de seguridad, se puede obtener el código fuente y examinarlo. Si se cree que se encontró un error es importante reportarlo.

El principal problema que puede plantearse con el uso de MySQL viene dado por el soporte, que garantizan otras empresas dedicadas a la venta comercial de sistemas de bases de datos; sin embargo, existen muchos recursos abiertos que brindan soporte a MySQL:

El manual de referencia de MySQL está incluido con todas las distribuciones MySQL, y también está disponible en línea. Oracle brinda la posibilidad de servicios de monitoreo, clases de entrenamiento y soporte técnico.

Existen innumerables listas de correo activas sobre MySQL a las que cualquier individuo puede afiliarse. Estas listas tienen muchos participantes colaborando de gran importancia, algunos de ellos son los desarrolladores de MySQL.

La comunidad de MySQL es muy participativa tanto para desarrolladores como para no desarrolladores. Las preguntas en las listas de correos suelen ser contestadas en cuestión de minutos. Cuando se reportan errores los desarrolladores normalmente los arreglan rápidamente y los parches están disponibles para descargarlos diariamente en Internet. Al contrastar esto con la experiencia normalmente obtenida de los grandes vendedores de sistemas de bases de datos, se observa una de las principales fortalezas de MySQL.

Las distribuciones de MySQL incluyen las siguientes herramientas:

- **Servidor SQL:** es el motor de fuerza de MySQL y proporciona acceso a las bases de datos.
- **Programas Clientes y de Utilidades:** incluyen un cliente interactivo que permite realizar consultas directamente y ver los resultados. También están disponibles varias herramientas administrativas y programas de utilidades que facilitan el mantenimiento de las bases de datos. Estos programas de utilidades por ejemplo facilitan el monitoreo y control del servidor, importación de datos, respaldo de datos, revisar tablas en búsqueda de problemas y más.
- **Librería cliente para escribir programas propios:** cualquier programador puede escribir programas clientes en C porque la librería está en C, sin embargo, la librería puede ser enlazada con lenguajes tales como Perl, PHP o Ruby para proporcionar las bases de las interfaces de MySQL en dichos lenguajes.

Adicionalmente al software que se incluye con MySQL, existen muchos programas escritos por personas talentosas y bien capacitadas que están dispuestos a compartir dicho software. Como resultado se tiene acceso a una variedad de aplicaciones que facilitan el uso de MySQL o que extienden su alcance en áreas tales como el desarrollo de aplicaciones Web. (Dubouis., 2009)

La Figura 1.6 representa gráficamente la estructura lógica de la arquitectura de MySQL. La capa superior contiene los servicios que no son exclusivos de MySQL. Son herramientas de tipo cliente/servidor. La segunda capa es donde se encuentra el núcleo de MySQL, incluyendo el código para el análisis de la consulta, análisis, optimización, el almacenamiento en caché, y todas las funciones incorporadas (por ejemplo, fechas, horas, matemáticas y cifrado). Cualquier funcionalidad proporcionada en motores de almacenamiento vive en este nivel: procedimientos almacenados, triggers, vistas entre otros. La tercera capa contiene los motores de almacenamiento. Son los responsables de almacenar y recuperar los datos. (Dubouis., 2009)

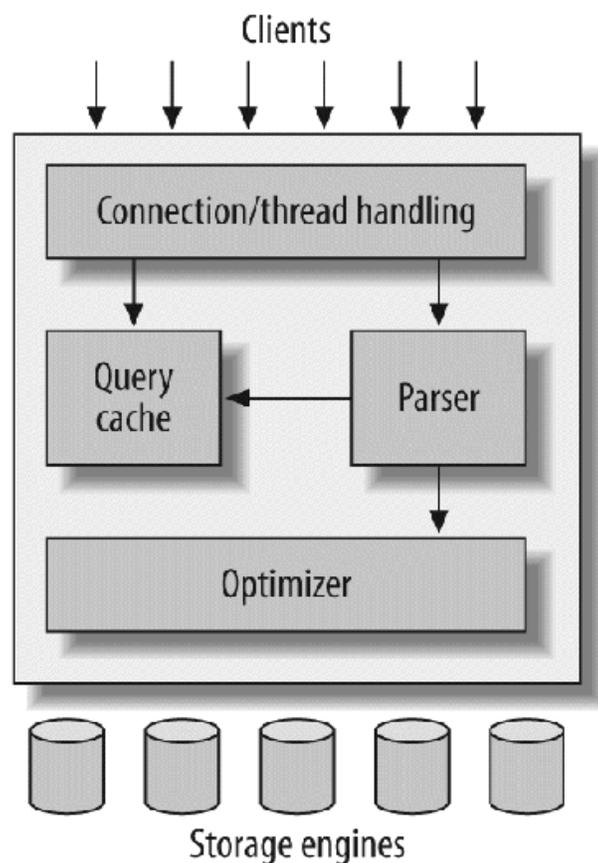


Figura 1.6 - Vista Lógica de la Arquitectura de MySQL (Dubouis., 2009)

1.13 GIT

Al iniciar un proyecto de desarrollo, muchos autores recomiendan como buena práctica, la creación de una estrategia de respaldo de datos; hoy en día son muchas las causas que pueden conllevar una pérdida total o parcial de la información por ende siempre es aconsejable tener respaldo de la misma. En el caso de los proyectos de programación, ésta estrategia

involucra el uso de un manejador de versiones. Un manejador de versiones no es más que una herramienta que gestiona las diferentes versiones de software o de otros contenidos dentro de un proyecto.

GIT, es un manejador de versiones particularmente poderoso, flexible y de bajos costos operativos que facilita enormemente el desarrollo colaborativo. Fue creado por Linus Torvalds para apoyar el desarrollo del kernel de Linux 1, pero desde entonces se ha convertido en una herramienta invaluable en el desarrollo de numerosos proyectos.

Dado que GIT es un sistema de control de versiones distribuido, es vital que garantice la integridad de la data y que la misma no se modifica de alguna forma. Para ello, GIT utiliza una función criptográfica muy común llamada SHA1 (Secure Hash Function), para nombrar e identificar los objetos dentro de su base de datos. Si bien este método no es absoluto, ha probado ser bastante fiable.

De igual manera, GIT lleva un registro de todos los cambios realizados sobre cada archivo. La información almacenada en esos cambios es actualizada por el programador. GIT se asegura de que los cambios no ocurran extrañamente a los archivos controlados, llevando una tabla de registros de cada cambio realizado en los mismos.

Los repositorios GIT contienen algunos objetos de datos que son inmutables. Estos objetos, una vez que han sido creados y colocados en la base de datos, no pueden ser modificados. El diseño de las bases de datos GIT implica que por ejemplo, el historial almacenado sea inmutable.

Las transacciones llevadas a cabo en GIT son atómicas; es decir, se almacenan transacciones completas y estados del repositorio discretos, que no pueden ser descompuestos en cambios individuales; con esto se asegura que la base de datos del control de versiones no quede en estado corrupto mientras ocurre algún cambio.

Dado que está hecho para controlar las versiones, GIT tiene muchas cosas en común con otros sistemas de control. Las nociones de agregar archivos, cambios del registro, repositorios, entre otras, son las mismas, y el flujo de trabajo también es similar a otras herramientas similares.

Un repositorio GIT es una base de datos que contiene toda la información necesaria para retener y manejar, las revisiones y el historial de un proyecto. En GIT, al igual que en otros

sistemas de control de versiones, el repositorio almacena una copia completa del proyecto durante todo su ciclo de vida. Sin embargo, a diferencia de otros sistemas de este tipo, el repositorio no sólo proporciona una copia funcional de todos los archivos, sino también de sí mismo.

GIT mantiene un conjunto de valores de configuración dentro de cada repositorio tales como el nombre y correo del usuario del repositorio. Estos valores de configuración, a diferencia de los otros datos y metadatos del repositorio, no son propagados de un repositorio a otro durante una operación de clonación o duplicación del mismo. En lugar de eso, GIT gestiona e inspecciona la información de configuración.

En último lugar, cabe mencionar que dentro de un repositorio GIT mantiene dos estructuras de datos primarias, la denominada tienda de objetos y el índice. Todos los datos del repositorio son almacenados en la carpeta raíz del proyecto, en una carpeta oculta llamada `.git`. La tienda de datos está diseñada para ser eficientemente copiada durante una operación de clonación como parte del mecanismo que soporta un manejador de versiones completamente distribuido. El índice es información transitoria, es privada a un repositorio y puede ser creada o modificada a petición. (Jon Loeliger, 2012)

1.14 Plataforma

En la figura 1.7 se presenta un diagrama que muestra cómo interactúan en la aplicación, las distintas tecnologías descritas en este capítulo. En dicha figura se aprecia que el usuario interactúa a través de clientes de navegación en los que se utiliza las tecnologías HTML5, CSS3 y JQuery enviando peticiones a través de internet al servidor donde se encuentra la herramienta; estas peticiones son procesadas por la aplicación desarrollada en Ruby on Rails que a su vez efectúa peticiones sobre una base de datos MySQL devolviendo los datos solicitados por la misma vía hasta llegar nuevamente al cliente de navegación del usuario.



Figura 1.7 - Diagrama de plataforma.

Finalmente, una vez descritas las principales tecnologías que engloban el presente trabajo de investigación, el cual se centra en el desarrollo de una aplicación Web, resulta de vital importancia destacar y explicar el conjunto de procedimientos y métodos de desarrollo que determinaran el curso de la investigación. Los mismos, serán brevemente descritos en el próximo capítulo.

Capítulo II. Métodos de Desarrollo de Software

El ciclo de vida de desarrollo de software describe la vida de un producto de software en términos de procesos desde su concepción hasta su desarrollo, implementación y mantenimiento. El proceso de desarrollo de software generalmente involucra las siguientes etapas: ingeniería de requerimientos, diseño, implementación, pruebas y mantenimiento. Cada una de estas etapas involucra a su vez una serie de actividades que conlleva a la puesta en producción del producto de software. (Asif Irshad Khan, 2011)

Actualmente, la industria del desarrollo de software posee distintos métodos entre los cuáles elegir según varíen sus necesidades. Incluso existen algunas organizaciones que optan por utilizar métodos propios para el desarrollo de software; sin embargo, la mayoría del sector concuerda en dividir estos métodos bajo dos grandes categorías: métodos pesados y métodos livianos.

La aplicación Web en que se fundamenta este trabajo de investigación tiene como objetivo permitir gestionar las actividades de Seminarios y TEG de la Escuela de Computación de la Facultad de Ciencias que utilicen métodos ágiles de desarrollo de software. Para ello es importante conocer algunos conceptos básicos de los distintos métodos de desarrollo existentes.

De igual manera, esta aplicación fue desarrollada utilizando uno de estos métodos ágiles; específicamente, el método ágil XP, que es también descrito en este capítulo.

2.1 Métodos de Desarrollo Ágiles

Los métodos ágiles, se basan en un conjunto de prácticas que les permiten a los programadores construir soluciones de manera más rápida y eficiente, respondiendo mejor ante los cambios de requerimientos emergentes.

Estos métodos se enfocan principalmente en desarrollos con cortos ciclos de vida, trabajando en conjunto con todas las partes involucradas en el proyecto, intentando mantener el desarrollo como un proceso simple. (Asif Irshad Khan, 2011)

El desarrollo ágil no se trata de un proceso específico que se pueda seguir a través de una serie de pasos. El desarrollo ágil se trata de una filosofía o una forma de concebir el proceso de desarrollo de software. La descripción canónica de esta filosofía se encuentra establecida en el

Manifiesto Ágil; este manifiesto, es una colección de 5 valores y 12 principios que deben ser puestos en práctica para desarrollar de forma ágil.

Los métodos de desarrollo ágiles consisten en conjuntos de elementos llamados convenientemente prácticas. Las prácticas incluyen hábitos como por ejemplo, estándares de codificación, controles de versiones o la elaboración de versiones demo para los integrantes del proyecto. La mayoría de estas prácticas ya existían previamente al desarrollo ágil. Estos métodos simplemente las combinan de maneras únicas, acentuando aquellas partes que dan pie a la filosofía ágil, descartando el resto, y mezclando estas prácticas con nuevas ideas. El resultado es un método poderoso que se enfoca en aumentar la productividad y la calidad en los procesos de desarrollo de software. (James Shore, 2008)

2.1.1 Valores de los Métodos Ágiles

La diferencia entre lo que se considera valioso y lo que es verdaderamente valioso genera desperdicios. El principal problema que ocurre con la gente con conocimientos básicos de desarrollo de software es que enfocan el proceso de desarrollo de forma individual. Lo verdaderamente importante es cómo los individuos se comportan como parte de un equipo y de una organización. Si todos los miembros de un equipo de desarrollo se enfocan en lo que es importante para todos en común, el éxito del desarrollo es más posible.

Los métodos de desarrollo ágil se cimentan sobre cinco valores que son fundamentales para la aplicación correcta de cualquiera de sus enfoques:

- **Comunicación:** lo más importante dentro de un equipo de desarrollo es la comunicación. Cuando surgen problemas, lo más probable es que algún miembro del equipo ya conozca alguna solución. La comunicación es importante para crear sentido de equipo así como cooperación efectiva. Sin embargo, si bien es muy importante, no es lo único necesario para lograr desarrollo efectivo de software.
- **Simplicidad:** es el más difícil de digerir de los valores ágiles. Hacer un sistema lo suficientemente simple para resolver con elegancia únicamente el problema actual es un trabajo duro. Cuando se tiene la necesidad de cambiar para ganar en simplicidad se debe encontrar una manera de llegar desde el estado actual al estado deseado. Los valores intentan compensarse y brindarse apoyo mutuamente, por lo que mejorar la comunicación conlleva a alcanzar mayor simplicidad eliminando requerimientos innecesarios o

postergando los que no son relevantes para las prioridades de un momento específico. Curiosamente, al alcanzar la simplicidad existe mucho menos que comunicar.

- **Retroalimentación:** en los métodos ágiles los cambios son inevitables y a su vez, crean la necesidad de retroalimentación. La idea no consiste en obtener la perfección de forma instantánea sino tener satisfacción por las mejoras realizadas en un momento dado. Se utiliza la retroalimentación como herramienta para acercar el estado actual al estado deseado. Los métodos ágiles intentan apoyarse tanto como pueden en la retroalimentación, y siempre de la forma más rápida posible. La retroalimentación a su vez, juega un papel crítico en la comunicación y contribuye significativamente a alcanzar la simplicidad. Mientras más simple sea el sistema, más fácil será obtener retroalimentación del mismo.
- **Coraje:** el coraje es una forma efectiva de enfrentar los temores. Las personas involucradas en el desarrollo de software suelen tener miedos. La manera en que manejan sus temores dictan si trabajan o no como partes efectivas de un equipo. El coraje como valor puede ser peligroso si no está en equilibrio con el resto de los valores, ya que puede conllevar a tomar decisiones sin importar las consecuencias, lo que no es efectivo en trabajo en equipo. Del mismo modo también es un valor muy poderoso si es de forma balanceada pues fomenta la comunicación y la confianza entre los miembros del equipo. El coraje necesario para descartar soluciones erróneas y buscar nuevas alternativas conlleva a la simplicidad. El coraje para buscar respuestas reales y concretas fomenta la retroalimentación.
- **Respeto:** es el valor sobre el que reposan los cuatro valores descritos anteriormente. Si los miembros de un equipo de trabajo no sienten respeto alguno entre sí o sus trabajos, las metodologías ágiles no funcionan. Para que el desarrollo de software crezca en productividad y humanidad, los aportes realizados por cada persona del equipo deben ser respetado.

Estos valores son los pilares de los métodos ágiles. Sin embargo, por sí mismos no proporcionan una forma concreta de cómo o qué se debe hacer durante el proceso de desarrollo. Para ello, cada método posee un conjunto de prácticas específicas que, basadas en unos principios específicos intentan alcanzar los valores descritos durante el proceso del ciclo de vida de un desarrollo de software. (Kent Beck, 2005)

2.1.2 Método por Prototipos

El método por prototipos consiste en un producto parcialmente desarrollado que permite a los miembros del equipo evaluar algunos aspectos del sistema propuesto y decidir si es adecuado para el producto final.

En este método, la construcción del producto comienza con el análisis de los requerimientos. Se definen los objetivos generales del producto, se identifican todos los requerimientos conocidos y posteriormente se realiza un diseño rápido que luego conlleva a la construcción de un prototipo.

El prototipo luego es evaluado en conjunto con todos los usuarios y se utiliza para refinar todos los requerimientos. Luego en una nueva iteración se construye un nuevo prototipo hasta que finalmente se alcanzan los resultados esperados (Asif Irshad Khan, 2011). En la Figura 2.1 se representa gráficamente el Método de Prototipos.

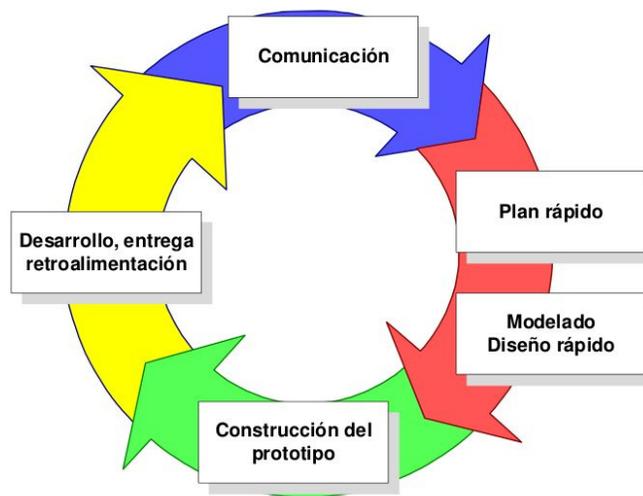


Figura 2.1 - Representación de Método por Prototipos (OSC, 2011)

2.1.3 Método RAD

RAD (Rapid Application Development), Es un método de desarrollo incremental que se fundamenta en ciclos de desarrollo muy cortos en conjunto y constante retroalimentación de todas las partes involucradas a través del ciclo de vida del desarrollo. (Asif Irshad Khan, 2011)

El objetivo principal de este método es evitar una extensa planificación, permitiendo a los programadores escribir software mucho más rápido y facilitando los cambios en los requerimientos. A continuación en la Figura 2.2, se representa gráficamente el Método RAD.

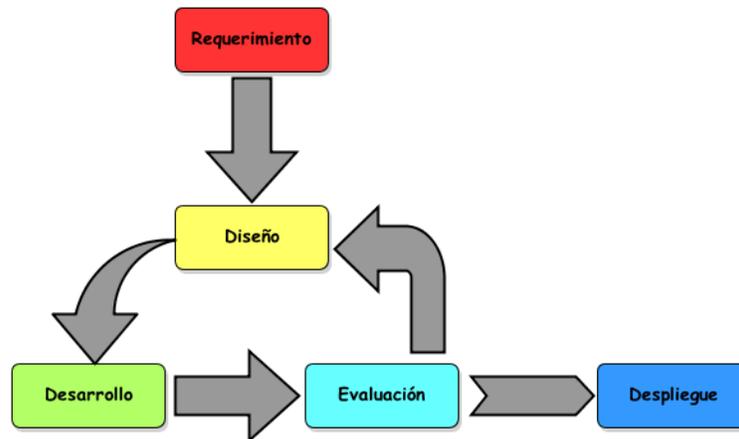


Figura 2.2 - Representación del Método RAD

2.1.4 Método de Programación Extrema (XP)

El método XP se basa en la excelencia de la aplicación de técnicas de programación, comunicación clara y trabajo en equipo, lo cual permite alcanzar resultados de alta calidad. XP está fundamentado sobre la filosofía de desarrollo ágil, por tanto está orientado a lograr los valores descritos por la misma. Adicionalmente posee un conjunto de prácticas que han demostrado ser útiles en el desarrollo de software. Cada una de estas prácticas se complementa entre sí ampliando sus efectos. Las mismas fueron elegidas como expresiones de los valores ágiles.

Un conjunto de principios y técnicas intelectuales son los que transforman los valores en las prácticas utilizadas por XP. Este método es un camino de mejora de la excelencia de las personas que se unen para desarrollar software. Algunas de sus diferencias respecto a otros métodos son:

- Ciclos de desarrollo cortos, que resultan en una retroalimentación temprana, concreta y constante.
- Enfoque de planificación incremental que permite desarrollar rápidamente un plan emergente que evolucione a lo largo de la vida del proyecto.

- Habilidad para flexibilizar la planificación de implementación de funcionalidades, dando respuesta a las necesidades cambiantes del negocio.
- Soporte a pruebas automatizadas escritas por programadores y usuarios para monitorear el progreso del desarrollo y permitir la evolución del sistema, así como la temprana detección de defectos.
- Soporte en comunicaciones oral para transmitir información referente a la estructura e intenciones del sistema.
- Diseño evolutivo que progresa mientras dure el desarrollo del proyecto.
- Trabajo de equipo colaborativo sin especialidades o talentos extraordinarios.
- Conjunto de prácticas que trabajan en armonía con los términos a corto plazo del equipo y los intereses a largo plazo del proyecto.

XP es un método liviano para equipos pequeños y medianos que desarrollan software cuyos requerimientos están constantemente cambiando. Es una disciplina que maneja los riesgos en todos los niveles del proceso de desarrollo, mientras intenta incrementar la productividad y producir software de alta calidad, sin agobiar a los miembros del equipo. (Kent Beck, 2005)

2.1.4.1 Prácticas

Las prácticas pueden ser incorporadas a un equipo de trabajo progresivamente a medida se empieza a aplicar el método XP para optimizar el proceso de desarrollo de software. La incorporación de unas u otras obedece a cada contexto específico, por lo que no existe un orden prioritario en las prácticas, sino una apreciación subjetiva de cuáles pueden ofrecer una mayor oportunidad de mejora en el proceso de desarrollo. (Kent Beck, 2005) Las prácticas establecidas por el Método XP son:

- **Espacio de Trabajo Conjunto:** desarrollar en un espacio abierto lo suficientemente grande para todo el equipo. De esta manera se facilita la comunicación dentro del equipo. Sin embargo, los equipos distribuidos en distintos lugares también pueden realizar XP manteniendo contacto cara a cara de forma constante y periódica de manera que se humanice el proyecto. En la Figura 2.3 se representa gráficamente un ejemplo de espacio de trabajo conjunto bajo el método XP.

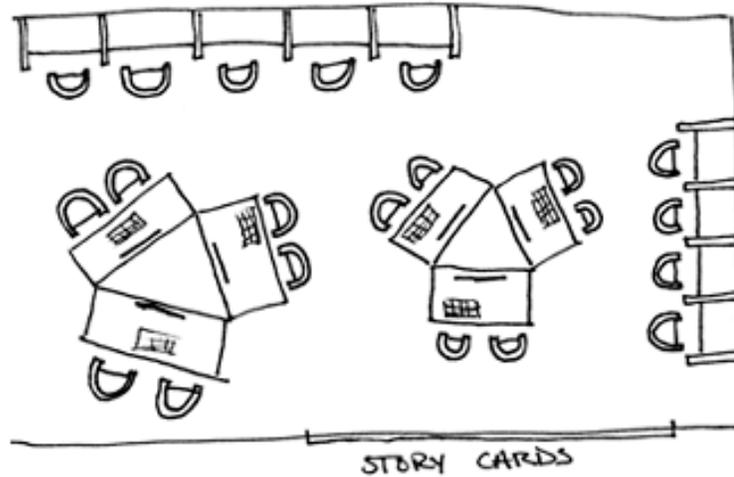


Figura 2.3 - Distribución del Espacio de Trabajo en XP

- **Equipo Completo:** en el equipo de trabajo se debe incluir individuos con todas las habilidades y perspectivas necesarias para que el proyecto tenga éxito.
- **Espacio Informativo:** debe existir un área del espacio de trabajo dedicado al estado del mismo. Una persona ajena al proyecto debería poder observar dicho área y tener una idea general de los avances del proyecto. Esta práctica puede consistir en colocar las historias en un lugar visible ordenadas convenientemente para que reflejen los avances generales. En la Figura 2.4 se muestra una representación ilustrativa de cómo pudiera ser este tipo de espacios informativos.

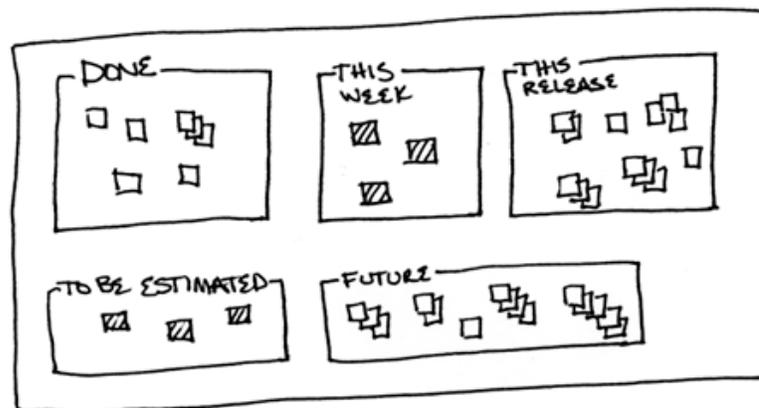
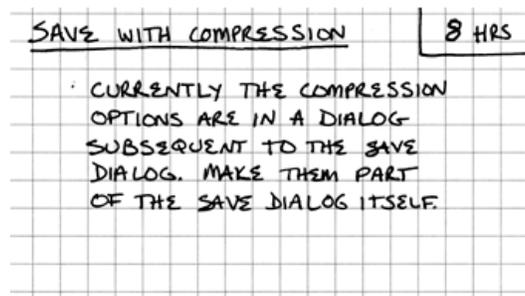


Figura 2.4 - Modelo de Espacio Informativo en XP (Kent Beck, 2005)

- **Trabajo Energizado:** sólo se deben trabajar tantas horas como se pueda mantener ser productivo y únicamente tantas horas como se pueda sostener. El desarrollo de software es una tarea que requiere una mente preparada, descansada y relajada.
- **Programación en Pares:** codificar todos los programas con parejas trabajando sobre una misma computadora. La programación en pares es un diálogo entre dos personas simultáneamente programando (analizando, diseñando y comprobando) e intentado programar mejor. La programación en pareja permite que se mantenga el enfoque en la tarea que se hace; las lluvias de ideas refinan el sistema; se mantiene claridad en las ideas; cuando uno de los programadores se atasca el otro puede tomar la iniciativa disminuyendo la frustración. Los pares son rotados frecuentemente tras ciertos períodos de tiempo. Puede ser una práctica agotadora pero a la vez gratificante. De esta manera todos los miembros del equipo llegan a estar involucrados en todo el proceso de desarrollo.
- **Emparejamiento y Espacio Personal:** el espacio y confort personal debe ser respetado por ambos miembros durante la programación en pares. La higiene personal y la salud son temas importantes en la programación en pares. Es importante respetar todas las diferencias personales.
- **Historias:** planificar actividades en función de funcionalidades visibles para el usuario. Tan pronto una historia es escrita, intentar estimar los esfuerzos de desarrollo necesarios para implementarla. La estimación temprana es una diferencia clave sobre otras prácticas de requerimientos. Dicha estimación permite que las perspectivas técnicas y de negocio interactúen, lo que permite crear valor agregado al proyecto más rápido. Las historias deben ser colocadas en el espacio informativo para conocimiento de todos los miembros del equipo de trabajo. La Figura 2.5 representa un ejemplo de una historia de usuario, describiendo un título general, un tiempo estimado de entrega y una descripción detallada.



SAVE WITH COMPRESSION | 8 HRS

CURRENTLY THE COMPRESSION
OPTIONS ARE IN A DIALOG
SUBSEQUENT TO THE SAVE
DIALOG. MAKE THEM PART
OF THE SAVE DIALOG ITSELF.

Figura 2.5 - Ejemplo de Historia de Usuario (Kent Beck, 2005)

- **Ciclos Semanales:** planificar el trabajo semana a semana. Para ello es prudente establecer reuniones al comienzo de cada semana. Durante esta reunión se revisan los progresos hasta la fecha, incluyendo cómo los progresos alcanzados se comparan con los esperados. Los usuarios seleccionan las historias que desean implementar en la nueva semana. Las historias son descompuestas en tareas para que los miembros de los equipos se comprometan y estimen los tiempos de las mismas. En un enfoque de desarrollo orientado a pruebas, la semana se debe comenzar escribiendo pruebas automatizadas que corran cuando las historias se hayan completado. Luego de escritas las pruebas, el resto de la semana se completan las historias y se hace que las pruebas sean válidas.
- **Ciclos Trimestrales:** planificar ciclos trimestrales uno a uno. Cada trimestre se debe evaluar el equipo, el proyecto, sus progresos y los avances alcanzados con respecto a las metas globales. Para la planificación trimestral se identifican los cuellos de botella, especialmente aquellos controlados fuera del equipo y se inician las reparaciones necesarias en ese sentido. Adicionalmente se planifica el tema o los temas para el siguiente ciclo trimestral. Se selecciona un conjunto de historias referente a los temas seleccionados y se hace una imagen general del proyecto.
- **Holgura:** en cualquier plan deben incluirse tareas menores que puedan ser descartadas si la planificación se atrasa de cualquier forma. Siempre pueden agregarse más historias después y entregar más de lo prometido. Fomentar dentro del equipo un sentido de responsabilidad para solo suscribir las tareas en las que sientan confianza suficiente para entregar en los tiempos pautados. Estos compromisos eliminan desperdicios. La comunicación honesta baja las tensiones y aumenta la credibilidad del equipo.
- **Compilación de 10 minutos:** automáticamente compilar el sistema completo y ejecutar todas las pruebas automatizadas en diez minutos.
- **Integración Continua:** integrar y comprobar cambios como máximo cada dos horas. La programación en equipos no tiene como fin dividir el problema. El proceso de integración es impredecible, pero puede tardar más incluso que el proceso de programación. Mientras más tiempo se espere para integrar, más costosos e impredecibles pueden resultar las integraciones.
- **Comprobación al Principio de Programación:** antes de comenzar a programar escribir casos de pruebas automáticas que fallen para cualquier código.

- **Diseño Incremental:** invertir en el diseño del sistema todos los días. Intentar mantener el diseño del sistema en excelentes condiciones para las necesidades del día. Trabajar gradual pero persistentemente en alinear el diseño con la comprensión del problema. Los equipos XP son confidentes en adaptar sus diseños a futuros requerimientos. Sin la atención diaria el diseño del sistema puede resultar perjudicado a mediano plazo haciendo los cambios mucho más costosos. El diseño incremental no debe ser minimizado por la carrera, sino que debe ser mantenido acorde con las necesidades del sistema en un momento dado. Como resultado se espera que los sistemas que empiezan como pequeños proyectos puedan crecer como sea necesario sin requerir costos exorbitantes en tiempo y dinero.

2.1.5 Método SCRUM

Scrum es un marco de trabajo para el desarrollo y mantenimiento de productos complejos que busca obtener productos del máximo valor posible de forma productiva y creativa. Los procesos de desarrollo de software no siempre generan las mismas salidas aun partiendo de las mismas entradas, Scrum se basa en esta afirmación de manera empírica, promoviendo el uso de la observación y la experimentación para inspeccionar y adaptar. Esto permite a los equipos de desarrollo observar regularmente la efectividad de sus prácticas de desarrollo y hacer cambios según sean requeridos.

A pesar que los procesos de desarrollo de software cuentan ya con más de 60 años, existen altas posibilidades de que un proyecto de mediano a gran tamaño falle. Por fortuna la industria finalmente comprende esta situación y se han empezado a tomar medidas para responder a esta problemática. La evidencia muestra cómo las prácticas de métodos ágiles tales como Scrum, están liderando los nuevos tiempos.

En 2010, se hizo pública la Guía Scrum de manera gratuita, en el dominio Scrum.org. Esta guía de apenas 15 páginas representa las reglas oficiales de Scrum, y es mantenida por sus creadores originales, Ken Schwaber y Jeff Sutherland. Sin embargo, a pesar de solo tener 15 páginas, puede resultar difícil de dominar. (Hundhausen, 2012)

El método Scrum consiste en los equipos Scrum, sus roles asociados, eventos, artefactos y reglas asociadas. Cada componente tiene un propósito específico dentro del marco de trabajo y es esencial para el éxito de Scrum y para su uso. Son las reglas de Scrum las que vinculan a los eventos, roles y artefactos, rigiendo las relaciones e interacciones entre ellos.

Scrum se fundamenta en la teoría empírica de control de procesos o empirismo. El empirismo asegura que el conocimiento proviene de la experiencia y de tomar decisiones a partir de lo que se conoce. Scrum emplea una aproximación iterativa e incremental para optimizar la predictibilidad y controlar el riesgo. (Ken Schwaber, 2011)

2.1.5.1 Pilares de SCRUM

Scrum se fundamenta sobre tres pilares que soportan toda la implementación del control empírico de procesos: transparencia, inspección y adaptación. (Ken Schwaber, 2011)

- **Transparencia:** los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de modo que los observadores compartan un entendimiento común de lo que se está viendo.
- **Inspección:** los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones no deseables. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando son realizadas de forma diligente por inspectores externos en el mismo lugar de trabajo.
- **Adaptación:** si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o material que está siendo procesado deben ser ajustados. Dicho ajuste debe ser realizado cuanto antes para minimizar desviaciones mayores. Para ello Scrum prescribe formalmente cuatro oportunidades para la inspección y adaptación:
 - Reunión de Planificación del Sprint.
 - Scrum Diario.
 - Revisión del Sprint.
 - Retrospectiva del Sprint.

2.1.5.2 Equipo SCRUM

Un equipo Scrum consta de un Dueño de Producto, un Equipo de Desarrollo y un Scrum Master. Los equipos Scrum son auto-organizados y multifuncionales. Los equipos auto-organizados eligen la mejor forma de llevar a cabo su trabajo, en lugar de ser dirigidos por otros externos al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otros que no son parte del equipo. El método de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los equipos Scrum entregan productos de forma iterativa e incremental maximizando las oportunidades para obtener retroalimentación. Las entregas incrementales de producto “Hecho” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto. (Ken Schwaber, 2011)

- **Dueño de Producto:** es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. Es el único responsable de gestionar la Pila de Producto. Es una única persona, no un comité. Las decisiones del Dueño de Producto se reflejan en el contenido y en la priorización de la Pila de Producto.
- **Equipo de Desarrollo:** son los profesionales que desempeñan el trabajo de entregar un incremento del producto “Hecho”, que sea potencialmente utilizable al final de cada Sprint. Sólo los miembros del Equipo de Desarrollo participan en la creación del Incremento. Los equipos de desarrollo tienen poder de auto-organizarse y auto-gestionar su propio trabajo.
- **Scrum Master:** es el responsable de asegurar que Scrum es entendido y llevado a cabo. Los Scrum Masters se aseguran de que el equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

2.1.5.3 Eventos de SCRUM

En Scrum existen eventos prescritos, que tienen como objetivo crear regularidad y reducir la necesidad de reuniones no definidas en Scrum. Se utilizan eventos en la forma de bloques de tiempo para asegurar que todos tengan una duración máxima. De esta manera se garantiza que se emplea un tiempo adecuado en planificación reduciendo el desperdicio en el proceso.

Además del evento denominado Sprint, que es un contenedor del resto de los eventos, cada uno de los eventos Scrum constituye una oportunidad para la inspección y adaptación de

algún aspecto. Estos eventos están diseñados para habilitar la transparencia e inspección. (Ken Schwaber, 2011)

- **Sprint:** es el núcleo del método Scrum. El Sprint, es un bloque de tiempo, generalmente de un mes o menos durante el cual se crea un incremento de producto “Hecho”, utilizable y potencialmente entregable. La duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza al finalizar el anterior. Cada Sprint consta a su vez una Reunión de Planificación del Sprint, los Scrums Diarios, el trabajo de desarrollo, la Revisión del Sprint y la Retrospectiva del Sprint.
- **Reunión de Planificación de Sprint:** el trabajo que se va a realizar durante el Sprint es planificado en esta reunión de planificación. En ella, se crea un plan mediante el trabajo colaborativo del Equipo Scrum al completo. La reunión está restringida a una duración máxima de ocho horas para un Sprint de un mes; para Sprints más cortos, el evento es proporcionalmente más corto. Esta reunión consta habitualmente de dos partes. Ambas partes de igual duración buscan responder a las interrogantes ¿Qué será entregado en el incremento resultante del Sprint que comienza? y ¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento?
- **Scrum Diario:** es una reunión que tiene una duración máxima de un bloque de tiempo de 15 minutos, para que el equipo de desarrollo sincronice sus actividades y cree un plan de actividades para las próximas 24 horas. Para ello se revisa el trabajo avanzado desde el último Scrum Diario y se hace una predicción del trabajo que podría ser completado antes del día siguiente.
- **Revisión del Sprint:** al final de cada Sprint se lleva a cabo una Revisión de Sprint, para inspeccionar el Incremento y adaptar la Pila del Producto si fuese necesario. Durante esta revisión, el Equipo Scrum y los interesados colaboran acerca de lo que se ha hecho para determinar qué podría hacerse. Es una reunión informal cuyo objetivo es facilitar la retroalimentación y fomentar la colaboración. El resultado de esta reunión es una Pila de Producto revisada, que define los elementos posibles para el próximo Sprint.
- **Retrospectiva de Sprint:** es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo, y crear un plan de mejoras que sean abordadas durante el siguiente Sprint. Esta retrospectiva tiene lugar después de la Revisión del Sprint y antes de la siguiente Reunión de Planificación de Sprint. Está restringida a un bloque de tiempo máximo de tres horas

para Sprints de un mes. Tiene como propósitos, inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas; identificar y ordenar los elementos más importantes que fueron bien, y posibles mejores; y por último crear un plan para implementar las mejoras para la forma en que el Equipo Scrum desempeña su trabajo. Para el final de la misma, el equipo Scrum debe haber identificado todas las mejoras que implementará en un próximo Sprint.

2.1.5.4 Artefactos de SCRUM

Los artefactos de Scrum representan trabajo o valor en diferentes formas que contribuyen a proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos Scrum están diseñados para maximizar la transparencia de la información clave necesaria para asegurar que los Equipos Scrum tengan éxito al entregar un incremento “Hecho”. (Ken Schwaber, 2011)

- **Pila de Producto:** es una lista ordenada de todo lo que podría ser necesario en el producto y constituye la única fuente de requerimientos para cualquier cambio a realizarse en el producto. Esta pila nunca está completa, evoluciona a medida que el producto y el entorno en el que será usado también lo hacen. Es una pila dinámica, que cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.
- **Pila de Sprint:** es el conjunto de elementos de la pila de producto que son seleccionados para ser llevados a cabo en un Sprint, más un plan para entregar el Incremento del producto y lograr el Objetivo del Sprint. Define el trabajo que el Equipo de Desarrollo llevará a cabo para convertir elementos de la Pila de Producto en un incremento “Hecho”. Contiene suficientes detalles como para que los cambios en el progreso puedan ser entendidos en el Scrum Diario. El equipo de desarrollo es el encargado de cambiar la Pila de Sprint durante un Sprint.
- **Incremento:** es la suma de todos los elementos de la Pila de Producto completados durante un Sprint y durante todos los Sprints previos. Al final de cada Sprint, la nueva parte del Incremento debe estar utilizable y cumplir la definición de “Hecho” establecida por el Equipo Scrum. Debe estar en condiciones de ser utilizada sin importar si el Dueño de Producto decide liberarla o no.

2.1.5.5 Definición de HECHO en SCRUM

Como se pudo notar en las descripciones anteriores en SCRUM se utiliza frecuentemente el término “Hecho” para definir un estado específico de los incrementos. Los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado para asegurar la transparencia.

El objetivo de cada Sprint es entregar incrementos de funcionalidad potencialmente utilizable, que se ajusten a la definición de “Hecho” actual del Equipo Scrum. Con el transcurrir del tiempo se espera que los equipos maduren y con ello se amplíe su definición de “Hecho” para incluir criterios más rigurosos para una calidad mayor. (Ken Schwaber, 2011)

2.2 Ventajas y Desventajas de los Métodos Ágiles

A continuación, en la tabla 2.1, se presentan de manera ilustrativa las ventajas de los métodos de desarrollo ágiles previamente descritos.

Tabla 2.1 - Ventajas de los Métodos de Desarrollo Ágiles (Asif Irshad Khan, 2011)

Método por Prototipos	Método RAD	Método XP	Método SCRUM
Desarrollo más rápido implica una entrega más rápida del producto, lo que se a su vez implica una reducción en costos.	El tiempo de entrega es mucho menor.	El método XP se adapta perfectamente a proyectos de pequeña a mediana envergadura. Involucra en el proceso de desarrollo a todas las partes, lo que se refleja en la calidad del producto final.	Proporciona un foro abierto, en el que todo el mundo conoce las responsabilidades del resto del equipo.
Puede integrarse con otros métodos como el método en cascada para producir resultados efectivos.	Los desarrollos rápidos permiten ahorrar costos y tiempo.	Se enfoca en la buena cohesión de un equipo.	Se enfoca en la comunicación del equipo, espíritu de equipo y solidaridad.
Existe un mayor y mejor involucramiento de los usuarios en el desarrollo.	Existe una mayor productividad cuando son equipos de pocas personas y poco tiempo.	Enfatiza el producto final.	Demostraciones frecuentes para tempranas retroalimentaciones con los usuarios involucrados.
	El progreso puede ser medible.	El enfoque basado en pruebas de los requerimientos es garantía de calidad.	Suele ser principalmente útil para proyectos rápidamente cambiantes o nuevos.

A continuación, en la tabla 2.2, se presentan de manera ilustrativa las desventajas de los métodos de desarrollo ágiles previamente descritos.

Tabla 2.2 - Desventajas de los Métodos de Desarrollo Ágiles (Asif Irshad Khan, 2011)

Método por Prototipos	Método RAD	Método XP	Método SCRUM
Carece de documentación de resultados, esto implica mayores costes en actualización y en soporte de la aplicación.	Solo es aplicable correctamente cuando se conocen bien y de antemano todos los requerimientos antes de iniciar cada ciclo de desarrollo.	Dificultad en los proyectos grandes donde la documentación es una parte esencial del mismo.	La toma de decisiones queda enteramente en las manos de los equipos.
Frecuentemente los resultados de este método son de un software de pobre calidad.	Requiere implicación de todos los usuarios a lo largo del ciclo de vida.	Requiere experiencia y habilidades para no degenerar en una simplificación basada en codificar y refactorizar.	Para entregar un proyecto a tiempo se requiere tener en el equipo únicamente miembros con experiencia.
Al enfocarse mayormente en experimentar con los requerimientos de los usuarios, puede existir un resultado basado en una mala comprensión de los mismos.	Solo es adecuado para proyectos que requieran menores tiempos de desarrollo.	Carece de diseño de documentación; adicionalmente la programación en pares resulta costosa.	El desarrollo del proyecto puede resultar enormemente afectado si alguno de los miembros del equipo abandona el mismo durante el ciclo de vida de desarrollo.
Se ahorra en costos y tiempos siempre y cuando los prototipos puedan ser reutilizados.	El producto final puede perder su competitividad por carencias en las funcionalidades internas y por ende ser de una baja calidad.	Los métodos XP no proporcionan lineamientos en la forma de reunir información. No existe formalmente un plan, medida o programa de gestión de calidad.	La existencia de miembros sin compromisos puede conllevar al fracaso del proyecto.
	El éxito depende de forma crítica en la existencia de desarrolladores bien calificados dentro del equipo.		Requiere un cierto nivel de entrenamiento para todos sus usuarios, lo que se traduce en un mayor costo del proyecto.

Capítulo III. Desarrollo de la Aplicación Web

En este capítulo se presenta la descripción del proceso de desarrollo de la aplicación Web para la gestión de las actividades de Seminarios y TEG de la Escuela de Computación, llevado a cabo utilizando el Método XP, del cual se seleccionaron las prácticas aplicables en el contexto actual.

3.1 Análisis de Métodos Ágiles de Desarrollo

Para desarrollar una aplicación Web que permita gestionar las actividades de Seminarios y TEG de la Escuela de Computación de la Facultad de Ciencias, se consideraron los requerimientos funcionales que debía cumplir para satisfacer las necesidades de los usuarios que aplican métodos ágiles de desarrollo de software, en la medida de lo posible.

Los métodos de desarrollo pesados no fueron considerados para la aplicación Web a desarrollar, dado que el contexto en el que son utilizados normalmente difiere del contexto académico en el que se desarrolla un Seminario o TEG. Adicionalmente, su concepción de diseño anticipatorio se contrapone a la filosofía predominante entre los Seminarios y TEG de la Escuela de Computación, en los que abundan diseños emergentes.

Una vez descritos detalladamente los métodos ágiles de desarrollo más utilizados a lo largo del capítulo anterior, resultó posible identificar las funcionalidades básicas que debía cumplir esta aplicación para la gestión de las actividades de desarrollo de Seminarios y TEG como fue planteada inicialmente.

Al analizar las teorías de cada uno de los métodos ágiles estudiados, se descartó el método de desarrollo por prototipos para las funcionalidades de la aplicación, ya que el mismo carece de documentación alguna para todo el ciclo de vida de desarrollo; además, este método posee un enfoque de experimentación con los requerimientos de los usuarios, el cual no es el más adecuado para proyectos de Seminarios y TEG, donde se espera una frecuente intervención de los profesores tutores.

Con los métodos restantes fue posible identificar que tanto el método RAD, como SCRUM y XP, manejan de alguna forma, listas de tareas que son elaboradas en conjunto con los usuarios para darles valor y prioridad antes o durante cada iteración. Esta premisa resultó ser

fundamental en el desarrollo de una interfaz de usuario única que se adapte a los tres métodos de forma transparente para el usuario.

Por otra parte, se evaluó una muestra aleatoria de 24 proyectos de TEG de la Escuela de Computación (tomada de la aplicación BUSCONEST), para conocer si los métodos ágiles previamente considerados, representan una parte significativa del total de los proyectos desarrollados en los últimos 5 años. De esta muestra se obtiene un gráfico que se ilustra en la figura 3.1, en el cuál se refleja el incremento en la utilización de métodos ágiles de desarrollo en los últimos tiempos.

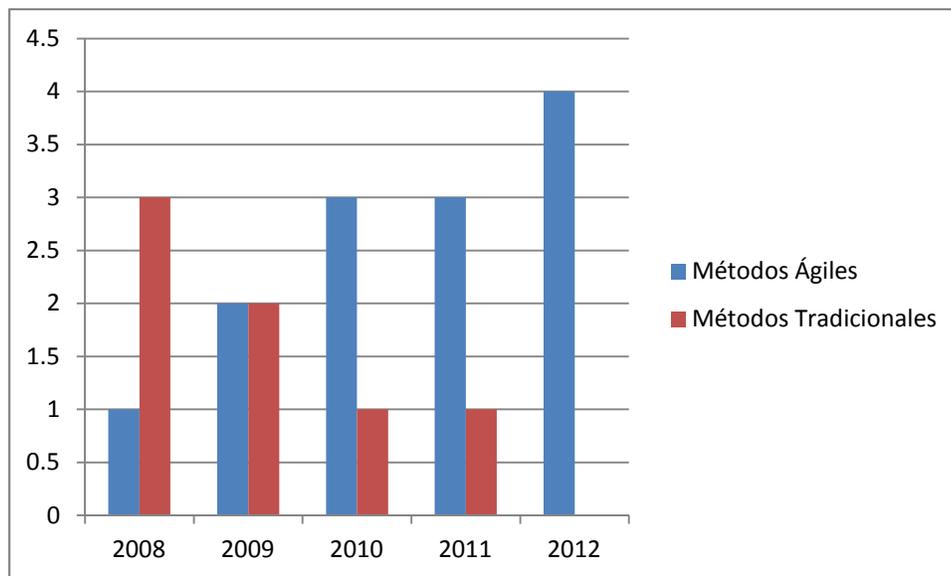


Figura 3.1 – Gráfico Comparativo del uso de los tipos de métodos de desarrollo en la Escuela de Computación.

3.2 Prácticas

Para la implementación del método XP en el desarrollo de un proyecto no es necesario utilizar todas las prácticas, por lo cual se seleccionaron las consideradas más relevantes y a su vez aplicables en el desarrollo de este proyecto, tomando en cuenta la realidad del mismo y el contexto en el que será desarrollado:

- **Programación en Pares:** para cada sesión los dos programadores trabajaron en un único equipo de forma conjunta y alternando el uso del teclado en turnos de dos horas; de esta manera se espera mantener en vigor el trabajo energizado por el resto de la sesión, dado que por un lado se evita la monotonía y por otro, los integrantes del par deben mantener la concentración al máximo para no entorpecer los procesos de desarrollo.

- **Espacio de Trabajo Conjunto:** se designaron como espacios de trabajo las residencias de cada programador; en cada una se dispuso de espacios conjuntos y suficientemente cómodos que permitieron a su vez la aplicación de otras prácticas.
- **Espacio Informativo:** en cada espacio de trabajo existió una cartelera en la que se colocaron las historias de usuario, dispuestas siguiendo las recomendaciones diagramadas descritas en capítulos previos para reflejar los avances generales del proyecto.
- **Trabajo Energizado:** no se trabajó en el desarrollo más de las horas establecidas por un horario laboral normal; adicionalmente para este desarrollo los programadores asumieron el compromiso de guardar suficientes horas de descanso previas a iniciar el desarrollo para no afectar la productividad.
- **Historias:** los requerimientos fueron plasmados en modelos de historias que permitieron estimar los esfuerzos de desarrollo requeridos para implementarlas. Las historias serán colocadas después de creadas en el espacio informativo designado. Todas las historias realizadas a lo largo del desarrollo serán digitalizadas y luego agregados al presente trabajo. En la siguiente tabla se ilustra el formato de Historia de Usuario definido para efectos de este proyecto.

Tabla 3.1 – Formato de Historia de Usuario utilizado en cada iteración del ciclo de desarrollo.

Título:	
Duración:	
Fecha de Inicio:	
Fecha de Culminación:	
Descripción	

- **Ciclos Semanales:** dado que se trata de un proyecto de mediano a largo alcance con un único par de programadores, se consideró que elaborar ciclos semanales era más perjudicial que beneficioso por lo corto de los períodos de desarrollo; en su lugar se realizaron ciclos quincenales para planificar el trabajo de cada quince días. Estas reuniones fueron pautadas con los tutores para seleccionar las historias de cada nuevo ciclo. Los resúmenes de estas reuniones quincenales serán digitalizados y agregados posteriormente al presente trabajo.
- **Ciclos Trimestrales:** se realizó un único ciclo trimestral en el que se establecieron, en conjunto con los tutores del presente trabajo, las metas globales así como el conjunto de historias globales que delimitan el alcance entero del desarrollo para obtener una imagen

general del proyecto. El resumen de esta reunión de planificación fue digitalizado y posteriormente agregado al presente trabajo.

- **Holgura:** se introdujeron algunas historias en cada ciclo quincenal que podían ser descartadas o pospuestas, en caso de que no poder ser entregadas en el tiempo establecido.
- **Integración Continua:** se utilizó el manejador de versiones GIT para llevar a cabo la integración continua de los avances en el desarrollo del proyecto.
- **Diseño Incremental:** los días viernes de cada semana del ciclo trimestral fueron utilizados principalmente en la refactorización de la aplicación para mantener su capacidad de evolucionar en conjunto con el desarrollo.

3.3 Planificación de Ciclo Trimestral

En esta sección se describen los alcances, lineamientos y requerimientos extraídos de la primera planificación trimestral tal y como fueron concebidos en su momento. Durante esta planificación se estableció que la aplicación Web debía permitir gestionar todos los procesos involucrados en la realización de Seminarios y Trabajos Especiales de Grado de la Escuela de Computación. Adicionalmente debía facilitar el proceso de asignación y búsqueda de temas y facilitar a los estudiantes o profesores, la obtención de información sobre sus progresos de manera eficiente y constante, a lo largo del desarrollo de sus proyectos.

A partir de estas premisas establecidas se definieron tres módulos fundamentales que debía contener la aplicación. En la figura 3.2 se representan dentro de las cajas los módulos y fuera de ellas los actores, que fueron identificados durante la reunión de planificación trimestral y que posteriormente son descritos en detalle.

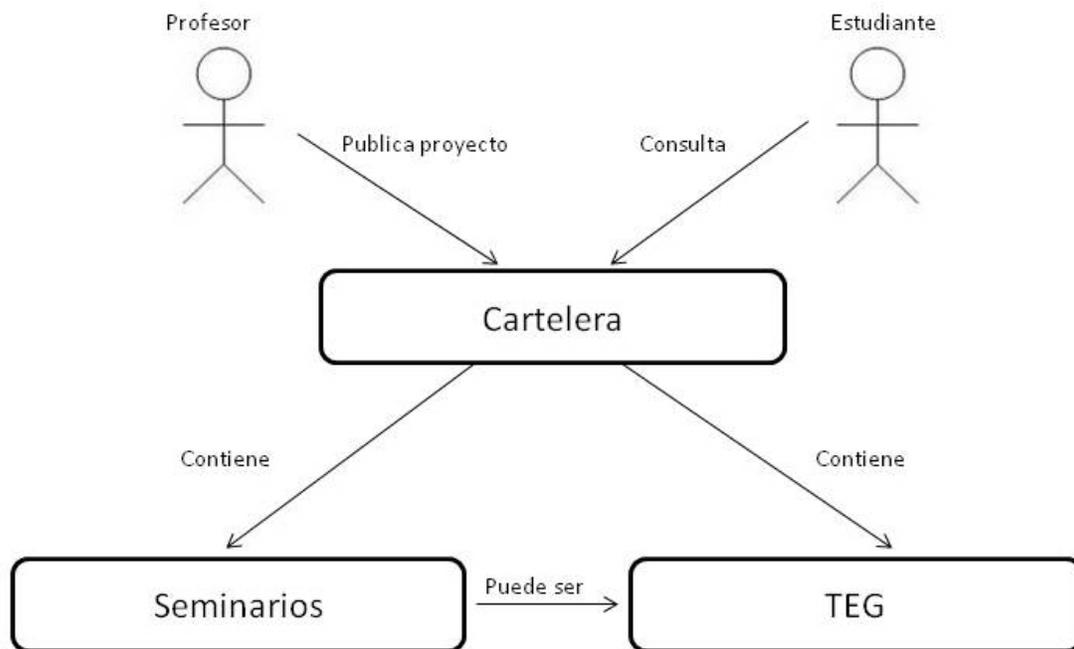


Figura 3.2 - Diagrama de módulos identificados en planificación trimestral.

3.3.1 Cartelera de Ofertas de Temas

El sistema debe permitir a los profesores gestionar la publicación de ofertas de temas para Seminarios y Trabajos Especiales de Grados. Para ello debe permitir publicar, eliminar y modificar publicaciones de temas. A su vez, los estudiantes podrán postularse a través del sistema como candidatos para estos temas. Para esta primera versión se contempla que por cada candidato postulado el profesor deberá recibir un correo de notificación para su aceptación.

Adicionalmente a las tres partes descritas se contempla que el sistema debe ser capaz de mostrar al profesor un panel global que permita ubicar rápidamente un tema por nombre o estudiantes, y en el mismo panel deben representarse gráficamente los avances de todos sus estudiantes, reflejando entre otros valores, los promedios de avances y total de estudiantes.

3.3.2 Trabajo de Seminario

El trabajo de Seminario se compone normalmente de dos partes; un documento de investigación y una presentación. Para gestionar correctamente los progresos en Seminarios, la aplicación debe permitir al profesor definir la estructura del documento de investigación y la

presentación. A partir de esta estructura el estudiante tiene la posibilidad de obtener retroalimentación, mediante la carga de archivos en forma de iteraciones; para una iteración, luego de que el estudiante ha subido los archivos, el profesor puede descargarlos, para accederlos y añadir comentarios con las revisiones pertinentes. Adicionalmente, al momento de cargarlo puede agregar comentarios por cada revisión y aprobar secciones de la estructura de contenidos previamente creada.

Por último, este módulo debe permitir tanto al estudiante como al profesor, conocer los avances alcanzados en el Seminario en cualquier momento.

3.3.3 Trabajo Especial de Grado

El Trabajo Especial de Grado (TEG), se compone normalmente de tres partes; un documento de investigación, una presentación y un producto. Al igual que en el módulo anterior, la aplicación debe permitir al profesor definir la estructura tanto del documento de investigación como de la presentación; del mismo modo, también obtener retroalimentación de los progresos a través de iteradas cargas de archivos en las que los profesores pueden añadir comentarios y aprobar secciones de la estructura de contenidos.

Como función agregada, este módulo debe permitir la gestión de los progresos en el desarrollo de productos mediante la creación de actividades. Para ello, se definen tres listas de actividades: tareas pendientes, tareas en desarrollo, tareas culminadas. Al momento de creación de una actividad, es agregada a la lista de Tareas Pendientes, y la misma contiene un título descriptivo, información detallada sobre lo que implica y una fecha estimada de entrega. Las actividades pueden ser cambiadas de listas a lo largo de su ciclo de vida para reflejar su estado, y a su vez pueden contener archivos que son agregados posteriormente por el estudiante durante o como comentarios que son agregados por el profesor.

Por último, el módulo TEG, al igual que el anterior, debe permitir tanto al estudiante como al profesor, conocer los avances alcanzados en cualquier momento.

En la figura 3.3 se representa un mapa conceptual en el que se pueden observar los requerimientos funcionales identificados al momento de realizar la planificación trimestral. En el centro del diagrama, el nombre del sistema; alrededor, las cajas color naranja representan los módulos y las amarillas o grises funcionalidades identificadas para estos módulos, durante la fase de planificación.

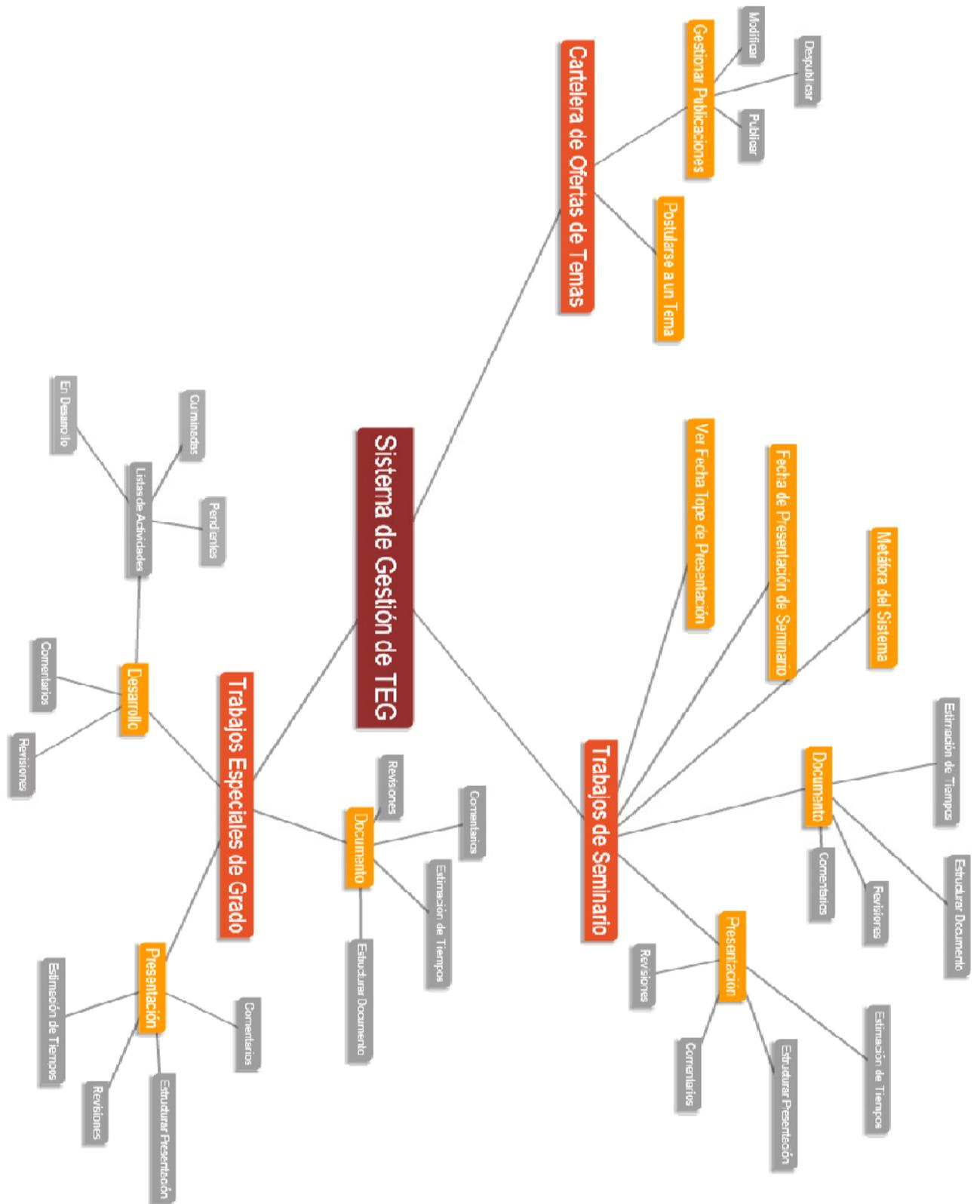


Figura 3.3 - Mapa Conceptual de la visión global del Sistema concebida para la Planificación del Ciclo Trimestral

3.4 Iteraciones

A continuación se describen las distintas iteraciones llevadas a cabo mediante la descripción de las actividades completadas en cada una.

3.4.1 Iteración 1 - (16-01-2013 al 31-01-2013)

Durante esta iteración se concibieron las primeras ideas de diseño de interfaz de usuario (IU) así como la idea general del sistema; se realizó un diseño de base de datos preliminar y se comenzó el desarrollo de las funcionalidades de autenticación y autorización de la aplicación.

3.4.1.1 Planificación

En la tabla 3.2 se detalla la planificación efectuada al inicio de esta iteración.

Tabla 3.2 – Planificación de Iteración 1.

Iteración 1		
Descripción	Elaboración de ideas conceptuales de la aplicación e instalación y configuración del entorno de desarrollo en Ruby on Rails.	
Período	16-01-2013 al 31-01-2013	
Fecha	Historia	Tipo
16-01-2013	Construcción de Wireframes.	Nuevo
18-01-2013	Diseño e implementación de base de datos.	Nuevo
21-01-2013	Desarrollo de métodos de autenticación.	Nuevo

3.4.1.2 Codificación

En la presente iteración se procedió a instalar todo lo necesario para establecer los ambientes de desarrollo en los equipos de trabajo, entre estas herramientas, se encuentran: MySQL, RubyMine, MySQL Workbench, RVM, Ruby on Rails y las dependencias necesarias. Los ambientes de desarrollo fueron configurados sobre una plataforma Linux (Ubuntu 12).

Una vez establecido el ambiente de desarrollo en cada equipo, se procedió a crear los esquemas de base de datos que serían utilizados a lo largo del ciclo de vida de desarrollo. De igual manera durante esta iteración se crearon las tablas necesarias para las funciones de autenticación y autorización de la aplicación, entre ellas se encuentran: usuarios, estudiantes y roles. Por último, después de la creación de las tablas de base de datos se desarrollaron los métodos de autenticación de usuario en la aplicación. Para ello se crearon los modelos, controladores y vistas correspondientes al usuario y a la sesión con cada uno de los métodos que fueron necesarios.

3.4.1.3 Diseño

Durante esta iteración se construyeron los wireframes correspondientes a las pantallas de la aplicación web en general, que permitirían en futuras iteraciones manejar una idea conceptual del sistema y así identificar nuevos requerimientos.

En las siguientes figuras se ilustran estos prototipos de diseño realizados utilizando la técnica de Wireframing. Esta técnica resultó ser de gran utilidad dado que una vez identificadas las funcionalidades que debían ser incluidas en el sistema, la dificultad se presentó en cómo lograrlo, sin perjudicar la navegación y la usabilidad a través del sistema. Estas maquetas o prototipos permitieron tener una mejor idea de lo que debía hacerse así como una idea de la posible distribución de los elementos de IU.

Las figuras 3.4 y 3.5 corresponden a los Wireframing de autenticación de usuarios, observándose en la figura 3.5 la barra de navegación presente para usuarios autenticados y otros elementos de IU que fueron considerados en su momento.

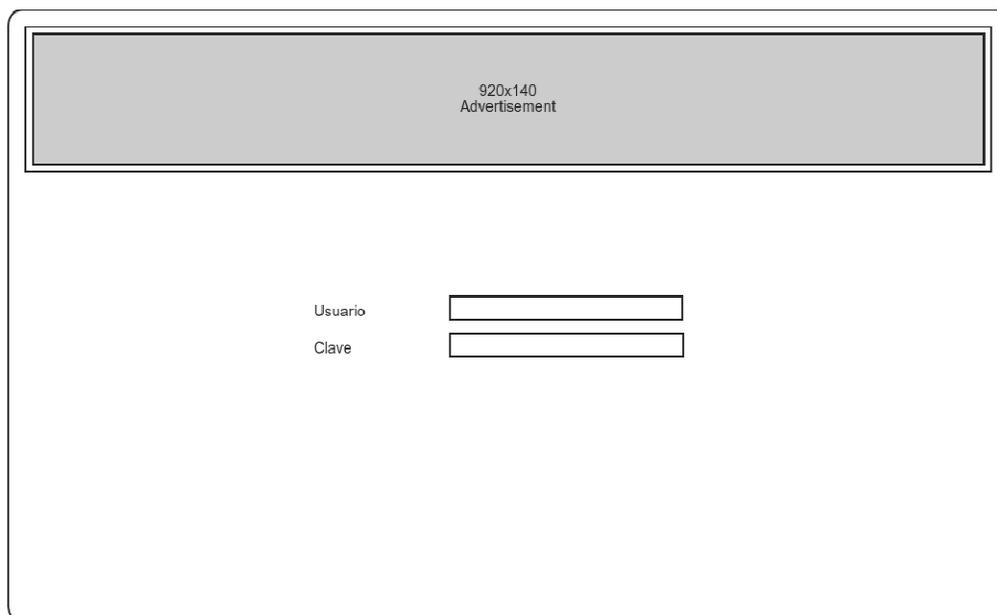


Figura 3.4 – Wireframe de Autenticación.



Figura 3.5 – Wireframe de Home – Profesor Autenticado.

La figura 3.6 corresponde al Wireframing de la pantalla que contiene el listado de estudiantes y sus progresos.

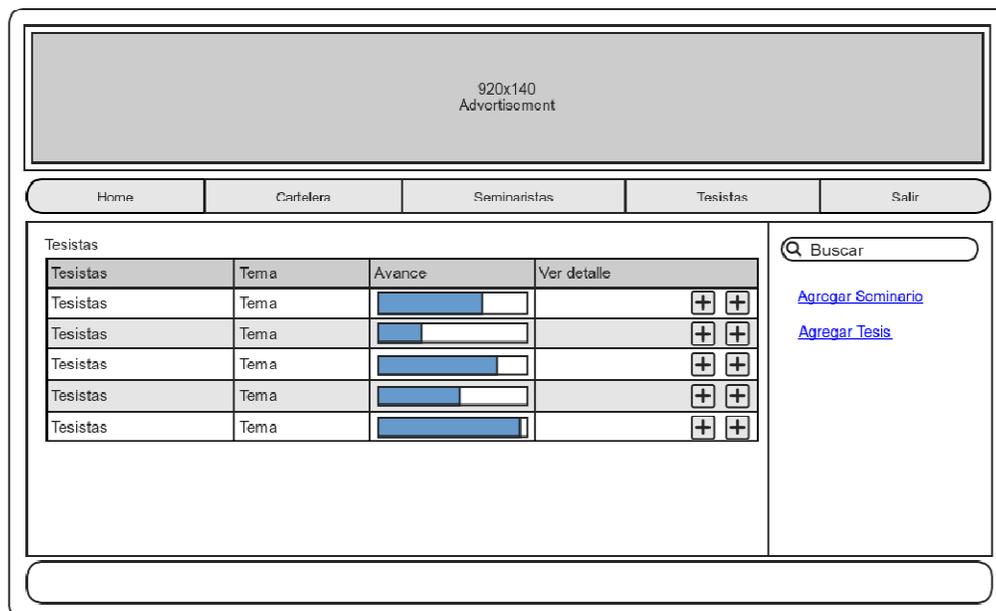


Figura 3.6 – Wireframe de Listado de Estudiantes – Profesor Autenticado.

La figura 3.7 corresponde al wireframe de la pantalla que contiene el listado de iteraciones de documentos pertenecientes a un tema en específico, así como la barra lateral con los controles de aprobación de contenidos disponibles para el profesor autenticado.

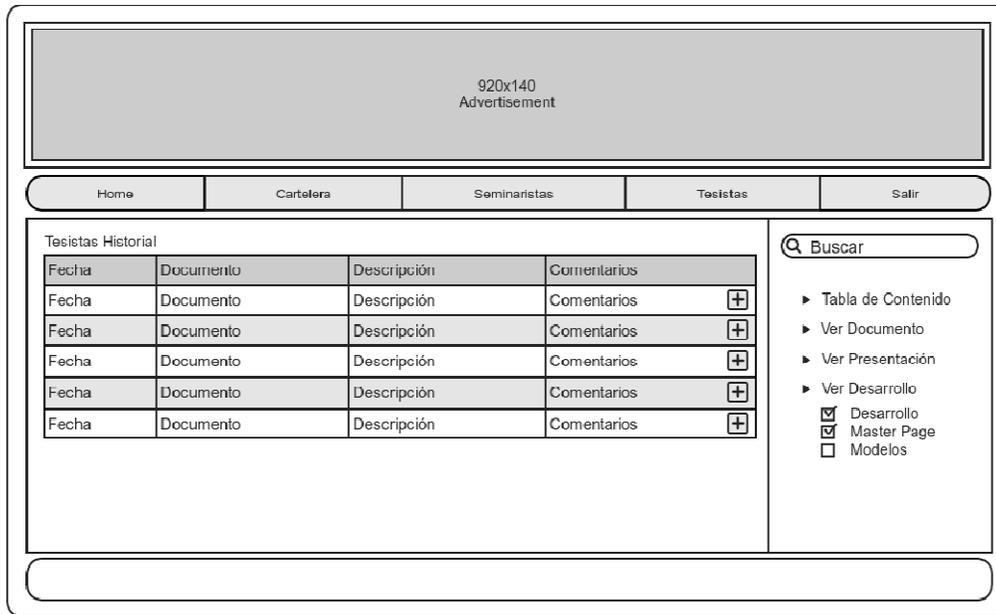


Figura 3.7 – Wireframe de Iteración de Documentos – Profesor Autenticado.

La figura 3.8 corresponde al wireframe de la pantalla de progresos de desarrollo, y se observa la idea de manejar tres paneles con las tareas pendientes para manipular sus estados, así como los atributos que puede tener una tarea en particular.

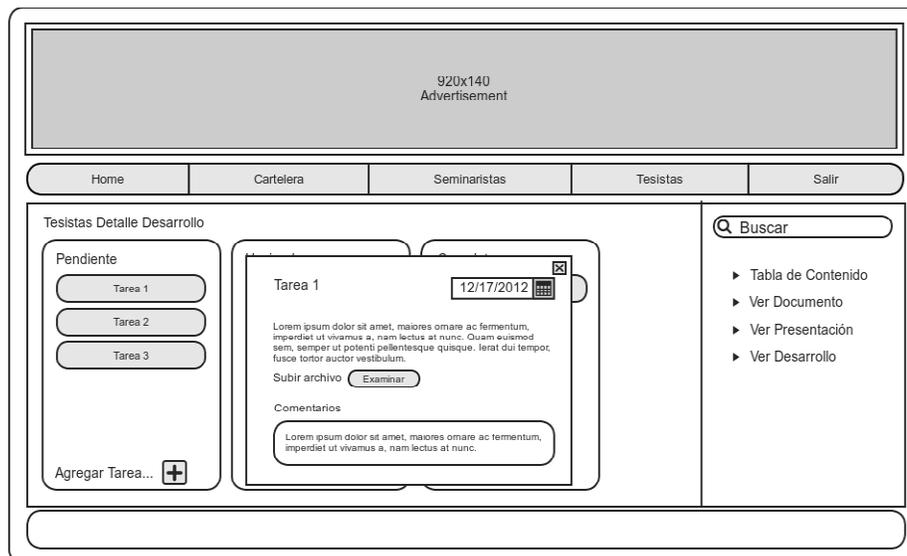


Figura 3.8 – Wireframe de Progresos de Desarrollo – Profesor Autenticado.

La figura 3.9 corresponde al wireframe de la pantalla que contiene el listado de iteraciones de presentaciones pertenecientes a un tema en específico, así como la barra lateral con los controles de aprobación de contenidos disponibles para el profesor autenticado.

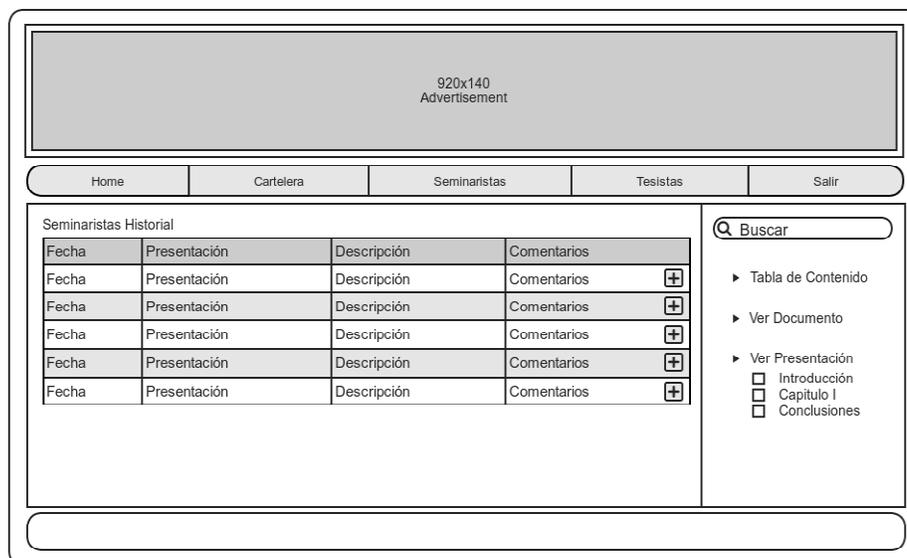


Figura 3.9 – Wireframe de Iteración de Presentaciones – Profesor Autenticado.

3.4.1.4 Historias

A continuación se detallan las historias de usuario correspondientes a esta iteración.

Título:	Construcción de Wireframes.
Duración:	4 horas.
Fecha de Inicio:	16-01-2013
Fecha de Culminación:	16-01-2013
Descripción	Para poder tener una idea más clara de lo que se desea hacer, se deben construir prototipos de diseño a través de wireframes. De esta manera se puede llegar a ilustrar una posible IU sin necesidad de comenzar el proceso de codificación.

Título:	Diseño e Implementación de Base de Datos
Duración:	8 horas
Fecha de Inicio:	18-01-2013
Fecha de Culminación:	20-01-2013
Descripción	Se debe diseñar e implementar la estructura de la base de datos. Esta estructura no es definitiva por lo que debe ser escalable y altamente cohesiva para garantizar que sea fácilmente modificable a lo largo del desarrollo.

Título:	Implementación de Autenticación.
Duración:	24 horas.
Fecha de Inicio:	21-01-2013
Fecha de Culminación:	31-01-2013
Descripción	La aplicación requerirá la funcionalidad de autenticación ya que el contenido no debe ser de dominio público; provisionalmente la aplicación tendrá un formulario de registro de usuarios. Por último se deben crear los mecanismos necesarios para que los usuarios no autenticados no puedan acceder.

3.4.1.5 Pruebas

A continuación se detallan las pruebas llevadas a cabo en esta iteración.

Descripción	Resultado Esperado	Resultado Obtenido
Registrar Usuario	El sistema deberá permitir a los usuarios registrarse a través de un formulario. El sistema deberá ser capaz de identificar que los datos de entrada sean válidos así como la unicidad de los correos electrónicos registrados así como de las cédulas registradas. Los datos del usuario deberán guardarse en la base de datos y los datos de contraseña deberán ser encriptados antes de ser guardados.	Se logró crear exitosamente un usuario. No se pudo crear dos usuarios utilizando el mismo correo electrónico ni la misma cédula de identidad del primer usuario creado.
Autenticar Usuario en la Aplicación.	El sistema deberá permitir a un usuario registrado autenticarse en la aplicación para acceder a contenidos restringidos.	El usuario se autenticó con éxito y el contenido restringido no puede ser visto por usuarios no autenticados.

3.4.2 Iteración 2 - (01-02-2013 al 15-02-2013)

Durante esta iteración se prosigue con la implementación de los métodos de autorización necesarios para manejar diferentes roles, se incorpora la barra de menú lateral a la IU y se realizan las primeras funcionalidades del sistema, implementando la creación de proyectos y por último la cartelera. La cartelera es concebida para permitir a los estudiantes consultar la oferta de temas publicados.

3.4.2.1 Planificación

En la tabla 3.3 se detalla la planificación efectuada al inicio de esta iteración.

Tabla 3.3 – Planificación de Iteración 2.

Iteración 2		
Descripción	Implementación de métodos de autorización y manejo de roles, creación de proyectos y cartelera de temas.	
Período	01-02-2013 al 15-02-2013	
Fecha	Historia	Tipo
01-02-2013	Implementación de autorización.	Nuevo
07-02-2013	Página de inicio para usuarios autenticados.	Modificación
08-02-2013	Menú de opciones laterales de IU.	Nuevo
10-02-2013	Creación de proyectos.	Nuevo
12-02-2013	Cartelera de temas.	Nuevo

3.4.2.2 Codificación

Durante esta iteración se desarrollaron todos los métodos de autorización y manejo de contenido por roles. Para ello fue necesario refactorizar la base de datos y los métodos de autenticación desarrollados durante la Iteración 1, para agregar la información específica de los estudiantes. Se incluyó la tabla estudiantes. Se desarrollaron los métodos para filtrar las acciones por roles.

De igual manera se agregó en la base de datos la tabla de proyectos. Posteriormente, se desarrollaron en la aplicación, el modelo de proyecto, el controlador de proyectos necesarios para la creación y edición de proyectos así como la consulta de los proyectos en cartelera.

3.4.2.3 Diseño

Durante esta iteración se modificaron las páginas de inicio para usuarios autenticados, modificando su contenido por roles. Posteriormente se realizaron las vistas para la cartelera y las correspondientes a la creación, edición y consulta de proyectos.

En las figuras 3.12 y 3.13 se ilustran, la página de inicio de los profesores y de los estudiantes, pudiéndose diferenciar los distintos grados de información y herramientas que poseen de inicio ambos tipos de usuario.



Figura 3.12 – Página de inicio de estudiantes.



Figura 3.13 - Página de Inicio de Profesores

En la figura 3.14 se puede observar la pantalla de creación de proyectos, únicamente disponible para profesores autenticados.

Figura 3.14 - Formulario de creación de proyectos.

En la figura 3.15 se puede ver la pantalla de cartelera; en ella, estudiantes y profesores pueden consultar los proyectos ofertados.

Área	Tipo	Tutor	Título
Aplicaciones con Tecnología Internet	Seminario	Ignacio Diago	Desarrollo de aplicación Web para la gestión de los procesos y actividades de Seminarios y TEG
Redes	Tesis	Ele Acosta	Segundo título
Bases de Datos	Seminario	Ignacio Diago	Lorem ipsum dolor sit amet

Figura 3.15 - Pantalla de Cartelera disponible para estudiantes y profesores.

3.4.2.4 Historias

A continuación se detallan las historias de usuario correspondientes a esta iteración.

Título:	Implementación de Autorización
Duración:	24 horas
Fecha de Inicio:	01-02-2013
Fecha de Culminación:	07-02-2013
Descripción	La aplicación debe permitir la separación de contenidos por roles de usuario. Se crean funcionalidades para la separación de contenidos.

Título:	Home Page para Usuarios Autenticados
Duración:	8 horas
Fecha de Inicio:	07-02-2013
Fecha de Culminación:	07-02-2013
Descripción	Se debe desarrollar la pantalla inicial del sistema para los usuarios autenticados. Los profesores autenticados podrán seleccionar entre otras opciones: seminaristas o tesis, por lo que deben realizarse las integraciones pertinentes en el menú superior.

Título:	Menú de Opciones Laterales
Duración:	4 horas
Fecha de Inicio:	08-02-2013
Fecha de Culminación:	08-02-2013
Descripción	Adicional a la autenticación, la aplicación debe poder permitir la separación de contenidos a través de roles de usuario. Para ello debe implementarse la tabla de roles así como todas las funcionalidades que luego permitan la implementación de contenidos por roles. Adicionalmente se debe colocar el botón que permita a los profesores autenticados crear nuevos proyectos.

Título:	Creación de Proyectos
Duración:	8 horas
Fecha de Inicio:	10-02-2013
Fecha de Culminación:	11-02-2013
Descripción	Se debe desarrollar un formulario que permita al profesor autenticado crear temas, bien sea para seminarios o trabajos especiales de grado. Los proyectos quedarán asociados a un tipo de proyecto, un área de investigación y a un profesor tutor.

Título:	Cartelera
Duración:	16 horas
Fecha de Inicio:	12-02-2013
Fecha de Culminación:	15-02-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un estudiante autenticado pueda consultar toda la oferta de temas disponibles publicados por todos los profesores. Los profesores autenticados podrán visualizar igualmente este contenido; adicionalmente tendrán una sección específica que les permita gestionar sus proyectos publicados.

3.4.2.5 Pruebas

A continuación se detallan las pruebas llevadas a cabo en esta iteración.

Descripción	Resultado Esperado	Resultado Obtenido
Crear Proyecto	El sistema deberá permitir a los usuarios registrados del rol profesor, crear proyectos a través de un formulario. El sistema deberá ser capaz de identificar que los datos de entrada sean válidos y que sólo los profesores tengan permiso de creación de proyectos.	Se logró crear exitosamente un proyecto. Se comprobó que las validaciones de los datos de entrada tanto del lado del cliente como del lado del servidor funcionaron correctamente.
Editar Proyecto	El sistema deberá permitir a los usuarios registrados del rol profesor, editar únicamente sus proyectos a través de un formulario. El sistema deberá ser capaz de identificar que los datos de entrada sean válidos y que sólo los profesores tengan permiso de edición de proyectos y a su vez, únicamente sobre sus propios proyectos.	Se logró editar exitosamente un proyecto. Se comprobó que las validaciones de los datos de entrada tanto del lado del cliente como del lado del servidor funcionaron correctamente. De igual manera se comprobó que un profesor únicamente podía modificar sus propios proyectos.
Consultar Cartelera	El sistema deberá permitir a los usuarios registrados consultar todos los proyectos creados a través de la aplicación en la sección de cartelera. Para ello únicamente es necesario validar que los usuarios estén autenticados sin importar su rol.	Se logró consultar el listado de proyectos existentes como profesor y como estudiante.

3.4.3 Iteración 3 - (15-02-2013 al 28-02-2013)

Durante esta iteración se prosigue con todos los desarrollos referentes a la postulación y asignación de temas; adicionalmente es necesario llevar a cabo un proceso de refactorización de unas funcionalidades previamente desarrolladas para incorporarles las nuevas funcionalidades.

3.4.3.1 Planificación

En la tabla 3.4 se detalla la planificación efectuada al inicio de esta iteración.

Tabla 3.4 – Planificación de Iteración 3.

Iteración 3		
Descripción	Implementación de métodos de postulación y asignación de temas.	
Período	15-02-2013 al 28-02-2013	
Fecha	Historia	Tipo
15-02-2013	Postulación para temas.	Modificación
21-02-2013	Asignación de temas.	Modificación
25-02-2013	Listados de estudiantes.	Nuevo
27-02-2013	Refactorización.	Modificación

3.4.3.2 Codificación

Durante esta iteración se hicieron tareas de refactorización del código existente para integrar las nuevas funcionalidades de asignación y postulación en temas de Seminario y TEG. Para ello fue necesario modificar la estructura de la base de datos agregando una tabla relacional entre usuario y proyectos para determinar los estudiantes asignados en un proyecto.

También fueron modificados los modelos de usuario y de proyectos existentes para agregar los métodos para contar los cupos disponibles en un proyecto así como obtener los estudiantes asignados en un proyecto y los métodos para obtener el tipo de proyecto (Seminario o TEG).

Junto a estos cambios también se desarrollaron las acciones del controlador y las vistas necesarias para la asignación y postulación. A partir de una de estas vistas surgió la importante necesidad de una función de autocompletar para el campo de cédula del estudiante en la vista de asignar tema. A razón de esta necesidad fue modificado el modelo de usuario para agregar el método que permitiera a la función de autocompletar obtener los estudiantes registrados que no estuviesen asignados a otros proyectos.

Resultó también necesario para concluir los procesos de asignación y postulación de temas crear los controles, vistas y métodos necesarios para enviar correos de notificación en los casos de asignación y postulación, a estudiantes y profesores, según correspondía.

3.4.3.3 Diseño

A lo largo de esta iteración fueron creadas las vistas necesarias para la solicitud y postulación de temas; de igual manera se crearon las vistas para que un profesor pueda listar sus proyectos, y por último se modificaron las vistas existentes para consultar en detalle los estudiantes asignados a un proyecto.

En la figura 3.16 se puede observar cómo un estudiante sin tema asignado solicita un nuevo tema mediante el uso de la cartelera de proyectos.

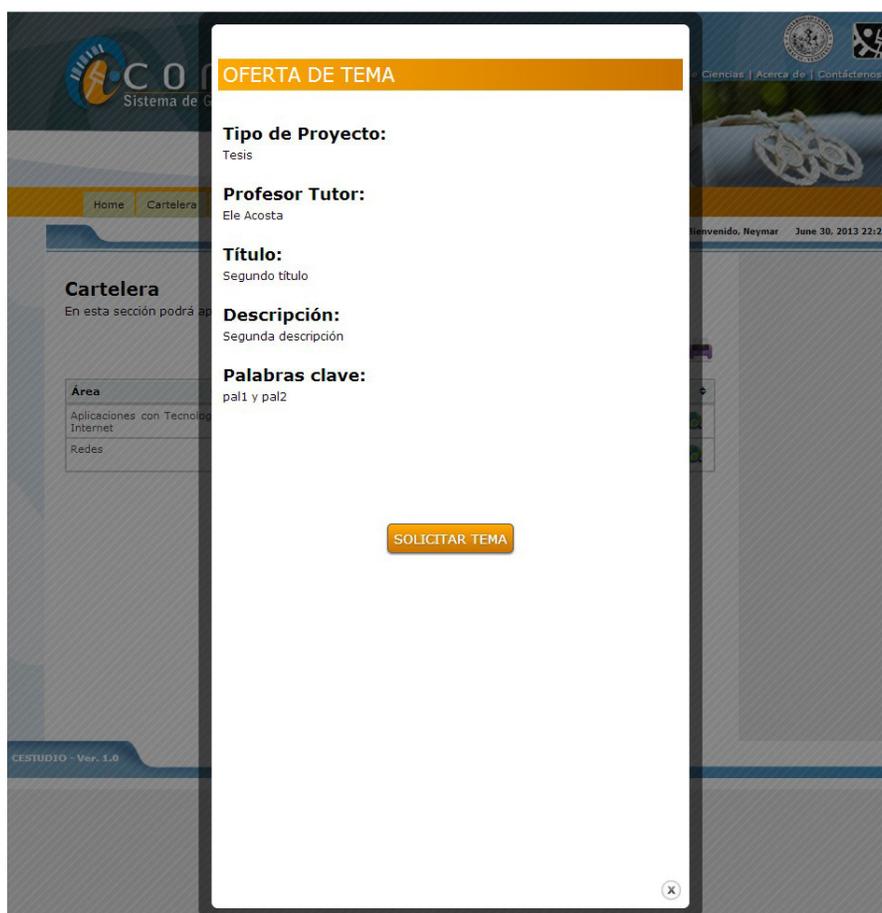


Figura 3.16 - Pantalla de Solicitud de Tema. A través de ella los estudiantes se postulan para un tema.

En la figura 3.17 se puede observar cómo un profesor asigna uno de sus proyectos a un estudiante, mediante el uso de la cartelera de proyectos.

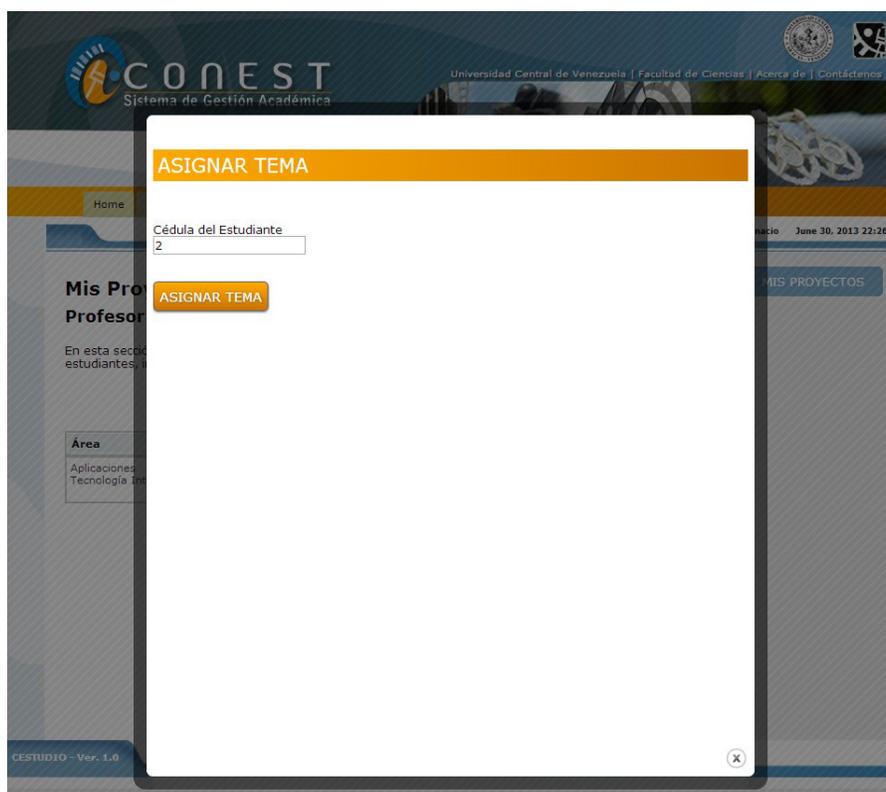


Figura 3.17 - Pantalla de Asignación de Temas.

En las figuras 3.18 y 3.19 se pueden observar las pantallas de gestión de proyectos del profesor, a través de la cuál un profesor puede consultar todos sus proyectos creados así como los estudiantes asignados a cada uno.

UNIVERSIDAD CENTRAL DE VENEZUELA
CONEST
 Sistema de Gestión Académica

Universidad Central de Venezuela | Facultad de Ciencias | Acerca de | Contáctenos

Home Cartelera Seminaristas Salir

Bienvenido, Ignacio June 30, 2013 21:45

Mis Proyectos

Profesor Ignacio Diago

En esta sección podrá asignar gestionar sus temas publicados con varias operaciones: asignar temas a estudiantes, iniciar proyectos, consultar proyectos en detalle, entre otros.

Área	Tipo	Título	Cupos					
Aplicaciones con Tecnología Internet	Seminario	Desarrollo de aplicación Web para la gestión de los procesos y actividades de Seminarios y TEG	1					
Bases de Datos	Seminario	Lorem ipsum dolor sit amet	1					

CESTUDIO - Ver. 1.0 Todos los Derechos Reservados. Grupo CONEST. Copyright 2007-2012

Figura 3.18 - Pantalla correspondiente al listado de proyectos de un profesor.

UNIVERSIDAD CENTRAL DE VENEZUELA
CONEST
 Sistema de Gestión Académica

Universidad Central de Venezuela | Facultad de Ciencias | Acerca de | Contáctenos

Home Cartelera

Bienvenido, Ignacio June 30, 2013 21:45

CONSULTAR PROYECTO

Tipo de Proyecto:
Seminario

Profesor Tutor:
Ignacio Diago

Título:
Desarrollo de aplicación Web para la gestión de los procesos y actividades de Seminarios y TEG

Descripción:
Se desea desarrollar una aplicación web que permita optimizar la forma en que se gestionan actualmente los procesos y actividades involucrados en los desarrollos de proyectos de Seminarios y Trabajos Especiales de Grado.

Palabras clave:
seminario tesis gestión aplicación web

Estudiante(s) Asignado(s):

 **Nombre:** David De Sousa
Eficiencia: 1
Promedio: 19
Año de ingreso: 2007

CESTUDIO - Ver. 1.0

Figura 3.19 - Detalle de proyecto consultado por un profesor.

En la figura 3.20 se puede observar que un proyecto cuenta con al menos un estudiante asignado, es iniciado.

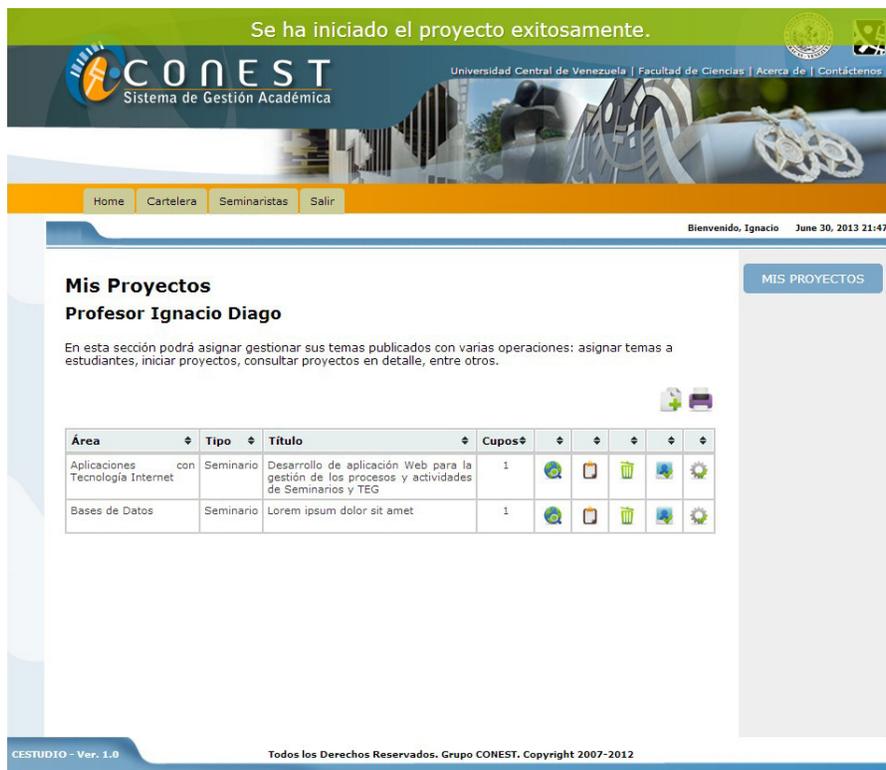


Figura 3.20 - Confirmación de que se ha iniciado un proyecto.

3.4.3.4 Historias

A continuación se detallan las historias de usuario correspondientes a esta iteración.

Título:	Postulación para Tema
Duración:	32 horas
Fecha de Inicio:	15-02-2013
Fecha de Culminación:	20-02-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un estudiante pueda consultar un tema a través de la cartelera y posteriormente postularse como candidato al mismo. Al momento de la postulación deberá recibir por correo electrónico una confirmación de su postulación, del mismo modo el profesor recibirá un correo detallando los datos del estudiante y el tema para el que fue postulado.

Título:	Asignación de Tema
Duración:	16 horas
Fecha de Inicio:	21-02-2013
Fecha de Culminación:	25-02-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado pueda asignar uno de sus temas creados a un estudiante a través de su cédula de identidad; para ello, el sistema debe tomar en cuenta que un estudiante no esté en dos proyectos simultáneamente, que un profesor no pueda asignar a estudiantes inexistentes y que un profesor no pueda asignar a más de dos estudiantes a un tema.

Título:	Listados de Estudiantes
Duración:	16 horas
Fecha de Inicio:	25-02-2013
Fecha de Culminación:	27-02-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado pueda visualizar los estudiantes que tiene asignados por proyecto. Los estudiantes estarán separados en dos pantallas según sea el tipo de proyecto que estén desarrollando.

Título:	Refactorización
Duración:	16 horas
Fecha de Inicio:	27-02-2013
Fecha de Culminación:	28-02-2013
Descripción	Se deben refactorizar algunas pantallas y secciones para incorporar las nuevas funcionalidades desarrolladas.

3.4.3.5 Pruebas

A continuación se detallan las pruebas llevadas a cabo en esta iteración.

Descripción	Resultado Esperado	Resultado Obtenido
Postular Estudiante en Tema	El sistema deberá permitir que únicamente los estudiantes registrados que no estén asignados en otros temas se postulen a los temas ofertas a través de la cartelera, siempre y cuando además el mismo esté publicado y haya disponibilidad en cupos. Una vez se haya postulado un estudiante se debe enviar un correo de confirmación al estudiante y un correo de notificación al profesor con los detalles del estudiante.	Se logró enviar una postulación de un estudiante sin tema. Se comprobó que únicamente los estudiantes sin tema se puedan postular y únicamente a proyectos con disponibilidad de cupos. Se verificó el funcionamiento de las notificaciones por correo electrónico.

Asignar Estudiante a Tema	El sistema deberá permitir a los profesores registrados asignar un proyecto que les pertenezca a un estudiante siempre y cuando haya cupos disponibles y el estudiante no se encuentre asignado a otros temas. Una vez se haya asignado un estudiante a un tema se debe enviar un correo de confirmación al estudiante.	Se logró asignar exitosamente un estudiante a un proyecto. Se comprobó que la cédula del estudiante existiera y que el estudiante no estuviera asignado a otro tema. Se comprobó el correcto funcionamiento de las notificaciones a través de correos electrónicos.
Autocompletar Cédula del Listado de Estudiantes en Asignar Tema	En la funcionalidad de autocompletar cédula del estudiante se espera que el sistema únicamente muestre cédulas válidas de estudiantes que no estén asignados a otros temas.	Se insertaron hasta 10 estudiantes en la base de datos y se asignó temas de prueba a 5 de ellos. Durante la prueba se comprobó que las listas de la función autocompletar únicamente mostraban las cédulas correctas que correspondían a los estudiantes sin temas.

3.4.4 Iteración 4 - (01-03-2013 al 15-03-2013)

Durante esta iteración se prosiguen realizando refactorizaciones para mantener el nivel de calidad del código e incorporar nuevas funcionalidades. Adicionalmente se desarrollan todas las funcionalidades relativas al proceso de gestión de presentaciones tanto de proyectos de Seminario como TEG.

3.4.4.1 Planificación

En la tabla 3.5 se detalla la planificación efectuada al inicio de esta iteración.

Tabla 3.5 – Planificación de Iteración 4.

Iteración 4		
Descripción	Gestión de actividades de presentación en Seminarios y TEG.	
Período	01-03-2013 al 15-03-2013	
Fecha	Historia	Tipo
01-03-2013	Refactorización.	Modificación
04-03-2013	Detalles de proyectos.	Nuevo
07-03-2013	Definición de estructura de contenido de la presentación.	Nuevo
07-03-2013	Marcar contenido de presentación como culminado.	Nuevo
13-03-2013	Barras de progreso de presentaciones.	Modificación

14-03-2013	Iteraciones de la presentación.	Nuevo
15-03-2013	Definición de estructura de contenido del documento.	Nuevo

3.4.4.2 Codificación

Al inicio de esta iteración se llevó a cabo un proceso de refactorización del código para mantener la calidad del mismo y seguir permitiendo su escalabilidad. Adicionalmente se crearon las tablas para las estructuras de contenidos de la presentación. De igual manera se crearon las tablas para almacenar las iteraciones de presentación de un proyecto.

Se desarrollaron los modelos de presentación y de iteración de presentación. Posteriormente se implementaron los métodos para marcar contenidos de la presentación como finalizados y una vez fueron culminados estos métodos se procedió a desarrollar los métodos para medir los progresos en la presentación.

3.4.4.3 Diseño

Durante esta iteración se desarrollaron los controles visuales para la visualización de los progresos de proyecto. Entre ellos las vistas globales de un proyecto y los controles de IU para marcar contenidos de presentación como finalizados y para el manejo de las presentaciones como iteraciones. Por último también fueron modificadas algunas vistas existentes para permitir agregar comentarios en las iteraciones.

En la figura 3.21 se puede observar la pantalla de progresos del proyecto donde tanto profesores como estudiantes, pueden visualizar los progresos alcanzados hasta un momento dado.



Figura 3.21 - Pantalla de Progresos de Proyecto.

En la figura 3.22 se puede observar la pantalla de creación de estructura de contenidos de documento o presentación disponible para los profesores.



Figura 3.22 - Pantalla de definición de estructura de contenidos para presentación o documento.

En la figura 3.23 se puede observar los controles creados para que los profesores puedan marcar contenidos de presentación como finalizados.



Figura 3.23 – Pantalla de iteraciones de presentación o documento.

3.4.4.4 Historias

A continuación se detallan las historias de usuario correspondientes a esta iteración.

Título:	Refactorización
Duración:	16 horas
Fecha de Inicio:	01-03-2013
Fecha de Culminación:	04-03-2013
Descripción	Se deben refactorizar algunas pantallas y secciones para incorporar las nuevas funcionalidades desarrolladas.

Título:	Detalle de Proyecto
Duración:	24 horas
Fecha de Inicio:	04-03-2013
Fecha de Culminación:	06-03-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado y un estudiante puedan visualizar en detalle la información de un proyecto en desarrollo. La página principal de detalle de proyecto debe contener los progresos alcanzados hasta el momento, así como los accesos a la presentación, documento y producto, si corresponde.

Título:	Definición de Estructura de Contenido de Presentación
Duración:	24 horas
Fecha de Inicio:	07-03-2013
Fecha de Culminación:	11-03-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado pueda crear la estructura de la presentación de cualquiera de sus proyectos. Para acceder a esta sección el profesor debió haber seleccionado previamente el proyecto y la sección referente a la presentación del mismo.

Título:	Marcar Contenido de Presentación como Culminado
Duración:	24 horas
Fecha de Inicio:	07-03-2013
Fecha de Culminación:	13-03-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado pueda marcar un contenido de una presentación como finalizado, en la sección de iteraciones de presentación de un proyecto. El estudiante no podrá marcar contenidos como culminados.

Título:	Barras de Progreso de Presentaciones
Duración:	12 horas
Fecha de Inicio:	13-03-2013
Fecha de Culminación:	15-03-2013
Descripción	A partir de la lista de contenidos culminados en la presentación es posible medir los avances de la presentación por lo que se debe realizar una barra de progreso que indique a los usuarios de manera visual los avances alcanzados por el estudiante.

Título:	Iteraciones de la Presentación
Duración:	8 horas
Fecha de Inicio:	14-03-2013
Fecha de Culminación:	15-03-2013
Descripción	Se debe desarrollar un formulario que permita al estudiante autenticado crear iteraciones, bien sea para seminarios o trabajos especiales de grado. Las mismas tendrán una descripción, la fecha y un archivo adjunto. Las iteraciones quedarán asociadas a un proyecto, un área de investigación y a un profesor tutor.

Título:	Definición de Estructura de Contenido de Documento
Duración:	8 horas
Fecha de Inicio:	15-03-2013
Fecha de Culminación:	15-03-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado pueda crear la estructura del documento de cualquiera de sus proyectos. Para acceder a esta sección el profesor debió haber seleccionado previamente el proyecto y la sección referente al documento del mismo.

3.4.3.5 Pruebas

A continuación se detallan las pruebas llevadas a cabo en esta iteración.

Descripción	Resultado Esperado	Resultado Obtenido
Crear estructura de presentación.	El sistema deberá permitir que los profesores puedan modificar la estructura de la presentación de sus proyectos iniciados a través de un formulario.	Se creó la estructura de la presentación exitosamente.
Marcar contenidos de presentación como finalizados.	El sistema deberá permitir que los profesores puedan marcar contenidos de la presentación como finalizados a través de un control de IU proporcionado en las vistas.	Los contenidos marcados como finalizados fueron almacenados exitosamente en la base de datos y en la IU.
Crear iteración de presentación	El sistema deberá permitir a los estudiantes crear iteraciones de presentación subiendo un archivo de presentación con una descripción de los trabajos realizados durante esa iteración.	Se creó la iteración exitosamente; la iteración recién creada es mostrada correctamente en el listado de iteraciones de presentación y los archivos de presentación son descargables sin problemas

3.4.5 Iteración 5 - (16-03-2013 al 30-03-2013)

En esta iteración desarrollan todas las funcionalidades relativas al proceso de gestión de documentos tanto de proyectos de Seminario como TEG así como los procesos de desarrollo en proyectos de TEG. Para ello también es necesario realizar refactorizaciones de procesos desarrollados previamente para incluir nuevas funcionalidades.

3.4.5.1 Planificación

En la tabla 3.6 se detalla la planificación efectuada al inicio de esta iteración.

Tabla 3.6 – Planificación de Iteración 5.

Iteración 5		
Descripción	Gestión de actividades de documentos en Seminarios y TEG y tareas de desarrollo en TEG.	
Período	16-03-2013 al 30-03-2013	
Fecha	Historia	Tipo
16-03-2013	Marcar contenido de documento como culminado.	Nuevo

17-03-2013	Barras de progreso de documentos.	Nuevo
19-03-2013	Iteraciones de documentos.	Nuevo
20-03-2013	Comentarios de iteraciones.	Modificación
21-03-2013	Exportar de seminario a tesis.	Nuevo
22-03-2013	Tareas de desarrollo.	Nuevo
23-03-2013	Paneles de tareas.	Nuevo
27-03-2013	Comentarios de tareas.	Modificación
28-03-2013	Archivos del desarrollo.	Nuevo

3.4.5.2 Codificación

Para esta iteración se agregaron al esquema de base de datos, las tablas que almacenarían las estructuras anidadas de los contenidos del documento; también fueron agregadas las tablas para almacenar las tareas de desarrollo y los archivos de estas tareas de desarrollo, así como también las tablas para almacenar los comentarios en tareas o en iteraciones de presentación o documento.

Una vez finalizados los cambios en base de datos fueron desarrollados los modelos de documento, iteración de documento, comentarios de iteración de documento, comentarios de iteración de presentación, tareas de desarrollo, archivos de tareas de desarrollo y comentarios de tareas de desarrollo. Posteriormente se implementaron los métodos para crear iteraciones de documento, marcar contenidos de documento como finalizados y añadir comentarios en las iteraciones de documento o presentación. Luego de finalizados estos métodos se procedió a desarrollar los métodos para medir los progresos en el documento e integrarlos con los métodos de medición de progreso de presentación para calcular finalmente el progreso global de un proyecto específico.

Además, en esta iteración se desarrollaron los métodos para crear tareas de desarrollo, y con estos también se crearon los métodos necesarios para agregar archivos de desarrollo a una tarea específica; se crearon los métodos para agregar comentarios en las tareas y finalmente se crearon los métodos para actualizar el estado de una tarea.

Por último se desarrollaron los métodos para convertir un proyecto de Seminario ya concluido con un 100% de progreso en un Trabajo Especial de Grado.

3.4.5.3 Diseño

En esta iteración fueron creadas las vistas para añadir comentarios a las iteraciones. En la figura 3.24 se puede observar la pantalla de comentarios disponible tanto para los estudiantes como para los profesores.

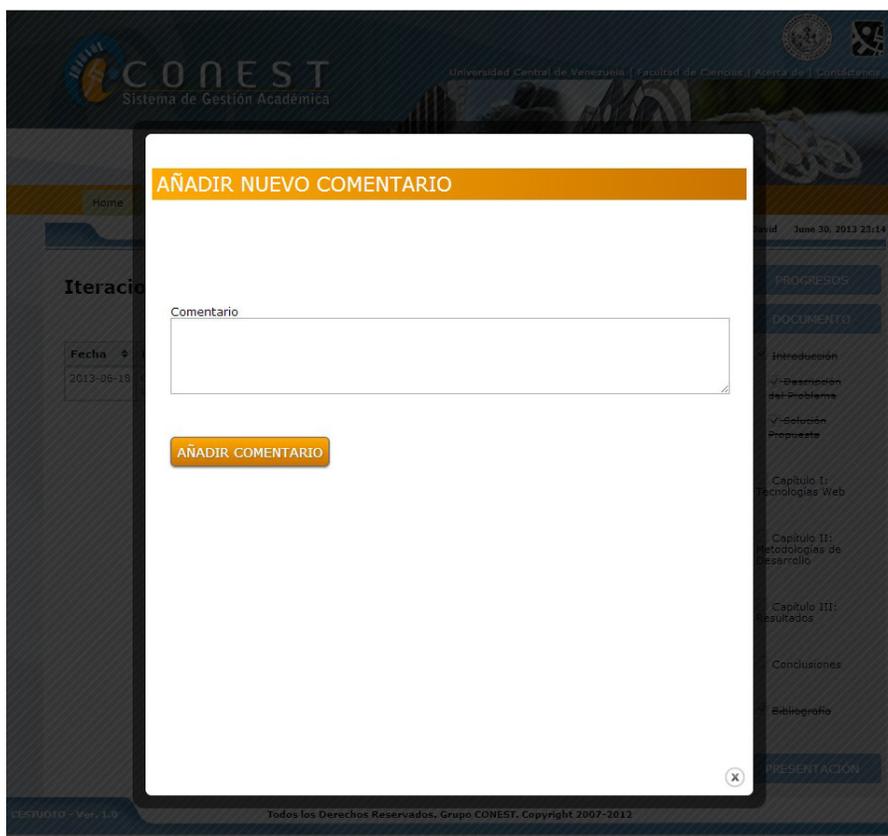


Figura 3.24 – Pantalla de Nuevos Comentarios en iteraciones de documento o presentación.

Una vez se ha concluido un proyecto de Seminario, se habilita la funcionalidad de exportación de Seminario a tesis. Para ello se modificaron las vistas existentes para habilitar esta nueva opción ya desarrollada. Cuando un Seminario es exportado se muestra un nuevo control como se observa en la figura 3.25.



Figura 3.25 - Pantalla de Progresos, en para exportar un seminario y continuarlo como tesis.

El principal desafío en esta iteración fue la elaboración de las vistas para permitir el manejo de las tareas de desarrollo entre tres listas a través de una simple interfaz de arrastre de elementos. También fueron agregadas notificaciones visuales para indicar tareas con retraso durante esta iteración.

Por último se crearon las vistas para permitir al usuario a través de un formulario crear tareas de desarrollo, agregar archivos y comentarios de una manera sencilla y en un único formulario. En las figuras 3.26 y 3.27 se pueden observar las pantallas donde se gestionan las tareas de desarrollo, estas pantallas están disponibles para los profesores y estudiantes.



Figura 3.26 - Pantalla de Gestión de Tareas de Desarrollo



Figura 3.27 - Formulario para agregar tareas de desarrollo, así como archivos y comentarios relacionados.

3.4.5.4 Historias

A continuación se detallan las historias de usuario correspondientes a esta iteración.

Título:	Marcar Contenido de Documento como Culminado
Duración:	8 horas
Fecha de Inicio:	16-03-2013
Fecha de Culminación:	16-03-2013
Descripción	Se deben desarrollar las funcionalidades necesarias para que un profesor autenticado pueda marcar un contenido de un documento como finalizado, en la sección de iteraciones de documento de un proyecto. El estudiante no podrá marcar contenidos como culminados.

Título:	Barras de Progreso de Documentos
Duración:	12 horas
Fecha de Inicio:	17-03-2013
Fecha de Culminación:	19-03-2013
Descripción	A partir de la lista de contenidos culminados en el documento, es posible medir los avances del documento por lo que se debe realizar una barra de progreso que indique a los usuarios de manera visual los avances alcanzados por el estudiante.

Título:	Iteraciones del Documento
Duración:	8 horas
Fecha de Inicio:	19-03-2013
Fecha de Culminación:	19-03-2013
Descripción	Se debe desarrollar un formulario que permita al estudiante autenticado crear iteraciones, bien sea para seminarios o trabajos especiales de grado. Las mismas tendrán una descripción, la fecha y un archivo adjunto. Las iteraciones quedarán asociadas a un proyecto, un área de investigación y a un profesor tutor.

Título:	Comentarios de las Iteraciones del Documento
Duración:	8 horas
Fecha de Inicio:	20-03-2013
Fecha de Culminación:	20-03-2013
Descripción	Se debe desarrollar un formulario que permita al estudiante y al profesor crear comentarios sobre las iteraciones, bien sea para seminarios o trabajos especiales de grado. Los comentarios de las iteraciones quedarán asociadas a un proyecto, un área de investigación.

Título:	Exportar de Seminario a Tesis
Duración:	8 horas
Fecha de Inicio:	21-03-2013
Fecha de Culminación:	21-03-2013
Descripción	Se debe desarrollar una función que permita al profesor, una vez finalizado el Seminario, exportarlo a TEG. Una vez hecho esto tanto el profesor como el estudiante podrán observar la última iteración del documento y presentación del Seminario.

Título:	Tareas del Desarrollo
Duración:	8 horas
Fecha de Inicio:	22-03-2013
Fecha de Culminación:	23-03-2013
Descripción	Se debe desarrollar formularios que permita al profesor o estudiante crear tareas del desarrollo.

Título:	Paneles de las Tareas del Desarrollo
Duración:	24 horas
Fecha de Inicio:	23-03-2013
Fecha de Culminación:	26-03-2013
Descripción	Esta sección cuenta con tres paneles (Pendiente – En Desarrollo – Finalizada), la tarea se crea por defecto en el panel Pendiente y la misma puede ser arrastrada mediante el uso de un control de interfaz de usuario de tipo Drag and Drop, dependiendo de su nuevo estatus, a cualquier de los otros paneles.

Título:	Comentarios de las Tareas del Desarrollo
Duración:	8 horas
Fecha de Inicio:	27-03-2013
Fecha de Culminación:	27-03-2013
Descripción	Se debe desarrollar un formulario que permita al estudiante y al profesor crear comentarios sobre las tareas. Los comentarios de las tareas quedarán asociadas a un proyecto, un área de investigación.

Título:	Archivos del Desarrollo
Duración:	8 horas
Fecha de Inicio:	28-03-2013
Fecha de Culminación:	28-03-2013
Descripción	Se debe desarrollar un formulario que permita al estudiante autenticado subir archivos. Los archivos quedarán asociadas a un proyecto, un área de investigación.

3.4.5.5 Pruebas

A continuación se detallan las pruebas llevadas a cabo en esta iteración.

Descripción	Resultado Esperado	Resultado Obtenido
Crear estructura de documento.	El sistema deberá permitir que los profesores puedan modificar la estructura del documento de sus proyectos iniciados a través de un formulario.	Se creó la estructura del documento exitosamente.
Marcar contenidos del documento como finalizados.	El sistema deberá permitir que los profesores puedan marcar contenidos y sub-contenidos del documento, como finalizados a través de un control de IU.	Los contenidos marcados como finalizados fueron almacenados exitosamente en la base de datos y en la IU.

Crear iteración de documento	El sistema deberá permitir a los estudiantes crear iteraciones de documento subiendo un archivo de tipo documento con una descripción de los trabajos realizados durante esa iteración.	Se creó la iteración exitosamente; la iteración recién creada es mostrada correctamente en el listado de iteraciones de documento y los archivos de documento son descargables sin problemas
Añadir comentario en iteración de documento	El sistema deberá permitir a usuarios registrados añadir comentarios relacionados a una iteración en forma de hilo de conversación.	Se guardaron los comentarios exitosamente y fueron mostrados correctamente en el listado de conversaciones de la iteración.
Exportar Seminario a Tesis	El sistema deberá permitir que un proyecto de Seminario que haya alcanzado un 100% de progreso sea exportado a Tesis para darle continuidad al mismo.	Se logró exportar con éxito un proyecto de Seminario culminado. La primera iteración de presentación y la primera iteración de documento en el nuevo proyecto de tesis fueron tomadas de las últimas iteraciones del Seminario.
Agregar Tarea de Desarrollo	El sistema deberá permitir a estudiantes y profesores crear nuevas tareas de desarrollo de un proyecto y a su vez añadirle tantos archivos o comentarios consideren necesarios. Una vez creada la tarea deberá reflejarse en la lista de tareas pendientes.	Se logró crear exitosamente una tarea con tres archivos y tres comentarios. La misma fue mostrada como se esperaba en la lista de tareas pendientes.
Modificar Tarea de Desarrollo	El sistema deberá permitir a estudiantes y profesores modificar las tareas de desarrollo de un proyecto y a su vez añadirle tantos archivos o comentarios consideren necesarios.	Se logró modificar exitosamente una tarea agregando nuevos archivos y comentarios.
Agregar Tarea de Desarrollo con Retrasos para Comprobar Alertas.	El sistema debería notificar cuando existen tareas vencidas que no han sido finalizadas.	Después de crear una tarea con fecha anterior a la fecha actual se comprobó que las notificaciones de retraso son mostradas correctamente.

3.4.6 Iteración 6 - (01-04-2013 al 10-04-2013)

Una vez culminados los desarrollos principales, surge la necesidad del usuario final de contar con nuevas funcionalidades. Para ello se planifica esta nueva iteración durante la cual las mismas son desarrolladas; entre estas funcionalidades figuran permitir al profesor convocar a los estudiantes a reuniones, generar planillas de solicitud de jurados, entre otros reportes en .pdf que son de relevancia para el usuario.

3.4.5.1 Planificación

En la tabla 3.7 se detalla la planificación efectuada al inicio de esta iteración.

Tabla 3.7 – Planificación de Iteración 6.

Iteración 6		
Descripción	Generar imprimibles en PDF y otros reportes relevantes.	
Período	01-04-2013 al 10-04-2013	
Fecha	Historia	Tipo
04-04-2013	Solicitud de reunión.	Nuevo
05-04-2013	Solicitud de jurados.	Nuevo
06-04-2013	Reporte de solicitud de nombramiento de jurado de TEG.	Modificación
07-04-2013	Reportes de resumen de proyecto.	Modificación
08-04-2013	Reporte de proyectos.	Modificación
09-04-2013	Reporte de ofertas de proyectos de seminarios y TEG.	Modificación

3.4.6.2 Codificación

Para esta última iteración se desarrollaron los métodos para generar los reportes en PDF que presenta la aplicación. Estos reportes surgen a raíz de los nuevos requerimientos identificados en la reunión de planificación.

Para estos reportes fue necesario refactorizar los modelos existentes para obtener los elementos que contiene cada reporte. Asimismo, se agregaron nuevos elementos de IU para generar las impresiones.

A nivel de diseño de la base de datos no fue necesario realizar ningún cambio más sin embargo se agregaron los nuevos métodos para permitir al profesor efectuar la convocatoria a

una reunión. De igual manera se desarrollaron los métodos de envío de correos para estas nuevas notificaciones.

3.4.6.3 Diseño

En esta iteración se diseñó la vista para la solicitud de reunión, la cual consta de un formulario donde el profesor puede convocar reuniones con sus estudiantes por proyecto cuando así lo considere oportuno.

Adicionalmente se diseñaron una serie de reportes en PDF. Entre ellos, el reporte de solicitud de jurados, que a partir de los datos insertados en un formulario genera la planilla de solicitud de jurados de TEG.

Por último se diseñaron y desarrollaron los reportes de resumen de proyecto, reporte de proyectos y reporte de ofertas de proyectos de seminarios y TEG. En el reporte de resumen de proyecto el profesor consolida los progresos de los estudiantes de un determinado proyecto. En el reporte de proyectos un profesor puede listar todos los proyectos en ejecución y en el reporte de ofertas de proyectos de seminarios y TEG se podrá observar todos los temas disponibles para ambas asignaturas.

En la figura 3.28 se puede observar la pantalla donde el profesor puede solicitar reuniones con sus estudiantes, consta de un formulario donde podrá seleccionar la fecha, la hora y e incluir una descripción.

The screenshot shows a web application interface with a central form titled "CONVOCAR A REUNIÓN". The form contains the following elements:

- A date input field labeled "Fecha".
- A time selection dropdown labeled "Hora" with the value "11 PM" and a minute selector set to "51".
- A text area labeled "Descripción".
- An orange button labeled "ENVIAR".

The interface also features a sidebar on the left with a "Home" link and a "Proyecto" section, and a sidebar on the right with a navigation menu including "PROGRESOS", "DOCUMENTO", "PRESENTACION", and "DESARROLLO". The top right corner shows a date and time: "June 30, 2013 23:51".

Figura 3.28 - Pantalla para convocar a reunión. Disponible para estudiantes y profesores.

En la figura 3.29 se puede observar el reporte de la solicitud de nombramiento de jurado TEG, el mismo contiene la fecha en que fue generado, el estudiante, el título del proyecto, el área de investigación, el resumen, las palabras claves, la fecha estimada de presentación y los jurados propuestos.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Solicitud de nombramiento de Jurado de TEG

Fecha
2013-04-19

Estudiante(s)
Ignacio Ortiz - C.I. 20

Título
Desarrollo de aplicación para el manejo de tesis y seminarios

Área de Investigación
Aplicaciones con Tecnología Internet

Resumen
Desarrollo de aplicación para el manejo de tesis y seminarios

Palabras Clave
tesis, seminario, desarrollo

Fecha Estimada de Presentación
2013-05-15

Propuesta de Jurado
Jurado Principal I: Eugenio Scalisse
Jurado Principal II: Sergio Rivas
Jurado Suplente I: Nestor Mendez
Jurado Suplente II: Lucia Ojeda

Firmas(s)

Jossie ZambranoRhadames Carmona

Figura 3.29 – Reporte de solicitud de nombramiento de jurado de TEG.

En la figura 3.30 se puede observar el reporte de resumen de proyecto, el mismo contiene la fecha en que fue generado, el tipo de proyecto, el título del proyecto, los integrantes del proyecto, el progreso de cada uno de los entregables del proyecto, el resumen, las palabras claves y por último el tutor.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Resumen de Proyecto

Fecha
2013-04-19

Tipo de Proyecto
Tesis

Título
Desarrollo de aplicación para el manejo de tesis y seminarios

Integrantes

Nombre	Cédula	Eficiencia	Promedio
Ignacio Ortiz	20	0.8	12

Progreso

Actividad	Progreso
Documento	100%
Presentación	100%
Desarrollo	100%

Promedio: 100%

Resumen
Desarrollo de aplicación para el manejo de tesis y seminarios

Palabras Clave
tesis, seminario, desarrollo

Profesor(a) Tutor(a)
Jossie Zambrano

Figura 3.30 – Reporte Resumen del Proyecto.

En la figura 3.31 se puede observar el reporte de ofertas de proyectos de Seminarios y TEG. En este reporte se podrá observar todos los proyectos ofertados, en el mismo contiene el área de investigación, el tipo, el tutor y el título del proyecto.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Oferta de Proyectos de Seminario y TEG

Área de Investigación	Tipo	Tutor	Título
Aplicaciones con Tecnología Internet	Tesis	Jossie Zambrano	Desarrollo de aplicación para el manejo de las inscripciones en la facultad de ciencias
Aplicaciones con Tecnología Internet	Tesis	Jossie Zambrano	Desarrollo de aplicación para el manejo de tesis y seminarios

Total de Proyectos Ofertados: 2

Figura 3.31 – Reporte de Ofertas de Proyectos de Seminarios y TEG.

3.4.6.4 Pruebas

A continuación se detallan las pruebas llevadas a cabo en esta iteración.

Descripción	Resultado Esperado	Resultado Obtenido
Solicitud de reunión	El sistema deberá permitir a los profesores solicitar reuniones con sus estudiantes cuando crea oportuno.	Se logró convocar a una reunión exitosamente. Al llenar el formulario el estudiante recibió un correo de notificación de la reunión.
Solicitud de jurado	El sistema deberá permitir a los profesores generar la planilla de solicitud los jurados para la presentación.	El profesor pudo generar satisfactoriamente la planilla de solicitud de jurados.

Reportes de resumen de proyecto.	El sistema deberá permitir tanto a estudiantes como profesores generar un reporte con el resumen del proyecto.	Se generó exitosamente el reporte. Al darle al icono de generar PDF tanto el profesor como el estudiante pudieron generar el reporte.
Reportes de proyectos.	El sistema deberá permitir tanto a estudiantes como profesores generar un reporte con sus proyectos.	Se generó exitosamente el reporte. Al darle al icono de generar PDF el profesor logró generar el reporte.
Reporte de ofertas de proyectos de seminarios y TEG.	El sistema deberá permitir tanto a estudiantes como profesores generar un reporte con todos los proyectos ofertados.	Se generó satisfactoriamente el reporte. El profesor como el estudiante al darle al icono de generar PDF pudieron generarlo sin problemas.

3.4.6.5 Historias

A continuación se detallan las historias de usuario correspondientes a esta iteración.

Título:	Solicitud de reunión
Duración:	4 horas
Fecha de Inicio:	04-04-2013
Fecha de Culminación:	04-04-2013
Descripción	Se debe desarrollar un formulario que permita al profesor solicitar reuniones.

Título:	Solicitud de jurado
Duración:	4 horas
Fecha de Inicio:	05-04-2013
Fecha de Culminación:	05-04-2013
Descripción	Se debe desarrollar un formulario que permita al profesor proponer el jurado.

Título:	Reporte de solicitud de nombramiento de jurado de TEG
Duración:	4 horas
Fecha de Inicio:	06-04-2013
Fecha de Culminación:	06-04-2013
Descripción	Reporte que muestra la solicitud de nombramiento de jurado de TEG.

Título:	Reporte Resumen del Proyecto
Duración:	4 horas
Fecha de Inicio:	07-04-2013
Fecha de Culminación:	07-04-2013
Descripción	Reporte que muestra el resumen del proyecto seleccionado.

Título:	Reporte de Proyectos
Duración:	4 horas
Fecha de Inicio:	08-04-2013

Fecha de Culminación:	08-04-2013
Descripción	Reporte que muestra todos los proyectos de un determinado profesor.
Título:	Reporte de Ofertas de Proyectos de Seminarios y TEG
Duración:	4 horas
Fecha de Inicio:	09-04-2013
Fecha de Culminación:	09-04-2013
Descripción	Reporte que muestra las ofertas de proyectos tanto para seminarios como para TEG.

3.5 Pruebas de Aceptación

Como pruebas de aceptación, se permitió que el tutor y un grupo de estudiantes interactuaran con la aplicación. Para ello la aplicación fue desplegada en un servidor público desde el cual se procedió a cargar los datos referentes a proyectos de prueba. Esta demostración involucró todos los procesos y funcionalidades de la aplicación.

Se registraron las cuentas del profesor y los estudiantes. Desde la cuenta del profesor fueron creados proyectos de Seminario y TEG. Posteriormente desde la cuenta de los estudiantes se procedió a solicitar los temas que inmediatamente después fueron asignados a cada uno desde la cuenta del profesor. Realizado esto el profesor procedió a dar inicio al proyecto de prueba cargado en el sistema a través de los controles proporcionados por la aplicación. Se creó la estructura del documento y la presentación del proyecto de prueba. Desde la cuenta de los estudiantes se procedió a crear algunas iteraciones, para posteriormente aprobar los contenidos finalizados, utilizando la cuenta del profesor. Por último se cargaron en la aplicación historias de usuarios de la primera y segunda iteración para simular los procesos del desarrollo. Durante esta prueba no se detectó error alguno, sin embargo, fueron realizadas algunas observaciones referentes a la interfaz de usuario: disposición de íconos, colores, textos de etiquetas, entre otros. También se detectaron nuevos requerimientos funcionales de reportes de información y documentos imprimibles tales como cartelera, resumen de proyectos, resumen de progresos, entre otros. Estos cambios y requerimientos se implementaron y probaron posteriormente.

La aplicación actualizada recibió el visto bueno de los profesores y estudiantes que participaron en las pruebas de aceptación.

En este capítulo se presentó la experiencia de desarrollo de la aplicación. A lo largo del mismo se asientan conocimientos sobre técnicas y métodos de desarrollo de aplicaciones Web que constituyen una parte fundamental de la Ingeniería de Software.

Conclusiones

El objetivo general del presente TEG fue alcanzado satisfactoriamente mediante el desarrollo de una aplicación Web que permite gestionar los procesos de desarrollo de proyectos de Seminarios y TEG desarrollados aplicando métodos ágiles de la Escuela de Computación de la Facultad de Ciencias.

Para poder alcanzar este objetivo se lograron cada uno de los objetivos específicos planteados; a partir de una comparación entre los métodos ágiles, se logró identificar las funcionalidades que debía tener la aplicación Web desarrollada. En este sentido todos los prototipos de interfaz de usuario (IU) elaborados fueron integrados de manera transparente a la ya existente de CONEST (Sistema de Gestión Académico Administrativo de la División de Control de Estudios de la Facultad de Ciencias) y otras aplicaciones desarrolladas en la Escuela de Computación.

La aplicación desarrollada puede ser de gran utilidad para los estudiantes durante el desarrollo de Seminarios o TEG, ya que desde el momento en que deben buscar un tema hasta el momento de la culminación del proyecto, se les proporcionan herramientas para medir y organizar sus progresos, así como obtener retroalimentación de manera organizada de parte de sus tutores. Del mismo modo esta aplicación permite a los profesores organizar de manera más eficaz y eficiente los proyectos de Seminarios o TEG que dirigen, proporcionándoles información real del progreso alcanzado por sus estudiantes y también un canal de comunicación directa con los mismos bajo un enfoque de proyectos.

Con el desarrollo de esta aplicación se espera, entre otras cosas, disminuir los tiempos de desarrollo de proyectos de Seminario y TEG así como la incidencia de abandono estudiantil en los últimos semestres de la carrera al facilitarse el proceso de culminación de la misma. Los estudiantes a partir de ahora podrán medir sus avances en conjunto con los profesores tutores de manera inmediata y constante.

Del mismo modo se espera que con esta nueva herramienta los profesores tutores puedan brindar una asesoría más oportuna y específica a cada estudiante a lo largo del desarrollo de sus proyectos de Seminario y TEG. Asimismo, dado que los profesores podrán gestionar los proyectos de manera más eficiente se espera entonces que cuenten con mayor disponibilidad de tiempo, y esto a su vez permita un incremento en la apertura de oferta de temas por parte de los mismos.

Cabe mencionar que para el desarrollo de la aplicación se utilizó el método ágil XP tomando en cuenta sus principales prácticas, así como sus valores fundamentales. El uso de este modelo, facilitó la planificación y entrega del proyecto dentro de los parámetros de tiempo establecidos, dividiendo los requerimientos en iteraciones, que se fueron desarrollando progresivamente según su prioridad, gracias a la utilización de la mayoría de las prácticas aplicables en el contexto de este proyecto.

La aplicación que constituye el producto final de este TEG, fue desarrollada en el lenguaje de programación Ruby en su versión 1.9.3, siendo ésta la última disponible al momento de iniciarse este proyecto. De igual forma se utilizó el framework de desarrollo Rails en su versión 3.2.9, por lo que para cumplir con los requisitos de desarrollo de dicho framework, la aplicación fue desarrollada siguiendo una arquitectura MVC. Entre las ventajas de utilizar una arquitectura MVC, se encuentra poder organizar el código mediante la separación de la capa de presentación de la lógica de negocio, lo cual facilita no solo la legibilidad del mismo sino también su escalabilidad y mantenimiento.

Durante cada iteración se llevaron a cabo una serie de pruebas de validación para comprobar la robustez del sistema. Del mismo modo, al finalizar todo el proyecto, se llevó a cabo una prueba de aceptación para validar el buen y correcto funcionamiento de la aplicación en conjunto con el tutor y otros estudiantes. Adicionalmente, se tomó como caso de estudio para validar esta aplicación, este proyecto, siendo la misma cargada en el servidor de pruebas, resultando exitosa la evaluación del mismo.

Para futuros desarrollos Web se sugiere la utilización de lenguajes de programación diferentes a Ruby sobre el framework Rails, que cuenten con una mayor compatibilidad con sus propias versiones anteriores, dado que si bien Ruby on Rails resultó ser efectivo en este caso, durante el uso del mismo se presentaron numerosas dificultades ante la escasez de documentación precisa y las constantes incompatibilidades entre versiones recientes y anteriores, lo que conllevó una ralentización en el desarrollo durante las primeras fases del mismo. Finalmente, se espera que tras el desarrollo de este TEG en el que se cumplieron satisfactoriamente los objetivos planteados inicialmente, se haya logrado un aporte significativo a la comunidad Docente y Estudiantil de la Escuela de Computación de la Facultad de Ciencias, ya que hasta el momento no contaba con una herramienta similar que permitiera gestionar los desarrollos de proyectos de Seminario y TEG.

Recomendaciones

Si bien a partir de ahora los estudiantes y profesores tutores de la Escuela de Computación de la Facultad de Ciencias, podrán beneficiarse del uso de la herramienta elaborada, de la experiencia de su elaboración pueden extraerse algunas recomendaciones que son brevemente mencionadas a continuación:

- Como primera recomendación tras culminar este proyecto, se sugiere implementar y fomentar su uso entre grupos pequeños de estudiantes y profesores, en manera de piloto, para extraer nuevos requerimientos y posibilidades para la herramienta.
- En la Escuela de Computación se encuentran en desarrollo numerosas aplicaciones Web que buscan optimizar los procesos internos de la misma. Esta cantidad de proyectos en desarrollo da pie a que pueda existir inconsistencia de datos y/o pérdidas de información. Como recomendación primordial tras la culminación de este proyecto, se sugiere darle continuidad al mismo con una necesaria y futura integración al Sistema CONEST, actualmente operativo en la Facultad.
- Durante el desarrollo de este proyecto se pudo distinguir una serie de requerimientos adicionales que pudieran dar pie a la extensión del mismo por varios semestres. Entre estos destacan los procesos que involucran la selección, asignación de jurados y revisión de los documentos por parte de los mismos, así como asignación de aulas y finalmente la fecha de presentación, validación de requisitos estudiantiles, entre otros.
- Dado que la aplicación desarrollada funciona actualmente con una constante y frecuente subida de archivos, los espacios de disco duro que consume pueden resultar en un problema a mediano plazo, por lo que se sugiere estudiar alternativas de alojamiento o inclusive alojamientos distribuidos para evitar el colapso de los servidores.

Referencias Bibliográficas

Asif Irshad Khan, R. J. (2011). *A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies*. *IJSI - International Journal of Computer Science Issues*, 8 (4), 441-450.

Bates, M. (2012). *Programming in CoffeeScript*. United States of America: Addison-Wesley.

Bear Bibeault, Y. K. (2010). *JQuery in Action*. United States of America: Manning.

Beighley, L. (2010). *JQuery for Dummies*. Indianapolis: Wiley Publishing Inc.

Brito, Á. (02 de 07 de 2009). *Círculo de Maquetadores*. Recuperado el 22 de 11 de 2012, de Las novedades de HTML 5 (Parte 1): <http://www.circulodemaquetadores.com/las-novedades-de-html-5-1/>

Craig Grannell, V. S. (2012). *The Essential Guide to HTML5 and CSS3 Web Design*. New York: Apress.

Dave Thomas, D. H. (2005). *Agile Web Development with Rails*. Raleigh, North Carolina: The Pragmatic Programmers LLC.

Dubouis., P. (2009). *MySQL*. Addison-Wesley.

Ediger, B. (2008). *Advanced Rails*. United States of America: O'Reilly.

Fisher, M. (13 de 07 de 2009). *...the personal blog of MIKE FISCHER*. Recuperado el 22 de 11 de 2012, de Essential Ruby on Rails plug-ins and gems: <http://mfischer.com/wordpress/blog/topics/computing/>

Fitzgerald, M. (2007). *Ruby Pocket Reference*. United States of America: O'Reilly.

HAML. (s.f.). Recuperado el 20 de 11 de 2012, de <http://haml.info/>

Hanselman, S. (22 de 07 de 2011). *Scott Hanselman's Computer Zen*. Recuperado el 22 de 11 de 2012, de CoffeeScript, Sass and LESS support for Visual Studio and ASP.NET with the Mindscape Web Workbench: <http://www.hanselman.com/blog/>

Hartl, M. (2011). *Ruby on Rails Tutorial: Learn Rails by Example*. Boston: Addison-Wesley.

Hundhausen, R. (2012). *Professional Scrum Development with Microsoft Visual Studio 2012*. Redmond, Washington: Microsoft Press.

Iván. (25 de 11 de 2011). *Ingeniería en Software*. Recuperado el 23 de 11 de 2012, de Modelo RUP: <http://ingenieriaensoftwareivan.blogspot.com/2011/11/software-blog.html>

James Shore, S. W. (2008). *The Art of Agile Development*. Sebastopol, California: O'Reilly.

Jeffrey Allan Hardy, C. C. (2007). *Beginning Rails: From novice to professional*. United States of America: Apress.

Jon Loeliger, M. M. (2012). *Version Control with GIT*. United States of America: O'Reilly.

Jonathan Chaffer, K. S. (2009). *Learning JQuery 1.3*. Birmingham: Packt Publishing.

Ken Schwaber, J. S. (10 de 2011). Scrum.org. Recuperado el 26 de 11 de 2012, de The Official Scrum Rulebook: <http://scrum.org/Scrum-Guides>

Kent Beck, C. A. (2005). Extreme Programming Explained: Embrace Change. Boston: Addison-Wesley.

Marco Casario, N. W. (2012). CSS3 Solutions - Essential Techniques for CSS3 Developers. New York: Apress.

OSC. (19 de 06 de 2011). OSC. Recuperado el 23 de 11 de 2012, de Análisis y Diseño de Sistemas - Modelos para el Desarrollo de Software: <http://osc.co.cr/analisis-y-diseno-de-sistemas-modelos-para-el-desarrollo-de-software/>

Sass-lang. (Syntactically Awesome Stylesheets) (s.f.). Recuperado el 20 de 11 de 2012, de <http://sass-lang.com/>

Wikipedia. (s.f.). Recuperado el 23 de 11 de 2012, de Component-based software engineering: http://en.wikipedia.org/wiki/Component-based_software_engineering