



UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

**Desarrollo del módulo de admisión a los Postgrados de
la Facultad de Ciencias de la Universidad Central de
Venezuela perteneciente al sistema de gestión
académica CONEST Postgrado**

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

Susana Carolina Hernández Calderón

Juan Victor Pita Cangalaya

para optar al título de
Licenciado en Computación

Tutor: Prof. Sergio Rivas

Caracas, Marzo 2015

Acta

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por los Bachilleres Susana Hernández C.I.: 18.093.593 y Juan Pita C.I.: 18.556.959, con el título “Desarrollo del módulo de admisión a los postgrados de la Facultad de Ciencias de la Universidad Central de Venezuela perteneciente al sistema de gestión académica CONEST postgrado”, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 17 de Marzo de 2015, a las 3:30 PM, para que sus autores lo defendieran en forma pública, en el aula PAIII de la Escuela de Computación, Facultad de Ciencias de la Universidad Central de Venezuela, lo cual estos realizaron mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 17 de Marzo de 2015, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor y Tutor Sergio Rivas.

Prof. Jossie Zambrano
(Principal)

Prof. Alejandro Crema
(Principal)

Prof. Sergio Rivas
(Tutor)

Dedicatorias y agradecimientos

Susana Hernandez:

Quiero agradecer y dedicar el presente trabajo a mis padres Betty Calderón y Aquiles Hernández, por haberme apoyado en los buenos y malos momentos, por estar siempre presente alentándome a continuar y alcanzar mis metas propuestas, a ustedes que han puesto todo de su parte para que pueda alcanzar este logro, los amo.

A mi gran amigo y compañero Juan Pita, porque has estado allí apoyándome en todo momento, porque trabajando juntos crecimos a nivel personal y profesional, me alegra mucho poder compartir este logro contigo.

A mi familia Calderón, a ustedes tías Judith, María, Consuelo y Nelly por ser como unas madres para mi, que me han apoyado a lo largo de todo el camino, dándome palabras de aliento, soportando mis distancias y ausencias, a ustedes tías queridas que de alguna manera contribuyeron en mi formación como persona y como profesional.

A la familia Pita Cangalaya, por todo su apoyo y atenciones durante la realización de este proyecto.

A mi prima hermana y amiga Jessica Navarro, por compartir esta gran aventura a mi lado, sin ti no habría sido lo mismo, gracias.

A los amigos y compañeros que he conocido a lo largo de mi carrera, por todo su apoyo en todo momento, por vivir juntos esta experiencia inolvidable, por todo el cariño que me han dado se han convertido más que en compañeros, mis más sinceras gracias.

A nuestros tutores por su apoyo incondicional y orientarnos a lo largo de este proyecto.

A todos nuestros profesores que aportaron sus conocimientos para formarnos como computistas íntegros. A nuestra universidad, la casa que vence la sombra, por permitirnos ser parte de ella y por ser testigo de nuestro crecimiento como profesionales.

A mi mascota, gran compañera y fiel amiga muñeca, por acompañarme cada noche de desvelo, cada día de estudio, por estar siempre allí.

A todos los perros de la universidad, quienes me acompañaron casi todas las clases, quienes estuvieron allí cuidándome en todo momento.

A todos aquellos que de alguna manera han colaborado y han formado parte de mi enriquecimiento personal.

A todos ellos muchas gracias.

Juan Pita:

Comenzaré por dar gracias a Dios por haber guiado y acompañado a lo largo de mi vida y carrera por ser mi guía y fortaleza en momentos de debilidad y brindarme una vida de experiencias, aprendizajes y sobre todo felicidad. Le doy gracias a mis padres Victor Manuel y Maria del Rosario, por los valores que me han inculcado y haberme dado la oportunidad de tener excelente educación en el transcurso de mi vida, sobre todo por ser un excelente ejemplo a seguir y su apoyo en las buenas y en las malas.

A mis hermanos Maria Eugenia y Rodrigo por representar un apoyo incondicional en este camino, sobre todo por representar la unidad familiar y demostrarme que soy su ejemplo a seguir.

A Sergio Rivas y Jossie Zambrano por todo su tiempo, apoyo y brindarme la oportunidad de desarrollar en CONEST, fue una buena experiencia que me hizo crecer profesionalmente.

A Susana Hernández por haber sido una buena compañera de tesis y amiga, por haberme tenido la paciencia necesaria y motivarme a seguir adelante en momentos de desesperación y sobre todo a su familia por el apoyo que brindo.

A Deyvid Martínez por ser un buen amigo, ya que brindo en mi presente realidad el apoyo y compadrazgo necesario para seguir adelante.

A mis hijos Wiliam Gonzáles, Daniel Dalke, Jorge Porras, Ramses Salazar y mi hija perdida Daniela Martínez por ser de los mejores amigos en este capítulo de mi vida llamado carrera universitaria y en el anterior estudios de bachillerato.

Por otra parte, este Trabajo Especial de Grado representa el fin y comienzo de una nueva etapa de mi vida, por lo que quiero expresar mi gratitud a todos quienes, de una u otra manera, me han acompañado en esta larga jornada.

Resumen

La Coordinación de Postgrado de la Facultad de Ciencias, es el ente, que se encarga de la administración y gestión de los distintos postgrados de la Facultad de Ciencias, no cuenta con un sistema automatizado para llevar a cabo sus procesos académicos y administrativos de manera ágil y estandarizada; actualmente se cuenta con información básica almacenada en hojas de cálculo y los expedientes se manejan en físico.

El presente Trabajo Especial de Grado tiene como objetivo desarrollar un sistema que permita automatizar el proceso para ingresar a un postgrado; se utilizó el método de programación ágil Programación Extrema (XP), así como tecnologías de desarrollo.

Entre las tecnologías de desarrollo se puede mencionar Ruby on Rails (RoR), que facilitó la realización de la parte práctica del presente trabajo de investigación de manera rápida, sencilla, eficiente, documentada y con un enfoque al usuario final.

Como producto final se obtuvo un módulo Web que sigue los estándares del Sistema de Control de Estudios (CONEST) para el área de postgrado denominado Postulación postgrado.

PALABRAS CLAVE: Universidad Central de Venezuela, Facultad de Ciencias, Coordinación de postgrado, académico, automatización, aplicación Web, postgrados, Ruby on Rails, preinscripción, postulación.

Índice general

Introducción	11
1. Postulación a los programas de postgrado	13
1.1. Coordinación de Postgrado	13
1.2. Sistema CONEST	20
1.2.1. Características de CONEST	21
1.2.2. Plataforma de CONEST	22
1.2.3. Estado actual de CONEST	23
2. Tecnologías para el desarrollo de aplicaciones Web	25
2.1. Aplicaciones Web	25
2.2. Tecnologías del lado del cliente	26
2.3. Tecnologías del lado del servidor	28
2.3.1. Ruby on Rails (RoR)	28
2.3.2. Sistema manejador de base de datos MySQL	31
3. Desarrollo del sistema	34
3.1. Descripción y adaptación del método XP	34
3.2. Fase de exploración	35
3.3. Fase de planificación	38

<i>ÍNDICE GENERAL</i>	7
3.4. Fase de diseño	40
3.5. Fase de codificación	41
3.6. Fase de Pruebas	42
3.7. Historias de usuario	43
3.8. Iteración 0 - Diseño y creación de la base de datos	50
3.9. Iteración 1 - Registro de postulantes.	52
3.10. Iteración 2 - Carga de currículum vitae del postulante	56
3.11. Iteración 3 - Carga de currículum vitae del postulante	62
3.12. Iteración 4 - Desarrollo del módulo del personal administrativo	66
3.13. Iteración 5 - Desarrollo del módulo del personal docente	70
3.14. Iteración 6 - Refactorización	73
3.15. Iteración 7 - Puesta en producción de la aplicación	75
Conclusiones	78
Recomendaciones	81
Bibliografía	82

Índice de figuras

1.1. Estructura organizacional de la Coordinación de Postgrado de la Facultad de Ciencias	14
1.2. Postgrados que ofrece la Facultad de Ciencias	15
1.3. Diagrama de actividad del proceso de admisión	16
1.4. Diagrama del proceso de selección	17
1.5. Planilla de preinscripción del postulante	18
1.6. Carta de aceptación a un postgrado del postulante	19
1.7. Reporte de postulación	20
1.8. CONEST 3.0	22
1.9. Plataforma de CONEST	23
2.1. Arquitectura Cliente/Servidor de tres capas	26
2.2. Estructura MVC de RoR (Rubysource.org, 2013)	29
2.3. Arquitectura de MySQL. Imagen tomada de Sicilia (2008)	33
3.1. Fase de exploración método de desarrollo XP	35
3.2. Registro del postulante y revisión de recaudos	37
3.3. Toma de decisión en la postulación	38
3.4. Fase de planificación adaptada método de desarrollo XP	39
3.5. Fase de codificación adaptada método de desarrollo XP	41

3.6. Diseño del Modelo de Datos	51
3.7. Vista registro de postulante	53
3.8. Código que registra postulante	54
3.9. Vista que permite cargar datos básicos	57
3.10. Diseño que permite la carga de estudios de pregrado	58
3.11. Diseño de vista que muestra los estudios de pregrado registrados	58
3.12. Código que permite listar, crear estudios de pregrado	59
3.13. Diseño de la vista de solicitud de postulación	63
3.14. Diseño de lista de solicitudes del postulante	64
3.15. Código que válida programa ya inscritos	65
3.16. Búsqueda de postulante desde el módulo administrativo	67
3.17. Vista del postulante desde el modulo administrador	68
3.18. Código que busca los datos del postulante desde el administrador	69
3.19. Diseño de la vista del postulante desde el módulo docente	71
3.20. Código que permite evaluar las solicitudes	72
3.21. Currículum vitae con fotografía	74
3.22. Código que permite incluir la foto del postulante al documento PDF	75
3.23. Diseño de la vista principal del modulo de postulación	77

Índice de tablas

3.1. Modelo de plan de iteración	39
3.2. Modelo de Historia de Usuario	40
3.3. Modelo de registro de pruebas de funcionamiento	43
3.4. Plan de iteración. Iteración 0	51
3.5. Plan de iteración. Iteración 1	52
3.6. Pruebas de funcionamiento. Iteración 1	55
3.7. Plan de iteración. Iteración 2	56
3.8. Pruebas de funcionamiento. Iteración 2	61
3.9. Plan de iteración. Iteración 3	62
3.10. Pruebas de funcionamiento. Iteración 3	66
3.11. Plan de iteración. Iteración 4	67
3.12. Pruebas de funcionamiento. Iteración 4	70
3.13. Plan de iteración. Iteración 5	71
3.14. Prueba de funcionamiento. Iteración 5	73
3.15. Plan de iteración. Iteración 6	73
3.17. Plan de iteración. Iteración 7	75
3.16. Pruebas de funcionamiento. Iteración 6	76

Introducción

La Coordinación de Postgrado de la Facultad de Ciencias de la Universidad Central de Venezuela, se encarga de la administración y gestión de los distintos programas de postgrados que la Facultad de Ciencias ofrece, además mantiene comunicación con los distintos comités académicos de cada postgrado específico.

Entre los distintos procesos que se llevan a cabo en la coordinación de postgrado se encuentra el proceso de preinscripción a los programas de postgrado, también llamado proceso de postulación; éste proceso no cuenta con un sistema automatizado que permita tener centralizada la información de los aspirantes a los postgrados así como mantener estandarizado el proceso, es por esta razón que se transportan documentos físicos entre los entes que intervienen en este proceso, lo que puede generar errores, aumenta la complejidad y los tiempos de respuesta en la realización de los trámites.

Para que el proceso de postulación sea ordenado surge la necesidad de mantener el control de los expedientes de los postulantes, así como de mantener informado en todo momento tanto al postulante como a personal docente y administrativo del estado del trámite.

Apoyándose en las bondades que ofrecen las aplicaciones Web, se pueden dar un conjunto de soluciones para satisfacer las necesidades que surgen a la hora de llevar a cabo este proceso, como lo es minimizar los errores que pudiesen existir, unificar el proceso en los diferentes comités de postgrado, facilitar el trabajo del personal involucrado, mejorar los tiempos de respuesta en la gestión del trámite, mejorar el desempeño del personal administrativo y disminuir los costos de recursos físicos.

Es por las razones mencionadas anteriormente se plantea el siguiente objetivo general "desarrollar un módulo Web que permita automatizar el proceso de postulación a los postgrados de la Facultad de Ciencias de la Universidad Central de Venezuela perteneciente al sistema de gestión académica CONEST postgrado".

Para llevar a cabo este objetivo se plantea:

1. Identificar el estado actual del proceso de ingreso a los programas de postgrado de la coordinación de postgrado.

2. Ampliar el modelo de datos del Sistema CONEST postgrado relacionado a la postulación.
3. Adaptar un método de desarrollo Web.
4. Desarrollar el módulo de postulación a los postgrados de la Facultad de Ciencias de la Universidad Central de Venezuela usando el método de desarrollo propuesto.
5. Colocar en producción el nuevo módulo de postulación del Sistema CONEST postgrado.

Para concretar lo planteado y cumplir con el objetivo mencionado se presentan cada uno de los capítulos que componen este documento.

- Capítulo I, Postulación a los programas de postgrado: comprende las entidades involucradas en el proceso de postulación a los programas de postgrado, como lo son la Coordinación de postgrado desde su perfil hasta el detalle de como se realiza actualmente el proceso de forma general y como es el intercambio de información con los distintos postgrados que la entidad coordina, sus principales funciones y las necesidades que presenta.
- Capítulo II, Tecnologías para el desarrollo de aplicaciones Web: comprende las bases conceptuales, donde se definen las tecnologías utilizadas para el desarrollo del módulo Web, así como sus bondades y las razones principales de su escogencia, entre estas tecnologías se puede mencionar Ruby on Rails (RoR) como la base donde se fundamenta el sistema desarrollado.
- Capítulo III, Marco Aplicativo: Se define el método de desarrollo ágil utilizado a lo largo del desarrollo del sistema, así como el detalle de la planificación, alcance y requerimientos del sistema, además comprende las distintas fases e iteraciones realizadas para lograr los objetivos propuestos y el cumplimiento de las funcionalidades del sistema.

Para finalizar se presentan las conclusiones y recomendaciones de éste trabajo, además de las referencias bibliográficas y consultadas.

Capítulo 1

Postulación a los programas de postgrado

La finalidad de este capítulo es presentar el contexto del problema, el cual se encuentra relacionado con el proceso de postulación a los programas de postgrado de la Facultad de Ciencias de la Universidad Central de Venezuela (UCV), así como también la justificación de realizar una herramienta Web para la automatización y estandarización del proceso mencionado.

1.1. Coordinación de Postgrado

La Coordinación de Postgrado de la Facultad de Ciencias de la UCV, es una estructura organizativa conformada por diversos postgrados que permiten la formación de especialistas altamente calificados que respondan a las demandas sociales, científicos que sirvan a los altos fines académicos de la universidad y del país, además permite la formación de docentes para el desarrollo de la propia institución y de otros institutos de educación superior.

El objetivo principal de la Coordinación de Postgrado de la Facultad de Ciencias de la UCV es coordinar y velar por el funcionamiento de todas las actividades concerniente a cualquiera de los postgrados que están bajo a su responsabilidad.

La estructura organizativa bajo la cual la Coordinación de Postgrado realiza sus funciones se puede observar en la Figura 1.1, donde se muestran cada una sus unidades.

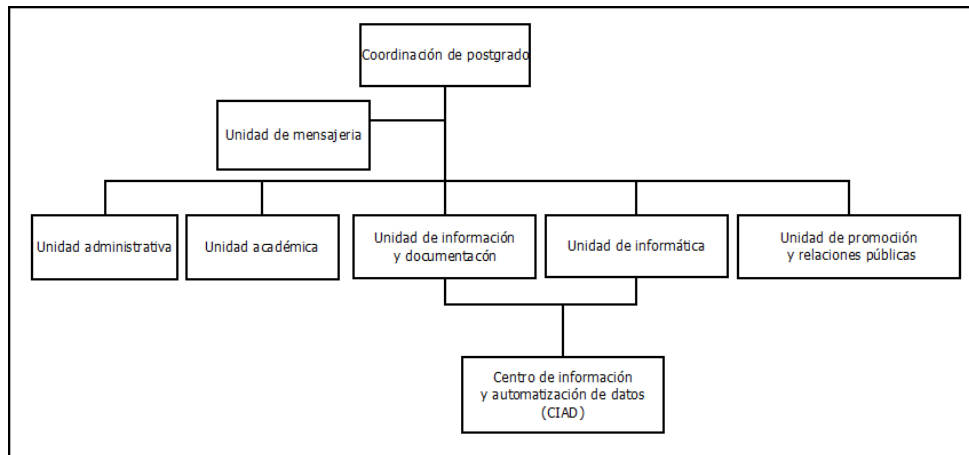


Figura 1.1: Estructura organizacional de la Coordinación de Postgrado de la Facultad de Ciencias

A continuación se describen los procesos que lleva a cabo la unidad académica; estos procesos permiten la postulación a los programas de postgrado.

Procesos de la unidad académica

El módulo o unidad académica tiene como objetivo principal controlar y ejecutar los trámites académicos pertenecientes a los distintos postgrados de la Facultad de Ciencias, así como prestar servicio y atención a los estudiantes durante dichos trámites, además permiten el manejo y control de su información durante su permanencia en el postgrado; cabe destacar que los procesos internos que se manejan en esta unidad involucran la intervención de las otras dependencias externas a la coordinación, pero que están en constante comunicación con control de estudios de cada postgrado.

En la Figura 1.2 se puede observar los distintos postgrados que se ofrecen en la Facultad de Ciencias.

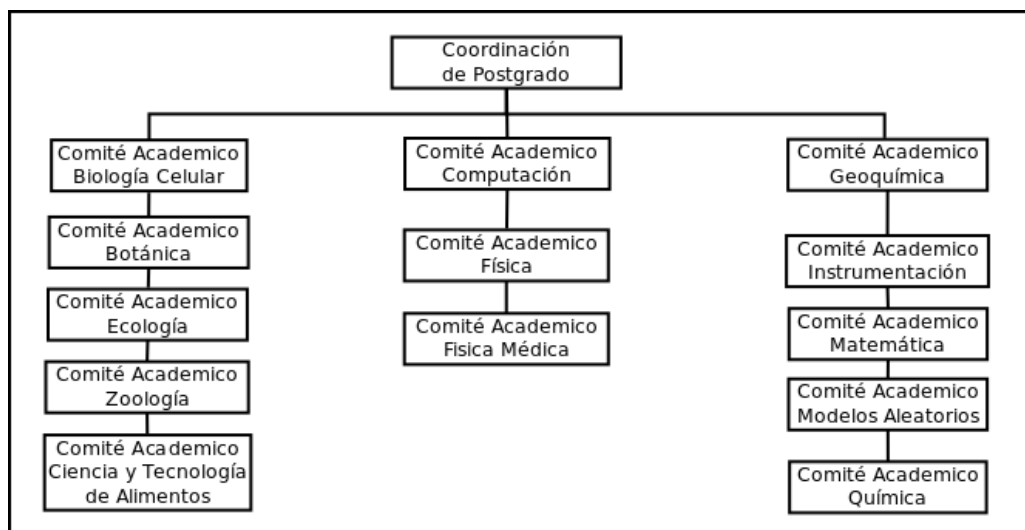


Figura 1.2: Postgrados que ofrece la Facultad de Ciencias

A continuación se describen los procesos involucrados en la postulación a los programas de postgrado que lleva a cabo la unidad académica:

- Procesos de admisión: proceso en el cual el aspirante al postgrado cancela en caja un arancel establecido por la Coordinación de Postgrado para poder postularse como estudiante. Éste llena una planilla de solicitud que obtiene en la página Web o en la oficina de la Coordinación de Postgrado, la cual entrega junto con los recaudos necesarios y una copia de la planilla en la recepción de la Coordinación de Postgrado, donde se revisa la documentación y se registra en una hoja de cálculo, se archiva el original y se envía la copia al control de estudios del postgrado solicitado. Este proceso se puede observar en la Figura 1.3.

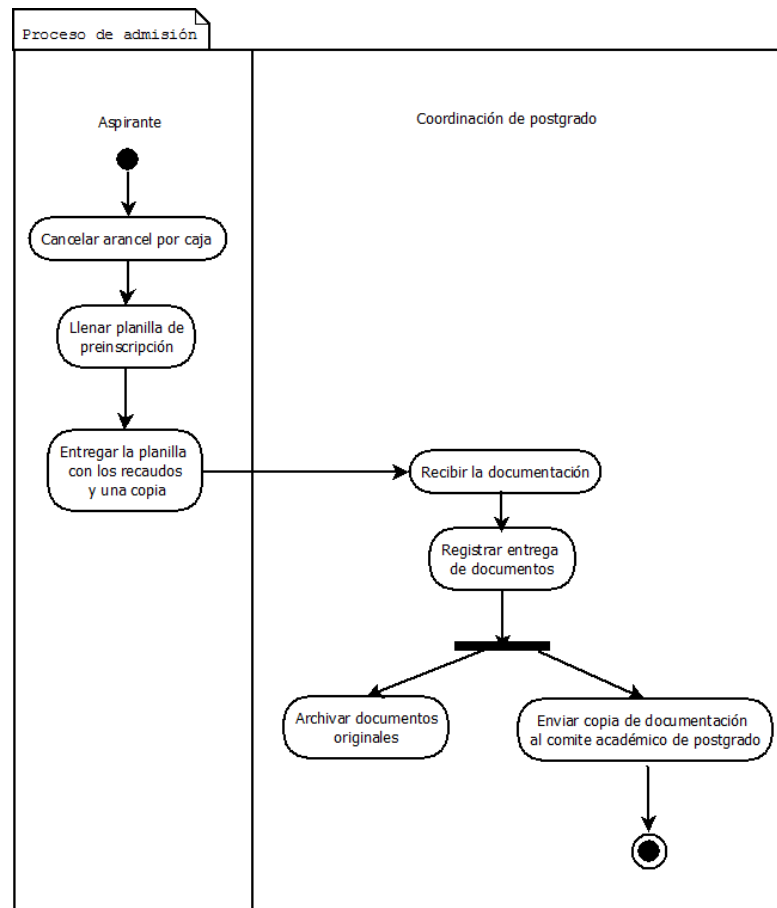


Figura 1.3: Diagrama de actividad del proceso de admisión

- Proceso de selección: proceso en el cual el aspirante al postgrado realiza una prueba de suficiencia de un idioma extranjero, evaluación de credenciales y entrevista. El comité académico del postgrado respectivo evalúa los resultados obtenidos, se redacta un oficio de admisión indicando si el mismo fue aceptado o no y se envía a la recepción de la Coordinación de Postgrado, en donde se registra el documento. En el caso de que el aspirante sea aceptado, se asigna un profesor guía y/o tutor, se crea el expediente del aspirante y un oficio de respuesta, enviando una copia al comité académico del postgrado y una copia al estudiante quien además retira una planilla en la Coordinación de Postgrado para formalizar la inscripción por Secretaría central, mientras que control de estudios del postgrado archiva el oficio de respuesta. En caso que el aspirante no sea aceptado, se desecha la documentación. Este proceso se puede observar en la Figura 1.4.

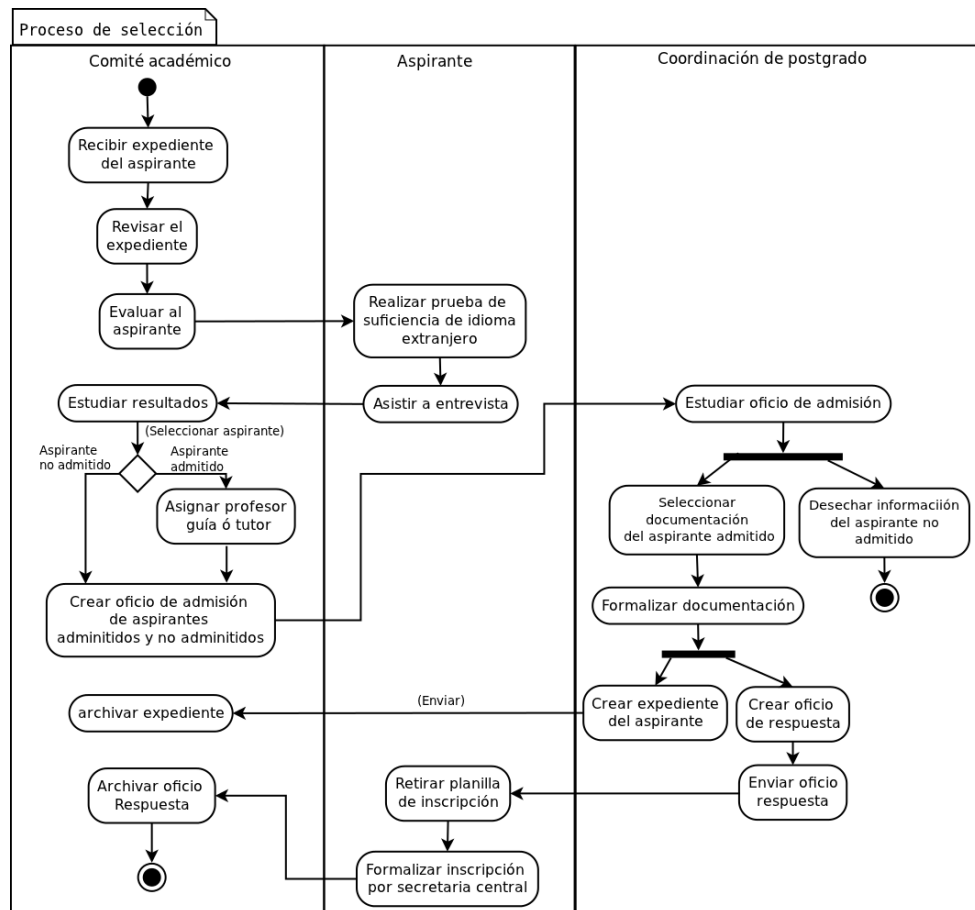


Figura 1.4: Diagrama del proceso de selección

Para recopilar información de estos procesos se realizan dos entrevistas, una en la Coordinación de postgrado y otra en un postgrado de la especialidad, con la finalidad de tener una visión global del proceso completo de admisión, estas entrevistas se describen a continuación:

- Entrevista 1: se realiza al personal administrativo de la Coordinación de Postgrado de la Facultad de Ciencias, donde explica las actividades que se llevan a cabo en la Coordinación en el proceso de admisión a los postgrados de la Facultad de Ciencias, esta admisión es el proceso que se menciona como postulación.
- Entrevista 2: se realiza al personal administrativo del postgrado de zoología, donde explica las actividades que se llevan a cabo en este postgrado durante el proceso de admisión a los postgrados de la Facultad de Ciencias.

En esta entrevista se proporcionan los documentos utilizados para este proceso, entre estos se tienen:

- La planilla de preinscripción: la cual se le da por nombre de ahora en adelante planilla de postulación a los postgrados, se puede observar en la Figura 1.5. En

este documento se detalla toda la información personal, académica, y profesional del postulante; así como también se detalla la información al postgrado que desea postularse.

Planilla de preinscripción

Encabezado

UNIVERSIDAD CENTRAL DE VENEZUELA CatEPG N° (postulante)

FACULTAD DE CIENCIAS

COMISION DE ESTUDIOS DE POSTGRADO

PLANILLA DE PREINSCRIPCION DE POSTGRADO

ESCRIBA CLARO Y EN LETRA DE IMPRENTA

FOTO

Foto del postulante

FACULTAD U/OTRO ORGANISMO ACADÉMICO	CÓDIGO	ESCUELA O DEPENDENCIA	CÓDIGO
DE NOMENCLATURA DEL CURSO	CÓDIGO	CLASE DE ESTUDIOS → AMPLIACIÓN <input type="checkbox"/> 1 MAESTRÍA <input type="checkbox"/> 3 ESPECIALIZACIÓN <input type="checkbox"/> 2 DOCTORADO <input type="checkbox"/> 4	

Datos del postgrado al que se postula

APELLIDO PATERNO DEL USU.	NOMBRE SOCIAL DEL USU.	N° CÉDULA O DE PASAPORTE
Sexo <input type="checkbox"/> F <input type="checkbox"/> M Estado Civil <input type="checkbox"/> Soltero <input type="checkbox"/> 1 <input type="checkbox"/> Casado <input type="checkbox"/> 2		NACIONALIDAD ACTUAL:
LUGAR DE NACIMIENTO, CIUDAD, ENTIDAD FEDERAL (O PAÍS)		FECHA DE NAC:
DIRECCIÓN DE HABITACIÓN		
CIUDAD	ENTIDAD FEDERAL	TELÉFONO

Datos Básicos y De contacto

TÍTULO(S) DE EDUCACIÓN SUPERIOR, CARRERAS	CÓDIGO	AÑO DE GRADO	INSTITUCIÓN DONDE OBTUVO
1-			
2-			

Estudios Pregrado

HA ESPECIALIZADO/OTROS CURSOS DE POSTGRADO SI <input type="checkbox"/> NO <input type="checkbox"/> 2 EN CASO AFIRMATIVO SEÑALE CUALES:									
CURSO	DENOMINACIÓN DEL CURSO	ENT. DONDE HIZO EL CURSO	PAÍS	DESDE	HASTA				
1- ESPECIALIZACIÓN				DÍA MES AÑO	DÍA MES AÑO				
2- MAESTRÍA				DÍA MES AÑO	DÍA MES AÑO				
3- DOCTORADO				DÍA MES AÑO	DÍA MES AÑO				
4- OTROS				DÍA MES AÑO	DÍA MES AÑO				

Estudios Postgrado

TRABAJA SI <input type="checkbox"/> NO <input type="checkbox"/> 2 EN CASO AFIRMATIVO DEDIQUE DONDE INDIQUE CERO (0) CUALQUIER COMERCIO POR LA MAS RECENTE, TODAS SUS ACTIVIDADES PROFESIONALES									
CARGO DE EMPLEADO	ADM. 1	CDC. 2	INSTITUCIÓN	DESDE	HASTA				
1-	<input type="checkbox"/>	<input type="checkbox"/>		DÍA MES AÑO	DÍA MES AÑO				
2-	<input type="checkbox"/>	<input type="checkbox"/>		DÍA MES AÑO	DÍA MES AÑO				
3-	<input type="checkbox"/>	<input type="checkbox"/>		DÍA MES AÑO	DÍA MES AÑO				
4-	<input type="checkbox"/>	<input type="checkbox"/>		DÍA MES AÑO	DÍA MES AÑO				
5-	<input type="checkbox"/>	<input type="checkbox"/>		DÍA MES AÑO	DÍA MES AÑO				
6-	<input type="checkbox"/>	<input type="checkbox"/>		DÍA MES AÑO	DÍA MES AÑO				

Experiencia Laboral

TRABAJO PUBLICADOS: TÍTULO, PÁG. FECHA, EDICIONES		DÍA MES AÑO
1-		
2-		

Trabajos publicados

FECHA DE ENTREGA →	FIRMA DEL SOLICITANTE	NOMBRE Y FIRMA DEL RECEPTOR DE LA PLANILLA
DÍA MES AÑO		

Información de La Solicitud

ACEPTADA _____

RECHAZADA _____

FIRMA DEL COORDINADOR DE POSTGRADO (SELLO)

Esta solicitud no garantiza su inscripción - Comité de Estudios para Candidato de la Facultad

Planilla de Pre-Inscripción - Universidad Central de Venezuela COORDINACIÓN DE POSTGRADO

Figura 1.5: Planilla de preinscripción del postulante

- La carta de aceptación: también se facilita la carta que se entrega a los postulantes al momento de ser aceptados en un postgrado, esta carta le permitirá inscribirse por Secretaría, se puede observar en la Figura 1.6.

Carta de aceptación

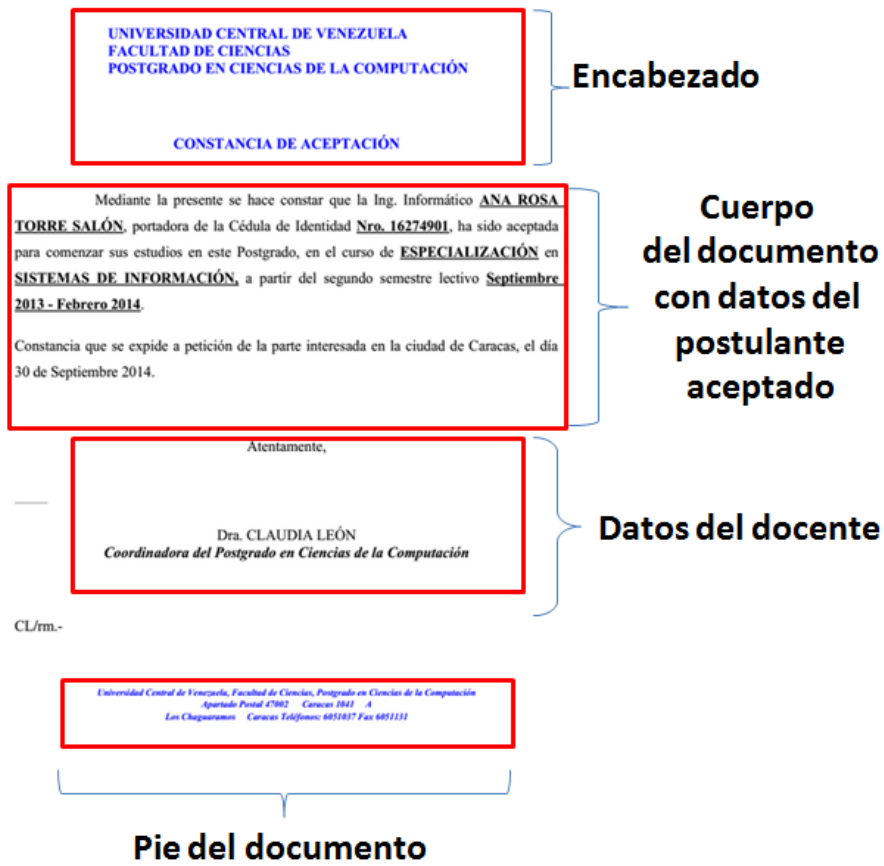


Figura 1.6: Carta de aceptación a un postgrado del postulante

- Reporte de postulación: es el reporte manejado por el personal administrativo de la Coordinación de Postgrado de la Facultad de ciencias, donde se detalla el registro de todos los postulantes en un período específico, se puede observar en la Figura 1.7.

TIPO DE PREINSCRI	SEMESTRE DE	CODIGO DE	CODIGO UXXI	NOMBRE DEL PROGRAMA	APELLIDOS	NOMBRES	TIPO DE DOCUM

NUMERO DE IDENTIDAD	NACIONALIDAD	ESTADO	SEXO	CODIGO	TELEFONO	CODIGO	TELEFONO	CODIGO	TELEFONO	E-MAIL

DIRECCIÓN	FECHA DE NACIMIENTO

Figura 1.7: Reporte de postulación

Debido a que la mayoría de los procesos llevados por la unidad académica sólo se apoyan en los documentos anteriormente descritos, como herramientas para llevar a cabo sus procesos; y estos documentos que se utilizan en los diferentes postgrados no son estándar, además de presentarse la necesidad de compartir documentación física entre los distintos postgrados, lo que puede generar errores y tiempos largos para establecer comunicación entre las partes involucradas, dificultad en la continuidad de los procesos y gran cantidad de documentos impresos; se decide realizar una solución Web.

La Facultad de Ciencias cuenta con un sistema para pregrado para la gestión académica CONEST 3.0 en el cual se maneja todo lo concerniente a los procesos académicos de pregrado de las diferentes entidades con las que se relaciona la División de Control de Estudios, las dependencias internas, externas, y las escuelas de la Facultad de Ciencias.

Debido a las bondades que presenta CONEST 3.0, así como sus características, estándares, y posibilidad de extensión se ha propuesto como sistema base en el desarrollo del módulo Web para satisfacer las necesidades de la Coordinación de Postgrado de la Facultad de Ciencias en el proceso de postulación a los postgrados. A continuación se presenta las características y la justificación del uso de CONEST como sistema base referencial.

1.2. Sistema CONEST

CONEST es el sistema de gestión académica de la Facultad de Ciencias de la Universidad Central de Venezuela, cuyo objetivo principal es automatizar los procesos

administrativos de la división de control de estudios, su visión es apoyar el cumplimiento de las normas académicas a través de las tecnologías de información y comunicación. Ha resuelto una problemática administrativa y ha sido fuente inagotable de conocimiento para quienes participan en su desarrollo: estudiantes, profesores y personal administrativo de la Facultad de Ciencias. Información que puede ser ampliada en Barrantes et al. (2011).

CONEST es el acronimo de Control de Estudios y dicho sistema surge de la necesidad de automatizar tareas y procesos académicos de diversos niveles jerárquicos con diferentes enfoques y alcances tales como inscribir, reincorporar y calificar estudiantes, generar reportes y documentos, disponer de la planta donde se imparten clases, coordinar el proceso de grado de los graduandos, entre otros. Información que puede ser ampliada en Sulbarán y Pedrozo (2009).

El sistema se encuentra formado por un conjunto de módulos que funcionan de manera integrada, donde continuamente se están integrando nuevas funcionalidades y haciendo mejoras a las existentes, lo que ha permitido una gran aceptación por parte de los usuarios del sistema.

La iniciativa nace de docentes de la licenciatura de Computación y estos en conjunto con sus estudiantes de laboratorios, pasantes, seminaristas, y tesisistas conforman el equipo de trabajo el cual a partir de octubre del 2006 comenzaron el análisis, diseño y desarrollo de lo que actualmente es una de las herramientas más importante de la Facultad de Ciencias, ha sido fuente inagotable de conocimiento para quienes participan en su desarrollo.

En la Figura 1.8 se puede evidenciar el sistema CONEST 3.0 en producción, información tomada de CONEST-UCV (2014).

1.2.1. Características de CONEST

El sistema CONEST presenta características que han permitido una gran aceptación por parte de sus usuarios, entre ellas se tienen:

- Implementado en Ruby on Rails: ya que RoR se maneja bajo la arquitectura Modelo/Vista/Controlador (MVC), permite organización e independencia de los componentes que conforman el sistema, así como flexibilidad a los cambios y extensibilidad de funcionalidades.
- Interfaz de usuario: posee una interfaz común para todos los módulos, la cual cumple con los estándares definidos con anterioridad, y basada en las mejores practicas de usabilidad, permitiendo uniformidad en los nuevos módulos.



Figura 1.8: CONEST 3.0

- Se maneja bajo la sincronización de código: ya se utiliza una herramienta de versiones de código que permite a los desarrolladores llevar el control de las modificaciones así como también trabajar en un ambiente colaborativo.
- Es un sistema escalable.
- Funcionalidades heredadas: debido a que es un sistema en un ambiente colaborativo, la mayoría de las funcionalidades generales son integradas continuamente.

Actualmente se encuentra en su versión 3.0, y a continuación se presentan las tecnologías que utiliza ya que en base a esta plataforma se reusa esta solución en un contexto similar para desarrollar el módulo Web para la postulación a los programas de postgrado de la Facultad de Ciencias.

1.2.2. Plataforma de CONEST

El sistema CONEST ha sido desarrollado con herramientas de software libre las cuales se mencionan a continuación:

- Plataforma: Linux Debian.
- Tecnologías del cliente tales como lo son: HTML5, JavaScript y JQuery , CSS3.
- Lenguaje de Desarrollo: Ruby 2.1.0.

- Framework de desarrollo: Rails 4.1.1.
- Servidores: SMTP exim para el envío de correo, MYSQL para el manejo de la base de datos, y Puma como servidor.
- Manejador de Versiones: GIT para mantener las distintas versiones de código.
- Algunas Gemas: SASS para el uso de estilos, CoffeScript para facilitar el manejo de código JavaScript, Haml para facilitar el manejo de código HTML, Prawn para la generación de documentos en formato PDF y Writeexcel para la generación de documentos en formato Excel, entre otros.

En la Figura 1.9 se puede evidenciar la plataforma del sistema CONEST, información tomada de Barrantes et al. (2011).

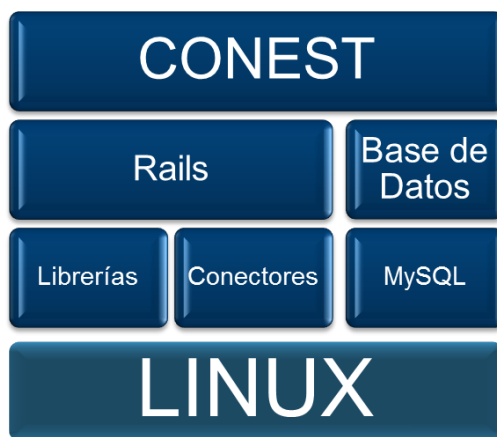


Figura 1.9: Plataforma de CONEST

Estas tecnologías y herramientas utilizadas han sido seleccionadas cuidadosamente para brindar un alto rendimiento al sistema, así como facilitar su desarrollo y permitir su extensibilidad, de forma rápida y estándar.

1.2.3. Estado actual de CONEST

En la actualidad CONEST se encuentra en producción en su versión 3.0, con más de 200 funcionalidades donde se gestionan los procesos administrativos de la división de control de estudios de la Facultad de Ciencias, esta versión presenta mejoras en relación a reingeniería de procesos internos de la aplicación que permiten eliminar redundancias de funcionalidades y fallas operativas, así como actualización de las herramientas a versiones estables, funcionalidades adicionales, rediseño y ajuste de la estructura de la base de datos,

una interfaz de usuario nueva que permite la consistencia y unificación de la misma en los distintos módulos, mejorando la usabilidad del sistema; presenta además un código organizado y la incorporación de herramientas para pruebas automatizadas.

Debido a las bondades que presenta CONEST y su aceptación por parte de los usuarios, es base fundamental en el desarrollo del módulo Web para automatizar el proceso de postulación a los postgrados de la Facultad de Ciencias de la Universidad Central de Venezuela; a continuación se presentan las tecnologías Web utilizadas durante el desarrollo de la aplicación.

Capítulo 2

Tecnologías para el desarrollo de aplicaciones Web

En este capítulo se presenta un conjunto de tecnologías con la finalidad de conocer las características de cada una ellas, dado que ofrecen un desarrollo de software ágil, maquetación acelerada, código limpio, solución a problemas comunes y compatibilidad entre navegadores, además ayuda al trabajo en equipo mediante bibliotecas y/o funcionalidades ya creadas para que sean utilizadas.

Gran parte de estas tecnologías han sido seleccionadas de forma tal que se cumpla con los estándares del sistema CONEST 3.0, el cual es base de desarrollo para el módulo de postulación a los postgrados de la Facultad de Ciencias.

2.1. Aplicaciones Web

Una aplicación Web (web-based application) es aquella que puede ser accedida por los usuarios a través de un navegador. Estas aplicaciones son ligeras, actualizables en tiempo real y mediante el uso de internet, el cliente puede acceder desde cualquier parte del mundo, información que puede ser ampliada en Techterms.com (2014).

Arquitectura Cliente/Servidor

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos, información que se puede ampliar en Ecured.cu (2014)Ecured.cu (2014).

En esta arquitectura se denomina cliente al proceso que solicita los recursos (navegador o explorador) y el servidor (servidor Web) al proceso que responde a las solicitudes, estas solicitudes se envían mediante un protocolo estandarizado (HTTP).

La variante de la arquitectura Cliente/Servidor más utilizada en la actualidad es la de tres capas, la cual se muestra en la Figura 2.1, y esta constituida por un código de presentación, de procesamiento de datos y de almacenamiento de datos.

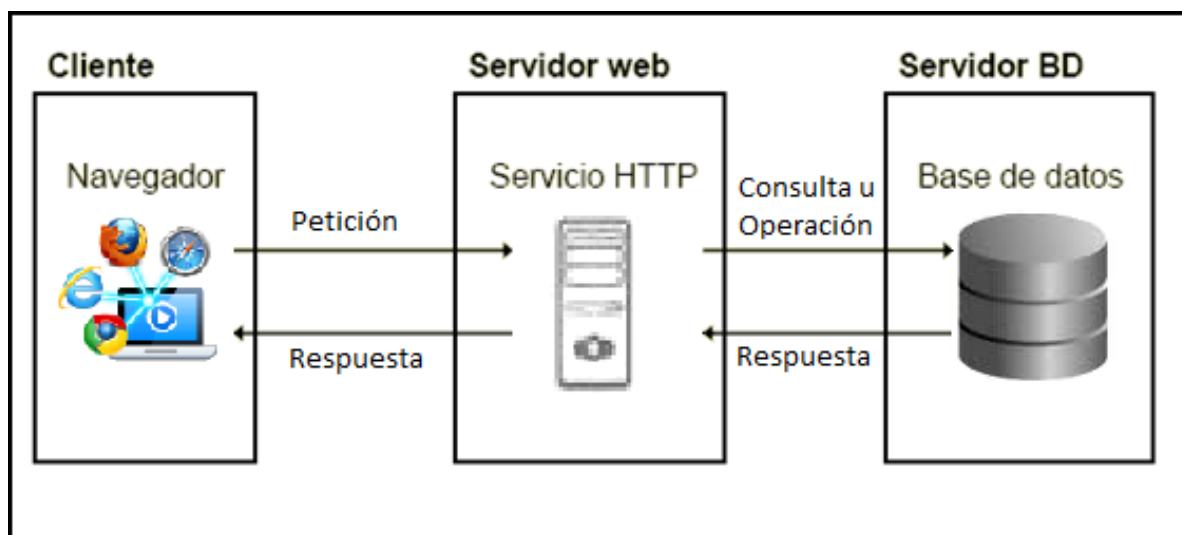


Figura 2.1: Arquitectura Cliente/Servidor de tres capas

A continuación se estudiarán las tecnologías de desarrollo de software más utilizadas en la actualidad, tanto del lado del cliente como del servidor.

2.2. Tecnologías del lado del cliente

Las tecnologías del lado del cliente son las páginas dinámicas que se ejecutan en el navegador del usuario. En estas páginas toda la carga de procesamiento de los efectos y funcionalidades la soporta el navegador. Usos típicos de las páginas de cliente son efectos especiales, validación de formularios, cálculos, etc, información que puede ser ampliada en Zenkius.blogspot.com (2008).

Para satisfacer la necesidad de integración y estandarización de las tecnologías del lado del cliente así como de agilizar el desarrollo de aplicaciones Web, y satisfacer las nuevas demandas como las móviles, se desarrolló HTML5 el cual se describe a continuación.

Lenguaje de marcas de hipertexto versión cinco

Durante los últimos años y en la actualidad HTML5, se usa para el desarrollo de muchas páginas Web, se puede definir como: “nuevo concepto para la construcción de sitios Web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.” (Gauchat, 2012). La simple versión de HTML propuesta sirvió como base para la creación de este estándar, además de la combinación con JavaScript lenguaje que permite al desarrollador crear aplicaciones dinámicas, complemente funcionales y CSS concentrado en convertir la estructura del documento utilizable y atractiva a la vista, “la combinación perfecta de estas tres tecnologías (HTML, CSS y Javascript) en una poderosa especificación está volviendo a Internet la plataforma líder de desarrollo.” (Gauchat, 2012). Otra de las bondades que brinda el nacimiento de este estándar es la posibilidad del uso de menos recursos, abaratando costos en tiempo de desarrollo y código más limpio.

Componentes de HTML5

“HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interfaces de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5.” (Gauchat, 2012).

- Lenguaje de marcas de hipertexto (HTML): Es un lenguaje de programación que permite describir la estructura y el contenido en forma de texto, mediante etiquetas; no es un lenguaje compilado por lo tanto los errores de programación no son detectados. Su principal base son los hipervínculos que permiten la navegación por el Word Wide Web. En HTML5 se agregan nuevas etiquetas que integran nuevas funcionalidades a HTML, permitiendo un desarrollo más simple y lógico en el código.
- Hojas de estilo en cascada (CSS): es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento (tamaño, color, fondo, bordes, entre otros), ya que “La Web demanda diseño y funcionalidad, no sólo organización estructural o definición de secciones” (Gauchat, 2012). El diseño del estilo de una aplicación Web es muy importante ya que HTML por sí mismo no brinda ésta funcionalidad. La responsabilidad de HTML5 es brindar mejoras a la última versión de CSS en la forma de organizar documentos y trabajar con sus elementos.
- Javascript: Este lenguaje forma parte de la base de HTML5, y es considerado uno de sus pilares mas fuertes. “Javascript es un lenguaje interpretado usado para múltiples

propósitos pero sólo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio.” (Gauchat, 2012).

Interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje de funciones elementales, esta nuevas APIs como (Web Storage, Canvas y otras), tienen como objetivo principal brindar poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la Web.

2.3. Tecnologías del lado del servidor

Las tecnologías que se encuentran del lado del servidor se encargan de procesar las solicitudes realizadas por parte del cliente, y permanecer en escucha de nuevas solicitudes, a continuación se mencionan alguna de estas tecnologías, información que puede ser ampliada en [Ecured.cu \(2014\)](#).

2.3.1. Ruby on Rails (RoR)

Ruby es un lenguaje de programación creado en el año 1993. Es interpretado, reflexivo y orientado a objeto, su distribución es bajo una licencia de software libre; su creador Yukihiro “Matz” Matsumoto, lo hizo con la intención de formar un nuevo lenguaje que incorporara tanto la programación funcional como la programación imperativa, información que puede ser ampliada en [Kehoe \(2013\)](#).

Rails es un framework de desarrollo Web para trabajar con el lenguaje de programación Ruby, esto significa que brinda una plantilla para simplificar el desarrollo de la programación de una aplicación, ofrece una amplia variedad de bibliotecas adicionales, llamadas gemas, que proporcionan muchas funcionalidades comunes en proyectos Web, por lo tanto, evita desarrollar código ya implementados por terceros.

Ruby on Rails en conjunto presentan un ambiente de trabajo completo, y es conocido como RoR; fue escrito por David Heinemeier Hansson a partir de su trabajo en Basecamp, una herramienta de gestión de proyectos.

La filosofía de Rails sigue dos grandes principios:

1. No te repitas (Dont Repeat Yourself, DRY): sugiere que escribir el mismo código una y otra vez es malo.
2. Convención sobre configuración (Covention Over ConFfiguration): se refiere a realizar siempre las mismas configuraciones para lograr un objetivo específico, RoR ofrece una convención para alcanzar dicha meta lo más rápido y fácil posible.

La rapidez en el desarrollo de aplicaciones en RoR está fundamentada en la idea de construir la aplicación basándose en el diseño de arquitectura Modelo/Vista/Controlador (MVC) 2.2, la correspondencia que se genera con el patrón MVC viene dada por:

- ActiveRecord: provee una interfaz entre las tablas de una base de datos relacional, y el código de la aplicación escrita en Ruby que manipula los registros de la base de datos.
- ActionView: es la encargada de recibir los datos de las clases del controlador, además existe una vista para cada acción del controlador y son éstas las que definen la presentación que tendrán los datos en los archivos HTML que se enviarán al navegador.
- ActionController: invoca a la lógica de la aplicación, que a su vez manipula los datos de la clase ActiveRecord y muestra los resultados usando la clase ActionView. Cada controlador es una clase de Ruby, y cada método dentro de dicha clase representa una acción.

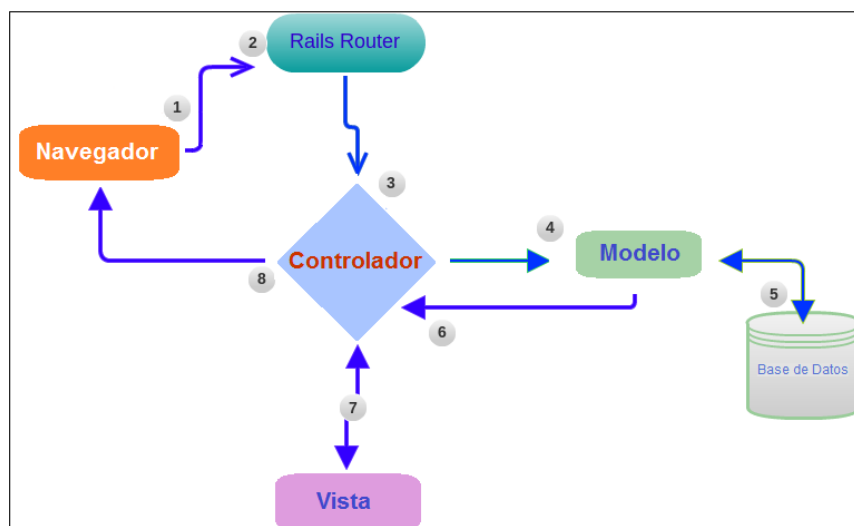


Figura 2.2: Estructura MVC de RoR (Rubysource.org, 2013)

A continuación se mencionan las principales bondades que ofrece el desarrollo en RoR:

- El enfoque orientado a objetos lo cual hace mantiene que el código sea limpia, eficaz y sencilla de entender.
- Interfaz para la manipulación de objetos dentro de bases de datos: por lo tanto todas las acciones/operaciones que se hacen dentro de la base de datos se realizan usando código Ruby solamente, por tanto la aplicación no es dependiente del tipo de base de datos que se este utilizando para almacenar la información.
- El ambiente de desarrollo de pruebas: tiene a su disposición un asistente semi-inteligente para comprobar el código y verificar si algo está fallando o todo está bien sin necesidad de realizarlo de forma manual.
- El desarrollo: RoR es una manera fácil, rápida y práctica de llevar a cabo proyectos Web en un ambiente dinámico y de rápido crecimiento al trabajar, incorporando la extensibilidad de la aplicación y la refactorización de la misma durante y después del desarrollo.

RoR es la base del Sistema de gestión académica CONEST para el área de pregrado y control de estudios; y dado que dicho sistema ha tenido gran aceptación se tomo como base para el desarrollo del módulo Web propuesto para el área de postgrado, por ende se utiliza RoR para poder incorporarlo a CONEST.

Gemas de RoR

Los paquetes de aplicación o bibliotecas de Ruby on Rails son llamadas RubyGems y se consiguen en el repositorio, las mismas facilitan la desarrollo de diversas funcionalidades a las aplicaciones en desarrollo de forma rápida y sencilla, información tomada de Rubygems.org (2014). Entre las gemas relevantes utilizadas para el desarrollo del módulo de postulaciones a los programas de postgrado de la Facultad de Ciencias, se tienen:

- Puma: es un simple, rápido y altamente concurrente servidor HTTP 1.1 para aplicaciones Web Ruby. Es usado tanto en entornos de desarrollo como en entornos de producción, información tomada de Rubygems.org (2014).
- Prawn: permite la generación de PDF de forma rápida y ágil, y debido a la necesidad de manejar documentos en este formato por parte de la Coordinación de Postgrado así como de los entes que la integran, es incorporado en el desarrollo de la aplicación, información tomada de Rubygems.org (2014). Algunas de las características que ofrece son:
 - Dibujo de vectores, líneas, polígonos, curvas, elipse.

- Soporte de imágenes PNG y JPG.
 - Herramientas para generar reportes y soporte de paginación.
- Writeexcel: permite la generación de documentos Excel de forma rápida y ágil, y debido a la necesidad de presentar reportes en este formato por parte de la Coordinación de Postgrado así como de los entes que la integran, es incorporado en el desarrollo de la aplicación, información tomada de Rubygems.org (2014). Algunas de las características que ofrece son:
- Generar un archivo compatible con el excel 97, 2000, 2002, 2003 y 2007; además de ser compatible con las aplicaciones de hoja de cálculo Gnumeric y OpenOffice.org.
 - Soporta varias hojas de cálculo por libro.
 - Soporta distintos formatos para el libro y las celdas.
 - Permite el manejo de texto, números, fórmulas, hipervínculos, imágenes y gráficos.
- Paperclip: permite el manejo y almacenamiento de documentos en disco tan fácil como sea posible, información tomada de Rubygems.org (2014). Alguna de las características que ofrece:
- Manejo de archivos como atributos de los modelos.
 - No se almacenan o eliminan archivos hasta que se llama ActiveRecord::Base save.
 - Gestiona validaciones basadas en el tamaño y la presencia, si es necesario.
 - Permite transformar imágenes mediante el uso de la aplicación ImageMagick.
 - Los archivos adjuntos se guardan en el sistema de archivos, y son fácilmente referenciados.

2.3.2. Sistema manejador de base de datos MySQL

MySQL es un sistema de gestión de bases de datos relacional, de código abierto, licenciado bajo GPL de la GNU, fundamentalmente escrito en lenguaje C/C++, y está disponible para las plataformas: Linux, Solaris, Ms Windows y Mac OS X; el lenguaje de programación que utiliza es el lenguaje de consultas estructurado (SQL), que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. Sicilia (2008).

Algunas características relevantes de MySQL 5.5 son:

- El principal objetivo es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas operativos o plataformas.
- Cada base de datos cuenta con tres archivos: uno de estructura, uno de datos y uno de índice, soporta hasta 32 índices por tabla.
- Arquitectura que soporta multi-hilos, múltiples clientes tienen acceso concurrente.
- Guarda en memoria cache los resultados de las consultas más comunes.
- Sistema de privilegios de usuarios, potente y flexible.
- Escalabilidad, bases de datos con contenidos de mas de 50 millones de registros.
- Soporte de varios lenguajes y juego de caracteres.
- Amplio soporte para aplicaciones que requieren bases de datos.
- Permite el uso de procedimientos almacenados (Store Procedures), disparadores (Triggers), vistas y planificador de eventos.
- Flexible sistema de contraseñas y gestión de usuarios, con buen nivel de seguridad en los datos.
- La base de datos MySQL y el motor de almacenamiento InnoDB han sido mejorados para brindar desempeño y escalabilidad óptimos, cuando se ejecutan en múltiples CPU, hardware de varios núcleos y sistemas operativos más modernos. Además, con la versión 5.5, InnoDB ahora es el motor de almacenamiento predeterminado de MySQL Database, lo que ofrece transacciones ACID, integridad referencial y recuperación ante averías.
- Mayor disponibilidad.

Arquitectura de MySQL

MySQL sigue el modelo de la arquitectura Cliente/Servidor, donde el cliente realiza peticiones al servidor (terminal de comandos, aplicaciones de escritorio, aplicaciones Web) y el servidor es el que gestiona las respuestas (MySQL server), como se puede observar en la Figura 2.3

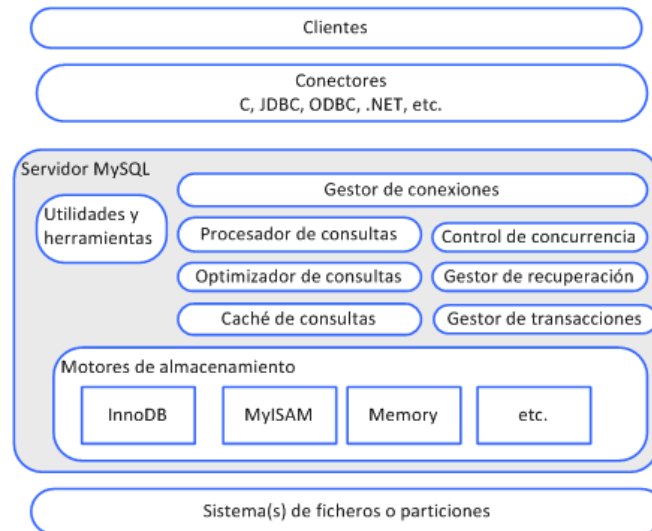


Figura 2.3: Arquitectura de MySQL. Imagen tomada de Sicilia (2008)

El manejo de tecnologías no es el único pilar en el desarrollo de software, también es necesario aplicar un método de desarrollo de software, es por esto que en el siguiente capítulo se describe el método utilizado en el desarrollo del módulo Web para las postulaciones a los postgrados de la Facultad de Ciencias.

Capítulo 3

Desarrollo del sistema

En este capítulo se describe el desarrollo del proceso de postulación y cómo se realiza la adaptación del método ágil de desarrollo de software Programación Extrema (XP), además se describe de forma detalla cada fase e iteración del desarrollo incremental del módulo, con la finalidad de poner en producción el módulo de postulación a los programas de postgrado para la Facultad de Ciencias.

3.1. Descripción y adaptación del método XP

La Programación Extrema (XP) es un método ligero, iterativo e incremental, el cual se utiliza para desarrollar software en grupos de pequeños programadores, donde la codificación es la actividad primordial sobre la documentación exhaustiva Escalante y Ovalles (2010). Este método formulado por Kent Beck, se basa en la simplicidad y tiene como objetivo principal la satisfacción del cliente; para lograrlo toma en cuenta los siguientes valores fundamentales: simplicidad, comunicación, retroalimentación, coraje y respeto.

Se diferencia de los métodos tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad, ya que se considera que los cambios de requisitos a lo largo de un proyecto son un aspecto natural, inevitable e incluso deseable en su desarrollo.

En esta sección se describen las fases que se llevaron a cabo y la adaptación de XP para el desarrollo del módulo de postulación de postgrado de la Facultad de Ciencias.

3.2. Fase de exploración

En esta fase el cliente le describe al equipo de desarrollo las funcionalidades con las que debe contar el sistema, estos requerimientos y funcionalidades son documentados en un formato llamado historias de usuario; las cuales se plantean y escriben junto con los desarrolladores.

Cada historia de usuario describe las características que deben ser adicionadas para las posteriores entregas del producto.

Luego de plantear a grandes rasgos las historias de usuario, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas a utilizar en el proyecto. Se prueban las tecnologías y se exploran las posibilidades de la arquitectura del sistema.

La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño del proyecto. En la figura (3.1) se muestra la fase de exploración adaptada en la cual se hace la investigación, levantamiento de información para elaborar las historias de usuario y propuesta del proceso para el desarrollo del modulo de postulación de postgrado. Luego que se generan todas las historias se pasa a la fase de planificación.

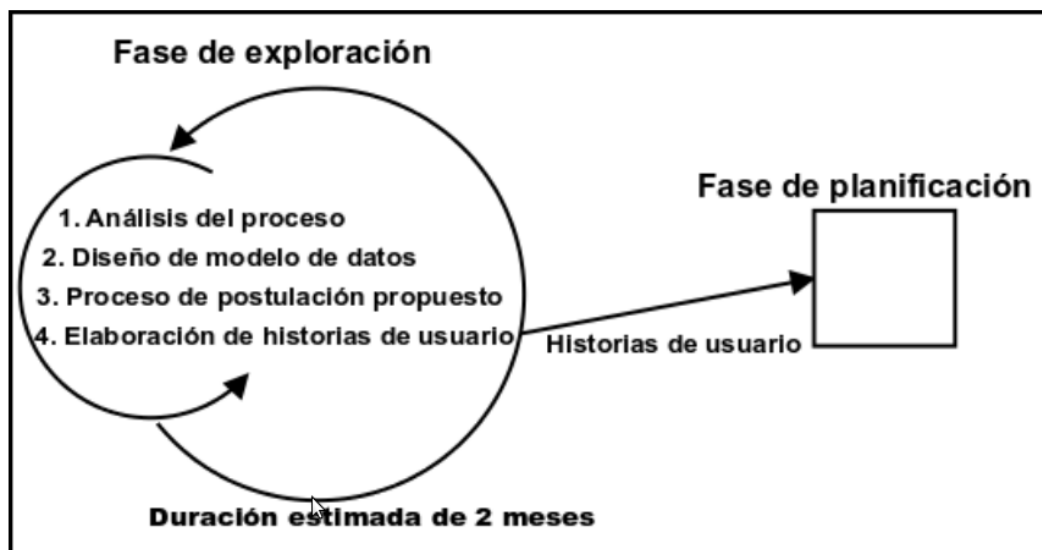


Figura 3.1: Fase de exploración método de desarrollo XP

Previo al desarrollo de la aplicación en la fase de exploración, se realiza un análisis del proceso de postulación que compete a la Coordinación de Postgrado y a los distintos postgrados de la Facultad de Ciencias de la Universidad Central de Venezuela, con la finalidad de optimizar el proceso, detectar fallas de comunicación entre los entes que intervienen en el proceso, facilitar, simplificar y adaptar a las nuevas necesidades el proceso actual.

Este análisis divide el proceso en tres actividades fundamentales:

- Registro del postulante: actividad en la cual el postulante se registra en el sistema en línea, almacena toda la información relacionada a su currículum vitae, junto con los documentos que avalan dicha información; una vez culminado el registro se procede a realizar la postulación a los programas de postgrado que seleccione el postulante con esto finalmente se puede genera el documento PDF que se debe presentar en la Coordinación de Postgrado de la Facultad de Ciencias, junto con los documentos originales y el recibo de pago respectivo, todo esto con la finalidad de formalizar la postulación. El sistema luego del registro de una solicitud de postulación la coloca en estatus nueva.
- Revisión de recaudos: actividad en la cual el personal administrativo de la Coordinación de Postgrado de la Facultad de Ciencias, recibe los recaudos del postulante, valida la documentación y en caso de estar correcta, registra en sistema el pago de la inscripción, archiva la documentación, firma y sella la planilla de inscripción entregando una copia al postulante. El sistema luego del registro de pago cambia el estatus de la solicitud nueva a entregada.

En la Figura 3.2 se puede observar las actividades: registro del postulante y revisión de recaudos.

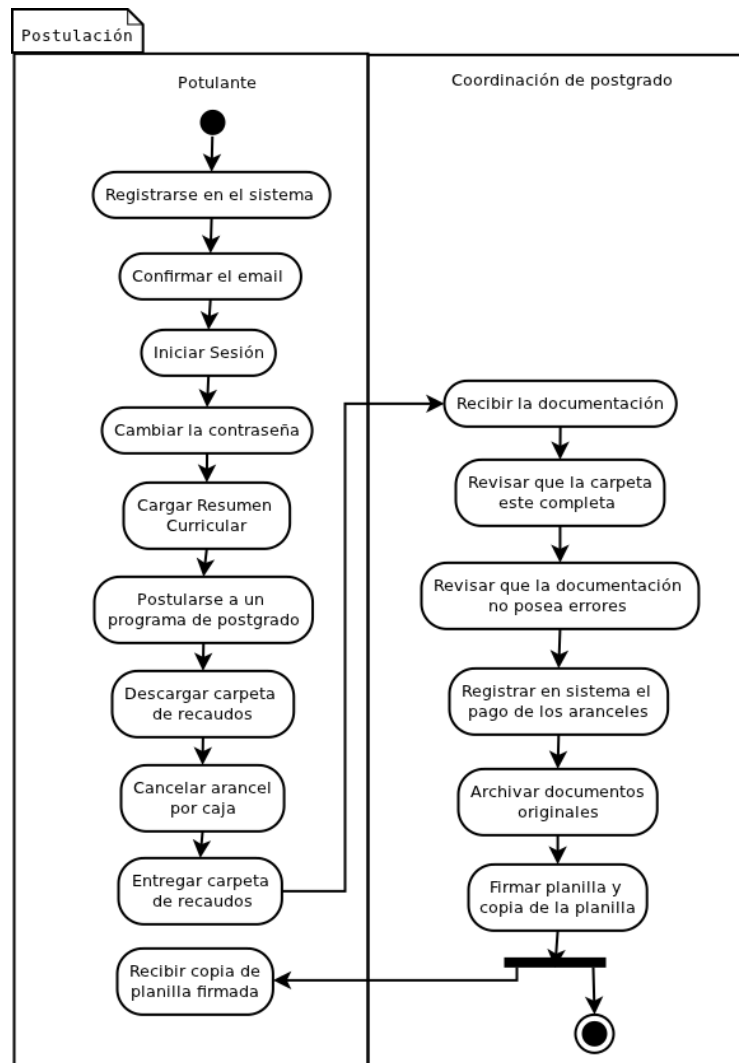


Figura 3.2: Registro del postulante y revisión de recaudos

- Toma de decisión: actividad en la cual el comité académico puede visualizar en el sistema las solicitudes de postulantes que cumplen con los recaudos completos, han formalizado el pago y tienen estatus de solicitud entregada. Permitiendo a los docentes agendar entrevistas, para luego admitir o no admitir las solicitudes, notificando al postulante por correo electrónico el estatus de su solicitud; en caso de que el estatus sea admitido activar las cartas de aceptación para que el postulante pueda descargarla. En el caso de que el estatus de la solicitud sea no admitido la Coordinación de Postgrado pueda deshacer toda la información registrada.

En la Figura 3.3 se puede observar la actividad de toma de decisión.

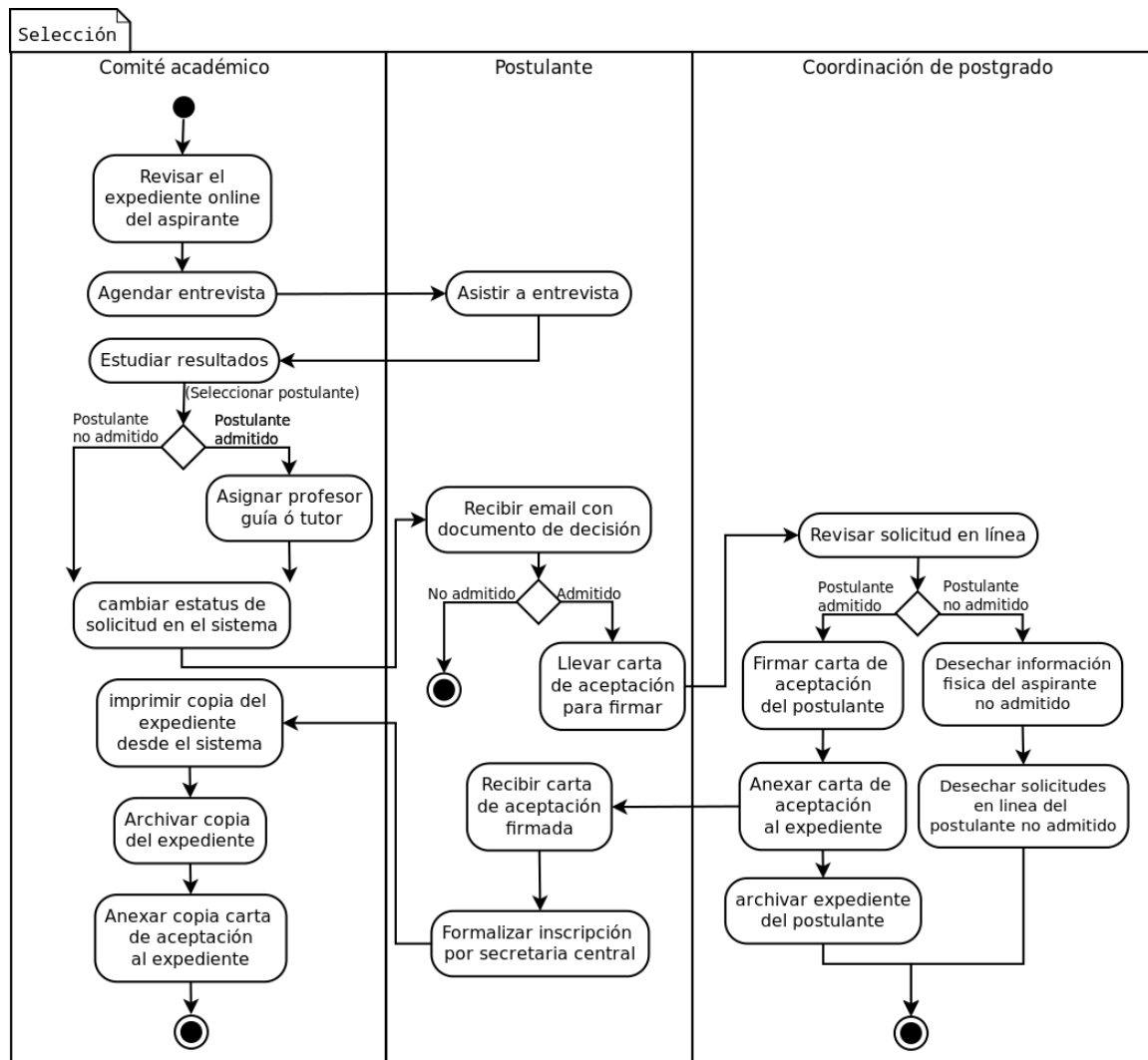


Figura 3.3: Toma de decisión en la postulación

Estas actividades permiten agilizar el proceso de postulación, así como también permiten a los postulantes registrarse cómodamente, facilita reportes al área administrativa y docente, evita de esta manera que viaje documentación física entre los distintos entes que intervienen en el proceso.

Cabe destacar que este análisis permite al grupo desarrollo tener una idea global del antes y después del proceso de postulación a los programas de postgrado de la Facultad de Ciencias, y con esto se procede a desarrollar la aplicación, se usa una adaptación de la fase de planificación del método de desarrollo ágil XP, la cual se describe a continuación.

3.3. Fase de planificación

En la fase de planificación se definen a través de las historias de usuario las principales actividades y funciones del sistema. En el desarrollo del módulo de postulación

de postgrado, esta fase permite establecer los planes de iteración y dividir el proyecto en iteraciones, con las que se estiman los esfuerzos para cumplir con el tiempo de entrega. En la figura (3.4) se muestra la adaptación de la fase de planificación y los artefactos que se usan.

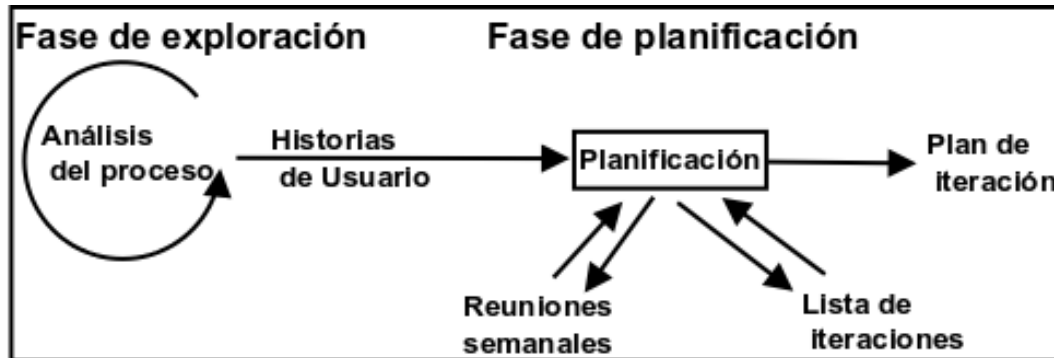


Figura 3.4: Fase de planificación adaptada método de desarrollo XP

- Plan de iteración: las iteraciones simbolizan los cambios incrementales generados a través de las pruebas y retroalimentaciones repetidas que a futuro dan como resultado un sistema estable pero en evolución. Estas pueden ser de dos tipos principalmente: por objetivo o por lapsos de tiempo. En este caso las iteraciones se realizan por objetivo, donde el cumplimiento del mismo, comprende la puesta en marcha de las historias de usuario. Para cada iteración la duración aproximada es de dos a tres semanas.

En principio se realizó un análisis global del estado actual del proceso y del levantamiento de información, del cual surgió una lista de historias de usuario, organizadas y desarrolladas en un conjunto de iteraciones.

Se utiliza un plan de iteración, como se aprecia en la tabla (3.1) al inicio de cada iteración, este contiene el número de iteración, una descripción, la fecha de inicio y la fecha de fin de la iteración, el número, la fecha de inicio del desarrollo de la historia, nombre y tipo de historia (Nueva, Modificación/mejora).

Tabla 3.1: Modelo de plan de iteración

Iteración:			
Descripción:			
Fecha inicio/ Fecha fin			
Número	Fecha	Historia	Tipo

- Historias de usuario: para el desarrollo de este módulo se usan historias de usuario para agrupar el conjunto de requerimientos funcionales, las cuales brindan una breve descripción del problema, reemplazan requisitos y dan una visión a la creación de las pruebas de funcionamiento.

Como se puede apreciar en el siguiente tabla (3.2), se utiliza una tabla que proporciona el número de historia, el nombre de la historia, estimación de tiempo de desarrollo y una descripción de la funcionalidad.

Tabla 3.2: Modelo de Historia de Usuario

Número:	Descripción
Estimación:	
Descripción:	

- Reuniones semanales: el objetivo de tener reuniones es mantener la comunicación entre el equipo y el cliente discutiendo problemas y soluciones. La agenda de reuniones con el cliente es cada dos semanas, estas reuniones, tienen como objetivo realizar las pruebas de funcionamiento. Las reuniones entre el equipo de desarrollo son semanales, permitiendo aportar ideas de solución a problemas en alguna historia de usuario en desarrollo, estas son netamente informativas.

3.4. Fase de diseño

El método de desarrollo XP hace especial énfasis en los diseños simples y claros. Esta fase se basa en los siguientes artefactos para el diseño:

- La simplicidad: en este Trabajo Especial de grado se usa el desarrollo de diagramas y modelos que permiten tener una fácil comprensión de los problemas asociados a cada iteración además de prototipos interfaz, en algunos casos se usa el diseño de tablas de base de datos, así como el diseño de algoritmos en papel de un proceso.
- La recodificación (“refactoring”): esta consiste en escribir nuevamente parte del código, sin cambiar su funcionalidad, a los efectos de hacerlo más simple. Se usa este concepto ya que, XP sugiere recodificar cada vez que sea necesario, pues si bien

puede parecer una pérdida tiempo innecesaria, los resultados de esta práctica tienen sus frutos en las siguientes iteraciones y en los casos donde es necesario ampliar o cambiar alguna funcionalidad.

3.5. Fase de codificación

XP propone que el grupo de desarrollo debe ser pequeño, dos (2) programadores y el cliente. Los programadores deben trabajar en pareja para así fomentar el intercambio constante de ideas a lo largo del desarrollo. Las pruebas unitarias deberán ser creadas antes del código fuente, así se asegura una programación ágil con bajas probabilidades de errores. Se deben realizar las integraciones constantemente para evitar obtener un software fragmentado y difícil de unir. Además también se evitan las confusiones de los programadores. Recomienda dejar para el final la optimización del código y también utilizar estándares de programación. En la figura 3.5 se muestra la adaptación de la fase de codificación.

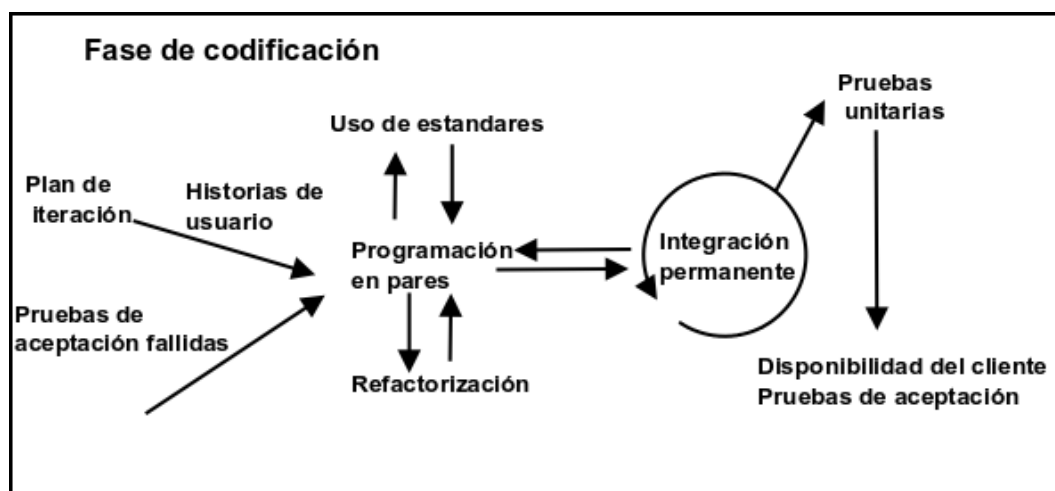


Figura 3.5: Fase de codificación adaptada método de desarrollo XP

- Disponibilidad del cliente: uno de los requerimientos de XP, es tener al cliente disponible durante todo el proyecto, esto no solamente como apoyo para el equipo de desarrollo, sino también formando parte del equipo.
- Uso de estándares: el método XP promueve la programación basada en estándares de manera que sea fácil y entendible por todo el equipo y que facilite la recodificación, se sigue el estándar de desarrollo que se usa en CONEST en la versión 3.0, tanto para el desarrollo como para el diseño e implementación de base de datos.
- Programación en pares: El método de desarrollo XP propone, que los programadores deben trabajar en parejas para así fomentar el intercambio de ideas a lo largo del desarrollo y dejar para el final la optimización del código.

En éste Trabajo Espacial de Grado, la codificación se realiza en parejas, pero no se trabaja en conjunto con la misma historia de usuario sino que, cada programador se encarga de realizar y alcanzar el objetivo deseado con cada requerimiento, dependiendo de la complejidad del mismo, es decir, mientras más complejo es el requerimiento, se trabaja en parejas en el mismo lugar de trabajo, mientras que, si la complejidad no es tan alta, cada programador realiza la programación por separado, manteniendo todo el tiempo la comunicación entre los integrantes del equipo, facilitando que la información fluya lo más eficientemente posible.

- Integraciones permanentes: XP promueve publicar lo antes posible las nuevas versiones estables del código libre de errores, aunque no sean las últimas.

3.6. Fase de Pruebas

Siguiendo los lineamientos del método XP, para esta fase se realizan las pruebas de funcionamiento, en las cuales el cliente verifica el correcto funcionamiento de lo que esta probando, cuando se pasan las pruebas de funcionamiento se considera que la correspondiente historia de usuario se ha completado y se inicia una nueva tarea, por otra parte se obvia la automatización de pruebas. Otra de las pruebas que propone XP, son las pruebas unitarias, cada uno de los módulos pasa por estas antes de ser liberado, luego se guardan para que un futuro sirvan en caso de recodificar, cambiar o corregir errores.

Con respecto a esta fase se realizan pruebas unitarias y pruebas de funcionamiento. Para las pruebas unitarias se emplea una técnica simple que consiste, en evaluar parámetros de entrada seleccionados por los programadores y observar las salidas constatando que cumplan con lo esperado. Las pruebas de funcionamiento se realizan por los clientes, con el fin de comprobar que sus requerimientos se cumplan satisfactoriamente. Además se utiliza un formato para registrar cada una de las pruebas como se puede observar en la siguiente tabla (3.3). Esta tabla proporciona el número del caso de prueba (Nro), historia de usuario involucrada (sólo el número), descripción del caso de prueba, resultado esperado, resultado obtenido y motivo de la falla o recomendación.

Tabla 3.3: Modelo de registro de pruebas de funcionamiento

Nro.	Historias de usuario involucradas.	Descripción del caso de prueba.	Resultado esperado.	Resultado obtenido.	Motivo de falla ó recomendación.

Análisis general del sistema

Haciendo uso de las fases y artefactos descritos anteriormente, en las iteraciones del presente Trabajo Especial de Grado, algunas de las fases serán opcionales. En la iteración cero se usan sólo las fases de planificación y diseño. De la iteración uno a la seis se hace uso de todas las fases. En la iteración siete se hace uso, sólo de las fases de planificación y pruebas. Con las historias de usuario se elaboran planes de iteración, donde se agrupan, en iteraciones según sus objetivos. Dichas historias se obtienen desde la primera reunión con el cliente y se presentan a continuación:

3.7. Historias de usuario

Número: 1	Identificación de tablas de conest_pregrado que pueden ser usadas en conest_postgrado
Estimación: 2 días	
Descripción:	Elaboración de un resumen de las tablas que se pueden usar en la base de datos del módulo de postulación de postgrado

Número: 2	Migración de tablas necesarias de conest_pregrado a conest_postgrado
Estimación: 2 días	
Descripción:	Migración de las tablas que pueden se usadas

Número: 3	Diseño y creación del nuevo modelo de datos para el módulo de postulaciones de postgrado
Estimación: 5 días	
Descripción:	Implementar el nuevo modelo de datos para el módulo de postulación de postgrado

Número: 4	Establecer la relaciones con los modelos asociados
Estimación: 5 días	
Descripción:	Implementar las claves primarias, índices y claves foráneas para optimizar las consultas en el modelo

Número: 5	Desarrollar una vista para el registro del postulante
Estimación: 1 día	
Descripción:	Desarrollar una vista que permita el registro de postulantes

Número: 6	Desarrollar validaciones para el formulario de registro de postulante
Estimación: 2 días	
Descripción:	Desarrollar validaciones en las ventanas usadas para el registro del postulante las cuales deben cumplir con ciertas condiciones

Número: 7	Desarrollar un módulo de cambio de contraseña
Estimación: 2 días	
Descripción:	Desarrollar un modulo para el cambio de contraseña de usuario, que solicite cambio de clave al inicio de sesión por primera vez

Número: 8	Desarrollar de un método para generar una clave temporal
Estimación: 3 días	
Descripción:	Desarrollar un método el cual genere de forma aleatoria una clave temporal para usuarios nuevos

Número: 9	Desarrollar método para inicio de sesión en módulo del postulante
Estimación: 2 días	
Descripción:	Adaptar la vista y lógica del inicio de sesión en el sistema de forma tal que permita el ingreso al módulo de postulante, si el usuario posee este rol

Número: 10	Desarrollar un método que permita generar notificaciones de correo electrónico
Estimación: 3 días	
Descripción:	Desarrollo de un método que permita el envío de correo electrónico personalizado

Número: 11	Desarrollar vistas para registrar toda la información pertinente al postulante
Estimación: 2 días	
Descripción:	Desarrollar vistas que permitan al postulante cargar toda su información personal, educativa y profesional en el sistema

Número: 12	Desarrollar ventanas modales para registra la información del postulante
Estimación: 5 días	
Descripción:	Desarrollar ventanas modales que permitan cargar en el sistema toda la información referente a datos académicos y experiencias profesionales del postulante.

Número: 13	Desarrollar validaciones para los formularios de carga información del postulante
Estimación: 3 días	
Descripción:	Desarrollar validaciones y mensajes informativos que ayuden al postulante durante el proceso y le informen en caso de ocurrir algún error

Número: 14	Desarrollar validaciones en el flujo de trabajo del registro de postulante
Estimación: 4 días	
Descripción:	Desarrollar validaciones que permitan controlar paso a paso la forma como se realizará la carga de información personal educativa y laboral del postulante, esto mediante el bloqueo y liberación de pasos según aplique

Número: 15	Generar de reporte en formato PDF del currículum del postulante
Estimación: 5 días	
Descripción:	Desarrollar un método que permita genera el currículum del postulante tomando los datos cargados previamente en el sistema

Número: 16	Desarrollar de un módulo para almacenar documentos digitales
Estimación: 3 días	
Descripción:	Desarrollar un módulo que permita almacenar en el sistema documentos digitales referentes a: documento de identidad, títulos de pregrado, diplomas de postgrado y cartas de trabajo del postulante

Número: 17	Desarrollar módulo de edición de datos de usuario
Estimación: 2 días	
Descripción:	Desarrollar un módulo que permita editar los datos de usuario, foto y la clave del mismo

Número: 18	Desarrollar vistas para la solicitud de postulación a un programa de postgrado
Estimación: 3 días	
Descripción:	Desarrollar vistas que permitan recopilar la información necesaria para crear una solicitud de postulación a un programa de postgrado

Número: 19	Desarrollar vista que lista solicitudes del módulo del postulante
Estimación: 2 días	
Descripción:	Desarrollar la vista que lista todas las solicitudes del postulante

Número: 20	Desarrollar validaciones para la postulación a programas de postgrado
Estimación: 3 días	
Descripción:	Desarrollar validaciones, donde se filtren los programas de postgrado por tipo de forma dinámica y además filtre los programas de postgrado que no se han inscrito mediante el uso de la tecnología Ajax

Número: 21	Desarrollar un método para generar postulaciones a programas de postgrado
Estimación: 3 días	
Descripción:	Desarrollar un método que permita generar postulaciones a programas de postgrado, almacenando la información referente al postgrado seleccionado, al postulante e información necesaria respecto a la solicitud y su estatus

Número: 22	Desarrollar un método para generar la planilla de postulación en formato PDF
Estimación: 2 días	
Descripción:	Desarrollar un método que permita, generar una planilla de solicitud a una postulación de programa postgrado en formato PDF, donde se debe generar una planilla por cada postulación con información referente a la solicitud al programa de postgrado y datos del postulante

Número: 23	Desarrollar módulo de la encuesta genérica
Estimación: 2 días	
Descripción:	Desarrollar un módulo que permita almacenar información referente a preguntas genéricas y respuestas de encuestas asociadas a un programa de postgrado

Número: 24	Desarrollar las vista que listan período académico
Estimación: 3 días	
Descripción:	Desarrollar las vistas que permita el listado de postulaciones por período académico visualizando cuales han sido aprobadas y cuales han sido rechazadas

Número: 25	Generar reporte en formato Excel de las postulaciones por período académico
Estimación: 2 días	
Descripción:	Desarrollar método que genere reportes en formato Excel de las postulaciones del período académico con toda la información detallada del postulante y de la solicitud

Número: 26	Desarrollar el módulo de búsqueda del postulante
Estimación: 2 días	
Descripción:	Desarrollar un módulo que mediante el uso de ventanas de auto completar, permita buscar un postulante de forma dinámica.

Número: 27	Desarrollar la vista del módulo administrativo al consultar un postulante
Estimación: 3 días	
Descripción:	Desarrollar las vistas necesarias para desplegar toda la información de un postulante al momento de ser consultado, estas vistas deben desplegar información personal, el listado de sus postulaciones y una opción para ver a detalle o descargar el currículum del postulante

Número: 28	Desarrollar validaciones para las vistas en el módulo administrativo
Estimación: 2 días	
Descripción:	Desarrollar validaciones y mensajes informativos que ayuden al personal administrativo durante el proceso y le informen en caso de ocurrir algún error

Número: 29	Desarrollar el módulo de registro de pago de una solicitud
Estimación: 4 días	
Descripción:	Desarrollar un módulo que permita cargar en el sistema la información referente a los aranceles a ser cancelados por cada solicitud realizada por un postulante, entre esta información se contempla el número de solicitud, la fecha y el monto

Número: 30	Desarrollar un módulo para editar programa de postgrado
Estimación: 2 días	
Descripción:	Desarrollar un módulo que permita editar el nombre de los programas de postgrado

Número: 31	Desarrollar un módulo para crear período académico
Estimación: 2 días	
Descripción:	Desarrollar un módulo que permita cargar la información referente a un nuevo período académico y además que permita indicar si esta activo o no.

Número: 32	Desarrollar método para la evaluación de solicitudes en el módulo docente
Estimación: 3 días	
Descripción:	Desarrollar un método que permita reflejar la evaluación obtenida por la Unidad Académica de las solicitudes de los postulantes, donde se determine si las mismas han sido admitidas o no admitidas

Número: 33	Desarrollar un método para el envío de cartas de aceptación
Estimación: 3 días	
Descripción:	Desarrollar un método que permita generar las cartas de aceptación, de aquellas solicitudes con estatus aceptada, además se mande esta carta por correo electrónico al postulante

Número: 34	Instalación del ambiente de desarrollo y producción
Estimación: 3 días	
Descripción:	Instalación en el servidor asignado de todas las herramientas y aplicaciones para el ambiente de desarrollo y producción de la aplicación

Número: 35	Configuración del servidor
Estimación: 2 días	
Descripción:	Configuración del servidor de correo, Web y base de datos.

Número: 36	Despliegue del sistema en ambiente de producción
Estimación: 3 días	
Descripción:	Despliegue en ambiente de producción de la aplicación

Número: 37	Colocar un mensaje funcional con un ejemplo de contraseña
Estimación: 2 días	
Descripción:	Modificación en la vista del registro para mostrar un mensaje funcional con un ejemplo de contraseña

Número: 38	Cambiar el campo cédula por documento de identidad
Estimación: 2 días	
Descripción:	Modificación del campo cédula por documento de identidad para permitir el registro de postulantes con pasaporte

Número: 39	Incluir la foto del postulante en el reporte PDF del currículum
Estimación: 5 días	
Descripción:	Modificación del currículum para agregar una fotografía del postulante

Número: 40	Configurar un orden específico para listar los programas de postgrado
Estimación: 2 días	
Descripción:	Modificación del orden de los programas de postgrado, para listarlos en un orden específico

Requerimientos generales del sistema

A continuación se presenta el desarrollo de la aplicación Web que permite la administración del proceso de postulación a los diferentes programas de postgrado de la Facultad de Ciencias, así como también la carga de recaudos necesarios para registrar una postulación. Sólo se necesita tener conocimientos del uso de un navegador Web para poder utilizarlo.

3.8. Iteración 0 - Diseño y creación de la base de datos

- Fase de planificación: en el siguiente plan de iteración que se puede apreciar en la tabla (3.4) las historias de usuario a desarrollar en esta fase.

Tabla 3.4: Plan de iteración. Iteración 0

Iteración 0			
Descripción		Diseño y desarrollo del modelo de datos para el módulo de postulaciones de postgrado	
Fecha Inicio/ Fecha Fin		15-10-2013/ 15-12-2013	
Número	Fecha	Historia	Tipo
1	15-10-2013	Identificación de tablas de conest_pregrado que pueden ser usadas en conest_postgrado	Nueva
2	25-10-2013	Migración de tablas necesarias de conest_pregrado a conest_postgrado	Nueva
3	30-11-2013	Diseño y creación del nuevo modelo de datos para el módulo de postulaciones de postgrado	Nueva
4	10-12-2013	Establecer las relaciones con los modelos asociados	Nueva

- Fase de diseño: en el modelo de datos que de la figura 3.6 se muestran las principales tablas de la base de datos del sistema, compuesta de cincuenta y siete tablas para almacenar información referente al proceso de postulación, la cual permite independencia entre las tablas para futuras adiciones al sistema, sin necesidad de realizar grandes cambios en el modelo de datos.

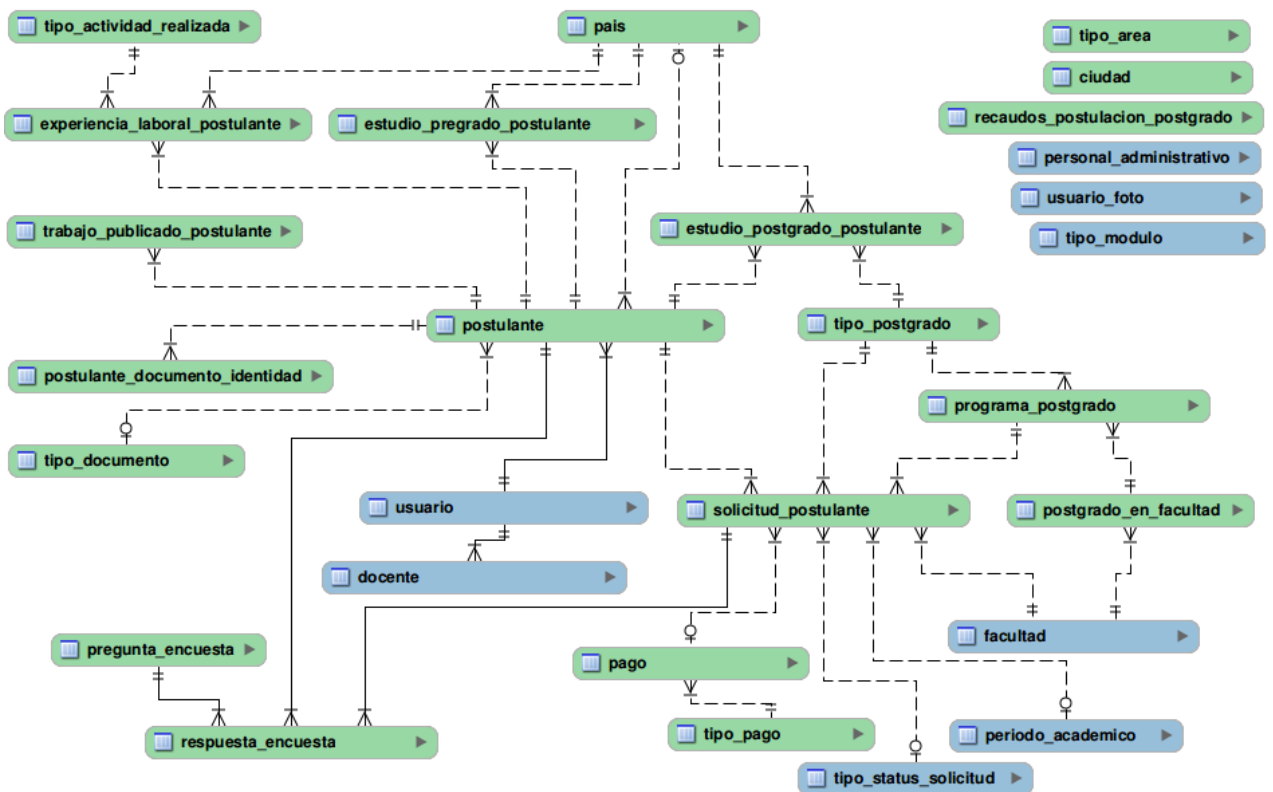


Figura 3.6: Diseño del Modelo de Datos

3.9. Iteración 1 - Registro de postulantes.

- Fase de planificación: en el siguiente plan de iteración de la tabla (3.5) se observan las tareas de programación a desarrollar en esta fase.

Tabla 3.5: Plan de iteración. Iteración 1

Iteración 1			
Descripción		Creación de las vistas y el método que permite el registro de postulantes de postgrado nuevos	
Fecha Inicio/ Fecha Fin		10-01-2014/ 07-02-2014	
Número	Fecha	Historia	Tipo
5	10-01-2014	Desarrollar las vistas para el registro del postulante	Nueva
6	11-01-2014	Desarrollar las validaciones para el formulario de registro de postulantes	Nueva
7	12-01-2014	Desarrollar el módulo de cambio de contraseña	Nueva
8	17-01-2014	Desarrollar un método para generar una contraseña temporal	Nueva
9	18-01-2014	Desarrollar un método para inicio de sesión en el módulo del postulante	Nueva
10	01-02-2014	Desarrollar un método que permita generar notificaciones de correo electrónico	Nueva

- Fase de diseño: en esta iteración se desarrolla la funcionalidad que permite el registro de postulantes en el sistema, la cual es necesaria para poder postularse en alguno de los programas de postgrado que se ofertan. A través de la siguiente interfaz se brinda la posibilidad al postulante de poder llenar sus datos básicos y le llegue una notificación de correo electrónico con su contraseña temporal. En la siguiente figura (3.7) se puede apreciar su diseño.



The image shows a web application interface with a modal window titled "Registro postulante". The modal contains the following fields and values:

- Primer nombre *: JUAN
- Primer apellido *: PITA
- Documento de Identidad *: V 18556959
- Correo Electrónico *: juan.pita@ciens.ucv.ve
- Repita el Correo Electrónico *: juan.pita@ciens.ucv.ve

A yellow "Guardar" button is located at the bottom of the modal, with a mouse cursor pointing to it. In the background, a login form is partially visible with fields for "Cédula ó Pasaporte número sin punto" and "Contraseña contraseña", and an "Iniciar" button.

Figura 3.7: Vista registro de postulante

- Fase de codificación: el método encargado de registrar al postulante recibe del formulario los siguientes parámetros: nombre, apellido, nacionalidad, documento de identidad, correo electrónico, estos datos pasan por el proceso de validación para almacenarlos en la base de datos, cabe desatacar que la dificultad de este método es el diseño del proceso ya que se hacen varias validaciones y la comprobación de existencia del usuario, su principal ventaja es la verificación de los datos del postulante, esto se logra mediante el envío de un correo electrónico para verificar la autenticidad del postulante, seguido de esta verificación es enviada la contraseña temporal para poder ingresar al sistema. En la figura (3.8) se puede apreciar el código que se encarga de validar y registrar los postulantes.

```

19
20 def validar_registro_postulante
21   @flujo = "validar_registro_postulante"
22   @titulo = "Postulación a Postgrado"
23   @usuario = Usuario.find_by_cedula(params[:usuario][:cedula])
24   @postulante = Postulante.find_by_documento_identidad(params[:usuario][:cedula])
25
26   if @usuario.nil?
27     @usuario = Usuario.new(params[:usuario].permit!)
28     clave = @usuario.generar_clave()
29     @usuario.encryptar_clave(clave)
30     @usuario.asignar_tipo_usuario(3)
31     @usuario.usuario_nuevo
32     @usuario.save
33     usuario_nuevo = true
34   else
35     usuario_nuevo = false
36   end
37
38   if @postulante.nil?
39     @postulante = Postulante.new(@usuario.attributes.select{|nombre, _| Postulante.attribute_names.include? nombre})
40     @postulante.documento_identidad = @usuario.cedula
41     @postulante.generar_token
42     @postulante.cambiar_clave = usuario_nuevo
43     if @postulante.save
44       @usuario.tipo_status_sistema_id = "A"
45       @usuario.save
46       postulante_nuevo = true
47     else
48       postulante_nuevo = false
49     end
50   else
51     postulante_nuevo = false
52   end
53
54   if @usuario && @postulante && postulante_nuevo
55     if usuario_nuevo
56       titulo = "Postulación al Postgrado de la Facultad de Ciencias"
57       mensaje = "Registro exitoso, favor siga el siguiente enlace para validar su correo: "
58       mensaje += url_for(controller: :principal_postulante, action: :validar_token, token: @postulante.token, host: '190.169.72.141:3000/postu
59       mensaje += "\n\nEn caso de no funcionar el enlace por favor copiar y pegar en su navegador"
60       bitacora(mensaje)
61       NotificacionCorreo.correo_general(@usuario, titulo, mensaje).deliver
62     else
63       titulo = "Postulación al Postgrado de la Facultad de Ciencias"
64       mensaje = "Registro exitoso, favor siga el siguiente enlace e inicie sesión con su usuario y contraseña actual: "
65       mensaje += url_for(controller: :inicio, action: :index, host: '190.169.72.141:3000/postulacion_postgrado').to_s
66       mensaje += "\n\nEn caso de de no recordar su clave actual por favor solicite cambio de contraseña"
67       mensaje += "\n\nEn caso de no funcionar el enlace por favor copiar y pegar en su navegador"
68       bitacora(mensaje)
69       NotificacionCorreo.correo_general(@usuario, titulo, mensaje).deliver
70       end
71       @registrado = true
72     else
73       mensaje = "Datos del Postulante no han podido ser registrados!"
74       bitacora(mensaje)
75       @registrado = false
76     end
77   end
78

```

Figura 3.8: Código que registra postulante

- Fase de pruebas: las pruebas de funcionamiento tienen como objetivo verificar que la interfaz solicite los datos necesarios para el registro inicial del postulante, comprobando tras varios registros y distintos casos de prueba, los resultados esperados, una vez se culmina esta prueba, se adapta la nueva funcionalidad al sistema, siendo esta una historia de usuario culminada y verificada satisfactoriamente, por otra parte,

en la primera prueba correspondiente a las historia de usuario uno y dos, se solicitan unos cambios como recomendación y mejora, estas observaciones se usan como nuevas historias de usuario y se pretende dar solución a estas historias de usuario en la iteración seis. En la tabla (3.6) se muestran las pruebas de funcionamiento que se ejecutan a este plan de iteración.

Tabla 3.6: Pruebas de funcionamiento. Iteración 1

Nro	Historias de usuario involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido	Motivo de falla o recomendación
01	5,6	Proporcionar los datos necesarios para el registro de postulantes, en el formulario de registro postulante	El sistema deberá indicar la carga de datos básicos para el registro del postulante, una vez indicado hacer la validación de cada uno de los campos	El sistema solicita los datos tal como se esperaba, validando finalmente que la data sea válida	Se recomienda colocar el tamaño de la contraseña mínimo requerido Modificar el campo cédula por documento de identidad para permitir el registro de postulantes con el número de pasaporte
02	7, 8, 10	Verificar la autenticidad del correo electrónico registrado	El sistema deberá enviar un enlace para verificar el correo electrónico registrado	El sistema envió correctamente el enlace para verificar el correo y este verifica correctamente la cuenta de correo del postulante	
03	9	Proporcionar un formulario para iniciar sesión en el sistema como postulante	El sistema deberá permitir iniciar sesión como postulante	El sistema permite el inicio de sesión como postulante	
04	7	Proporcionar un formulario con los datos necesarios para recuperar la contraseña	El sistema deberá permitir recuperar la contraseña, validando los datos registrados	El sistema válida y permite el cambio de contraseña en caso de ser un usuario registrado	

3.10. Iteración 2 - Carga de currículum vitae del postulante

- Fase de planificación: en el siguiente plan de iteración en la tabla (3.7) se observaban las historias de usuario a desarrollar en esta fase.

Tabla 3.7: Plan de iteración. Iteración 2

Iteración 2			
Descripción		Desarrollo de vistas del módulo de carga de currículum postulante y generación de reporte PDF del currículum del postulante	
Fecha Inicio/ Fecha Fin		14-02-2014/ 11-04-2014	
Número	Fecha	Historia	Tipo
11	14-02-2014	Desarrollar vistas para registrar toda la información pertinente al postulante	Nueva
12	15-02-2014	Desarrollar ventanas modales para registrar información del postulante	Nueva
13	16-02-2014	Desarrollar validaciones para los formularios de carga información del postulante	Nueva
14	01-03-2014	Desarrollar validaciones en el flujo de trabajo del registro del postulante	Nueva
15	02-03-2014	Generar de reporte en formato PDF del currículum del postulante	Nueva
16	14-03-2014	Desarrollar un módulo para almacenar documentos digitales	Nueva
17	04-04-2014	Desarrollar un módulo de edición de datos de usuario	Nueva

- Fase de diseño: en esta iteración se desarrolla la funcionalidad que permite la carga de los datos básicos, datos contacto, estudios de pregrado, estudios de postgrado, experiencias laborales, trabajos publicados, los cuales se registran para que el personal administrativo y docente realice su evaluación, seguidamente al final de cargar toda esta información se puede descargar el currículum del postulante en formato PDF, esto comprende la utilidad que le permite al postulante obtener el archivo digital su currículum, con el contenido generado. A través de la siguiente interfaz se le brinda la posibilidad al postulante de poder cargar sus datos e ir navegando por cada uno de los pasos que tiene el sistema. En la figura (3.9) se muestra el formulario donde se solicitan los datos básicos del postulante nombres, apellidos, sexo, estado civil, fecha de nacimiento, país de nacimiento y una copia del documento de identidad.

The screenshot displays the 'CONEST Sistema de Gestión Académica' interface. At the top, there are navigation tabs: 'Inicio', 'Resumen de curricular', and 'Postulación'. The user is logged in as 'BANESCO MARIA' with ID 'V-18556958'. The page title is 'Datos Personales' and the date is 'Viernes Friday, 10 de octubre de 2014, 05:22pm'. A yellow warning box states: 'Recuerde que los datos de preinscripción en el sistema deben ser iguales a los datos de los documentos a consignar'. Below this is a breadcrumb trail: '[Datos personales] > Datos contacto > Pregrados > Postgrados > Empleos > Publicaciones'. The main form area is titled 'Llenar, por favor, sus datos personales completos y sin abreviaturas.' and contains the following fields:

- Documento de Identidad *: V 18556958
- Primer nombre *: ANA
- Segundo nombre: MARIA
- Primer apellido *: GONZALES
- Segundo apellido: GARCIA
- Sexo *: FEMENINO
- Estado civil *: SOLTERO
- Fecha de nacimiento *: 26/09/2014
- País de nacimiento *: VENEZUELA
- Copia del documento de identidad: No se eligió archivo

A 'Guardar y continuar >>' button is located at the bottom right of the form.

Figura 3.9: Vista que permite cargar datos básicos

Para el diseño de la vista que permite la carga de los estudios de pregrado, estudios de postgrado, experiencias laborales y trabajos publicados, se tiene un diseño con dos vistas, una de las cuales es un formulario en el que se pide, el contenido de los estudios, experiencias o trabajos publicados según sea el caso y la siguiente vista una tabla con la lista de los estudios que han sido agregados al perfil del postulante, en esta vista la tabla brinda una opción que permite eliminar por completo el contenido registrado, brindando al postulante un enfoque de que esta guardando un contenido y hacer un poco mas clara y rápida la carga de información, en la figura (3.10) se puede apreciar la venta modal con el formulario que permite la carga de estudios de pregrado. Por otra parte en la figura(3.11) se muestra la vista que lista los estudios de pregrado del postulante.

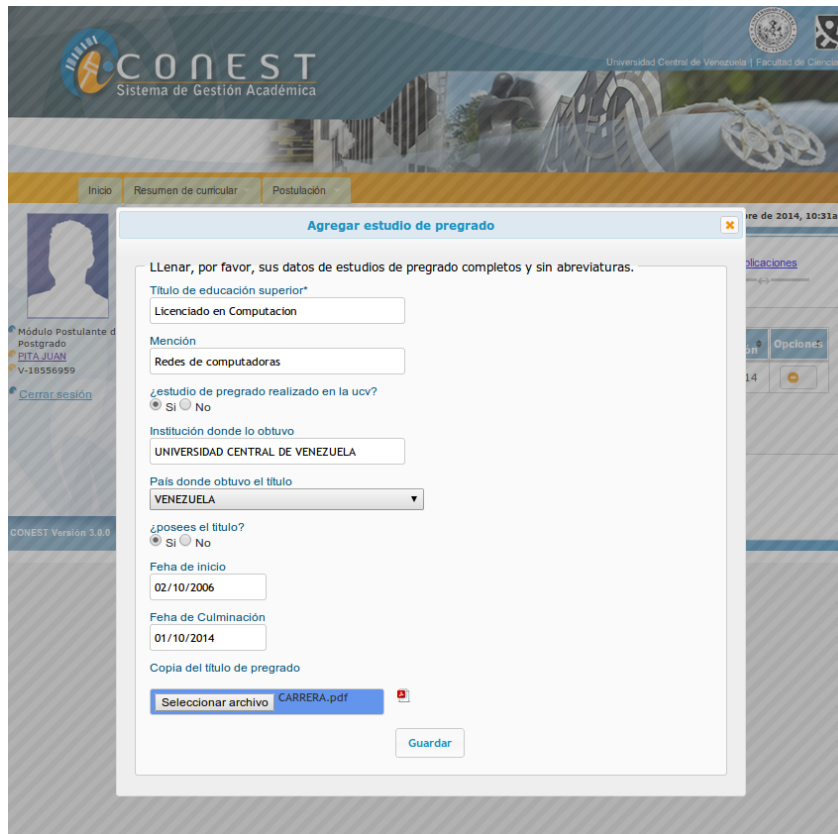


Figura 3.10: Diseño que permite la carga de estudios de pregrado



Figura 3.11: Diseño de vista que muestra los estudios de pregrado registrados

- Fase de codificación: el código encargado de almacenar los estudios del postulante, esta compuesto por varias secciones, ya que el diseño es modular, la ventaja de este

componente es que se puede reutilizar para cualquiera de los módulos en los que se quiera registrar datos de un postulante. Este módulo no sólo permite la carga de datos del postulante sino también la carga de archivos en distintos formatos. Al momento de almacenar los datos de los estudios de un postulante, se verifica si el postulante existe por medio de una frase clave (token) asignada en el registro inicial, en caso de que exista, se reciben todos los parámetros del postulante que son pasados por el formulario para luego hacer el salvado de los mismos, al culminar esta función, se hace la redirección a la página que lista los estudios de pregrado. A continuación en la figura (3.12) se puede apreciar el código que muestra y guarda los estudios de pregrado de un postulante.

```

208
209
210 def estudios_pregrado
211   @postulante = Postulante.find_by_token(session[:postulante].token)
212   @titulo_pagina = "Estudios pregrado"
213   @opcion = true
214   @paso_actual = 4
215 end
216
217 def modal_estudio_pregrado
218   @postulante = Postulante.find_by_token(session[:postulante].token)
219   @paises = Pais.order('nombre')
220   @opcion = true
221   render layout: false
222 end
223
224 def crear_estudio_pregrado
225   @postulante = Postulante.find_by_token(session[:postulante].token)
226   @estudio_pregrado = EstudioPregradoPostulante.new(:documento_identidad=> @postulante.documento
227   @estudio_pregrado.update(params[:estudio_pregrado_postulante].permit!)
228   documento_pregrado = @estudio_pregrado.build_estudio_pregrado_documento(params[:estudio_pregra
229   @estudio_pregrado.save
230   mensaje = "Se agregaron los datos del estudio de pregrado correctamente"
231   bitacora(mensaje, @postulante.documento_identidad)
232   flash[:mensaje] = mensaje
233
234   @postulante.paso = 4 if @postulante.paso != 4
235   @postulante.save
236   redirect_to action: 'estudios_pregrado'
237 end
238
239 def eliminar_estudio_pregrado
240   @postulante = Postulante.find_by_token(session[:postulante].token)
241   @estudio_pregrado = EstudioPregradoPostulante.find(params[:id])
242   @estudio_pregrado.estudio_pregrado_documento.destroy if @estudio_pregrado.estudio_pregrado_doc
243   @estudio_pregrado.delete
244   @postulante.paso = 2 if @postulante.estudio_pregrado_postulantes.length == 0
245   @postulante.save
246   mensaje = "Se elimino estudio de pregrado correctamente"
247   bitacora(mensaje, @postulante.documento_identidad)
248   flash[:mensaje] = mensaje
249   redirect_to action: 'estudios_pregrado'
250 end

```

Figura 3.12: Código que permite listar, crear estudios de pregrado

- Fase de pruebas: las pruebas de funcionamiento tienen como objetivo verificar que la interfaz solicitara, los datos necesarios para el registro completo del currículum del postulante, comprobando tras varios registros y distintos casos de prueba, los resultados esperados, una vez culminada la prueba, se adapta la nueva funcionalidad al sistema, siendo esta una historia de usuario culminada y verificada satisfactoria-

mente, además se refleja en la prueba ocho correspondiente a la historia de usuario doce, que se solicitan unos cambios como recomendación y mejora, esta observación se usa como una nueva historia de usuario y se da solución a esta en la iteración seis. En la tabla (3.8) se muestran las pruebas de funcionamiento que se realizan.

Tabla 3.8: Pruebas de funcionamiento. Iteración 2

Nro.	Historias de usuario involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido	Motivo de falla o recomendación
05	11, 13	Proporcionar los datos necesarios para el registro del postulante.	El sistema deberá guardar los datos del postulante	El sistema guarda correctamente los datos del postulante	
06	12, 13	Proporcionar los datos necesarios para el registro de información desde ventanas modales	El sistema deberá registrar los datos desde una venta modal para luego listarlos en una tabla	El sistema guarda y lista correctamente los datos desde la venta modal	
07	14.	Registrar las solicitudes si se llenan completamente los tres primeros pasos	El sistema deberá bloquear los pasos de registro de estudios de postgrado, experiencias y trabajos	El sistema bloqueó los pasos de registro de estudios de postgrado experiencias laborales y trabajos correctamente	
08	15	Seleccionar ver currículum para generar el currículum del postulante en formato PDF	El sistema deberá cargar en el navegador el documento en formato PDF solicitado. El mismo deberá contener el encabezado, el título del documento, los datos del postulante y el postgrado, la fecha de emisión, paginación, datos de estudios, experiencias laborales y trabajos publicados además un pie de página	El sistema carga en una nueva instancia del navegador el currículum en formato PDF. El mismo contiene todos los datos esperados en dicho documento.	Se recomienda colocar la foto del postulan en el currículum
09	16	Seleccionar cargar documentos para subir los documentos solicitados	El sistema deberá permitir la carga de documentos	El sistema carga correctamente los documentos enviados	
10	17	Seleccionar el nombre del postulante para editar sus datos básicos de usuario en el sistema	El sistema deberá cargar una vista en la que se permite editar su contraseña, datos básicos y foto personal	El sistema carga correctamente, la vista que permite editar los datos del usuario y actualiza correctamente estos datos	

3.11. Iteración 3 - Carga de curriculum vitae del postulante

- Fase de planificación: en el siguiente plan de iteración de la tabla (3.9) se observaban las historias de usuario a desarrollar en esta fase.

Tabla 3.9: Plan de iteración. Iteración 3

Iteración 3			
Descripción		Desarrollo del módulo de postulación a los programas postgrado y generación de comprobante de postulación PDF	
Fecha Inicio/ Fecha Fin		15-04-2014 / 14-06-2014	
Número	Fecha	Historia	Tipo
18	15-04-2014	Desarrollar las vistas para la solicitud de postulación a un programa de postgrado	Nueva
19	18-04-2014	Desarrollar la vista que lista solicitudes del módulo del postulante	Nueva
20	25-04-2014	Desarrollar validaciones para la postulación a programas de postgrado	Nueva
21	02-04-2014	Desarrollar un método para generar postulaciones a programas de postgrado	Nueva
22	09-04-2014.	Desarrollar un método para generar la planilla de postulación en formato PDF	Nueva
23	16-05-2014	Desarrollar el módulo de la encuesta genérica	Nueva

- Fase de diseño: en la presente iteración se realiza, el desarrollo del módulo de registro de solicitudes a programas de postgrado que se ofrecen, en cuanto al diseño se debe tener dos listas de tipo selección dependientes que listen los tipos de postgrado y luego de una selección se active la otra lista, cargando únicamente los programas de postgrado que se ofrecen para ese tipo de postgrado, en la figura (3.13) se muestra el diseño de la vista de solicitud de postulación con la encuesta genérica y las listas dependientes para elegir el programa de postgrado.

The image shows a web application interface for 'CONEST Sistema de Gestión Académica'. The main content is a form titled 'Postularse a un postgrado'. The form is divided into several sections:

- Header:** 'Ingrese los datos del curso de postgrado al que desea postularse'.
- Form Fields:**
 - 'Clase de estudios *': A dropdown menu with 'Seleccione' as the current selection.
 - 'Nombre del programa de postgrado *': A dropdown menu with the instruction 'Seleccione primero una clase de estudios'.
- Encuesta Section:**
 - 'Motivo por el cual desea ingresar al postgrado': A text input field.
 - 'Entre las distintas areas de trabajo, indique la de su preferencia': A text input field.
 - 'Cual fue el motivo de su escogencia': A text input field.
- Buttons:** A 'Postularse' button at the bottom right of the form.

On the right side of the page, there is a table with the following data:

Estatus
ACEPTADA
NEGADA
ENTREGADA
ENTREGADA
ENTREGADA
ENTREGADA
NEGADA
NEGADA
NUEVA

Figura 3.13: Diseño de la vista de solicitud de postulación

Con respecto a la interfaz de usuario que lista las solicitudes realizadas, se tiene una tabla que lista las postulaciones detallando lo siguiente, tipo de postgrado, nombre del programa, fecha de solicitud, estatus de la solicitud. en la figura (3.14) se puede apreciar el diseño con la vista que lista las solicitudes realizadas de un postulante.

The screenshot displays the CONEST Sistema de Gestión Académica interface. At the top, there is a navigation bar with 'Inicio', 'Resumen de curricular', and 'Postulación'. The main content area is titled 'Postulaciones' and shows a table of 'Solicitudes'. The table has four columns: 'Tipo de postgrado', 'Nombre del curso', 'Fecha Solicitud', and 'Estatus'. The data rows are as follows:

Tipo de postgrado	Nombre del curso	Fecha Solicitud	Estatus
ESPECIALIZACION	FÍSICA MÉDICA	09/06/2014	ACEPTADA
ESPECIALIZACION	SISTEMAS DE INFORMACIÓN	20/07/2014	NEGADA
CURSO DE PERFECCIONAMIENTO PROFESIONAL	ECOLOGÍA DE POBLACIONES Y COMUNIDADES	16/07/2014	ENTREGADA
CURSO DE PERFECCIONAMIENTO PROFESIONAL	FÍSICA AVANZADA	05/06/2014	ENTREGADA
CURSO DE PERFECCIONAMIENTO PROFESIONAL	PROTECCIÓN RADIOLÓGICA	16/07/2014	ENTREGADA
DOCTORADO	CIENCIAS DE LA COMPUTACIÓN	23/08/2014	ENTREGADA
DOCTORADO	BOTÁNICA	27/09/2014	NEGADA
MAESTRIA	FÍSICA MÉDICA	30/09/2014	NEGADA
ESPECIALIZACION	MODELOS ALEATORIOS	30/09/2014	NUEVA

Additional interface elements include a user profile sidebar for 'PITA JUAN' (ID: V-18556959) and a footer with 'CONEST Versión 3.0.0' and 'Todos los Derechos Reservados. Grupo CONEST. Copyright 2007-2013'.

Figura 3.14: Diseño de lista de solicitudes del postulante

- Fase de codificación: el código encargado de generar la solicitud de postgrado, esta compuesto por varios métodos que permiten guardar la selección del postulante, validando previamente que el programa de postgrado al que se pretende postular no sea el mismo, cabe destacar que la principal dificultad es la configuración de las listas de tipo de postgrado y programas de postgrado, pues son dos listas que dependen de hacer una consulta en base de datos al seleccionar el tipo de postgrado, para esto se cuenta con el apoyo de la tecnología ajax que permite hacer la consulta dinámica a base de datos sin necesidad de recargar la ventana modal, esta función hace el paso de parámetros vía GET al método buscar_programa_postgrado, con el id del tipo de postgrado seleccionado, en este método se hace la consulta a base de datos y se retorna una vista con el resultado que se quiere, en la figura (3.15) se muestra el código que hace el envío de los datos al método que hace el hace la consulta y valida los programas ya inscritos.

```

75 #programas_postgrados = $('#solicitud_postulante_codigo_u XXI').html()
76 # $(".error").hide()
77 $('#programa_postgrado_error_todos_inscritos').hide()
78 $('#clase_estudio_error').show()
79 $('#solicitud_postulante_codigo_u XXI_id').attr('disabled', 'disabled')
80 $('#solicitud_postulante_tipo_postgrado_id').change ->
81 $('#programa_postgrado_error').hide()
82 tipo_postgrado = $(this).val()
83 $.ajax {
84   url: 'http://190.169.72.141:3000/postulacion_postgrado/principal_postulante/buscar_programa_postgrado?id='+tipo_postgrado,
85   url: 'http://10.0.1.11:3000/postulacion_postgrado/principal_postulante/buscar_programa_postgrado?id='+tipo_postgrado,
86   success: (data) ->
87     $('#programa').html data
88     if $('#solicitud_postulante_tipo_postgrado_id :selected').text() is 'Seleccione' then $('#clase_estudio_error').show()
89     if $('#solicitud_postulante_codigo_u XXI_id option').size() == 1
90       $('#solicitud_postulante_codigo_u XXI_id').attr('disabled', 'disabled')
91     else
92       $('#solicitud_postulante_codigo_u XXI_id').removeAttr("disabled")
93       $('#programa_postgrado_error').show()
94       $('#clase_estudio_error').hide()
95       $('#todos_inscritos_error').hide()
96
97 $('#solicitud_postulante_codigo_u XXI_id').change ->
98 $('#programa_postgrado_error').hide()
99 if $('#solicitud_postulante_codigo_u XXI_id :selected').text() is 'Seleccione' then $('#programa_postgrado_error').show() and $('#to
100
101 $('#encuesta_postgrado_mate').hide()
102 $('#curso_mate_Si').click ->
103   $('#encuesta_postgrado_mate').show()
104   $('#encuesta_postgrado').hide()
105 $('#curso_mate_No').click ->
106   $('#encuesta_postgrado').show()
107   $('#encuesta_postgrado_mate').hide()

```

Figura 3.15: Código que válida programa ya inscritos

- Fase de pruebas: las pruebas de funcionamiento que se realizan tienen como objetivo verificar que la interfaz solicite los datos necesarios para el registro completo de la solicitud del postulante a un programa de postgrado, comprobando tras varios registros y distintos casos de prueba, los resultados esperados, una vez culminada la prueba, se adapta la nueva funcionalidad al sistema, siendo esta una historia de usuario culminada y verificada satisfactoriamente, en la prueba once correspondiente a la historia de usuario dieciséis, se solicitan unos cambios como recomendación y mejora, esta observación se usa como una nueva historia de usuario y se pretende dar solución a esta en la iteración seis. En la tabla (3.10) se muestran las pruebas de funcionamiento que se realizan.

Tabla 3.10: Pruebas de funcionamiento. Iteración 3

Nro	Historias usuario involucradas	Descripción caso de prueba	Resultado esperado	Resultado obtenido	Motivo de la Falla o recomendación
11	18,20,21	Proporcionar los datos necesarios para el registro de una solicitud de alguno de los programas de postgrado	El sistema deberá guardar los datos de la solicitud, validando que sean correctos	El sistema guardo correctamente los datos de la solicitud	Colocar un orden específico para listar los tipos de postgrados
12	19, 20	Proporcionar una vista con una tabla que liste las solicitudes registradas por el postulante.	El sistema deberá listar las solicitudes luego del registro de la solicitud, mostrar una tabla con la lista de las solicitudes generadas	El sistema lista correctamente los datos de la solicitud de postulante luego del registro de una solicitud	
13	22	Seleccionar descargar comprobante, para generar la constancia de inscripción	El sistema deberá genera una constancia de solicitud que contenga encabezado título de documento y datos del postulante con los datos de la solicitud además de fecha de generación, paginación y pie de página	El sistema carga en una nueva instancia del navegador la constancia de solicitud en formato PDF. El mismo contiene todos los datos esperados en dicho documento	
14	23	Proporcionar una vista con el formulario de un encuesta genérica	Se deberá guardar los datos de la encuesta de la solicitud	El sistema guarda correctamente los datos de la encuesta	

3.12. Iteración 4 - Desarrollo del módulo del personal administrativo

- Fase de planificación: en el siguiente plan de iteración, en la tabla (3.11) se observaban las historias de usuario a desarrollar en esta fase.

Tabla 3.11: Plan de iteración. Iteración 4

iteración 4			
Descripción		Desarrollar módulo administrativo	
Fecha Inicio/ Fecha Fin		19-08-2014 / 31-08-2014	
Número	Fecha	Historia	Tipo
24	19-10-2014	Desarrollar las vistas que listan postulaciones por período académico	Nueva
25	20-08-2014	Generar reportes en formato Excel de las postulaciones por período académico	Nueva
26	24-08-2014	Desarrollar el módulo de búsqueda del postulante.	Nueva
27	25-08-2014	Desarrollar la vista del módulo administrativo al consultar un postulante	Nueva
28	26-08-2014	Desarrollar validaciones para las vistas en el módulo administrativo.	Nueva
29	27-08-2014	Desarrollar el módulo de registro de pago de una solicitud	Nueva.
30	29-08-2014	Desarrollar un módulo para editar los nombres de programas de postgrado	Nueva
31	30-08-2014	Desarrollar un módulo para crear período académico	Nueva

- Fase de diseño: con respecto a las vistas en el módulo administrativo, al momento de hacer la búsqueda de algún postulante se despliega una ventana modal donde se pide, número documento de identidad o nombre del postulante, para luego de buscar y direccionar a la vista que se encarga de listar sus datos. En la siguiente figura (3.16) se puede apreciar el diseño de la venta modal que permite buscar postulantes.

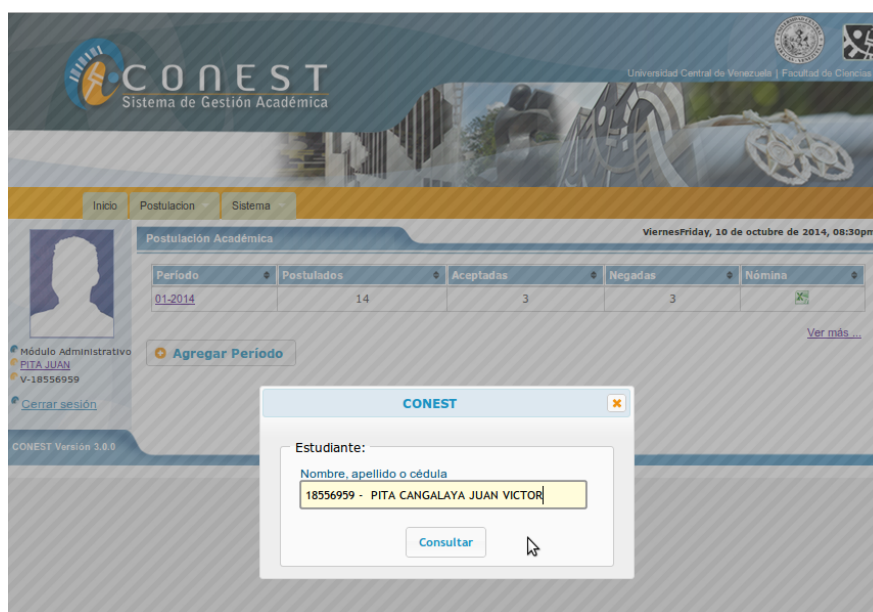


Figura 3.16: Búsqueda de postulante desde el módulo administrativo

La interfaz de usuario del postulante desde el modulo del personal administrativo, esta dividida en dos secciones dentro de la misma interfaz, en la cual en la parte

superior se tiene el curriculum vitae del postulante para poder desplegarlo completamente, se tiene un enlace a ver más, esta opción lleva al currículum del postulante, sección en la cual se puede descargar en formato PDF, en la parte inferior de la vista se muestra una tabla en la cual se tienen todos los postgrados a los que el postulante está solicitando ingresar, brindando una opción para el registro de pago, en la figura (3.17) se puede ver el diseño general de la vista de un postulante desde el módulo del personal administrativo.

The screenshot displays the user interface for the CONEST Sistema de Gestión Académica. At the top, there is a navigation bar with 'Inicio', 'Postulación', and 'Sistema' menus. The main content area is titled 'Datos del Curriculum' and shows the profile of 'JUAN VICTOR PITA CANGALAYA'. The profile includes personal data such as birth date (10/12/1985), sex (MASCULINO), and marital status (CONCUBINO). Below the profile, there is a table of 'Solicitudes' (Applications) with columns for 'Tipo de postgrado', 'Nombre del curso', 'Fecha Solicitud', 'Estatus', 'Recaudos', and 'Pago'. The table lists seven applications, including 'ESPECIALIZACION' in 'FÍSICA MÉDICA' and 'SISTEMAS DE INFORMACIÓN', and 'DOCTORADO' in 'CIENCIAS DE LA COMPUTACIÓN' and 'BOTÁNICA'. A 'Pago' button is visible at the bottom right of the table.

Tipo de postgrado	Nombre del curso	Fecha Solicitud	Estatus	Recaudos	Pago
ESPECIALIZACION	FÍSICA MÉDICA	09/06/2014	ACEPTADA	CONFORME	Ver pago
ESPECIALIZACION	SISTEMAS DE INFORMACIÓN	20/07/2014	ENTREGADA	CONFORME	Ver pago
CURSO DE PERFECCIONAMIENTO PROFESIONAL	ECOLOGIA DE POBLACIONES Y COMUNIDADES	16/07/2014	ENTREGADA	CONFORME	Ver pago
CURSO DE PERFECCIONAMIENTO PROFESIONAL	FÍSICA AVANZADA	05/06/2014	ENTREGADA	CONFORME	Ver pago
CURSO DE PERFECCIONAMIENTO PROFESIONAL	PROTECCIÓN RADIOLÓGICA	16/07/2014	ENTREGADA	CONFORME	Ver pago
DOCTORADO	CIENCIAS DE LA COMPUTACIÓN	23/08/2014	ENTREGADA	CONFORME	Ver pago
DOCTORADO	BOTÁNICA	27/09/2014	NUEVA	REVISAR	Pago

Figura 3.17: Vista del postulante desde el módulo administrador

- Fase de codificación: Una de las tareas más importantes en cuanto a este módulo es el desarrollo de la interfaz de usuarios del postulante desde el rol del personal administrativo, pues se debe consultar el curriculum vitae y la solicitud de cada uno de los postulantes. Una vez realizado el proceso de búsqueda dinámica del postulante, se deben mostrar todos sus datos de postulación en una tabla. El desarrollo de este método representa una mayor complejidad debido a una serie de condiciones y validaciones que se realizan para lograr satisfactoriamente el listado

de todas las postulaciones, cabe destacar que con la ayuda de métodos definidos en el código perteneciente al módulo de cargar de currículum vitae del postulante se puede generar la vista del currículum vitae y la vista de las solicitudes, evitando así recodificar, en la figura (3.18) se puede apreciar el método que se encarga de hacer la consulta en el modelo de los datos postulante enviando como parámetro vía GET el documento de identidad del postulante.

```
20
21 def estudiante
22   termino = params[:term]
23   cantidad = termino.strip.split.size
24   arreglo = []
25   if cantidad == 1
26     objetos = Usuario.where(["cedula like ? OR primer_nombre like ? OR primer_apellido like ?", "#{
27     objetos.select!{|x| Postulante.where(documento_identidad:x.cedula).take }
28     objetos.each{|x| arreglo<<x.nombre_autocomplete}
29   else
30     term1, term2, demas = termino.split
31     objetos = Usuario.where(["(cedula like ? OR primer_nombre like ? OR primer_apellido like ?) AND
32     objetos.select!{|x| Postulante.where(documento_identidad:x.cedula).take }
33     objetos.each{|x| arreglo<<x.nombre_autocomplete}
34   end
35   render text:arreglo.to_json
36 end
37
```

Figura 3.18: Código que busca los datos del postulante desde el administrador

- Fase de pruebas: las pruebas de funcionamiento tienen como objetivo verificar que la interfaz permita buscar y detallar las solicitudes de un postulante, así como generar reportes y registrar el pago, comprobando tras varios registros y distintos casos de prueba, los resultados esperados, una vez culminada la prueba, se adapta la nueva funcionalidad al sistema, siendo esta una historia de usuario culminada y verificada satisfactoriamente, cabe destacar que en la prueba once correspondiente a la historia de usuario dieciséis, se solicitan unos cambios como recomendación y mejora, esta observación se usa como una nueva historia de usuario y se pretende dar solución a esta en la iteración seis. En la tabla (3.12) se muestran las pruebas de funcionamiento que se realizan.

Tabla 3.12: Pruebas de funcionamiento. Iteración 4

Nro	Historias usuario involucradas	Descripción caso de prueba	Resultado esperado	Resultado obtenido	Motivo de la falla o recomendación
15	24	Proporcionar una vista con una tabla que liste los postulantes por periodo académico	Se listen todos los postulantes con solicitudes registradas	Se listan correctamente las solicitudes	
16	25	Seleccionar nomina para generar reporte Excel	Al seleccionar descargar nomina el sistema deberá carga un reporte en formato Excel de con los datos de los postulantes y de la solicitud	El sistema descarga en formato Excel una documento con los datos de los postulantes	Agregar un campo adicional al reporte Excel de las postulaciones que indique el estatus de la solicitud
17	26, 28	Proporcionar una vista con una venta modal que busque al postulante	El sistema deberá buscar al postulante por número de documento de identidad o por nombre	El sistema busca correctamente el postulante solicitado	
18	27	Proporcionar una vista con los datos del postulante y su solicitud	El sistema deberá mostrar la información del postulante solicitado	El sistema muestra correctamente los datos del postulante	
19	28, 29	Proporcionar una vista para registrar los pagos de las solicitudes	El sistema deberá guardar toda la información del pago de la solicitud	El sistema guarda correctamente los datos del pago la solicitud	
20	29,30	Proporcionar una vista para editar los datos de los programas postgrado	El sistema deberá guardar sólo el nombre del programa de postgrado	El sistema guarda correctamente sólo el nombre del programa	
21	31	proporcionar una vista para registrar los periodos de postulación	El sistema deberá guardar los datos del periodo académico a registrar	El sistema guarda correctamente los datos del periodo académico	

3.13. Iteración 5 - Desarrollo del módulo del personal docente

- Fase de planificación: en el siguiente plan de iteración en tabla (3.13) se observaban las historias de usuario a desarrollar en esta iteración.

Tabla 3.13: Plan de iteración. Iteración 5

iteración 5			
Descripción		Desarrollar módulo docente	
Fecha Inicio/ Fecha Fin		05-09-2014 / 08-09-2014	
Número	Fecha	Historia	Tipo
32	05-09-2014	Desarrollar método para la evaluación de solicitudes en el módulo docente	Nueva
33	07-09-2014	Desarrollar método para creación y envío de cartas de aceptación.	Nueva

- Fase de Diseño: para realizar la evaluación de la solicitud, se necesita que el docente seleccione el botón de evaluación y llene en la ventana modal que se despliega, si acepta o niega la postulación, además guardar una observación de la evaluación. Para la evaluación del postulante se proporciona una vista con una tabla y un botón que permite evaluar la solicitud, tal como se aprecia en la siguiente figura (3.19)

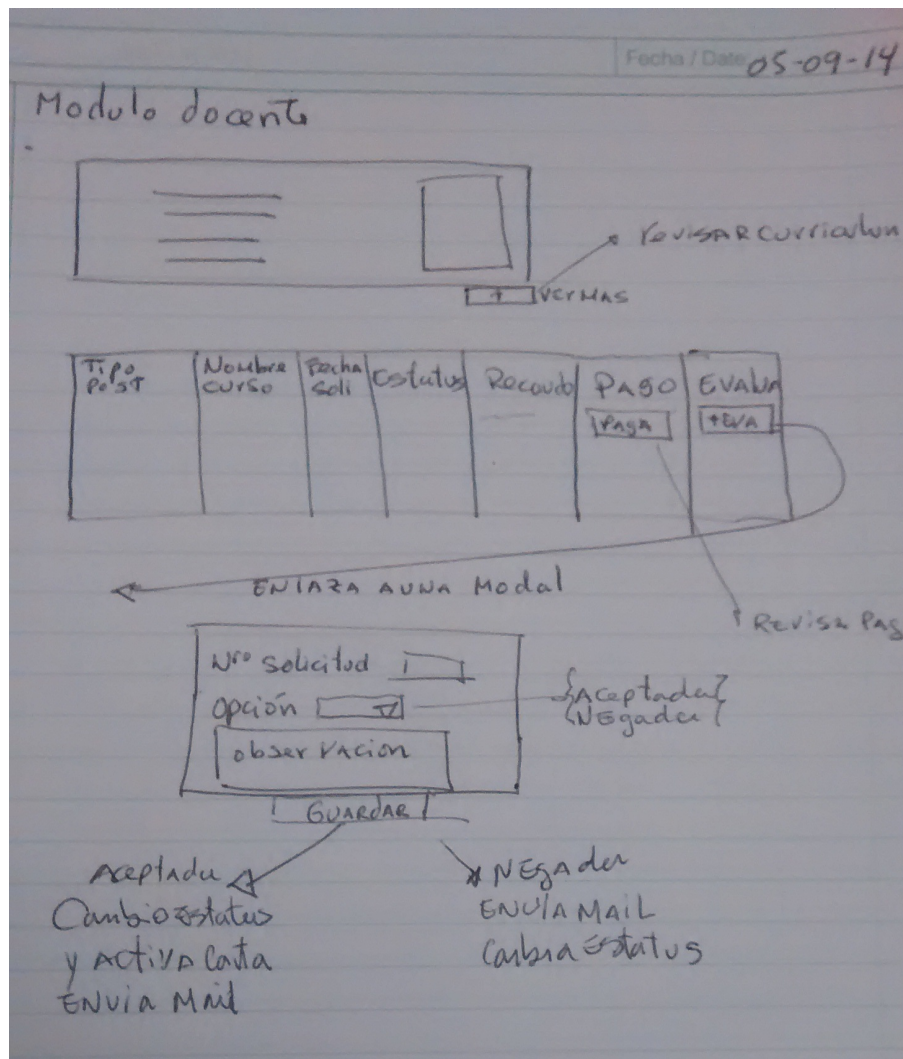


Figura 3.19: Diseño de la vista del postulante desde el módulo docente

- Fase de codificación: en esta fase el método encargado de hacer la evaluación de los

postulantes, en cuanto a desarrollo no es de mayor dificultad ya que sólo recibe por parámetro el número de solicitud y se hace una actualización del estatus en sistema, agregando la fecha de respuesta en la que la solicitud fue evaluada, luego de esto se salvan las opciones seleccionas las cuales son número de solicitud, estatus y alguna observación de la evaluación, a continuación en la figura (3.20) se puede apreciar el código que evalúa la solicitud del postulante.

```
90 def registrar_estatus
91   @solicitud = SolicitudPostulante.find_by_id(params[:solicitud][:id])
92   if @solicitud.update(params[:solicitud].permit!)
93     @solicitud.fecha_respuesta = Time.now
94     @solicitud.save
95     mensaje = "Se realizó el cambio de estatus correctamente"
96     bitacora(mensaje, @solicitud.documento_identidad)
97     flash[:mensaje] = mensaje
98     redirect_to action: :datos_postulante, "estudiante" => {"cedula" => @solicitud.documento_iden
99   else
100    flash[:mensaje] = "El cambio de estatus no pudo ser procesado"
101    redirect_to_index
102    return
103  end
104 end
```

Figura 3.20: Código que permite evaluar las solicitudes

- Fase de pruebas: las pruebas de funcionamiento tienen como objetivo verificar que la interfaz de usuario permita aceptar y rechazar solicitudes de los postulantes al postgrado, así como las funcionalidades que se tienen en el perfil de personal docente, comprobando tras varios registros y distintos casos de prueba, los resultados esperados, una vez culminada la prueba, se adapta la nueva funcionalidad al sistema, siendo esta una historia de usuario culminada y verificada satisfactoriamente. A continuación en la siguiente tabla (3.14) se muestran las pruebas de funcionamiento que se realizan.

Tabla 3.14: Prueba de funcionamiento. Iteración 5

Nro	Historias usuario involucradas	Descripción caso de prueba	Resultado esperado	Resultado obtenido	Motivo de la Falla o recomendación
22	32	Proporcionar una vista que permita registrar la evaluación de la solicitud	El sistema guarde los datos de la evaluación de la solicitud	El sistema guardo correctamente los datos de la evaluación del postulante	
23	33	Proporcionar un enlace para habilitar la descarga del reporte en formato PDF con la carta de aceptación	El sistema debe permitir que se descargue el PDF con la carta de aceptación. Soló si el estatus de la solicitud es aceptada	El sistema genera correctamente el reporte en formato PDF con la carta de aceptación del postulante aceptado	

3.14. Iteración 6 - Refactorización

En esta iteración se planifica la refactorización de cada una de las historias de usuario, en las que se requiere mejorar o solventar alguna falla durante el desarrollo.

- Fase de planificación: en el siguiente plan de iteración en la tabla (3.15) se observaban las historias de usuario a refactorizar en esta fase.

Tabla 3.15: Plan de iteración. Iteración 6

Iteración 6			
Descripción		Refactorizar y modificar de las fallas en las iteraciones 1, 2, 3, 4	
Fecha Inicio/ Fecha Fin		12-09-2014 / 19-09-2014	
Número	Fecha	Historia	Tipo
37	12-09-2014	Colocar mensaje funcional indicando un ejemplo de contraseña	Modificación
38	13-09-2014	Cambiar el campo cédula por documento de identidad	Modificación
39	14-09-2014	Cargar foto en el PDF del currículum	Modificación
40	15-09-2014	Configurar un orden específico para listar los programas de postgrados	Mejora

- Fase de diseño: para mostrar la fotografía del postulante en el currículum se proporciona una interfaz de usuario con una tabla donde se muestran del lado izquierdo

de los datos básicos del postulante y del lado derecho la fotografía correspondiente, tal como se aprecia en la siguiente figura (3.21)

Módulo Administrativo
PITA JUAN
V-18556959
Cerrar sesión

Miércoles 15 de octubre de 2014, 06:50am

Datos personales

Datos básicos

Nombres	SUSANA CAROLINA
Apellidos	HERNANDEZ CALDERON
Cédula ó Pasaporte	V-18093593
Fecha de nacimiento	16/11/1990
Pais de nacimiento	VENEZUELA
Sexo	FEMENINO
Estado civil	SOLTERO

Datos de contacto

Datos contacto

Dirección	CARACAS VENEZUELA
Entidad Federal	DISTRITO CAPITAL
Ciudad	CARACAS
Teléfono habitación	02121234567
Teléfono celular	02121234567
Teléfono trabajo	02121234567
Correo electronico	sana@gmail.com
Correo electronico alternativo	sana@hotmail.com

Estudios pregrado

Institucion universitaria	Título de educacion superior	Mención	Lugar	Fecha de ingreso	Fecha de culminación
UNIVERSIDAD CENTRAL DE VENEZUELA	LICENCIATURA DE COMPUTACION	CALCULO CIENTIFICO	VENEZUELA	01/09/2014	02/09/2014

Estudios postgrado

Clase de estudios	Nombre del programa	Institucion universitaria	Pais	Fecha de inicio	Fecha de culminación
DOCTORADO	PROFESIONAL	UNIVERSIDAD CENTRAL DE VENEZUELA	VENEZUELA	01/09/2005	06/09/2010

[Ver menos...](#)

[Descargar curriculum](#)

CONEST Versión 3.0.0 Todos los Derechos Reservados. Grupo CONEST. Copyright 2007-2013

Figura 3.21: Currículum vitae con fotografía

- Fase de codificación: en esta fase se modifica el método que se encarga de generar el currículum vitae del postulante en formato PDF, de acuerdo a la información previamente cargada en el sistema, este método es recodificado de forma que permite mostrar la fotografía del postulante en una ubicación específica en el PDF, esta genera un grado de dificultad en el desarrollo al momento de colocar la imagen en una ruta en disco, para que la gema prawn pueda leerla desde esa ruta, ya que según el estándar de CONEST la imagen debe leerse desde una ruta temporal, pero en el desarrollo se desea que la imagen sea leída desde la ruta del expediente digital en disco del postulante; otra dificultad que se presenta es posicionar la imagen dentro del PDF, ya que anexa una hoja en blanco al momento que se solicita el reporte en formato PDF. Esta dificultad se supera y se logra esta mejora de forma satisfactoria, a continuación en la figura (3.22) se muestra el código donde se incluye la foto del

postulante en la generación del currículum en formato PDF.

```

68
69 def dibujar_foto_postulante(postulante)
70   bounding_box([485, bounds.top-70], :width => 75, :height => 85) do
71     archivo = buscar_imagen(postulante,postulante.documento_identidad)
72     image archivo, width: 60, :position => :center, :vposition => :center
73     rio(archivo).rm if archivo = "#{Rails.root}/tmp/#{postulante.documento_identidad}.jpg"
74     transparent(0.9) { stroke_bounds }
75   end
76 end
77

```

Figura 3.22: Código que permite incluir la foto del postulante al documento PDF

- Fase de pruebas: las pruebas de funcionamiento que se realizan tienen como objetivo verificar que la interfaz genere el currículum en formato PDF con los datos necesarios para la solicitud del postulante, se agregue el mensaje funcional del campo contraseña, se listen en el orden deseado los programas de postgrado y se agregue el campo adicional al reporte Excel, todo esto comprobando tras varias pruebas y distintos casos de prueba, los resultados esperados, una vez culminadas las pruebas, se adapta la mejora al sistema, siendo esta una historia de usuario culminada y verificada satisfactoriamente. A continuación en la siguiente figura (3.16) se muestran las pruebas de funcionamiento que se realizan.

3.15. Iteración 7 - Puesta en producción de la aplicación

- Fase de planificación: en el siguiente plan de iteración en la tabla (3.17) se observaban las historias de usuarios a desarrollar en esta fase.

Tabla 3.17: Plan de iteración. Iteración 7

Iteración 7			
Descripción		Puesta en producción del sistema de postulación	
Fecha Inicio/ Fecha Fin		09-09-2014 / 12-09-2014	
Número	Fecha	Historia	Tipo
34	09-09-2014	Instalación del ambiente de desarrollo y producción	Nueva
35	10-09-2014	Configuración del servidor	Nueva
35	12-09-2014	Despliegue del sistema de postulación en ambiente desarrollo y elaboración de pruebas	Nueva

- Fase de pruebas: las pruebas de funcionamiento, tienen como objetivo verificar la correcta instalación de todos los componentes para poder explorar todo el sistema

Tabla 3.16: Pruebas de funcionamiento. Iteración 6

Nro	Historias usuario involucradas	Descripción caso de prueba	Resultado esperado	Resultado obtenido	Motivo de la falla o recomendación
24	37	Proporcionar un mensaje funcional para introducir la contraseña	El sistema debe mostrar un mensaje funcional que indique un ejemplo de contraseña permitido	El sistema muestra un mensaje funcional con un ejemplo de contraseña	
25	38	Permitir el registro de postulantes con cédula o pasaporte, pedir documento de identidad	El sistema debe mostrar un mensaje que indique que se permite el registro con un documento de identidad	El sistema permite correctamente el registro de postulantes con cédula o pasaporte, mostrando el mensaje funcional cédula o pasaporte	
26	39	Proporcionar en el currículum en formato PDF, la foto del postulante	El sistema deberá mostrar la fotografía del postulante, en el currículum en formato PDF	El sistema muestra correctamente la fotografía del postulante en el currículum en formato PDF	
27	40	Proporcionar una lista con el orden específico de los postgrados	El sistema debe mostrar la listas de programas de postgrado en un orden específico	El sistema muestra en el orden específico correctamente la lista de los programas de postgrado	

por medio de un recorrido visual y funcional, con el fin de constatar su correcto funcionamiento. Se logra instalar correctamente el ambiente de desarrollo y producción, se verifica el correcto funcionamiento del servidor y se despliega el aplicativo correctamente. En la figura (3.23) se muestra la vista principal del módulo de postulación de postgrado puesto en producción.



Figura 3.23: Diseño de la vista principal del modulo de postulación

Conclusiones

El desarrollo del módulo de postulación para programas de postgrados de la Facultad de Ciencias de la Universidad Central de Venezuela incidió positivamente en el funcionamiento de los procesos de la Coordinación de postgrado en la automatización del proceso de postulación, ya que permite minimizar los errores que pudiesen existir a la hora de llevar a cabo este proceso, así como unificar el mismo en las diferentes escuelas, facilitar el trabajo de cada una de las coordinaciones de postgrado de las distintas escuelas de la Facultad de Ciencias, mejorar los tiempos de respuesta en el proceso, mejorar el desempeño del personal administrativo y disminuir los costos de recursos físicos; permitiendo alcanzar el objetivo principal de forma satisfactoria; para lograrlo se contó con el apoyo del personal de la coordinación de postgrado para el levantamiento de información del proceso, así como también en el análisis de requerimientos del sistema; se coordinaron y planificaron las entregas de avances, cabe destacar, que demostraron gran interés y disposición de participar en todas las fases del desarrollo de la aplicación.

La Coordinación de Postgrado al inicio de la investigación no tenía bien definido el proceso de postulación a los programas de postgrado, con lo cual se tuvo que invertir tiempo en la correcta definición del proceso así como de los requisitos y necesidades que se estaban presentando en la coordinación para este proceso; durante este tiempo en el cual se levanta información sobre el proceso que se llevaba actualmente, se contó en todo momento con la participación del personal administrativo de la Coordinación de Postgrado, donde en conjunto se definen los alcances del sistema y sus funcionalidades principales, cabe destacar que contar con su apoyo influye en ellos de forma positiva ya que disminuye la resistencia al cambio y para los desarrolladores facilitando la información necesaria para el cumplimiento de las actividades.

Durante el levantamiento de información del proceso, se determinó que la Coordinación de Postgrado almacenaba toda la información referente al proceso de postulación en un archivo excel, por lo que se diseñó un modelo de datos para el Sistema CONEST postgrado basado en el modelo de datos utilizado en CONEST 3.0 en pregrado, con la finalidad de automatizar el proceso postulación acorde con las necesidades de los usuarios y los requisitos del sistema; así mismo se contempló la posibilidad de un crecimiento

continuo del sistema; ésta actividad no fue tarea fácil ya que se tuvo que invertir tiempo para validar que se estuvieran siguiendo los estándares de CONEST 3.0 a su cabalidad.

Trabajar bajo un ambiente colaborativo proporcionó grandes ventajas al momento del desarrollo ya que constantemente se tenían actualizaciones y funcionalidades que podían ser tomadas en cuenta para incorporarlas en el sistema de postulación, pero a su vez el hecho de seguir el estándar de CONEST 3.0 hizo necesaria la comunicación con desarrolladores de CONEST para garantizar que estos estándares se estuvieran cumpliendo en su totalidad, así como para aclarar dudas presentadas en su momento.

Para lograr el desarrollo del módulo de admisión de manera ágil, bien planificado y que se garantice la usabilidad del mismo, se utilizó una adaptación del método XP con las siguientes modificaciones: en lugar de desarrollar en pares de programadores, se desarrollara en pares bajo un mismo ambiente de trabajo, se incluirá en el proceso de desarrollo el principio de usabilidad ya que se considera un requisito indispensable en el desarrollo de software, para asegurar que este principio se cumpla se incluirán durante todo el proceso artefactos de AgilUs como lo son las guías de estilo Acosta (2011) . Se tomará en cuenta las sesiones de sincronización para mantener al equipo de trabajo actualizado del método SCRUM Proyectosagiles.org (2013); se debe resaltar que el uso de este método adaptado favorece positivamente en el seguimiento del desarrollo de la aplicación debido a que permite entregas del producto final con lo que el usuario quiere.

Durante el desarrollo de la aplicación se experimento una curva de aprendizaje y adaptación a los estándares preestablecidos, para de esta forma evitar redundancia y reutilizar las funcionalidades existentes; este período de adaptación podría ser acortado si se presentara una documentación actualizada y compartida con el listado de funcionalidades del sistema y como utilizarlo; una especie de API. Durante cada iteración en la cual se avanza de forma iterativa e incremental el desarrollo de la aplicación se experimento una grata satisfacción al confirmar la receptividad que se tenía por parte de los usuarios del sistema durante todo el proceso.

Cabe destacar que el uso de pruebas unitarias y pruebas realizadas con los usuarios finales además de detectar comportamientos no deseados en el sistema, permite en cada paso garantizar que el sistema se adapte a sus necesidades y requerimientos, así como también permite visualizar nuevas mejoras al proceso y funcionalidades no contempladas en el proceso original; para realizar esta actividad se presento dificultad en un momento dado donde la coordinación de las pruebas se vio interrumpida por el mantenimiento de los servidores de la facultad además de la época de vacaciones colectivas del personal de la Facultad de Ciencias.

La puesta en producción del sistema se realizó satisfactoriamente, donde la Coordinación de Postgrado de la Facultad de Ciencias facilitó el recurso, donde se configuró

la base de datos, el aplicativo Web, las herramientas para el desarrollo y mantenimiento del sistema, también todo lo necesario para el correcto funcionamiento del módulo; la instalación no fue trivial, ya que se deben seguir normas y procedimientos de seguridad en la red de la Universidad Central de Venezuela y de la Facultad de Ciencias, los cuales tienen tiempos de espera y tiempos de respuestas definidos.

La utilización del framework Ruby on Rails y la practicidad con la cual se maneja el lenguaje Ruby, permitieron que la solución sea presentada haciendo uso de las convenciones manejadas por el framework, pero para esto se necesitó de un período de adaptación, configuración del ambiente de desarrollo, sincronización de los desarrolladores y una investigación sobre las gemas que podrían ser utilizadas para satisfacer ciertas necesidades del sistema.

La escogencia inadecuada de las gemas en RoR en lugar de agilizar el desarrollo puede retrasarlo, como sucedió en la una oportunidad con la gema wiked; la cual se intentó usar para llevar el control en los pasos de postulación al momento de cargar la información del postulante, pero luego al notar la complejidad que implicaba adaptar el sistema a la utilización de la gema, se desistió de utilizarla. A pesar de este inconveniente el uso de RoR en base a esta experiencia fue realmente positiva, tanto al momento de desarrollo como al momento de las pruebas, de la adaptación y refactorización.

Por último el análisis previo al desarrollo sobre el proceso de postulación a los programas de postgrado de la Facultad de Ciencias, permite facilitar esta información a la Coordinación de Postgrado para ser analizada y de esta forma realizar una reingeniería al proceso para optimizarlo, y luego en base este proceso optimizado presentar la solución tecnológica del módulo Web de postulación a los programas de postgrado.

Toda esta experiencia presenta una fuente de conocimiento enriquecedora tanto a nivel personal como profesional para los autores de este Trabajo Especial de Grado; así como también representa un aporte a la comunidad de la Facultad de Ciencias al permitir al automatización del proceso de postulación a los programas de postgrado de la Facultad de Ciencias.

Recomendaciones

Como toda aplicación que se encuentra en constante crecimiento, y como CONEST postgrado no es la excepción, es importante dejar sentadas las bases que simplifiquen el diseño de nuevas funcionalidades y la realización de las futuras mejoras a la aplicación, tales como:

- Realizar las validaciones pertinentes en cuanto a las tres cartas de recomendación, enviando la solicitud de la carta de recomendación a través de un enlace Web que invite al profesor a recomendar al postulante.
- Mejorar la vista de la aplicación en los dispositivos móviles.
- Desarrollo de un modulo de estadísticas e indicadores que permita obtener reportes sobre procesos de postulación previos.

Además se deja la siguiente recomendación para el uso del sistema:

- Para poder visualizar en la aplicación los históricos de los procesos de postulación, se recomienda almacenar esos históricos en la base de datos del sistema.

Bibliografía

- Acosta, A. E. (2011). Agilus: un método ágil de desarrollo de software que incorpora la usabilidad.
- Barrantes, D., Zambrano, J., y Rivas, S. (2011). Conest 3.0.
- CONEST-UCV (2007-2014). conest.ciens.ucv.ve. URL <http://conest.ciens.ucv.ve/webapp/>.
- Ecured.cu (2014). Arquitectura cliente servidor. URL http://www.ecured.cu/index.php/Arquitectura_Cliente_Servidor.
- Escalante, S. y Ovalles, G. (2010). Desarrollo de los módulos de constancias y de comprobantes para el sistema conest postgrado.
- Gauchat, J. (2012). *El gran libro de html5 ccs3 y javascript*. Tecnicas marcombo.
- Kehoe, D. (2013). What is ruby on rails? URL <http://railsapps.github.io/what-is-ruby-rails.html>.
- Proyectosagiles.org (2013). Qué es scrum. URL www.proyectosagiles.org/.
- Rubygems.org (2014). rubygems.org. URL rubygems.org.
- Rubysource.org (2013). Getting started with mvc. URL <http://rubysource.com/getting-started-with-mvc/>.
- Sicilia, M. (2008). Visión general de la arquitectura de mysql 5.1. URL <http://cnx.org/content/m18938/latest/>.
- Sulbarán, G. y Pedrozo, N. (2009). Desarrollo de un módulo que permita manejar eventos dentro de los procesos modelados para el prototipo de la aplicación web de gestión de procesos académicos de la coordinación de postgrado de la facultad de ciencias de la universidad central de venezuela.
- Techterms.com (2014). Web application. URL http://techterms.com/definition/web_application.

Zenkius.blogspot.com (2008). Tecnologías del lado del cliente. URL <http://zenkius.blogspot.com/2008/02/tecnologias-del-lado-del-cliente.html>.