



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**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**

Trabajo Especial de Grado
presentado ante la ilustre
Universidad Central de Venezuela
por los bachilleres

**Glendy Patricia Sulbarán Goyo
Néstor David Pedrozo Pedrozo**

para optar al título de Licenciado en Computación.

Tutor: Prof. Andrés Sanoja

Caracas, Venezuela
Octubre 2009

Dedicatoria y agradecimientos

Néstor David Pedrozo:

Antes que nada, quiero agradecer a mis padres Leopoldo y Rosalba, por haber confiado en mí, por haberme guiado por el mejor camino, por el apoyo, comprensión que han tenido y por estar siempre en los buenos y malos momentos de mi vida.

A mis hermanas Mónica y Xiomara que siempre estuvieron pendientes y atentas desde el inicio hasta el fin de la carrera.

A todos mis tíos, tías que siempre me han tenido presente en sus oraciones y sinceras bendiciones pidiendo a dios me guie por el camino del bien y por mantenerme como buen ejemplo para el resto de la familia.

A todos mis primos que han sido mis mejores amigos y compañeros.

A mi amigo Gabriel Plaza por el apoyo prestado durante la trayectoria de la carrera y por ser un amigo motivador durante la realización de muchos proyectos entre ellos el seminario, trabajo especial de grado, campo laboral y vida personal.

A mi amiga Virmary Rojas por su apoyo de forma incondicional en el momento mas necesitado durante la realización del Trabajo Especial de Grado.

A la familia Martínez Sánchez por quererme y apoyarme tanto y por haberme brindarme el cariño que siempre han demostrado y por abrirme la puerta de su casa, corazón y por apoyarme desde el inicio de esta meta y por los sabios consejos por parte de cada uno de sus integrantes.

A la familia León Jiménez por haber confiado en mí y haberme brindado el apoyo desde inicio de mis estudios.

A mi amigo Edwin Díaz quien fue motivador importante antes y durante la realización y cumplimiento de esta meta.

A mi amiga y compañera durante esta aventura Glendy Sulbarán, por la confianza de-

positada en mi durante la ejecución de la carrera y por ser testigo de muchas aventuras y anécdotas.

A nuestro tutor Andrés Sanoja por haber confiado en nosotros y haber dirigido en este importante proyecto.

A todos los mencionados y faltantes, mis más sinceras gracias.

Glendy Sulbarán:

Quiero agradecer y dedicar el presente Trabajo Especial de Grado primeramente a mi familia por que siempre me han apoyado e incentivado para que logre mis metas. En especial a mi madre Gladys Goyo, mi padre Jesús Sulbarán, y mi tío Orlando Goyo porque me han dado con amor las herramientas claves para mi formación y me han sabido indicar el camino del bien.

A mi hermana Gledys Sulbarán, quien a pesar de nuestras diferencias, forma parte fundamental de mi crecimiento y ha sido mi guía y ejemplo a seguir.

A la familia Ramírez Porras por haberme mostrado el calor de un segundo hogar, por sus consejos y por compartir con emoción todos mis logros.

A la familia Pedrozo Pedrozo por tanto apoyo durante la realización de este proyecto, en especial a la Sra. Rosalba por todas su atenciones.

A las familias Sanchez Esaa y Piñero Gandarillas por siempre recibirme con los brazos abiertos y brindarme cariño y motivación.

A mis compañeros y grandes amigos Gabriel Plaza (Plazita) y Néstor Pedrozo (Señor Pedrozo) por su confianza, paciencia, cariño y preocupación para con mi persona. Por tantos momentos de aprendizaje y diversión, y por mostrarme que puedo contar con ellos en cualquier ámbito (Universidad, Trabajo y Vida personal).

A mi profesor y Tutor Andres Sanoja, por depositar su confianza en nosotros y haber aceptado liderar este proyecto. Por su dedicación, atención y paciencia.

A los profesores Jossie Zambrano y Sergio Rivas por su gran disposición y aporte a nuestra investigación.

A los profesores Wilfredo Rangel y Concettina Di Vasta por siempre estar pendientes de mí y por su cariño.

A mis amigos Pablo Mirabella, Jesús Sanchez, Loy Ramírez y Virmary Rojas por su ayuda desinteresada y sincera en los momentos más difíciles del desarrollo de este Trabajo.

A todos los amigos que he conocido a lo largo de mi carrera por todos los instantes y las enseñanzas compartidas y por todo el amor que me han dado.

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

A todos ellos muchas gracias.

Resumen

En la unidad académica de la Coordinación de Postgrado de la Facultad de Ciencias de la Universidad Central de Venezuela intervienen un determinado número de procesos donde destaca la complejidad y flujo constante de información afectando de manera directa o indirecta el tiempo de respuesta para llevar a cabo la realización efectiva de diversos trámites internos y externos que allí son demandados. El presente Trabajo Especial de Grado tiene como objetivo desarrollar un sistema de workflow u herramienta que permita monitorear los procesos y controlar su ejecución, encargándose de crear una lista de trabajo según los procesos vayan iniciándose, y según sea el perfil entre el conjunto de actores como lo son estudiantes, docentes y personal administrativo que labora en dicha organización.

La metodología de desarrollo utilizada para la gestión de esta aplicación fue una adaptación de la Programación Extrema (XP), la cual permitió la elaboración de la parte práctica del presente trabajo de investigación. Dentro de las necesidades contempladas en la aplicación se tiene lo siguiente: Mantener informado a los actores involucrados del flujo de trabajo con la información pertinente dependiente del tipo de proceso que se estuviese llevando a cabo en determinado momento. El desarrollo de la aplicación contempló el estudio e incorporación del conocimiento previamente desarrollado en los sistemas referentes a: el Sistema de Control de Estudios (CONEST) y el Sistema de Gestión de Postgrado (SIGEPOST) y el nuevo sistema Conest Postgrado.

Palabras Claves: Coordinación de Postgrados, workflow, estudiantes, docentes, Conest Postgrado.

Índice general

Introducción	1
1. Marco Referencial	4
1.1. Coordinación de Postgrado	4
1.1.1. Perfil de la Coordinación de Postgrado	5
1.1.2. Estructura Organizacional	6
1.1.3. Procesos de la Unidad Académica	10
1.2. CONEST	19
1.2.1. Introducción a CONEST	19
1.2.2. Tecnologías de CONEST	21
1.2.3. Estado actual de CONEST	22
1.3. CONEST POSTGRADO	23
1.3.1. Introducción CONEST POSTGRADO	23
1.3.2. módulos de CONEST POSTGRADO	24
1.4. Resumen del capítulo	26
2. Marco Metodológico	27
2.1. Adaptación del Proceso de Desarrollo Programación Extrema	27
2.2. Análisis general del sistema	30

	VI
2.2.1. Historias de Usuario	30
2.3. Metáfora del sistema	36
2.4. Especificaciones técnicas	37
2.5. Resumen del capítulo	38
3. Marco Aplicativo	39
3.1. Contexto del desarrollo	39
3.2. Plan de Iteración	39
3.3. Iteración 0	39
3.4. Iteración 1	46
3.5. Iteración 2	52
3.6. Iteración 3	57
3.7. Iteración 4	62
3.8. Iteración 5	65
3.9. Iteración 6	69
3.10. Iteración 7	73
3.11. Iteración 8	78
3.12. Iteración 9	83
4. Conclusiones	91
4.1. Conclusiones	91
4.2. Resultados	92
4.3. Aportes a la Investigación	95
Bibliografía	96

Índice de figuras

1.1. Estructura Organizativa de la Coordinación de Postgrado	6
1.2. Descripción del proceso de admisión	11
1.3. Descripción del proceso de selección	12
1.4. Descripción del proceso de inscripción	13
1.5. Descripción del proceso de calificación	14
1.6. Descripción del proceso de retiro de materia	15
1.7. Descripción del proceso de desincorporación	16
1.8. Descripción del proceso solicitud de constancias	17
1.9. Descripción del proceso de programación docente	18
1.10. Ambiente y usuarios de CONEST	20
1.11. Plataforma de CONEST	22
1.12. Esquema general de Conest Postgrado	24
2.1. Metáfora del sistema	37
3.1. Diagrama de actividades del flujo de trabajo para el proceso de Admisión	41
3.2. Diagrama de actividades del flujo de trabajo para el proceso de Selección	42
3.3. Diagrama de actividades del flujo de trabajo para el proceso de Retiro de Materia	43

3.4. Diagrama de actividades del flujo de trabajo para el proceso de Desincorporación	44
3.5. Diagrama de actividades del flujo de trabajo para el proceso de Calificación	45
3.6. Clases de implementación Iteración 1. Módulo Caso.	49
3.7. Métodos crear y agregar_destinatarios de la clase Caso	50
3.8. Método actualizar_status	50
3.9. Clases de implementación Iteración 2. Módulo Solicitud.	54
3.10. Método asociar_caso	55
3.11. Clases de implementación Iteración 3. Módulo Oferta Materia.	59
3.12. Método asociar_caso	60
3.13. Clases de implementación Iteración 4. Módulo Plan Docente.	63
3.14. Métodos de cambio de estado para Plan Docente	64
3.15. Método asociar_caso	66
3.16. Clases de implementación Iteración 5. Módulo Admisión.	67
3.17. Clases de implementación Iteración 6. Módulo Selección.	70
3.18. Método seleccion	71
3.19. Método cambiar_estado_caso	72
3.20. Clases de implementación Iteración 7. Módulo Evaluación.	75
3.21. Método asociar_caso para una evaluación	76
3.22. Clases de implementación Iteración 8. Módulo Calificación.	80
3.23. Método asociar_caso para una calificación	81
3.24. Clases de implementación Iteración 9. Módulo Inscripción.	85
3.25. Clases de implementación Iteración 9. Módulo Administrativo.	86
3.26. Método por_categoria	86
3.27. Método notificar_caso	87

3.28. Método archivar_caso	87
3.29. Método obtener_no_inscritos	88
3.30. Método Asociar_Caso_Inscripcion	88
4.1. Resultados de rendimiento Web para el caso de estudio Solicitudes de Retiro Total de Semestre. 100 peticiones por iteración.	93
4.2. Gráfico de resultados de rendimiento Web para el caso de estudio Solicitudes de Retiro Total de Semestre. 100 peticiones por iteración.	94

Índice de cuadros

2.1. Formato seleccionado para la bitácora de desarrollo	28
2.2. Formato seleccionado para manejar Historias de Usuario	28
2.3. Formato de registro de pruebas del lado del cliente	29

Introducción

Las exigencias del mercado y la presión de la competencia obligan a las organizaciones a ser más eficaces y eficientes en todas las áreas de servicio al cliente, producción, servicios internos y control. Los computadores de escritorio y las aplicaciones de ofimática han incrementado la eficacia y desempeño de los individuos en las organizaciones a niveles extraordinarios. Ahora, gracias a la interconexión de los PCs mediante las redes y la tecnología Web, es posible incrementar también los niveles de eficacia y desempeño de los grupos de trabajo mediante las intranets.

La realización de actividades coordinadas en las que intervienen dos a más miembros de un equipo de trabajo de acuerdo con reglas de negocio establecidas (workflow) suelen ser costosas y demandan importantes recursos a las organizaciones tanto al nivel de personal, materiales y equipos de oficina como logística. La automatización de los procesos que normalmente se realizan pasando información en papel de una persona a otra, puede representar a sus organización retrasos de procesos e insatisfacción por parte de usuarios interesados en la realización y/o culminación de trámites que permitan notificar a tiempo estados de procesos que se están en ejecución constante y cambiando sus status con cierta regularidad.

Actualmente los diferentes postgrados que se dictan en la Facultad de Ciencias y que son dirigidos por medio de la dirección de postgrado de esta Facultad, cuentan con un sistema que cumple con ciertos aspectos de los procesos llevados a cabo, sin embargo gracias a cambios que han tenido los mismos y también al surgimiento de otros nuevos, se vio la necesidad de mejorar los ya existente y automatizar todos aquellos nuevos. Por otra parte gran mayoría de los procesos que se llevan a cabo en la Coordinación de Postgrados entre estudiantes, profesores y personal administrativo ya existe un prototipo elaborado por un conjunto de desarrolladores.

En tal sentido, el objetivo fundamental de este Trabajo Especial de Grado consiste en la implementación de un sistema de workflow para la Coordinación de Postgrado que ten-

ga como función principal ayudar a manejar procesos proveyendo a los entes y usuarios involucrados tener información instantánea sobre el estado de cualquier proceso, asegurando que las tareas sean priorizadas según el status y dependiendo de a quien vaya dirigida, con la finalidad de asegurar consistencia y confiabilidad cada vez que se ejecuta un proceso, de esta manera los errores son reducidos y la aplicación garantizara el cumplimiento de las reglas de negocio preestablecidas.

Para concretar lo planteado y cumplir con el objetivo mencionado anteriormente se dará una visión de cada uno de los capítulos presentados a lo largo de este documento.

- **Capítulo I, Marco Referencial:** comprende las entidades importantes involucradas en dicho sistemas, como lo son la Coordinación de postgrado desde su perfil hasta como se realizan actualmente las tareas de forma general y como es el intercambio de información con los distintos postgrados que la entidad coordina. También se describe una entidad muy importante para el sistema como lo es el sistema CONEST, sistema referencial que aporta gran ayuda e impulso en la realización del sistema planteado.

- **Capítulo II, Marco Metodológico:** en este capítulo se explican y documentan todos los pasos que llevaron para cumplir con la meta establecida la cual fue planteada según la metodología de desarrollo ágil expuesta por la programación extrema la cual establece puntos específicos tales como planificación, diseño, codificación y pruebas. También es presentado el análisis inicial del sistema prototipo creado, la metáfora planteada y especificaciones técnicas.

- **Capítulo III, Marco Aplicativo:** en este capítulo se dan los detalles de las iteraciones efectuadas para el cumplimiento de la funcionalidades del sistema, en ella se precisa el plan de desarrollo planteado al cliente y las tareas para completar dicho plan.

- **Capítulo IV, Conclusiones:** en este capítulo se muestran los resultados, recomendaciones y aportes de la aplicación desarrollada para futuros trabajos relacionados con el sistema planteado.

Para finalizar con la estructura del Trabajo Especial de Grado se presentaran las referencias bibliográficas consultadas durante el desarrollo del documento y aplicación.

Objetivo General:

Analizar, diseñar y construir un sistema automatizado y orientado a la Web, que permita llevar el control de los procesos académicos que se llevan a cabo en la Coordinación de

Estudios de Postgrado, logrando mantener un flujo de trabajo de los procesos involucrados de tal manera de mantener informado a los entes involucrados el estado del flujo a través de notificaciones automáticas que informan a los usuarios de los elementos que requieren que se le preste atención, contribuyendo en la disminución de los tiempos de respuesta inmersos en el desarrollo de los procesos y las actividades relacionadas.

Objetivos Específicos:

- Adaptar la metodología XP para el desarrollo del sistema que gestiona el flujo de trabajo de los procesos académicos de la Coordinación de Postgrado.
- Implementar funcionalidades que permitan llevar a cabo un sistema de workflow a través de notificaciones asociadas a los procesos Académicos de la Coordinación de Postgrado.
- Desarrollar e implementar el modelo de base de datos que permitan reflejar la información concerniente a las notificaciones generadas en el intercambio de información de los procesos académicos.
- Someter a pruebas de funcionalidad a la aplicación y a pruebas de rendimiento a un módulo de la aplicación para verificar el correcto funcionamiento y la eficiencia del sistema.

Capítulo 1

Marco Referencial

La finalidad de este capítulo es presentar las bases referenciales que sirven de fundamento para el desarrollo de este Trabajo Especial de Grado. El mismo se divide en tres secciones, las cuales se describen a continuación.

En la primera sección, se hace una descripción de la organización a la cual estará destinada el sistema a implementar, en nuestro caso el mismo estará orientado a la Coordinación de Postgrado. Se tocarán puntos referentes a esta coordinación, su estructura organizacional y los procesos involucrados en el sistema.

En la segunda sección de este capítulo, se hace referencia al sistema CONEST de la División de Control de Estudio de la Facultad de Ciencias de la Universidad Central de Venezuela. Se hará un descripción del sistema, mencionando los módulos, su plataforma tecnológica y funcionamiento.

En la tercera sección se habla del nuevo sistema Conest Postgrado, en donde se describen sus módulos los cuales han sido base para el desarrollo del presente trabajo.

1.1. Coordinación de Postgrado

En esta parte del capítulo se describe el perfil de la coordinación, las unidades que constan de ella así como también las funciones de cada una y los procesos que intervienen en dicha coordinación a nivel académico.

1.1.1. Perfil de la Coordinación de Postgrado

La Coordinación de Postgrado de la Facultad de Ciencias de la Universidad Central de Venezuela(UCV), tiene por objetivo primordial coordinar y velar por el funcionamiento de todas las actividades concerniente a cualquiera de los postgrados que están bajo su responsabilidad, para ello las funciones principales que son llevadas a cabo por la coordinación son:

- Dirigir y representar al postgrado de la facultad de ciencias y velar por su buen funcionamiento.
- Convocar y presidir las reuniones de la comisión de estudio de postgrado de la Facultad de Ciencias.
- Preparar el orden del día de cada reunión de la comisión y la minuta del acta de la sesión precedente.
- Informar periódicamente a la comisión de las gestiones propias de su cargo
- Asistir a las reuniones de consejo de estudio de postgrados de la UCV y a las del Consejo de Facultad
- Elaborar el presupuesto para el funcionamiento del postgrado de la facultad.
- Ejecutar el presupuesto ordinario del postgrado de la facultad.
- Informar y divulgar a la comunidad universitaria y nacional, todo lo relativo a los programas de postgrado de la Facultad que ameriten difusión.
- Asistir al coordinador de postgrado en las reuniones de comisión de mesa y en la comisión de estudio de postgrado
- Elaborar minutas ordinarias, complementarias, puntos varios y de admisión inherentes a las reuniones de la comisión de estudios de postgrado así como también, las dirigidas al Consejo de Facultad
- Organizar y preparar el material que se utilizara en cada reunión de la comisión.
- Elaborar las actas resultantes de las reuniones de la comisión.
- Elaborar oficios dirigidos al Consejo de Facultad solicitando nombramientos de jurados, inclusión de profesores a la planta profesoral, aperturas de asignaturas, aprobación de proyecto entre otros.

- Emitir oficios para el consejo de estudios de postgrado solicitando la designación de jurados para tesis doctorales, reconocimientos de créditos y correspondencia en general
- Atender a coordinadores, profesores, estudiantes, secretarias y público en general, para brindar información relacionada con la unidad académica.
- Poner en práctica programas destinados a la consecución de recursos financieros
- Representar los postgrados de la Facultad de Ciencias ante el Consejo de Facultad

1.1.2. Estructura Organizacional

La Coordinación de la Facultad de Ciencias está a cargo de un director y a su vez esta dividida en cinco unidades de trabajo y el CIAD (Centro de Información y Automatización de Datos). Véase figura 1.1

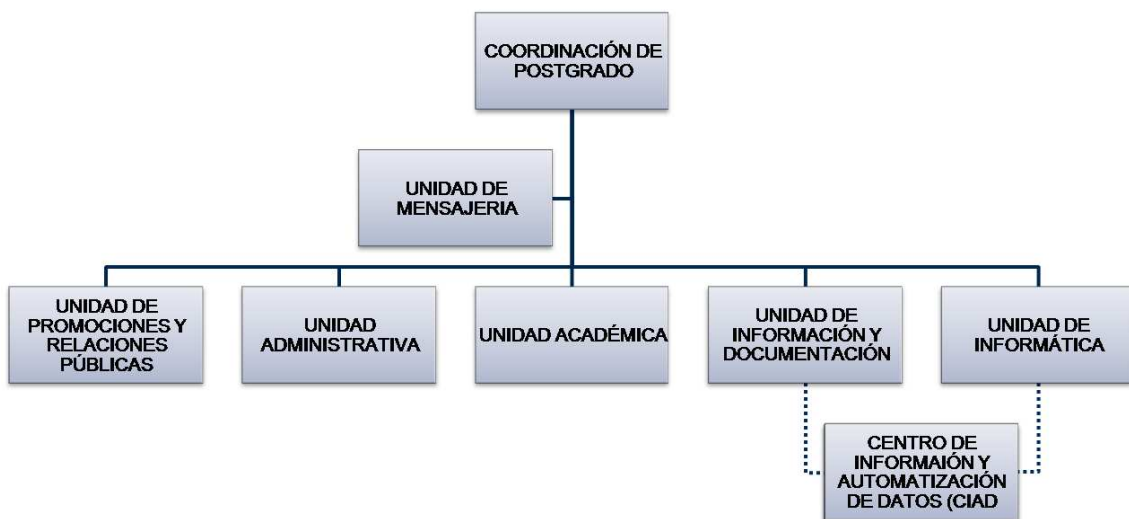


Figura 1.1: Estructura Organizativa de la Coordinación de Postgrado

Cada una de estas unidades tiene unas funciones específica dentro de la coordinación, la cual a grandes rasgos se indica a continuación:

- **Unidad Administrativa:** Tiene como fin, controlar y ejecutar las acciones necesarias que permitan prestar los servicios y la atención adecuada a cada uno de los

postgrados de la Facultad de Ciencias, a través de las asignaciones ordinarias entregadas por el C.N.U.(Centro Nacional de Universidades) e ingresos propios que controla la Coordinación de Postgrado. Para cumplir con este objetivo la unidad administrativa debe manejar todos los ingresos y egresos generados en la misma, llevar un control de la cuenta de ingresos propios de la Facultad de Ciencias y de la cuenta del consejo de estudio de postgrado de la Facultad de Ciencias.

Asimismo se encarga de la elaboración de solicitudes de cheques, viáticos, órdenes de pedidos y pagos, requisiciones, elaboración de minutas y actas de la comisión de presupuesto, adquisición y compra de materiales, equipos y suministros, seguimiento y actualización de los pagos de estudiantes, seguimiento de subvenciones de matriculas y becarios, control de caja chica, atención de estudiantes y profesores.

● **Unidad Académica:** Esta unidad se encarga de las siguientes actividades:

- Elaboración de minutas y actas de las reuniones de la CmEPG (Comisión de Estudios de Postgrado).
- Seguimiento de los trámites tratados en las reuniones de la CmEPG.
- Preparación y montaje de las reuniones del CmEPG.
- Realización de las minutas de las reuniones de la Comisión de Mesa.
- Envío de trámites al Consejo de Estudio de Postgrado. Envío de trámites al Consejo de Facultad.
- Elaboración de oficios para la Coordinación de Postgrado.

● **Unidad de Información y Documentación:** Esta unidad se encarga de las siguientes actividades:

- Proveer a los postgrados de la Facultad de Ciencias de los recursos tecnológicos, materiales y humanos que permitan la organización y difusión de los fondos documentales de sus archivos.
- Diseñar productos y servicios que permitan el acceso y difusión de la información de postgrado.
- Servir de interfaz en la transferencia de información entre la CmEPG, Consejo de Estudios de Postgrado y el Consejo de Facultad.
- Llevar memoria institucional de los postgrados de la Facultad de Ciencias.
- Aperturar, organizar, preservar y controlar los expedientes de los estudiantes regulares, irregulares, egresados y profesores de los postgrados de la Facultad de Ciencias.
- Realizar informes de gestión.
- Coordinar proyectos archivísticos.

- Apoyar a los postgrados de la Facultad de Ciencias en todo lo relacionado con la información.

- **Unidad de Informática:** Esta unidad se encarga de las siguientes actividades:

- Realizar mantenimiento a la red informática de la Coordinación del Postgrado.
- Asesorar a los postgrados de la Facultad de Ciencias en las adquisiciones y montaje de equipos informáticos.
- Programar, diseñar y mantener las bases de datos de la Coordinación de Postgrado.
- Diseñar y mantener las páginas Web de los postgrados de la Facultad de Ciencias.
- Servir de intermediario entre la Coordinación de Postgrado, el Centro de Computación de la Facultad de Ciencias y la Dirección de Informática de la Universidad Central de Venezuela.
- Promocionar los postgrados de la Facultad de Ciencias vía Internet.
- Prestar servicios de atención al público.
- Dictar cursos de computación al personal de la Coordinación de Postgrado.
- Controlar la seguridad de la red de la Coordinación de Postgrado.

- **Unidad de Promociones y Relaciones Públicas:** Esta unidad se encarga de las siguientes actividades:

- Promocionar los postgrados de la Facultad de Ciencias a nivel nacional e internacional.
- Preparar los actos de graduación de los postgrados.
- Atender a los graduandos, asesorándolos y facilitándoles la realización de los trámites para la graduación.
- Diagramar y publicar el material de difusión de la Coordinación de Postgrado y los actos de graduación.
- Realizar el proceso de pre-inscripción de los postgrados.
- Atender al público y servir de intermediario en relaciones públicas con personalidades y entidades públicas o privadas.

- **Unidad de Mensajería:** Esta unidad se encarga de las siguientes actividades:

- Distribuir y entregar la correspondencia tales como: oficio, memorandos, libros, periódicos, folletos y cheques entre las distintas dependencias a las cuales van dirigidas.

- Reproducir material académico y administrativo en la fotocopidora.
- Preparar y clasificar correspondencia general.
- Manejar equipos de oficina como: encuadernadora, plastificadora, fotocopidora, guillotina, engrapadora industrial, entre otros.
- Instalar equipos de computación.
- Trasladar artículos de oficina de una dependencia a otra, dentro de las instalaciones de la institución.
- Colaborar en cualquier eventualidad que se presente en las otras unidades de la Coordinación de Postgrado siguiendo instrucciones dada por el supervisor inmediato.
- Cumplir con las normas y procedimientos de seguridad integral establecidos por la organización.

● **Centro de Información y Automatización de Datos(CIAD):** Es un centro que brinda a la comunidad de postgrado de la Facultad de Ciencias de la UCV(Universidad Central de Venezuela), servicios informáticos que faciliten las labores académicas y de investigación. Este centro está conectado con el SIDEPUCV

1.1.3. Procesos de la Unidad Académica

1.1.3.1 Descripción del servicio

El módulo académico tiene como fin controlar y ejecutar los trámites académicos que permiten prestar servicio y atención a los estudiantes pertenecientes a los distintos postgrados de la facultad.

1.1.3.2 Descripción del proceso

Maneja y controla la información de los estudiantes durante su permanencia en el postgrado. Así mismo, se encarga de la elaboración de minutas y actas de las reuniones de la Comisión de Estudios de Postgrado (CmEPG), seguimiento de los trámites tratados en las reuniones de la CmEPG, preparación y montaje de las reuniones de la CmEPG, envío de trámites al Consejo de Estudios de Postgrado, envío de trámites al Consejo de Facultad, elaboración de oficios para la Coordinación de Postgrado, entre otros.

Cabe destacar que los procesos internos que se manejan en esta unidad, involucra la intervención de otras dependencia externas a la coordinación, pero que están en constante comunicación, como son el control de estudio de cada postgrado de la Facultad de Ciencias, siendo en total trece (13).

A continuación se muestran los procesos internos. La representación de los mismos fue mediante diagramas de actividades, con una breve descripción de cada uno.

Proceso de Admisión.

En el proceso de admisión, el postulante al postgrado, cancela en caja un monto establecido por la coordinación para obtener el material de preinscripción . Este llena una planilla de solicitud de admisión que se obtiene desde la página Web del postgrado, y la entrega junto con otros recaudos a la Coordinación de Postgrado de la facultad de ciencias. En la recepción de la Coordinación de Postgrado se revisa la documentación, la misma se registra en una sistema (hoja de cálculo), se hace un duplicado del mismo, se archiva el original y se envía copia al control de estudio del postgrado solicitado.

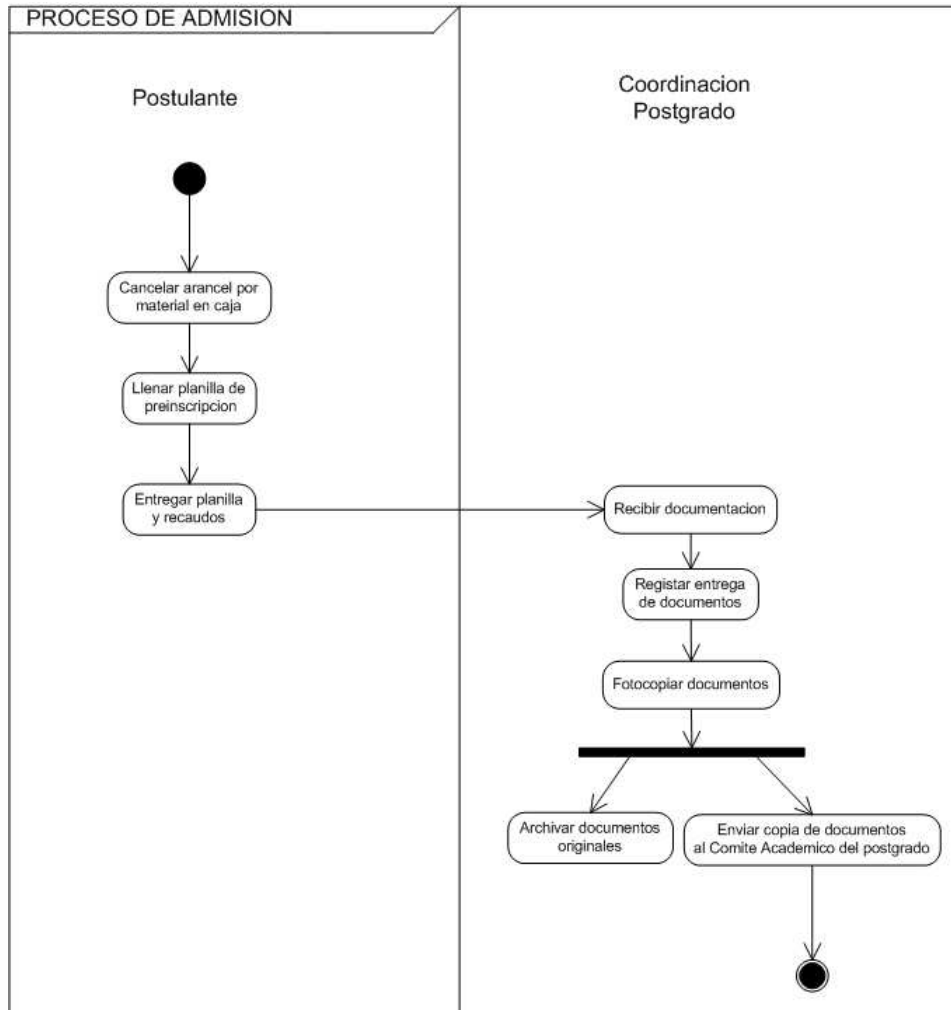


Figura 1.2: Descripción del proceso de admisión

Proceso de Selección.

En el proceso de selección, el postulante realiza una prueba de suficiencia de un idioma extranjero, evaluación de credenciales y entrevista. El comité académico del postgrado correspondiente evalúa los resultados obtenidos y en caso de considerarse que el aspirante necesite nivelación, se le manda a cursar materia(s) requerida(s) en el pregrado. Se crea un oficio de admisión indicando si el postulante fue aceptado o no y se envía a la recepción de la Coordinación de Postgrado, en donde se registra en sistema el documento. En el caso en el que el postulante fue aceptado, se crea el expediente del postulante y se crea un oficio-respuesta, enviando una copia al comité académico del postgrado y otra al postulante, este retira en la Coordinación de Postgrado una planilla para formalizar su inscripción por secretaria central, mientras que el comité académico del postgrado archiva el oficio-respuesta.

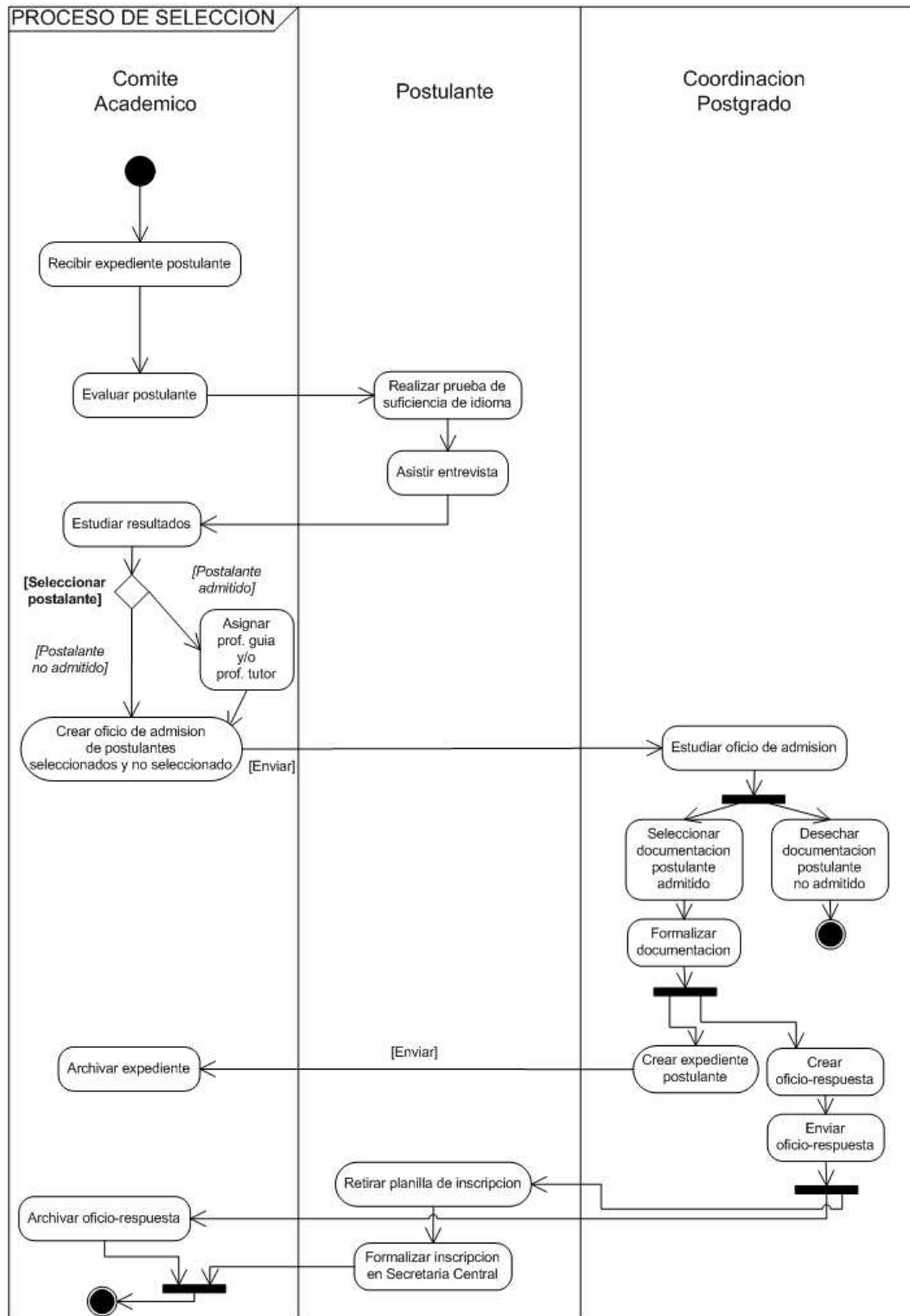


Figura 1.3: Descripción del proceso de selección

Proceso de Inscripción.

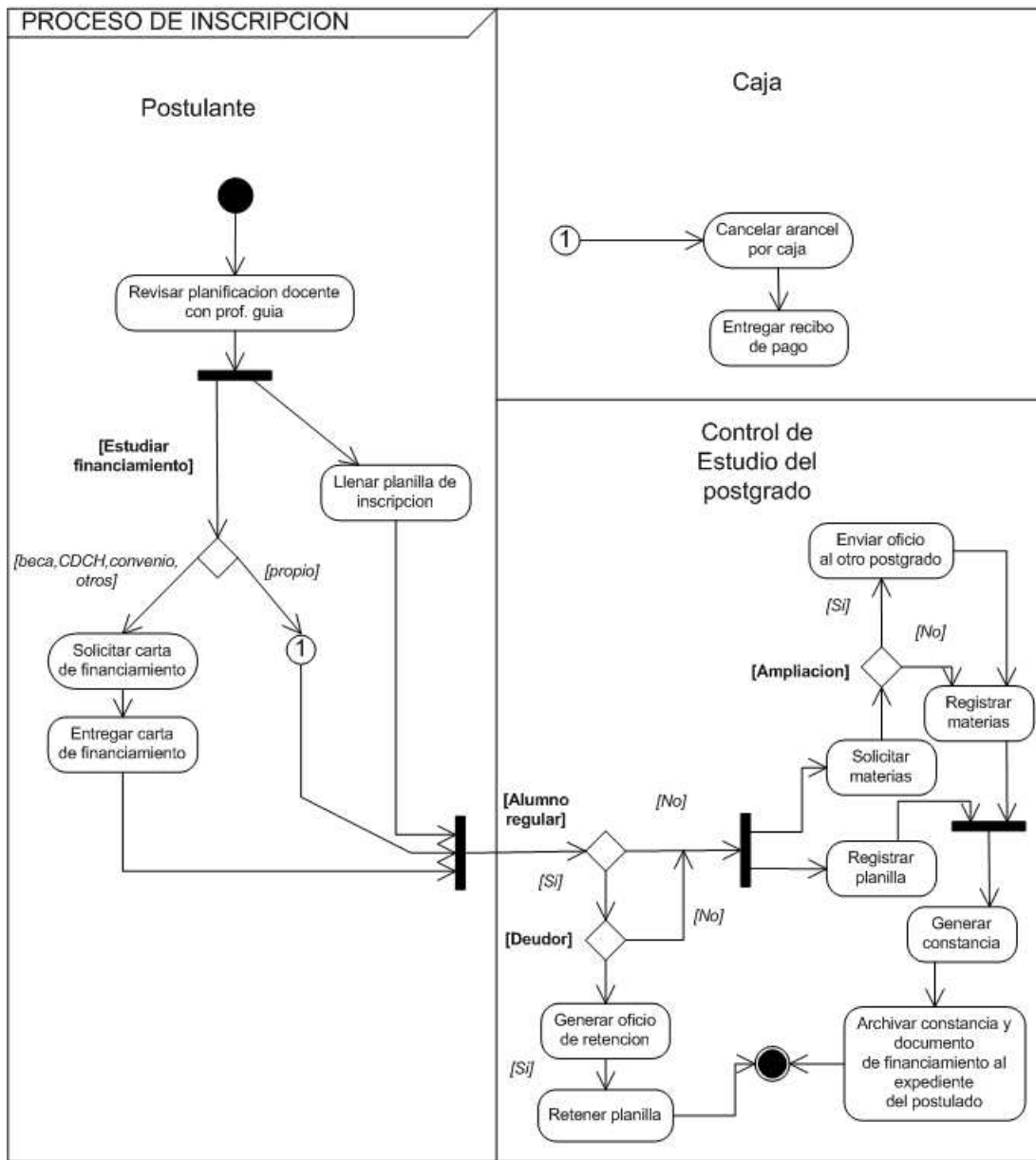


Figura 1.4: Descripción del proceso de inscripción

En el proceso de inscripción, el estudiante revisa la planificación docente y en conjunto a su profesor guía planifica y organiza las materias a inscribir. En caso de financiamiento propio, cancela por caja el arancel respectivo o en su defecto si es por terceros, solicita una carta de financiamiento (CDCH, beca, convenio). En el control de estudio del postgrado, entrega el recibo de pago o carta de financiamiento según sea el caso, y la planilla

con la firma del profesor guía y la firma del profesor tutor si es requerido. En caso de ser estudiante regular se chequea el expediente para verificar que no tenga deudas. De poseer deuda se retiene la planilla, se crea un oficio indicando que el estudiante presenta deuda, y se cancela la inscripción. Se registra los datos de la planilla en sistema. Se le informa al estudiante las materias abiertas en la programación docente del periodo a comenzar. El mismo menciona cuales materias va a inscribir y se registra en sistema. Para inscripción de materias de estudiantes de otros postgrado (ampliación) se registra y se envía un oficio al postgrado correspondiente. Se le entrega una constancia de inscripción indicando las asignaturas inscritas. Se envía un oficio y todos los originales del estudiante inscrito a la Coordinación de Postgrado.

Proceso de Calificación.

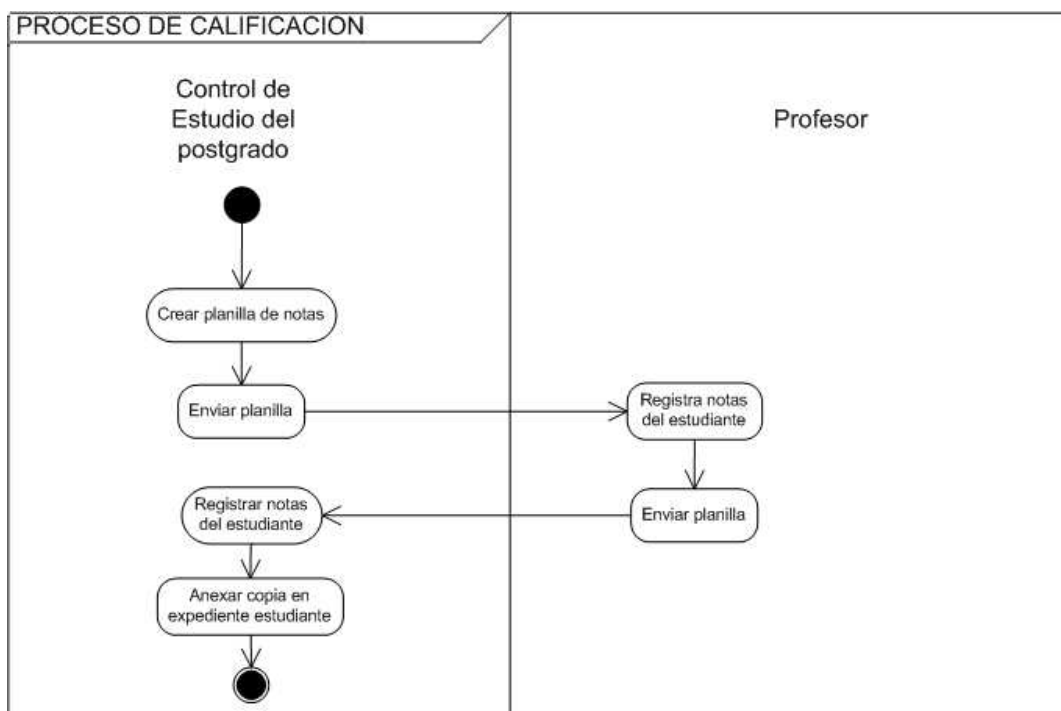


Figura 1.5: Descripción del proceso de calificación

En el proceso de calificación, el control de estudio del postgrado, se genera las planillas de notas a mitad de semestre y se reparte a cada profesor. El profesor califica los estudiantes y envía planilla al control de estudio, en donde se carga las notas en el record de notas de cada estudiante y se anexa constancia en el expediente del mismo.

Proceso de Retiro de Materia.

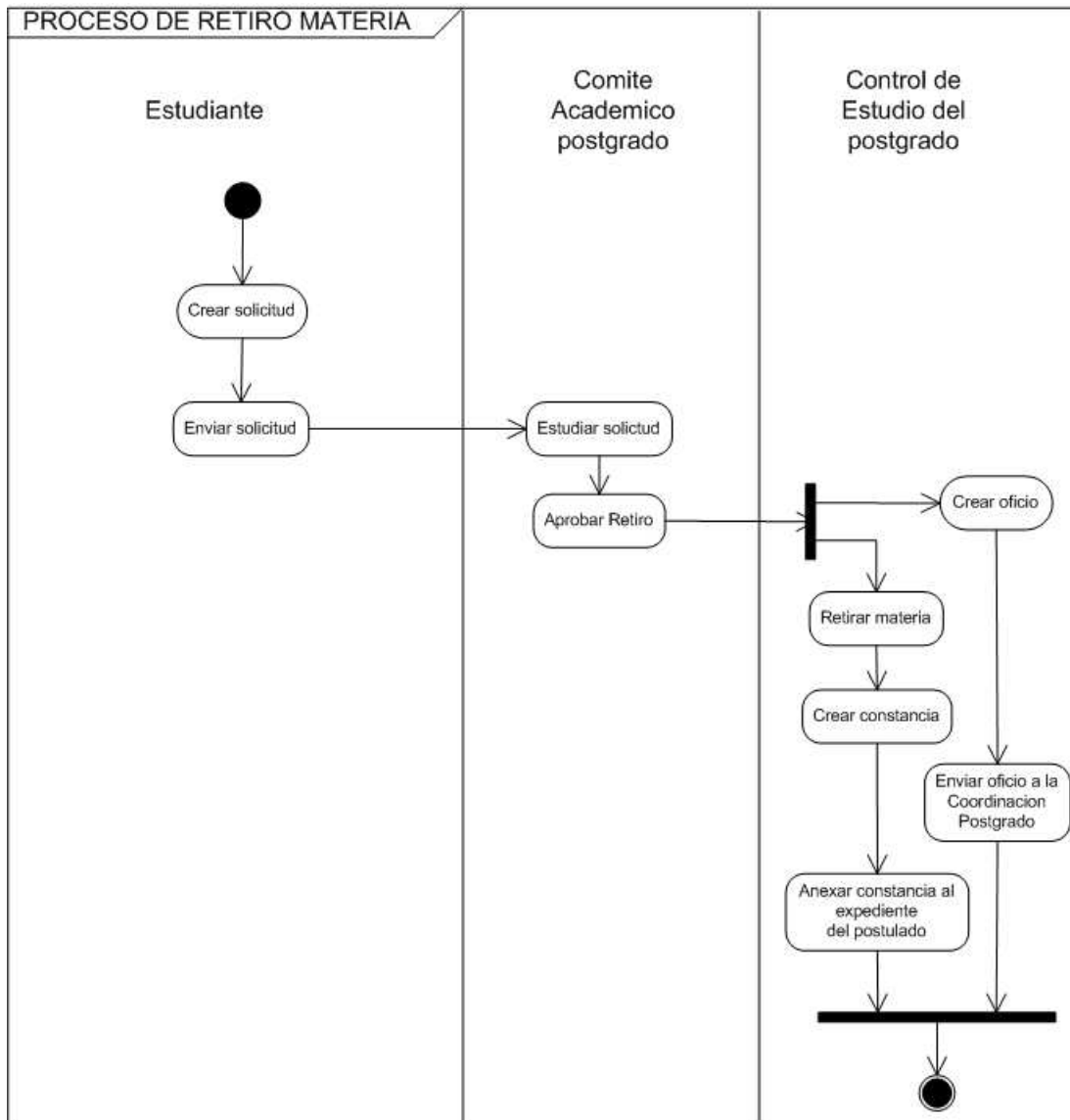


Figura 1.6: Descripción del proceso de retiro de materia

En el proceso de retiro de una materia, el estudiante realiza una carta indicando la materia a retirar, entrega la carta a la comisión académica, luego se estudia la solicitud y se procede a retirar la materia del sistema, se crea un oficio el cual se envía una copia a la Coordinación de Postgrado y otra se anexa al expediente del estudiante.

Proceso de Desincorporación.

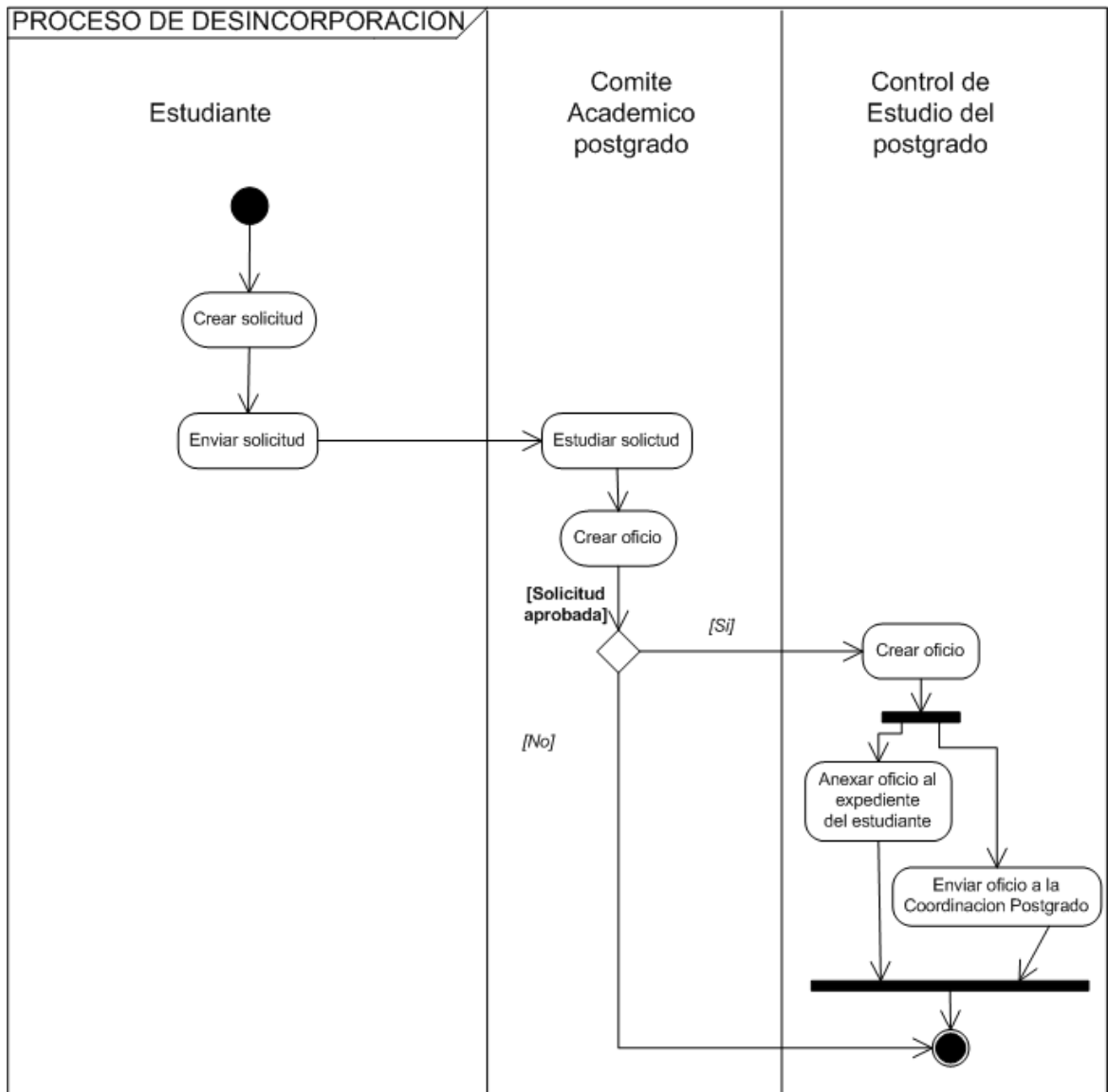


Figura 1.7: Descripción del proceso de desincorporación

En el proceso de desincorporación, el estudiante realiza un oficio indicando los motivos, avalado por su tutor o profesor guía. entrega el oficio a la comisión académica en donde se evalúa la solicitud, de aprobar la desincorporación se cambia el estado del estudiante y se crea un oficio enviándolo a la Coordinación de Postgrado.

Proceso de Solicitud de Constancias.

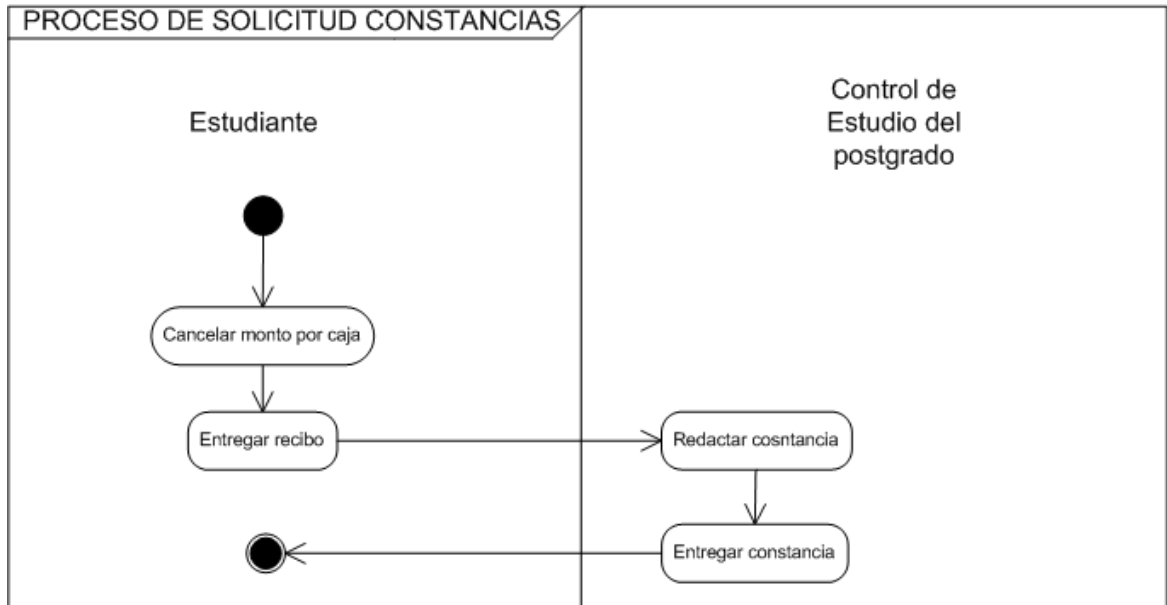


Figura 1.8: Descripción del proceso solicitud de constancias

En el proceso de solicitud de constancias, el estudiante cancela un arancel por la constancia en caja, entrega el recibo al control de estudio en donde se redacta la constancia y se entrega al estudiante.

Proceso de Programación Docente.

En el proceso de programación docente, el profesor que desee proponer una materia envía un comunicado dirigido al comité académico, este comité propone una planificación docente por ejecutar, la cual envía a la coordinación de estudios de postgrados. Este estudia dicha programación, crea un oficio con las revisiones de la misma y lo envía de vuelta al comité donde se hacen los ajustes descritos en dicha revisiones y se define la programación docente ejecutada, es enviado una copia a la coordinación donde se verifica, archiva y finalmente se publica.

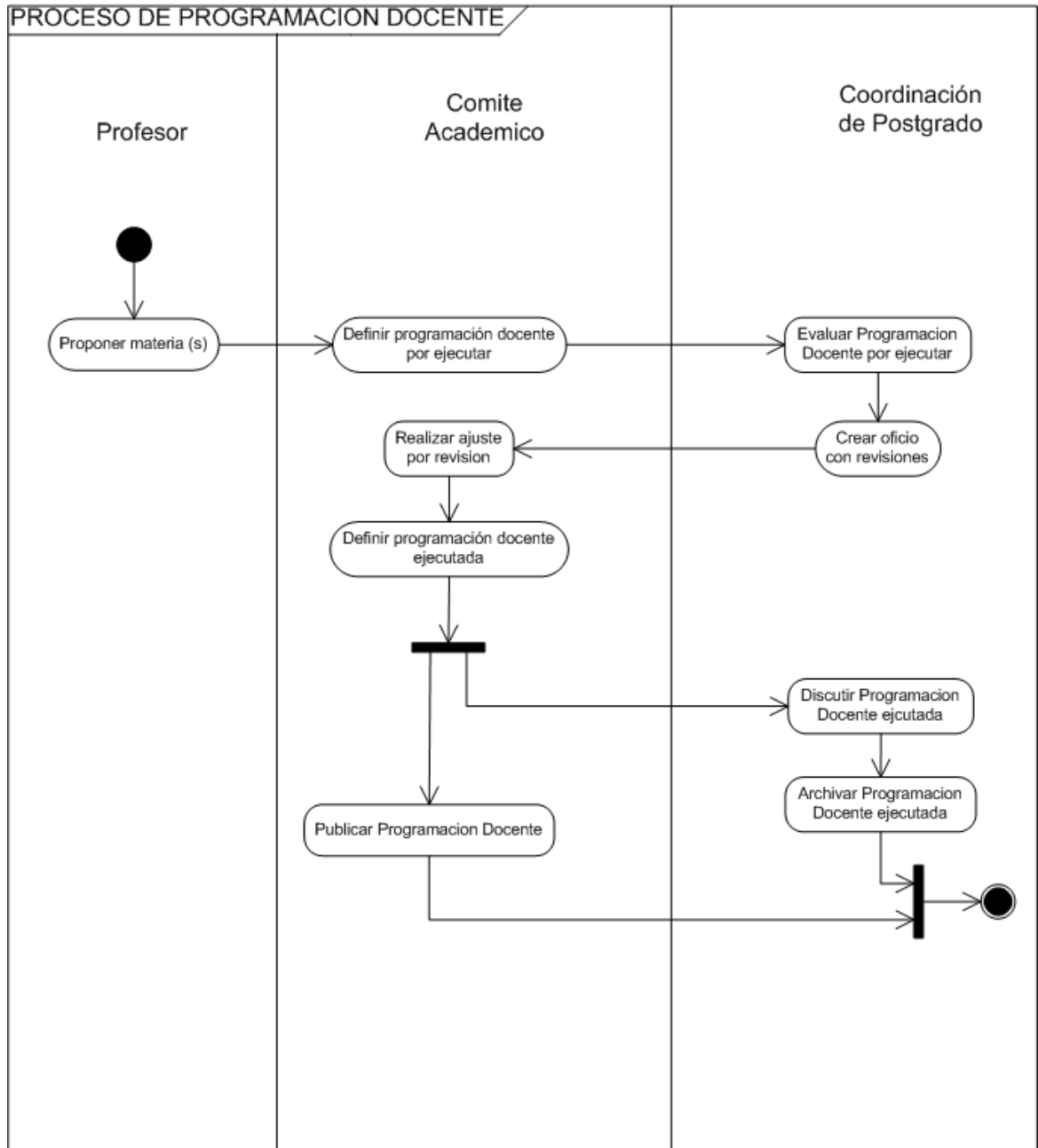


Figura 1.9: Descripción del proceso de programación docente

1.2. CONEST

Esta parte del capítulo se reseña CONEST en cuanto a su desarrollo, uso y práctica, su estructura, las tecnologías implementadas en ella y la situación actual de dicha aplicación.

1.2.1. Introducción a CONEST

CONEST es un sistema encargado de todo lo concerniente a los procesos académicos de las diferentes entidades con las que se relaciona la División de Control de Estudios (DCE), como lo son las dependencias internas y externas, y las escuelas de la Facultad de Ciencias. Su sigla significa *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.

CONEST administra, gestiona y mantiene datos de aproximadamente 3600 estudiantes de las licenciaturas de Biología, Computación, Física, Geoquímica, Matemática y Química. Es un sistema orientado a servicios de usuarios y cuenta con un modelo de datos de 5 millones de registros, 140 tablas que contienen la información de más de 37 mil estudiantes que han pasado por la Facultad de Ciencias

La iniciativa nace de docentes de la licenciatura de computación, y éstos en conjunto con sus estudiantes de laboratorios, pasantes, seminaristas, y tesistas conforman el equipo de trabajo el cual a partir de octubre del 2006 comenzó con el análisis, diseño y desarrollo de lo que hoy es una de las herramientas más importante de la comunidad de la Facultad de Ciencias.

El sistema es una aplicación Web para ser usada por estudiantes, docentes y personal administrativo de las diferentes entidades en donde se aprovecha de la tecnología para minimizar el uso de los recursos materiales y humanos (Ver figura 1.10). Se encuentra formado por un conjunto de módulos que funcionan de manera integrada, y se obtiene una mayor satisfacción de los usuarios debido al mejoramiento y la ampliación de los servicios prestados.

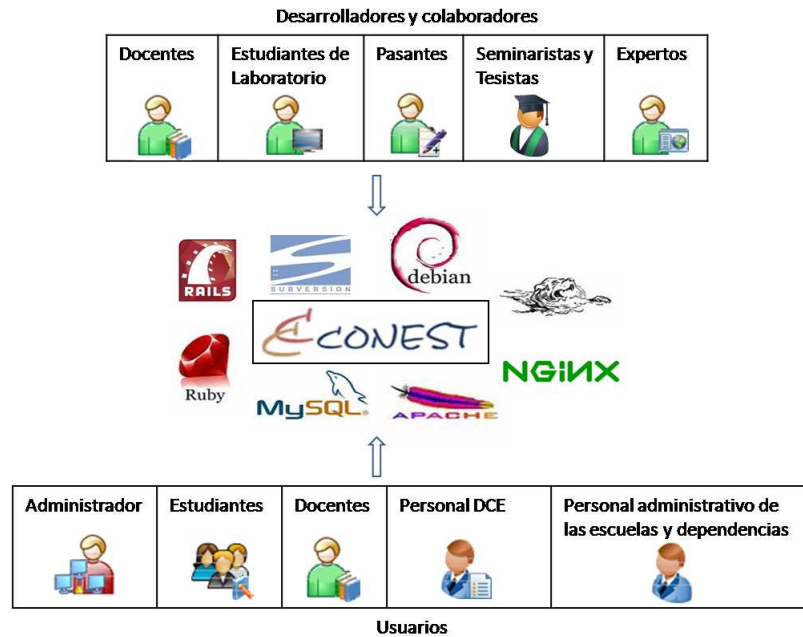


Figura 1.10: Ambiente y usuarios de CONEST

El enfoque de CONEST ha sido el renovar los servicios que ofrece y para ello se creó con ciertas particularidades que facilitarían su desarrollo y actualización. Algunas características de CONEST son:

- Implementado en Ruby on Rails bajo la arquitectura Modelo Vista Controlador (MVC), permitiendo una buena organización del trabajo debido a la independencia de los componentes que la conforman y mucha flexibilidad en el proceso de realización de cambios.
- Manejado con la herramienta Subversión (SVN) que permite a los desarrolladores establecer sincronización de versiones del código y que trabajen en ambientes colaborativos donde pueden aprender del trabajo realizado por otros integrantes del grupo.
- La base de datos utiliza un repositorio centralizado, normalizado, flexible y de alta configuración permitiendo a futuro que la aplicación pueda ser adaptada en otros contextos académicos similares.
- Posee una interfaz de usuario que es común para todos los módulos en cuanto a la diagramación, colores, banners, fotos, estilo, etc., cumpliendo con los principios, lineamientos y estándares definidos con anterioridad y a los cuales deben adaptarse los desarrolladores.

- Tiene establecida una estructura y organización para su fácil comprensión como por ejemplo estándares en la base de datos (nombre de tablas, atributos, etc.), haciendo que el mantenimiento del sistema sea sencillo.
- Sus funcionalidades pueden ser accedida de forma remota desde cualquier navegador las 24 horas del día, los 365 días del año. Entre sus funcionalidades no solo está los procesos de inscripción, grado y calificación, sino también ver el estado de dichos procesos y generar reportes relacionados en distintos formatos, además del envío de información a estudiantes y docentes a través de sus correos electrónicos como por ejemplo la fecha de inicio y fin del proceso de inscripción.

1.2.2. Tecnologías de CONEST

Para el desarrollo de CONEST se pensó en utilizar herramientas y tecnologías con licenciamiento libre. Entre los lenguajes en el que ha sido desarrollado está XHTML, CSS, JavaScript, XML, y Ruby. Tiene también el servidor SMTP Exim para el envío de correos, MySQL como sistema manejador de base de datos y SVN como manejador de versiones. CONEST funciona sobre el sistema operativo Linux Debian.

Las aplicaciones Web han sido desarrolladas con el framework de Rails, se realiza la conexión a la base de datos MySQL a través de su componente ActiveRecord. Adjuntamente Mongrel será el encargado de atender las peticiones HTTP que lleguen del servidor web Apache, esto se da a través de Nginx quien redirige y balancea la carga hacia los Mongrels, cada Mongrel ejecuta la aplicación Rails en el intérprete de Ruby. En Ruby es requerido unos componentes adicionales para ser utilizados en ciertos módulos, como por ejemplo en la generación de reportes.

En la Figura 1.11 se puede apreciar la disposición de cada una de las herramientas que conforman la plataforma CONEST, las cuales están ordenadas según su niveles de instalación y acceso, comenzando por el sistema operativo Linux sobre el cual se instala dichas herramientas que logran el funcionamiento de la aplicación.

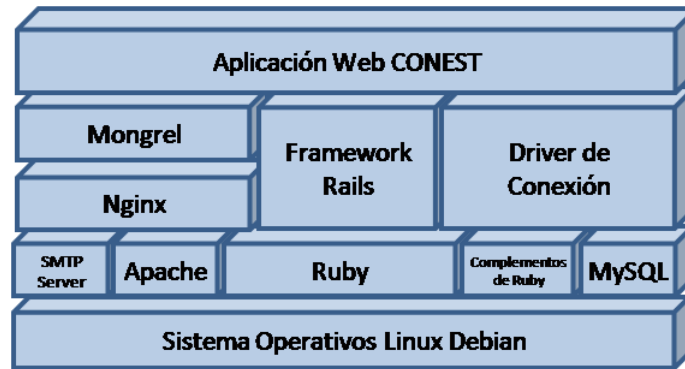


Figura 1.11: Plataforma de CONEST

Para la ejecución de la aplicación basta con que el usuario use cualquier navegador y haga la petición y conexión al servidor de CONEST para visualizar las páginas y acceder a las distintas funcionalidades. Las peticiones viajan por Internet y son recibidas por el servidor Web Apache.

1.2.3. Estado actual de CONEST

Actualmente el desarrollo del sistema CONEST apunta hacia la creación de una nueva versión llamada CONEST 2 la cual abarca la reingeniería de los procesos internos de la aplicación y donde se piensa en la actualización de algunas herramientas de este sistema a versiones estables, contemplando una serie de funcionalidades adicionales y la reutilización de ciertos métodos, aprovechando las potencialidades que ofrece el entorno Web. También se rediseña y ajusta la estructura de la base de datos, mejorando y creando relaciones y campos necesarios, y se están estableciendo nuevos estándar en las interfaces, en la programación, y en la base de datos.

1.3. CONEST POSTGRADO

En esta parte del capítulo se expone el nuevo sistema Conest Postgrado, los módulos implementados y el estado del mismo.

1.3.1. Introducción CONEST POSTGRADO

Dado que las actividades llevadas a cabo en los procesos académicos de los diferentes postgrados y de la Coordinación de Postgrado son en gran parte manuales, y que en el flujo de trabajo que engloba la ejecución de las tareas había pasos repetidos, se presentaban fallas ocasionando aumento de costos en cuanto a recursos físicos y de tiempo. Esto pasa a ser considerado el problema principal, agregando otro como la comunicación entre diversas unidades que por la demanda que presentaban, cada una por separado acumulaba sus tareas y requerían la creación de informes y reportes de distintos tipos que generaban retrasos.

La investigación fue enfocada en la Coordinación de Postgrado, Ente central donde llegan las solicitudes, notificaciones y reportes originados en los controles de estudio de los postgrados de la Facultad de Ciencias, dicha documentación llega sin formato estándar, lo que genera actividades extra de transcripción al personal administrativo de la Coordinación mencionada, incrementando la carga laboral y la demora en el flujo de trabajo.

El estudio de las actividades de la Coordinación de Postgrado inicia en julio del 2008 y una vez identificada la problemática, en febrero del 2009 se propone sistematizar los procesos académicos de esta Coordinación y ayudar a la comunidad involucrada en dichos procesos. En este estudio se evidenció la necesidad de que cierta documentación y datos requerían ser registrados y procesados, para llevar control y estadísticas que puedan usadas a futuro dentro de la Coordinación de Postgrado y que puedan ser solicitadas por otros entes dentro de la universidad. Además se observa que como la información es compartida hay que darle prioridad a la estandarización.

Se implementa entonces una aplicación Web que tenga como función automatizar la información institucional de los procesos académicos y administrativos de la Coordinación de Postgrado. A esta aplicación se le denomina como Sistema Conest Postgrado y va a estar compuesta de los procesos académicos estudiantiles como: preinscripción, inscripción, selección, solicitud de constancias, solicitud de retiros de materia, solicitud de desincorporación y de los procesos académicos docentes como: ofertar materias, planificación docente, calificación de forma definitiva y calificación de forma continua.

Para la realización de esta aplicación es tomado como base el Sistema de Control de Estudio (CONEST), del cual se obtiene el modelo de datos y los lineamientos generales,

logrando establecer un único sistema de gestión de los procesos administrativos de la docencia, que sea funcional a nivel de pregrado y postgrado. También se investiga el Sistema de Gestión de Postgrado (SIGEPOST) del cual se toma aspectos del modelo de datos y los registros.

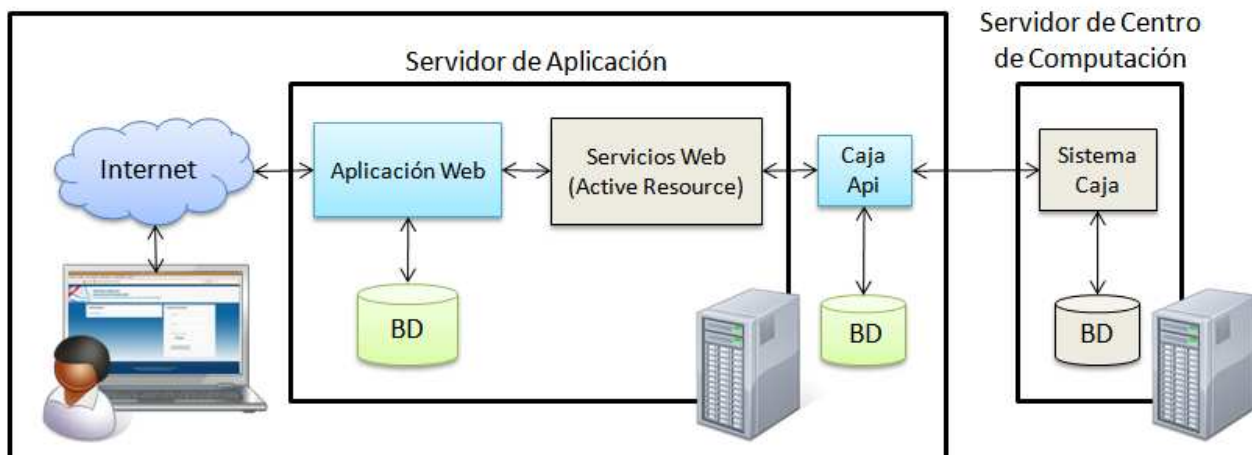


Figura 1.12: Esquema general de Conest Postgrado

1.3.2. módulos de CONEST POSTGRADO

La aplicación Conest Postgrado inicia con un conjunto de módulos independientes entre sí que pueden ser accedidos según los roles que cumplan los usuarios en el sistema postgrado de la Facultad de Ciencias y ofrecen sus servicios para solventar problemas a dichos usuarios. A continuación se describe los módulos implementados en el nuevo sistema Conest Postgrado:

- **Módulo Admisión:** Implementa funcionalidades que permiten llevar a cabo procesos inherentes a la admisiones estudiantiles. Acá el aspirante registra en el sistema datos personales e información de la carrera que aspira, además dato de un recibo de pago del material de preinscripción que se le es entregado previa cancelación en caja. Luego es generado un comprobante de preinscripción y se da opción de descargar el material para formalizar la solicitud. Cuando es consignado dicho material en la Coordinación de Postgrado, el personal administrativo los digitaliza en el sistema y archiva los originales. Por último la dependencia del postgrado solicitado es notificada del registro de un nuevo aspirante.
- **Módulo Selección:** Implementa funcionalidades que permiten llevar a cabo procesos inherentes a la selecciones estudiantiles. Una vez realizado el registro del aspirante y la recepción de documentos, el comité académico procede a evaluarlo,

para ello es requerido el ingreso al sistema donde se visualiza la lista de aspirantes, y se puede acceder al detalle y documentos de cada uno. Luego se acepta o no al aspirante, y ese resultado es notificado a la Coordinación de Postgrado, de ser aceptado se guarda como un usuario más del sistema pasando formar parte de la matrícula estudiantil y asignándosele el docente guía y opcionalmente del docente tutor.

- **Módulo Inscripción:** Implementa funcionalidades que permiten llevar a cabo procesos inherentes a la inscripciones estudiantiles. El estudiante ingresando al sistema podrá seleccionar de un listado las materias a inscribir, en donde indica la forma de pago bien sea propia o financiada. En caso de ser propia deberá introducir el número del ticket de pago, y en caso de ser financiada indicar el nombre de la institución que se hará responsable del pago. Por último podrá descargar el comprobante de inscripción

- **Módulo Solicitud:** Implementa funcionalidades que permiten llevar a cabo los procesos inherentes a las solicitudes estudiantiles. Las solicitudes que se pueden realizar son retiro de materia, desincorporación y solicitud de constancia.

Para el retiro de materia el estudiante ingresa al sistema, ve el listado de las materias activas y selecciona las que desee retirar, siempre y cuando mantenga cursando al menos una. Luego de esto se notifica al docente a cargo de la materia y al comité académico para que evalúe la solicitud. En caso de ser aceptada el sistema retira las materias seleccionadas y el estudiante es notificado.

Para la desincorporación, el estudiante indica el motivo por el cual desea desincorporarse. Luego la solicitud es estudiada por el comité académico, y en caso de ser aceptada el sistema procede al retiro formal de las materias y se cambia el estado del estudiante, este es notificado de la aprobación o no de la solicitud.

Para la solicitud de constancia el estudiante selecciona la que desea generar e introduce el número de ticket entregado por la caja de la facultad, luego de validado se puede descargar la constancia solicitada.

- **Módulo Oferta Materia - Planificación Docente:** Implementa funcionalidades que permiten llevar a cabo los procesos inherentes a ofertar materia por docente. El proceso inicia, con la oferta de materias por parte del docente, quien al ingresar al sistema seleccionará las materias que desee dictar para el período en curso, si desea ofertar una materia que no se encuentre en el listado, podrá proponerla ingresando la información básica de la misma. Con este conjunto de propuestas, el comité académico, seleccionará cuales deben formar parte de la oferta académica o planificación docente del período en curso. Al aceptar una materia, se deberá asignar el coordinador a cargo, quien será notificado vía correo electrónico, el número de secciones a abrir y el número de estudiantes por sección.

- **Módulo Evaluaciones:** Implementa funcionalidades que permite al coordinador de una materia asociarle un conjunto de evaluaciones, cada una de éstas identificadas con un tipo de evaluación (parcial, quiz, proyecto, entre otros), un porcentaje y una fecha. Al sumarse el porcentaje de cada evaluación debe dar un total de 100 %. También se da la opción de editar evaluaciones existente para una materia.
- **Módulo Calificaciones:** Implementa funcionalidades que permiten al coordinador de una materia llevar a cabo la calificación de sus estudiantes. La calificación puede ser continua, en el cual se le asigna una nota a cada evaluación que tenga la materia y se calcula el acumulado, o definitiva en la cual se asigna un nota global de la materia. Para cuando la calificación es definitiva se da la opción de guardar temporalmente la calificación de algunos estudiantes o formalizar la calificación de todos los estudiantes.

1.4. Resumen del capítulo

En este capítulo, se definió los conceptos fundamentales para la elaboración del presente Trabajo Especial de Grado. Se hace referencia a los entes y sistemas que sirvieron como guía para la identificación del problema y el planteamiento de la solución, como lo son la Coordinación de Postgrado de la Facultad de Ciencias, organización en el que se enfoca la investigación, describiendo su perfil, estructura organizacional y los procesos académicos que se llevan a cabo en ella. Otra referencia es el sistema CONEST de la División de Control de Estudios de la Facultad de Ciencias, del cual hace una reseña de su plataforma, su funcionamiento y su estado. Por último se hace referencia al sistema prototipo CONEST POSTGRADO el cual pasó a ser nuestro sistema base por poseer la implementación de los procesos académicos a los cuales se le modelará el sistema workflow.

Capítulo 2

Marco Metodológico

En este capítulo se presenta la metodología utilizada para la realización del desarrollo de la aplicación, la cual fue la adaptación de Programación Extrema (XP), metodología ágil de desarrollo que permite según su concepción definir tareas según la interacción que se tengan con los usuarios, estableciendo tiempo de realización y culminación de dichas tareas y entre el conjunto de desarrolladores definir historias de usuarios asociadas a las tareas previamente definidas. No se enfoca en la documentación sino en los requerimientos planteado por el cliente.

En el marco metodológico de este Trabajo Especial de Grado se presenta la descripción del proceso de desarrollo a seguir y su adaptación, seguidamente las etapas del proceso, actividades, resultados y estrategias involucradas en el desarrollo de la aplicación.

2.1. Adaptación del Proceso de Desarrollo Programación Extrema

En esta oportunidad, el proceso de desarrollo fue Programación extrema (XP), la cual se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ágiles, que son aquellas en la que se da prioridad a las tareas que dan resultados directos y que reducen la burocracia.

La aplicación de las tareas XP busca obtener el éxito con comunicación constante con los clientes, para ello se debe tomar en cuenta las fases fundamentales que son:

- **Planificación:** En esta fase se definen entre el conjunto de desarrolladores y usuarios del sistema a través de relatos las respectivas historias de usuarios, se establece el plan de entregas parciales (release planning) según un cronograma las cuales deben ser frecuentes y se divide el proyecto en fases de iteración y la misma debe permitir estimar el tiempo de desarrollo del mismo.

El desarrollo se divide en iteraciones, cada iteración comienza con un plan para que se elijan las historias de usuarios a desarrollar, cabe destacar que las iteraciones durante el desarrollo de las mismas pueden surgir cambios, dichos cambios son adaptados o mejorados rápidamente sin que afecte la planificación del release.

En esta etapa se estableció una bitácora de desarrollo para planificar, en base a días o semanas, esta bitácora esta compuesta por fecha de inicio y una fecha de fin, tanto por modulo como por tarea, la precedencia de la tarea, el porcentaje de culminación, días estimados y días estimados y días realizados. El formato tomado en consideración se observa en el cuadro 2.1.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Das estimados	Das realizados
-	-	-	-	-	-	-

Cuadro 2.1: Formato seleccionado para la bitácora de desarrollo

El formato de presentacin de las historias de usuario se observa en el cuadro 2.2.

Número: -	Nombre: -
Prioridad: -	Estimación: -
Decripción: -	

Cuadro 2.2: Formato seleccionado para manejar Historias de Usuario

Se crearon una lista de historias de usuario durante el desarrollo previo, de la automatización de los procesos por parte de un grupo de tesista que también estaban inmersos en la solución del problema.

- **Diseño:** Una vez obtenidos todos los requerimientos de la aplicación, se da paso a modelar una implementación parcial no definitiva de diagramas, modelos donde permita tener una visión clara de lo que el cliente espera, esta modelación debe ser de fácil entendimiento para la documentación del sistema, en esta etapa de desarrollo de la metologis XP las revisiones de hacen de forma continua según el ritmo que se vayan agregando nuevas funcionalidades al sistema.

- **Codificación:** La clave del proceso de desarrollo de XP es la comunicación. La gran mayoría de los problemas en los proyectos de desarrollo son provocados por

falta de comunicación en el equipo, así que se pone un gran énfasis en facilitar que la información fluya lo mas eficientemente posible, para ello se establecieron las siguientes puntos:

- Como primer punto a destacar fue, desarrollar de manera modular el sistema, para que cada programador pueda ubicarse y entender el código programado y así ser mas rápido a la hora de hacer algún cambio necesario
- Con respecto a la nomenclatura y variables, se optó por desarrollo en español. Para variables con dos o más palabras deben estar separadas por `_`. Si la palabra tiene ñ se sustituye por n, si tiene acentos se omite, siempre en minúscula
- Con respecto a los métodos, se definen siempre en minúscula, si son varios nombres, estarán separados por separados por `_`, si el método retorna un lógico (boolean) debe terminar en `?`, si es un método que modifica el objeto debe terminar en `!` y si este método muestra una excepción al error, éste debe terminar en `!`
- Para la base de datos, las tablas tipo o tablas para clasificar algo, los nombres deben estar precedidos por el prefijo `tipo_`, los id son preferiblemente `char(10)` y no son auto-numéricos, las tablas tipo tienen un campo descripción `varchar(255)`, los campos String son `varchar(255)` a menos que sean largos los cuales son `longtext` y se aceptaron claves compuestas, además, se declaran siempre en minúscula y en singular. Para las relaciones se mantiene las convenciones de Rails, `nombre_tabla_nombre_campo`

Para la etapa de codificación de cada iteración se asignaban diferentes tareas por historias de usuario a cada programador, al finalizar la tarea se registraban en el manejador de versiones.

- **Pruebas:** En esta etapa de modelación ágil, las pruebas de aceptación se realizan frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios elegidas para la iteración, y son “pruebas de caja negra”, en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario ha completado y se pasa a una nueva tarea

No. Caso Prueba	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
-	-	-	-	-	-

Cuadro 2.3: Formato de registro de pruebas del lado del cliente

2.2. Análisis general del sistema

El análisis comienza desde el levantamiento de la información la cual está plasmada en el primer capítulo del presente documento, se considera los requerimientos necesarios para el desarrollo de la aplicación y a partir de estos se crean las historias de usuarios que llevarán el seguimiento del sistema. Dichas historias se obtienen desde la primera reunión con el cliente y se presentan a continuación:

2.2.1. Historias de Usuario

Número: 1	Nombre: Desarrollar método guardar en el modelo Caso.
Prioridad: Alta	Estimación: 10 días
Descripción: Guardar la información referente a un caso	

Número: 2	Nombre: Desarrollar interfaz de usuario para modelo Caso.
Prioridad: Alta	Estimación: 3 días
Descripción: Desarrollar interfaz de usuario, donde se solicite la información necesaria para crea un caso	

Número: 3	Nombre: Desarrollar validaciones para formulario del modelo.
Prioridad: Alta	Estimación: 2 días
Descripción: Validar que la información ingresada sea valida y proceder con la creación de un caso	

Número: 4	Nombre: Desarrollar método que permita asociar un documento a un caso.
Prioridad: Alta	Estimación: 3 días
Descripción: Asociar a un caso los documentos que formen parte de este	

Número: 5	Nombre: Desarrollar modelo que permita asociar un comentario a un caso.
Prioridad: Alta	Estimación: 2 días
Descripción: Agregar comentarios al caso, identificando fecha y usuario que lo realice.	

Número: 6	Nombre: Desarrollar componente Javascript para listar los casos.
Prioridad: Alta	Estimación: 3 días
Descripción: Ofrecer el listado de casos filtrado por los estados y según el perfil del usuario.	

Número: 7	Nombre: Desarrollar vista para un Caso.
Prioridad: Alta	Estimación: 2 días
Descripción: Desarrollar una vista donde se muestre la información de un caso seleccionado.	
Número: 8	Nombre: Desarrollar componente ajax para visualizar lista de casos por ciertas categorías, como por ejemplo listar por fecha de creación.
Prioridad: Alta	Estimación: 4 días
Descripción: Dar opción para listar caso por alguna categoría seleccionada por el usuario.	
Número: 9	Nombre: Desarrollar componente ajax para visualizar un caso en particular.
Prioridad: Alta	Estimación: 2 días
Descripción: Permitir la opción de una lista de casos poder visualizar el detalle de uno en particular.	
Número: 10	Nombre: Desarrollar componente ajax para visualizar los comentarios asociados a un caso.
Prioridad: Alta	Estimación: 2 días
Descripción: Permitir visualizar los comentarios asociados a un caso con el detalle de cuando y quien lo creo	
Número: 11	Nombre: Desarrollar componente ajax para visualizar los documentos adjuntos.
Prioridad: Alta	Estimación: 4 días
Descripción: Permitir visualizar la documentación asociada a un caso, así mismo de permitir poder cambiarla.	
Número: 12	Nombre: Desarrollar con javascript la paginación de la lista de casos.
Prioridad: Alta	Estimación: 3 días
Descripción: Desarrollar un componente javascript que permita paginar los casos listados.	
Número: 13	Nombre: Desarrollar el método asociar_caso que permita crear un caso relacionado a la solicitud.
Prioridad: Alta	Estimación: 6 días
Descripción: Crear el caso referente a una solicitud, estableciendo la relaciones con el modelo Caso)	

Número: 14	Nombre: Aplicar el método agregar_destinatario de la solicitud.
Prioridad: Alta	Estimación: 3 días
Descripción: Determinar a quienes debe llegar el caso relacionado con la solicitud, y agregarlo como destinatario.	
Número: 15	Nombre: Aplicar el método agregar_seguimiento_caso de la solicitud.
Prioridad: Alta	Estimación: 3 días
Descripción: Permitir llevar el historial del caso de una solicitud.	
Número: 16	Nombre: Desarrollar el método actualizar_status de la solicitud.
Prioridad: Alta	Estimación: 6 días
Descripción: Permitir al usuario autorizado cambiar el estado del caso de la solicitud.	
Número: 17	Nombre: Modificar método aprobar y no_aprobar de la solicitud.
Prioridad: Alta	Estimación: 5 días
Descripción: Modificar métodos para que establezcan el estado final del caso referente a la solicitud.	
Número: 18	Nombre: Actualizar la lista de casos para que los usuarios sepan que se actualizó la solicitud.
Prioridad: Baja	Estimación: 2 días
Descripción: Reflejar en la vista de los usuarios involucrados en la solicitud que el caso ha cambiado de estado	
Número: 19	Nombre: Desarrollar el método asociar_caso que permita crear un caso asociado a Oferta Materia.
Prioridad: Alta	Estimación: 6 días
Descripción: Crear el caso referente a una oferta materia, establecer relación con el modelo Oferta Materia	
Número: 20	Nombre: Cuando otro profesor oferte una materia ya ofertada agregarlo como destinatario al caso asociado
Prioridad: Media	Estimación: 7 días
Descripción: Asociar un profesor a un caso ya existente de una materia ya ofertada y colocar como no leído a los otros profesores que hayan ofertado previamente	
Número: 21	Nombre: Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso
Prioridad: Alta	Estimación: 3 días
Descripción: Determinar a quienes debe llegar el caso relacionado con la oferta materia, y agregarlo como destinatario.	

Número: 22	Nombre: Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso
Prioridad: Alta	Estimación: 3 días
Descripción: Permitir llevar el historial del caso de una solicitud.	
Número: 23	Nombre: Modificar los métodos que actualizan el estado de una oferta materia para que actualice el estado del caso.
Prioridad: Alta	Estimación: 5 días
Descripción: en el modelo cambiar los métodos relacionados a la actualización de estado de una oferta materia, para que el caso pueda ser cambiado de estado	
Número: 24	Nombre: Actualizar la lista de casos para que los usuarios sepan que se actualizó el plan docente.
Prioridad: Media	Estimación: 2 días
Descripción: Colocar el caso que cambió de estado como no leído para que el usuario vuelva a leerlo.	
Número: 25	Nombre: Desarrollar el método asociar_caso que permita crear un caso asociado a la admisión que a su vez tiene una relación con el aspirante.
Prioridad: Alta	Estimación: 6 días
Descripción: Crear el caso referente a una admisión, estableciendo las relaciones necesarias con los modelos involucrados (Admisión, Aspirante).	
Número: 26	Nombre: Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso
Prioridad: Alta	Estimación: 3 días
Descripción: Determinar a quienes debe llegar el caso relacionado con la admisión, y agregarlo como destinatario.	
Número: 27	Nombre: Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso
Prioridad: Alta	Estimación: 3 días
Descripción: Permitir llevar el historial del caso de una admisión.	
Número: 28	Nombre: Desarrollar método cambiar_estado_caso que permita modificar el estado del caso referente a la selección, según el paso y el perfil.
Prioridad: Media	Estimación: 3 días
Descripción: Crear un método parametrizado que realice los cambios de estado del caso en todo el proceso de selección.	

Número: 29	Nombre: Modificación del método tiene_documento para cambiar el estado a “por evaluar”.
Prioridad: Baja	Estimación: 1 días
Descripción: Editar el método tiene_documento que cambia el estado del aspirante a “por evaluar” para que también cambie el estado del caso asociado a “por evaluar”.	
Número: 30	Nombre: Modificación del método seleccion para cambiar el estado a “aceptado” o “no aceptado”.
Prioridad: Baja	Estimación: 1 días
Descripción: Editar el método seleccion para que una vez evaluado el aspirante por coordinador de área se cambie el estado del caso a “aceptado” o “no aceptado”.	
Número: 31	Nombre: Modificación del método formalizar_admision para formalizar después de ser aceptado.
Prioridad: Baja	Estimación: 1 días
Descripción: Editar el método formalizar_admision para que después de ser “aceptado” un aspirante la secretaria de coordinación pueda formalizar la selección, cambiado el estado del caso a “formalizada”	
Número: 32	Nombre: Desarrollar método que permita guardar una evaluación asociada a una materia y sus diferentes tipos de evaluación.
Prioridad: Alta	Estimación: 5 días
Descripción: Hacer un método que cree una evaluación con los parámetros que identifique la materia a la que va dirigida dicha evaluación y los diferentes tipo de evaluaciones que tendrá.	
Número: 33	Nombre: Crear el método asociar_caso que cree un caso asociado a una evaluación.
Prioridad: Alta	Estimación: 5 días
Descripción: Crear el caso referente a una evaluación, estableciendo las relaciones necesarias con los modelos involucrados (Evaluacion, EvaluacionOfertaAcademica).	
Número: 34	Nombre: Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso
Prioridad: Alta	Estimación: 2 días
Descripción: Determinar a quienes debe llegar el caso relacionado con la evaluación, y agregarlos como destinatario.	
Número: 35	Nombre: Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso
Prioridad: Alta	Estimación: 2 días
Descripción: Permitir llevar el historial del caso de una evaluación.	

Número: 36	Nombre: Modificar método crear_actualizar para que se cree un nuevo caso al cambiar las evaluaciones de una materia.
Prioridad: Baja	Estimación: 2 días
Descripción: Cuando se modifique la evaluación de una materia, crear un nuevo caso para que los usuarios involucrados se enteren.	
Número: 37	Nombre: Crear método guardar en el modelo Calificación. Solo cuando es calificación definitiva.
Prioridad: Alta	Estimación: 5 días
Descripción: Hacer un método que cree una calificación con los parámetros que identifique la materia a la que va dirigida.	
Número: 38	Nombre: Desarrollar el método asociar_caso que permita crear un caso relacionado a una calificación.
Prioridad: Alta	Estimación: 5 días
Descripción: Crear el caso referente a una calificación, estableciendo las relaciones necesarias con los modelos involucrados (Calificacion, EvaluacionHistorialAcademico).	
Número: 39	Nombre: Aplicar el método agregar_destinatarios que determine a quienes debe llegar ese caso.
Prioridad: Alta	Estimación: 2 días
Descripción: Determinar a quienes debe llegar el caso relacionado con la calificación, y agregarlos como destinatario.	
Número: 40	Nombre: Desarrollar método agregar_seguimiento_caso que permita llevar el historial del caso.
Prioridad: Alta	Estimación: 2 días
Descripción: Permitir llevar el historial del caso de una calificación.	
Número: 41	Nombre: Crear la metáfora para identificar que un caso tiene documentos asociados.
Prioridad: Baja	Estimación: 2 días
Descripción: Identificar a cada caso de la bandeja con una metáfora cuando este tenga documentos adjuntos.	
Número: 42	Nombre: Crear una alerta que notifique cada vez que llegue un nuevo caso o cuando un caso existente sea actualizado.
Prioridad: Alta	Estimación: 4 días
Descripción: Revisar si hay casos nuevos o actualizados para el usuario actual y notificarlo mediante alertas.	

Número: 43	Nombre: Crear nuevas categorías para que muestre el caso según el módulo que represente, tales como Solicitud, Oferta Materia, Admisión, Inscripción, Evaluación y Calificación.
Prioridad: Media	Estimación: 2 días
Descripción: Organizar los casos según el módulo desde donde fue generado.	
Número: 44	Nombre: Crear check-box para seleccionar los casos que se quieran archivar.
Prioridad: Baja	Estimación: 2 días
Descripción: Permitir archivar casos seleccionados por el usuario, y solo visualizar los casos guardados en la opción de ver 'Todos'.	
Número: 45	Nombre: Finalizado el proceso de inscripción notificar al personal administrativo correspondiente, de los estudiantes que no se hayan inscrito.
Prioridad: Alta	Estimación: 4 días
Descripción: Obtener aquellos estudiantes que aún cumpliendo con las restricciones para inscribirse en el periodo actual, no lo hicieron. Y notificar a los usuarios pertinentes de estos estudiantes.	
Número: 46	Nombre: Crear un módulo activación que permita a la secretaria de Postgrado activar/desactivar los módulos de inscripción y calificación.
Prioridad: Alta	Estimación: 2 días
Descripción: Crear para la secretaria de Postgrado una interfaz para que pueda activar y desactivar los módulos inscripción y calificación.	

2.3. Metáfora del sistema

El objetivo es mejorar la comunicación entre los entes involucrados en la Coordinación Académica de Postgrado mediante la asignación a través de notificaciones de tareas que forman parte de un conjunto de trabajo permitiendo así la colaboración entre personas que intervienen en procesos, recuperando de esta manera información necesaria para completar tareas. La metáfora del sistema planteado se basa en su totalidad, en la comunicación consecuencia de un evento el cual es activado tras la puesta en marcha y posterior ejecución de un evento, que es activado cuando es iniciado alguno de los procesos que se encuentran involucrados en la gestión académica de los postgrados que son dictados en la Facultad de Ciencias de la Universidad Central de Venezuela y que interactúan entre sí cada vez que los eventos son iniciados y cambian de estados mediante los entes que intervienen como estudiantes, comisión académica, coordinaciones y secretarías.

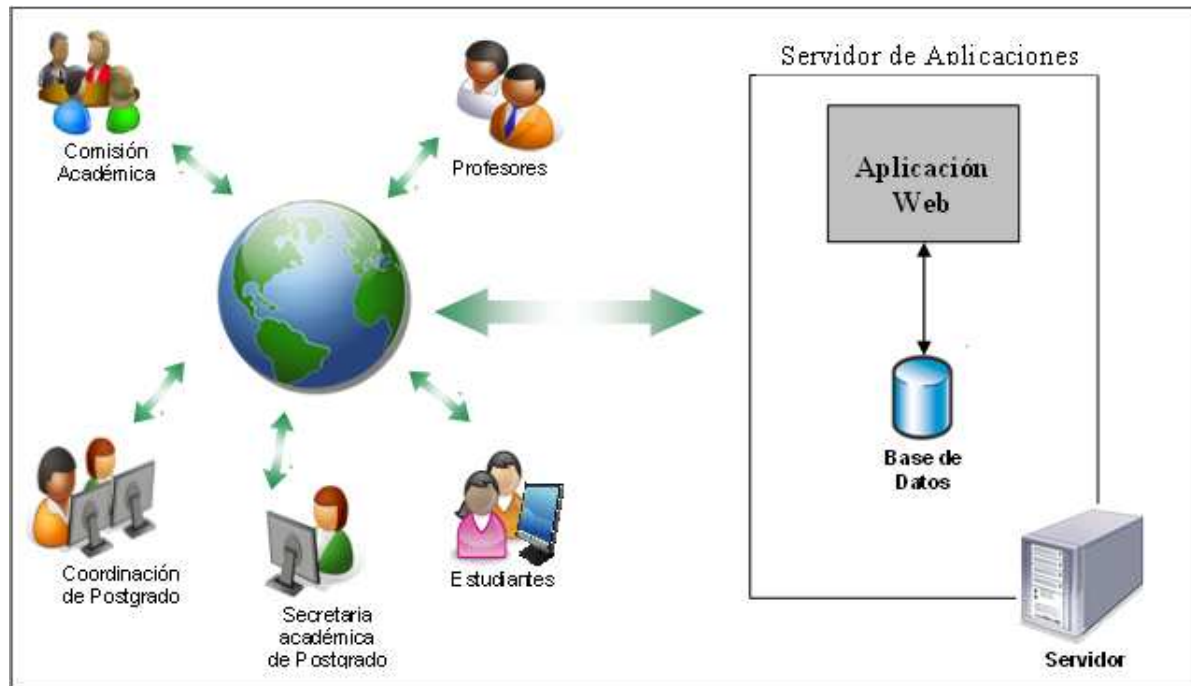


Figura 2.1: Metáfora del sistema

2.4. Especificaciones técnicas

El desarrollo del sistema se implantará utilizando las siguientes herramientas tecnológicas:

- Servidor Mongrel de Rails (V-1.1.5)
- Procesamiento de Eventos: JavaScript, Prototype, JQuery y Scriptaculous
- Intercambio de información cliente-servidor: AJAX
- Gestión de usuarios, solicitudes y peticiones: SMD MySQL (V-5.0)

Después de ser planteado los requerimientos básicos, de haber sido diseñado una estructura general del sistema y de ser seleccionadas las historias de usuario, se programa un conjunto de 9 iteraciones que abarcaran dichas historias y las cuales serán descritas en el que capítulo que siguiente.

2.5. Resumen del capítulo

En este capítulo, se presentó una adaptación del proceso de desarrollo de software aplicado a la presente investigación, basado en el enfoque o metodología ágil Programación Extrema (XP) para la construcción de la aplicación Web. Para tales efectos, se describe el contexto de desarrollo, descripción de cada una de las fases del método de desarrollo utilizado para la realización del sistema de workflow de los procesos académicos, definiéndose este como todas las actividades relacionadas con los procesos de admisión y selección de aspirantes, inscripción de materias, solicitudes estudiantiles y planificación docente, procesos dirigidos en la Coordinación de Postgrado en conjunto a los trece postgrados que lo componen.

Capítulo 3

Marco Aplicativo

3.1. Contexto del desarrollo

El presente proyecto ha sido elaborado en el contexto del Trabajo Especial de Grado presentado ante la ilustre Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación bajo la dirección del profesor Andrés Sanoja, para optar al título de Licenciado en Computación, por parte de los bachilleres Glendy Patricia Sulbarán Goyo y Néstor David Pedrozo Pedrozo.

3.2. Plan de Iteración

Debido a la metodología de desarrollo utilizada para el presente Trabajo Especial de Grado, fue necesario la definición y posterior creación de un conjunto de iteraciones presentadas “releases”, las cuales a cada una de éstas en su momento de creación se les estableció fecha de inicio y fecha de culminación por parte del equipo de trabajo, debidamente asesorados por parte de nuestro tutor, quien fue guía para la definición de las llamadas historias de usuario que en su función formarían parte importante de la implementación dentro de un período de tiempo estipulado.

3.3. Iteración 0

En esta iteración se detalla el planteamiento del problema, el cual se basa en la no existencia de representación de procedimientos de trabajos, trámites entre procesos mediante

el flujo de documentos y/o notificaciones de forma electrónica u automática.

- **Planteamiento del problema:** Para el momento de la realización de este Trabajo Especial de Grado se cuenta con un prototipo que próximamente estará en funcionamiento y que permitirá llevar a cabo la ejecución de forma automatizada de todos los procesos académicos de la coordinación de postgrado y sus respectivos postgrados los cuales son dictados en la Facultad de Ciencias de la Universidad Central de Venezuela, agilizando de esta manera la ejecución de los procesos involucrados, sin embargo dicho sistema aun no cuenta con una herramienta que permita manejar el flujo de trabajo que involucran a los procesos y que maneje como actividad principal la notificación de acontecimientos como transferencia de documentos e información concerniente a cambios de estados de procesos que interactúan a medida que los actores involucrados cambian los estatus de los mismos con la finalidad de medir y hacer seguimiento sobre el trabajo que se realiza en los distintos postgrado y sobre el ente que los coordina cuyos actores principales son los estudiantes, profesores y coordinadores.

- **Propuesta de solución:** La solución propuesta fue realizar un sistema de workflow denominados por el grupo y el líder del proyecto, como sistema de casos, que permita apoyar procesos estructurados orientados a la administración de procesos a través de seguimientos constantes según la lógica de negocio que se establece entre los procesos académicos, cuya información fue obtenida durante la recolección de datos para el desarrollo de este Trabajo Especial de Grado consiguiendo de esta manera aumento de la productividad, mejorar la calidad de los trámites dado al tiempo de obtención de la información a través de las notificaciones oportunas ,mejorando el servicio entre la coordinación, los postgrados y los estudiantes, reduciendo costos y manteniendo un entorno cambiante.

- **Proceso de admisión:** El proceso de admisión inicia con la cancelación del material de pre-inscripción por parte del aspirante. Con el recibo de pago que la caja entrega, el aspirante se registrará en el sistema prototipo, el cual le solicitará, datos personales, información referente a la carrera por la cual aspira. Dicho prototipo generara un comprobante de preinscripción y la opción de descargar el material necesario para formalizar su solicitud.

Una vez registrado el aspirante en el sistema prototipo es el momento de que la aplicación propuesta inicie el workflow con la creación de un caso, dicho caso será notificado a la Coordinación de Postgrado y al Comité Académico mediante una notificación la cual tendrá como primer estado del flujo PENDIENTE, dicha notificación solamente llegará a la bandeja de entrada de casos a la dependencia del postgrado correspondiente y al Comité Académico del Postgrado que el aspirante desee iniciar estudios.

El aspirante al consignar los requisitos ante la Coordinación de Postgrado, el personal administrativo de dicha instancia procederá a digitalizar los documentos y almacenarlos u asociarlo al aspirante mediante el sistema prototipo. Al realizar esta actividad el caso cambia de estado, el mismo pasa del estado PENDIENTE estado POR EVALUAR.

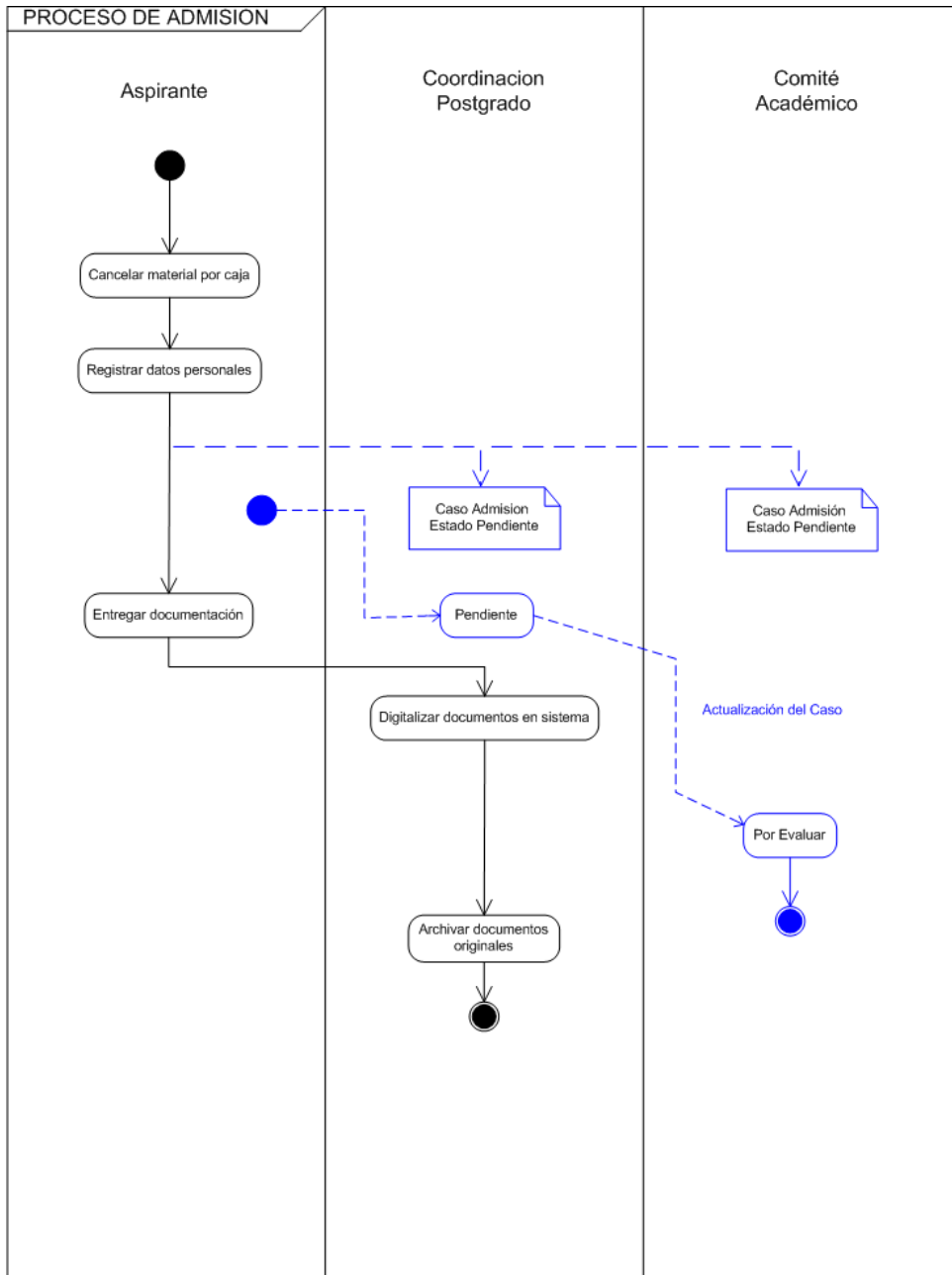


Figura 3.1: Diagrama de actividades del flujo de trabajo para el proceso de Admisión

- Proceso de selección:** Una vez que el caso es remitido a la Coordinación de Postgrado y se anexan los documentos necesarios, el caso cambia de estado PENDIENTE al estado POR EVALUAR, entonces es cuando el Comité Académico del Postgrado correspondiente procede a evaluar a cada aspirante. Para tal efecto el Comité Académico correspondiente ingresa en el sistema prototipo y selecciona del listado el aspirante a evaluar. Al realizar la evaluación el comité podrá emitir una observación y la correspondiente aceptación o no de su admisión. Si el aspirante es aceptado el caso será actualizado y pasará de estado POR EVALUAR a estado ACEPTADA, en caso contrario el caso pasa a estado NO ACEPTADA.

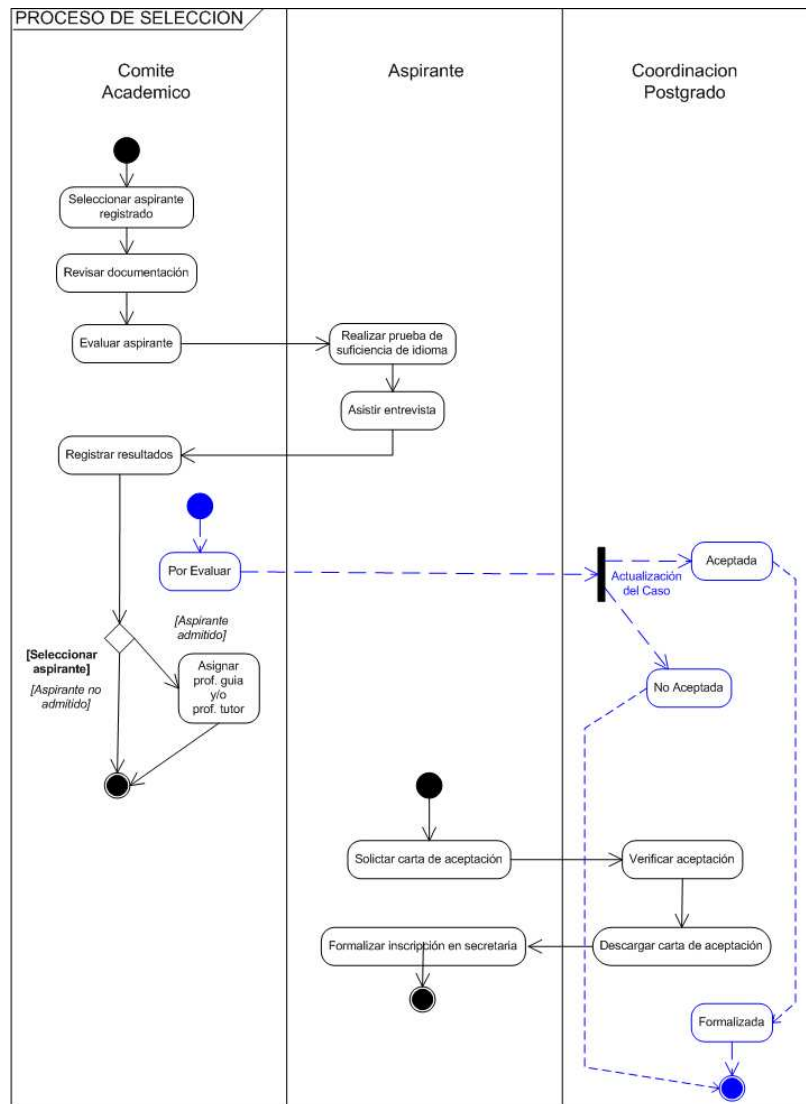


Figura 3.2: Diagrama de actividades del flujo de trabajo para el proceso de Selección

- Proceso de retiro de materia:** Al estudiante ingresar al sistema prototipo, el mismo podrá visualizar el conjunto de materias inscritas. Al momento que el estudiante en cuestión realice la solicitud de retiro(s) de materia(s) por el sistema prototipo, en este punto se enviara una notificación a través del sistema de casos, cuyo primer estado será el estado PENDIENTE, dicha notificación permitirá dar a conocer al comité académico del postgrado correspondiente que el estudiante ha decidido retirar una o mas materias de las inscritas por él al momento de realizar su inscripción. Una vez que el comité académico revisa el caso creado, el mismo podrá pasar al estado ACEPTADO o al estado NO ACEPTADO. En cualquiera sea el próximo estado, el estudiante será notificado a través del correo electrónico y en su bandeja de casos se podrá visualizar el estado de su solicitud.

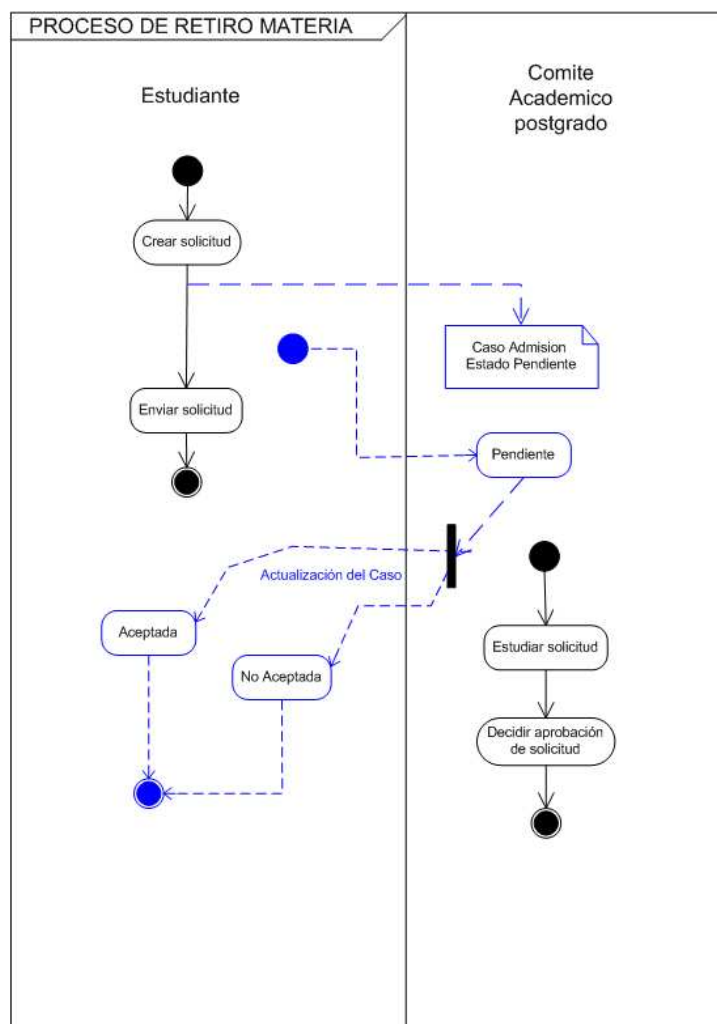


Figura 3.3: Diagrama de actividades del flujo de trabajo para el proceso de Retiro de Materia

- Proceso de desincorporación:** De manera análoga a la solicitud de retiro de materia, el estudiante puede hacer una solicitud de retiro total del semestre, en cuyo caso el estudiante debe dar una breve descripción del motivo por el cual esta haciendo dicha solicitud de tal manera que el comité del postgrado al cual el estudiante realiza estudios pueda evaluar y estudiar la solicitud y así poder tomar la decisión de aceptar o no el retiro total del semestre, él estudiante será notificado a través del correo electrónico y en su bandeja de casos se podrá visualizar el estado de su solicitud.

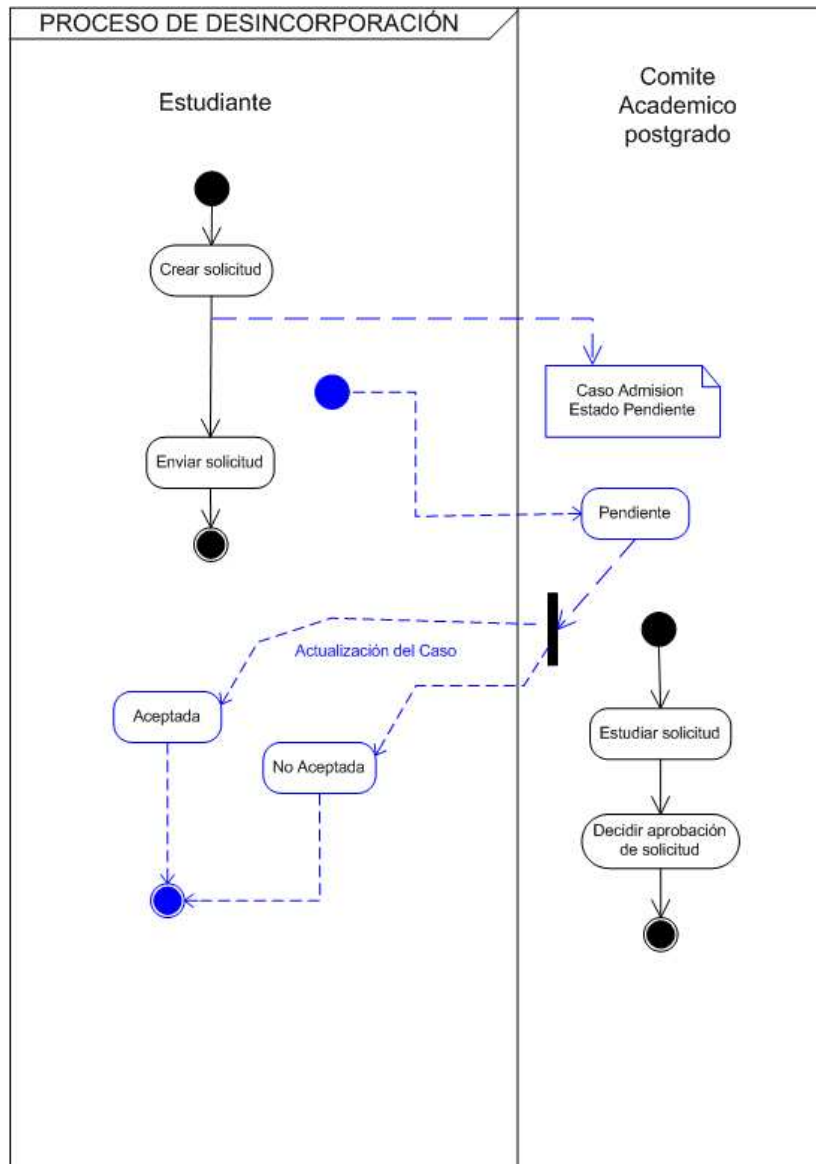


Figura 3.4: Diagrama de actividades del flujo de trabajo para el proceso de Desincorporación

- **Proceso de calificación:** El proceso de calificación inicia cuando es notificado al docente que esta habilitado el modulo para realizar las respectivas calificaciones, el modo de calificación puede ser de 2 maneras, continua y definitiva, siendo esta última la que se le lleva el sistema de casos propuesta en este trabajo especial de grado. El sistema de caso para el proceso de calificación inicia cuando el docente califica de forma definitiva a cada estudiante de la materia que acaba de dictar en el período actual, en cuyo caso tras a la evaluación será notificado a través del modulo de caso del estudiante y también a través del correo electrónico. Cabe destacar que esta notificación solamente manejará un sólo estado dado que en este caso se trata de una notificación que no amerita el cambio de estado.

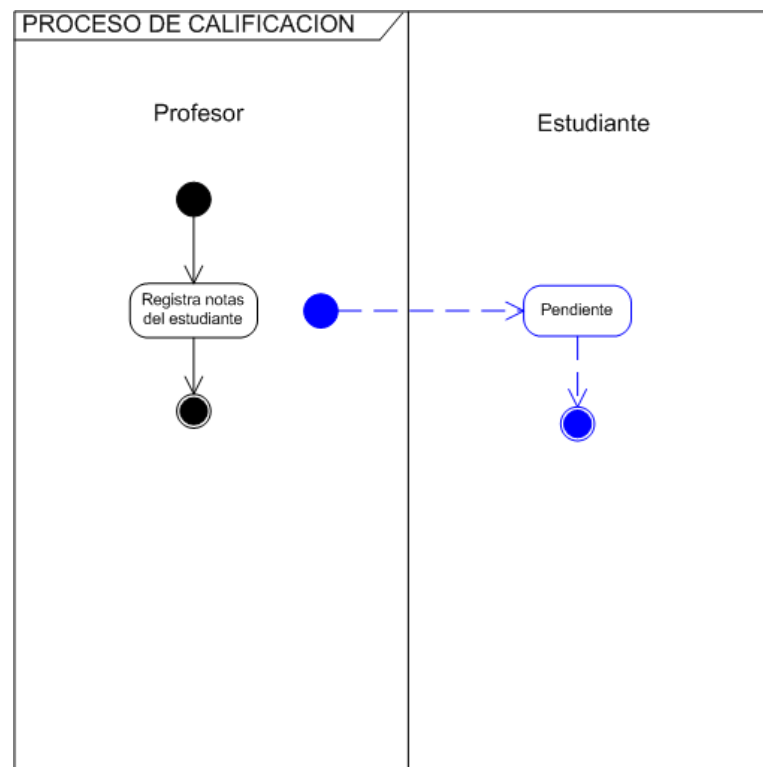


Figura 3.5: Diagrama de actividades del flujo de trabajo para el proceso de Calificación

3.4. Iteración 1

- Planificación

Iteración 1	
Descripción	Desarrollo del módulo Caso, módulo para el manejo de la bandeja de casos relacionados con el postgrado
Historias de Usuario a Desarrollar	1- Desarrollar método guardar en el modelo Caso 2- Desarrollar interfaz de usuario para modelo Caso 3- Desarrollar validaciones para formulario del modelo 4- Desarrollar método que permita asociar un documento a un caso 5- Desarrollar modelo que permita asociar un comentario a un caso 6- Desarrollar componente Javascript para listar los casos 7- Desarrollar vista para un Caso 8- Desarrollar componente ajax para visualizar lista de casos por ciertas categorías, como por ejemplo listar por fecha de creación 9- Desarrollar componente ajax para visualizar un caso en particular 10- Desarrollar componente ajax para visualizar los comentarios asociados a un caso 11- Desarrollar componente ajax para visualizar los documentos adjuntos 12- Desarrollar con javascript la paginación de la lista de casos
Tiempo Estimado	40 días
Fechas Inicio/Fin	15-04-2009 / 25-05-2009

- Tareas por Historia de Usuario

HU 1.- Desarrollar método guardar en el modelo Caso

- Crear componente para guardar información del caso.
- Establecer las relaciones con los modelos asociados (Organización, PeriodoAcademico, TipoStatusCaso) al momento de guardar.
- Establecer el tipo de caso y relacionarlo con la Clase correspondiente.
- Establecer el estado inicial según la clase a la que este relacionado el caso.

HU 2.- Desarrollar interfaz de usuario para modelo Caso

- Montaje de las plantillas (CSS + HTML).
- Diseño base de la página.
- Agregar todo el contenido básico para la generación del formulario.

HU 3.- Desarrollar validaciones para formulario del modelo

- Definir las validaciones para los campos de basicos del caso, (asunto, descripción, etc).

- Validar las relaciones como el usuario que crea el caso, la categoría y el estado.
- Crear y mostrar los mensajes de erros correspondientes.

HU 4.- Desarrollar método que permita asociar un documento a un caso

- Agregar contenido para que el aspirante pueda asociar los documentos.
- Cargar documentación para un caso.
- Permitir modificar la documentación del aspirante con su respectiva asociación.
- Componente que permita asociar los archivos a un documento.

HU 5.- Desarrollar modelo que permita asociar un comentario a un caso

- Hacer formulario para que el usuario pueda ingresar un comentario.
- Hacer las validaciones en el modelo formulario.
- Obtener fecha de registro del comentario y asociarlo.

HU 6.- Desarrollar componente Javascript para listar los casos

- Generar consulta de casos por estado
- Establecer vista de listado de caso
- Mostrar lista de casos asociados al perfil del usuario logeado.
- Organizar los casos cronologicamente.
- Identificar cada item de la lista de casos, con el numero del caso y el asunto.

HU 7.- Desarrollar vista para un Caso

- Diseño de la arquitectura para previa visualización.
- Montaje de las plantillas (CSS + HTML).
- Hacer diseño base de la página.

HU 8.- Desarrollar componente ajax para visualizar lista de casos por ciertas categorías, como por ejemplo listar por fecha de creación

- Generar consultas por categoría.
- Mostrar al usuario las opciones por la cual puede catalogar.
- Integrar las vistas con las consultas.

HU 9.- Desarrollar componente ajax para visualizar un caso en particular

- Consultar información asociada al caso.
- Mostrar la información asociada al caso.
- Hacer el enlace para visualizar un caso seleccionado de la lista de casos.

HU 10.- Desarrollar componente ajax para visualizar los comentarios asociados a un caso

- Componente que permita mostrar los comentarios con su detalle
- Paginar los comentarios.

HU 11.- Desarrollar componente ajax para visualizar los documentos adjuntos

- Al momento de subir los documentos para un caso, éstos pueden observarse.
- Paginar los documentos.

HU 12.- Desarrollar con javascript la paginación de la lista de casos

- Al momento de mostrar los casos según alguna categoría, paginarlos.

- Bitácora de desarrollo

La bitácora de desarrollo se hizo tomando en cuenta el conjunto de historias de usuario, generalizando así lo realizado desde el inicio hasta la culminación de la iteración 1.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	15/04/2009	23/04/2009	8	8
2	Montaje de las plantillas(CSS + HTML)	-	23/04/2009	30/04/2009	7	7
3	Análisis y diseño de la implementación	-	30/04/2009	12/05/2009	10	12
4	Desarrollo de la interfaz	-	12/05/2009	19/05/2009	5	7
5	Realización de pruebas	-	19/05/2009	25/05/2009	5	6

- **Diseño** En la figura 3.6 se visualizan las clases de implementación Ruby que dan solución a las historias de usuario mencionadas en la planificación de la iteración

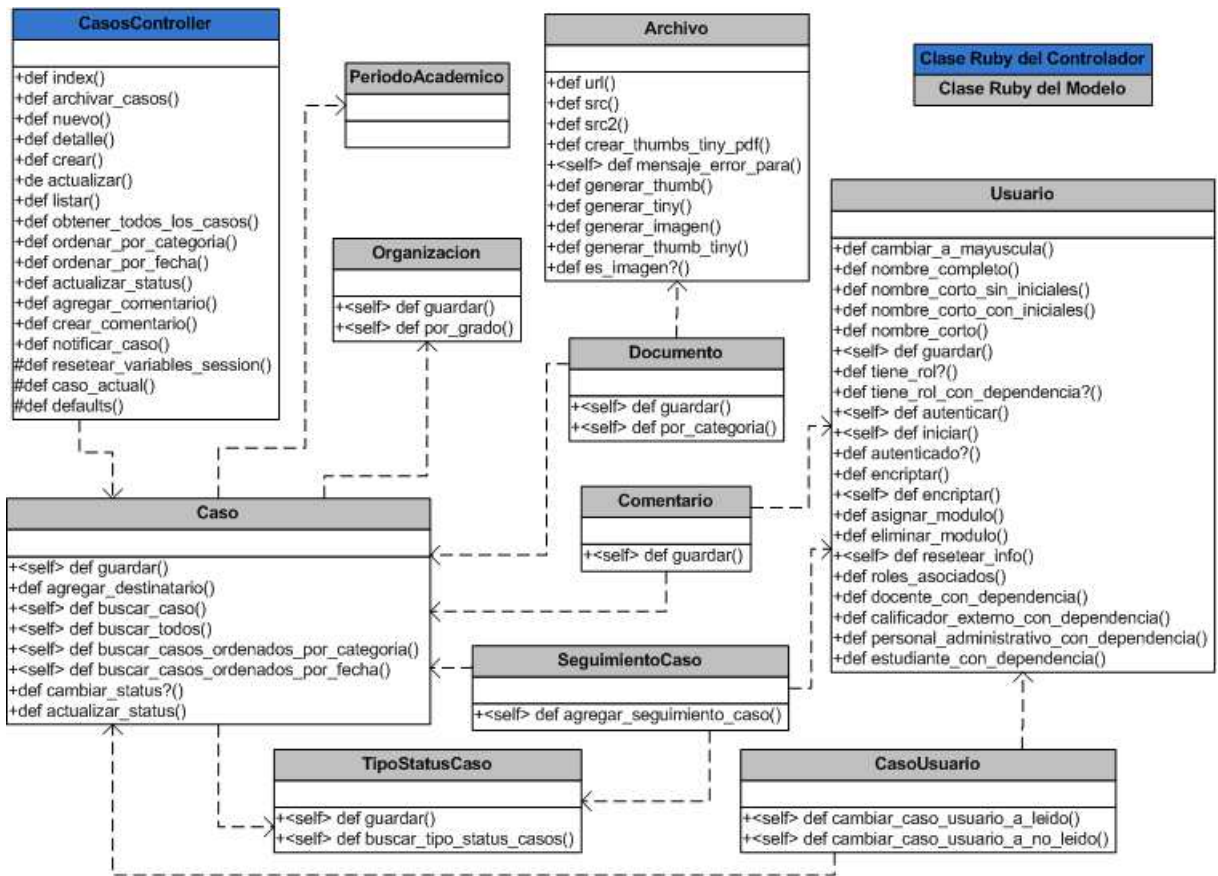


Figura 3.6: Clases de implementación Iteración 1. Módulo Caso.

- **Codificación** En la figura a continuación se presenta la implementación del método que guarda los casos, dicho método se encuentra en la clase Caso y el mismo se encarga de crear un caso, adicionalmente en la misma clase se encuentra un método que se encarga de asociar el caso a uno o más destinatarios dependiendo el proceso que se este llevando a cabo.

```
def self.guardar(attr, opciones = {})

  attr[:tipo_status_caso_id] ||= TipoStatusCaso.find(:first,
    :conditions => ["aplica_a = ? and nombre = ?", opciones[:categoria], "PENDIENTE"])

  caso = new(attr)

  exito = caso.valid?

  if exito
    caso.save
  end

  return {:exito => exito, :retorno => caso}
end

def agregar_destinatario(info=[])
  info[:arreglo_destinatario] = [info[:arreglo_destinatario]]
  unless Array===info[:arreglo_destinatario]
    info[:arreglo_destinatario].each do |destinatario_cedula|
      usuario = Usuario.find_by_cedula(destinatario_cedula)
      usuario.casos << self
    end
  end
end
```

Figura 3.7: Métodos crear y agregar_destinatarios de la clase Caso

En la siguiente figura se observa el método que permite cambiar de estado el caso permitiendo además hacerle seguimiento al mismo guardando su historial en el modelo SeguimientoCaso para saber las distintas etapas por el cual ha pasado el caso.

```
def actualizar_status(opciones={})
  update_attribute(:tipo_status_caso_id, opciones[:tipo_status_caso][:id])
  SeguimientoCaso.agregar_seguimiento_caso(:caso_id => self[:id],
    :tipo_status_caso_id => opciones[:tipo_status_caso][:id],
    :usuario_cedula => opciones[:usuario][:cedula])
end
```

Figura 3.8: Método actualizar_status

- Pruebas

Las pruebas fueron realizadas para esta clase en todo momento del desarrollo de la aplicación, cada vez que el cliente verificaba el desarrollo del proyecto surgían observaciones, las observaciones formaron parte importante de la guía que se tuvo con las mejoras para la aplicación.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Caso	Un usuario tras la puesta en marcha de algún proceso que requiera hacerle seguimiento genera un caso.	Dependiendo el proceso que se lleve a cabo se cree el caso con un estado inicial PENDIENTE	Se creo el caso correspondiente y con el estado inicial como era el esperado	
2	Caso	Una vez creado un caso se intenta modificar el mismo cambiandolo de estado	Que el caso cambie de estado y el resultado sea verificado en la interfaz gráfica de la aplicación.	El caso cambio de estado como se esperaba y se creo el historial para el caso.	

3.5. Iteración 2

- Planificación

Iteración 2	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Solicitudes
Historias de Usuario a Desarrollar	<p>13- Desarrollar el método asociar_caso que permita crear un caso relacionado a la solicitud.</p> <p>14- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso.</p> <p>15- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso.</p> <p>16- Desarrollar el método actualizar_status que permita al usuario autorizado cambiar el estado de la solicitud.</p> <p>17- Modificar método aprobar y no_aprobar para que establezcan el estado final del caso.</p> <p>18- Actualizar la lista de casos para que los usuarios sepan que se actualizó la solicitud.</p>
Tiempo Estimado	25 días
Fechas Inicio/Fin	26-05-2009 / 20-06-2009

- Tareas por Historia de Usuario

HU 13.- Desarrollar el método asociar_caso que permita crear un caso relacionado a la solicitud

- Identificar si la solicitud es retiro de materia o retiro total de materia, y asociar el historial académico de la materia o las materias a retirar.
- Obtener el asunto relacionado a la solicitud y asociarlo al caso.
- Obtener la descripción relacionada a la solicitud y asociarlo al caso.
- Crear el caso y asociarlo a la solicitud.

HU 14.- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso

- Obtener y consultar los usuarios involucrados en la solicitud.
- Asociar los usuarios como destinatarios del caso, en el modelo caso_usuario.
- Identificar y asociar al caso el usuario que crea la solicitud como otro destinatario.

HU 15.- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso

- Identificar el estado inicial del caso como equivalente del estado inicial de la solicitud.
- Asociar el historial con el estado inicial de la solicitud al caso, y con el usuario que crea el caso.

HU 16.- Desarrollar el método actualizar_status que permita al usuario autorizado cambiar el estado de la solicitud

- Cambiar el estado, y guardar el historial con dicho cambio y con la identificación del usuario que realizó el cambio.

HU 17.- Modificar método aprobar y no_aprobar de la solicitud.

- Editar el método aprobar para que modifique también el estado del caso.
- Editar el método aprobar para que guarde el historial de la modificación del estado del caso.
- Editar el método no_aprobar para que modifique también el estado del caso.
- Editar el método no_aprobar para que guarde el historial de la modificación del estado del caso.

HU 18.- Actualizar la lista de casos para que los usuarios sepan que se actualizó la solicitud.

- Colocar el caso como no leído para todos los usuarios relacionados con la solicitud.

- **Bitácora de desarrollo** La bitácora de desarrollo se hizo tomando en cuenta las historias de usuario para la iteración 2, y la misma se detalla a continuación.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	26/05/2009	04/06/2009	9	9
2	Análisis y diseño de la implementación	-	04/06/2009	09/06/2009	5	5
3	Desarrollo de la interfaz	-	09/06/2009	17/06/2009	8	8
4	Realización de pruebas	-	17/06/2009	20/06/2009	3	3

- **Diseño** En la figura 3.9 se visualizan las clases de implementación Ruby que dan solución a las historias de usuario mencionadas en la planificación de la iteración

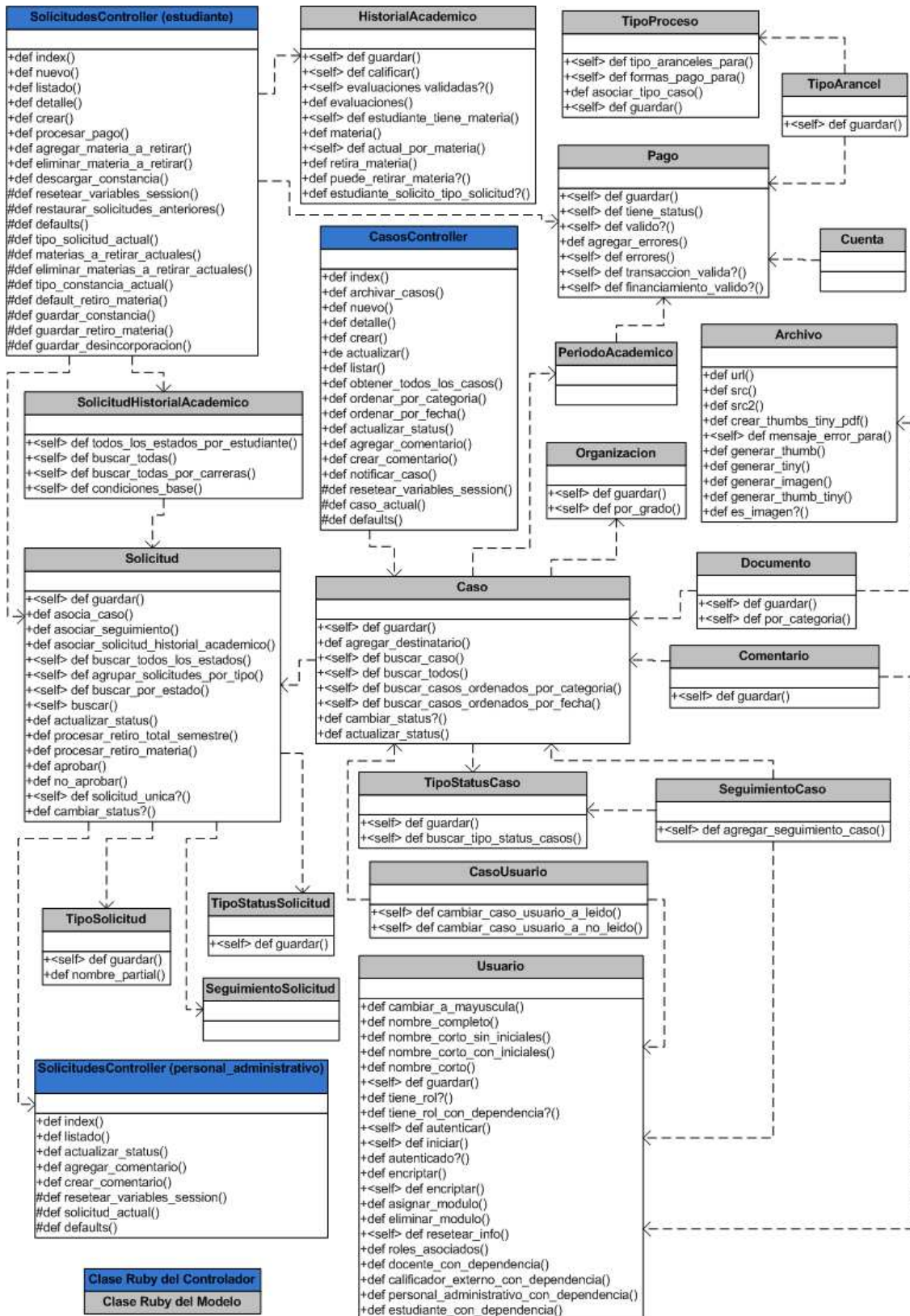


Figura 3.9: Clases de implementación Iteración 2. Módulo Solicitud.

- **Codificación** En la siguiente figura se muestra el extracto de código que permite crear un caso tras la puesta en marcha de un proceso de solicitud de retiro de materia o retiro total de semestre.

```
def asociar_caso(opciones)
  personal_administrativo_destinatario = PersonalAdministrativo.first(:conditions =>
    ["tipo_rol_id = ? AND
     organizacion_id = ?",
     TipoRol::OFICINISTA_1,
     opciones[:organizacion_id]], :joins => :personal_administrativo_en_organizaciones)

  caso = Caso.guardar({:asunto => tipo_solicitud[:descripcion],
    :descripcion => motivo,
    :organizacion_id => opciones[:organizacion_id],
    :categoria_id => id,
    :categoria_type => Solicitud.to_s,
    :fecha_creacion => Time.now,
    :fecha_actualizacion => Time.now,
    :usuario_cedula => opciones[:estudiante_cedula]},
    {:categoria => "Solicitud"})

  SeguimientoCaso.agregar_seguimiento_caso(:caso_id => caso[:retorno][:id],
    :tipo_status_caso_id => TipoStatusCaso::PENDIENTE_SOLICITUD,
    :usuario_cedula => opciones[:estudiante_cedula])

  caso[:retorno].agregar_destinatario(:arreglo_destinatario =>
    personal_administrativo_destinatario[:cedula])

end

def asociar_seguimiento(usuario_cedula)
  SeguimientoSolicitud.create( :solicitud_id => id,
    :tipo_status_solicitud_id => tipo_status_solicitud[:id],
    :usuario_cedula => usuario_cedula,
    :fecha_creacion => Time.now,
    :fecha_actualizacion => Time.now)

end
```

Figura 3.10: Método asociar_caso

- Pruebas

Las pruebas realizadas a la aplicación se basó en la creación de solicitudes estudiantiles tales como retiro de materias y retiro total de semestre, arrojando los resultados esperados por parte del cliente. En dichas pruebas surgieron observaciones las cuales fueron tomadas como referencia para la culminación del sistema de caso asociado al módulo de solicitudes.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Solicitud	Realizar una solicitud de retiro de materia.	Agregar un caso inducido por el estudiante a través del retiro de la materia y agregarlo en el historial académico del mismo.	Fue notificado el ente correspondiente del evento a través de su bandeja de casos y reflejado el resultado de la misma notificando al estudiante.	
2	Solicitud	Realizar una solicitud de retiro total de semestre.	Agregar un caso inducido por el estudiante notificando que necesitará realizar el retiro del semestre.	Fue notificado el ente correspondiente del evento a través de su bandeja de casos y reflejado y el resultado de la misma notificando al estudiante.	

3.6. Iteración 3

- Planificación

Iteración 3	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Oferta Materia
Historias de Usuario a Desarrollar	19- Desarrollar el método asociar_caso que permita crea un caso relacionado a una OfertaMateria 20- Cuando otro profesor oferte una materia ya ofertada agregarlo como destinatario al caso y colocar como no leído a los otros profesores que hayan ofertado previamente 21- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso 22- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso
Tiempo Estimado	19 días
Fechas Inicio/Fin	21-06-2009 / 10-07-2009

- Tareas por Historia de Usuario

HU 19.- Desarrollar el método asociar_caso que permita crea un caso relacionado a una OfertaMateria

- Obtener el asunto relacionado a la oferta materia y asociarlo al caso.
- Obtener la descripción relacionada a la oferta materia y asociarlo al caso.
- Establecer la relación de una oferta materia con el modelo caso, y hacer que al momento de guardar una oferta materia se cree un caso.

HU 20.- Cuando otro profesor oferte una materia ya ofertada agregarlo como destinatario al caso y colocar como no leído a los otros profesores que hayan ofertado previamente

- Validar que si un docente ofertó previamente, no se cree un caso nuevo sino que se actualice el caso relacionado a dicha oferta.
- Agregar el docente que esté ofertando actualmente al caso que se está actualizando
- Colocar el caso como no leído a los otros profesores que hayan ofertado previamente para que puedan notar que el caso ha sido modificado y lo lean de nuevo.

HU 21.- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso

- Obtener y consultar los usuarios involucrados en la oferta materia.
- Asociar los usuarios como destinatarios del caso, en el modelo caso_usuario, como lo son el docente que oferta y el docente con el cargo de "jefe de departamento".
- Identificar y asociar al caso el usuario que crea la oferta materia como otro destinatario.

HU 22.- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso

- Identificar el estado inicial del caso como equivalente del estado inicial de la oferta materia.
- Asociar el historial con el estado inicial de la oferta materia al caso, y con el usuario que cree el caso.

- **Bitácora de desarrollo** La bitácora de desarrollo se hizo generalizando las de historias de usuario de la iteración 3, y se describe a continuación.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	21/06/2009	28/06/2009	7	7
2	Análisis y diseño de la implementación	-	28/06/2009	01/07/2009	3	3
3	Desarrollo del modulo materia	-	01/07/2009	04/07/2009	3	3
4	Desarrollo de la interfaz	-	04/07/2009	07/07/2009	3	3
5	Realización de pruebas	-	07/07/2009	10/07/2009	3	3

- **Diseño** En la figura 3.11 se muestran las clases involucradas en la solución a las historias de usuario del módulo Oferta Materia

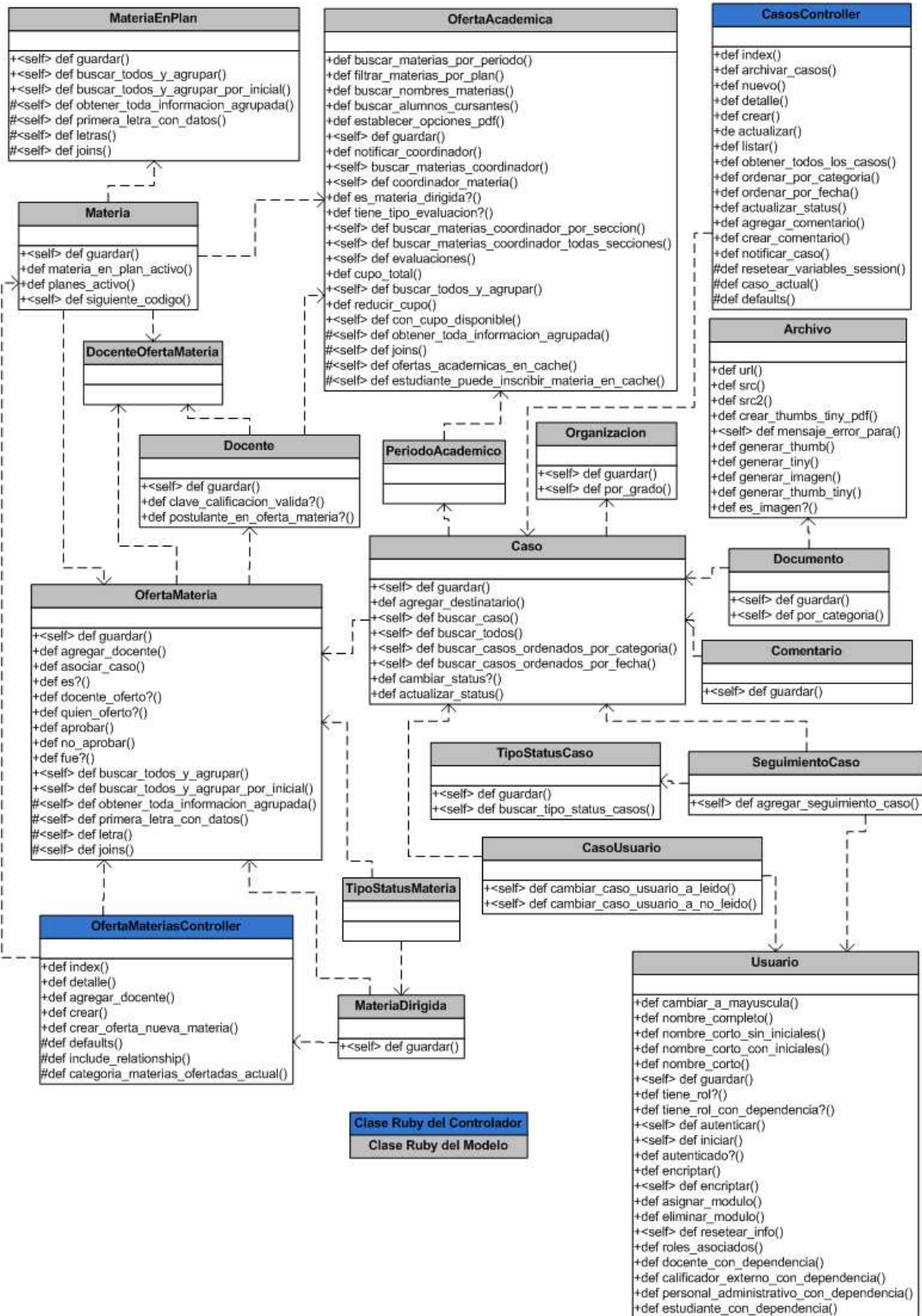


Figura 3.11: Clases de implementación Iteración 3. Módulo Oferta Materia.

- **Codificación** En la siguiente figura se muestra el extracto de código que permite crear un caso al momento que un docente se postular para dictar una materia.

```
def asociar_caso(options={})
  if self.caso.nil?
    arreglo_docente_cargo = DocenteEnCargo.find(:all,
      :conditions => {:organizacion_id => options[:organizacion_id],
        :tipo_cargo_docente_id => "JD"})
    cedulas_docentes = arreglo_docente_cargo.collect{|cp| cp[:docente_cedula]}
    unless cedulas_docentes.include?(options[:docente_cedula])
      cedulas_docentes << options[:docente_cedula]
    end
    docente = Usuario.find(options[:docente_cedula])

    asunto = "Materia "+self.materia[:nombre_corto]+" Ofertada por "+
      Organizacion.find(options[:organizacion_id])[:nombre]
    descripcion = "El docente "+ docente[:primer_nombre]+" "+
      docente[:primer_apellido]+" para el periodo "+
      self[:periodo_academico_id]+" oferto la materia "+self.materia[:nombre_corto]

    fecha = Time.now.to_formatted_s(:db)
    retorno = Caso.guardar({:asunto => asunto,
      :descripcion => descripcion,
      :organizacion_id => options[:organizacion_id],
      :categoria_id => self[:id],
      :categoria_type => self.class.to_s,
      :fecha_creacion => fecha,
      :fecha_actualizacion => fecha,
      :usuario_cedula => options[:docente_cedula]},
      {:categoria => "OfertaMateria"})
    caso = retorno[:retorno]
  else
    cedulas_docentes = options[:docente_cedula]
    caso = self.caso
  end
  SeguimientoCaso.agregar_seguimiento_caso(:caso_id => caso[:id],
    :tipo_status_caso_id => TipoStatusCaso::PENDIENTE_OFERTA_MATERIA,
    :usuario_cedula => options[:docente_cedula])
  caso.agregar_destinatario(:arreglo_destinatario => cedulas_docentes)
end
```

Figura 3.12: Método asociar_caso

- Pruebas

Las pruebas que se realizaron para este módulo se basaron en la postulación de un docente a materias del plan docente, dichas solicitudes o peticiones para dictar materias generaron un caso el cual debe llegar a la bandeja de casos de cada coordinador de la materia el cual el docente postulante quiere dictar.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Oferta Materia	Se realiza una postulación para dictar materias del plan docente.	Se debe crear un caso dirigido al comité académico del postgrado notificando las postulaciones de los docentes a las materias del plan docente.	Se creó el caso y tras la aceptación de la postulación se logró ver que el caso cambió de estado.	

3.7. Iteración 4

- Planificación

Iteración 4	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Plan Docente
Historias de Usuario a Desarrollar	23- Modificar los métodos que actualizan el estado de una oferta materia para que actualice el estado del caso 24- Actualizar la lista de casos para que los usuarios sepan que se actualizó la plan docente.
Tiempo Estimado	8 días
Fechas Inicio/Fin	11-07-2009 / 19-07-2009

- Tareas por Historia de Usuario

HU 23.- Modificar los métodos que actualizan el estado de una oferta materia para que actualice el estado del caso

- Cambiar el método aprobar para que el estado del caso indique que fue “aceptada” la oferta materia, pasando a formar del plan docente.
- Cambiar el método no_aprobar para que el estado del caso indique que fue “no_aceptada” la oferta materia.

HU 24.- Actualizar la lista de casos para que los usuarios sepan que se actualizó el plan docente.

- Colocar el caso como no leído para todos los usuarios relacionados con la plan docente.

- **Bitácora de desarrollo** La bitácora de desarrollo abarca las historias de usuario para la iteración 4, y es detallada a continuación.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	11/07/2009	13/07/2009	2	2
2	Montaje de las plantillas(CSS + HTML)	-	13/07/2009	15/07/2009	2	2
3	Análisis y diseño de la implementación	-	15/07/2009	17/07/2009	2	2
4	Desarrollo de la interfaz	-	17/07/2009	18/07/2009	1	1
5	Realización de pruebas	-	18/07/2009	19/07/2009	1	1

- **Diseño** En la figura 3.13 se muestran las clases involucradas en la solución a las historias de usuario del módulo Plan Docente

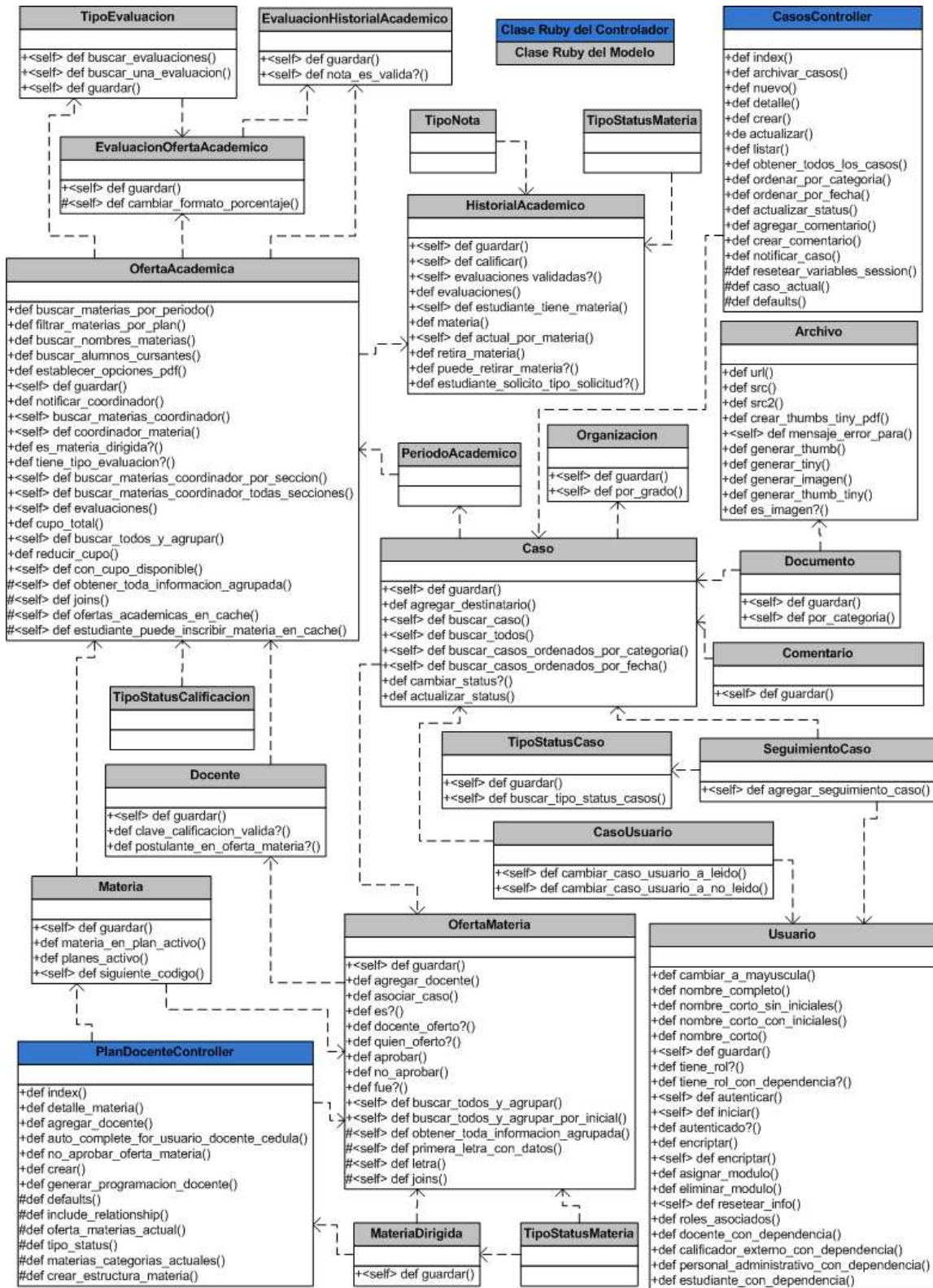


Figura 3.13: Clases de implementación Iteración 4. Módulo Plan Docente.

- **Codificación** En la siguiente figura se muestra el extracto de código que permite el cambio de estado de un caso para un plan docente.

```
def aprobar(opciones={})
  aprobada = TipoStatusOfertaMateria::APROBADA
  update_attribute(:tipo_status_oferta_materia_id, aprobada)
  completado_aceptado = TipoStatusCaso::ACEPTADA_OFERTA_MATERIA
  self.caso.update_attribute(:tipo_status_caso_id, completado_aceptado)
  SeguimientoCaso.agregar_seguimiento_caso(:caso_id => self.caso[:id],
                                           :tipo_status_caso_id => completado_aceptado,
                                           :usuario_cedula => opciones[:usuario_cedula])
end

def no_aprobar(opciones={})
  no_aprobada = TipoStatusOfertaMateria::NO_APROBADA
  update_attribute(:tipo_status_oferta_materia_id, no_aprobada)
  completado_no_aceptado = TipoStatusCaso::NO_ACEPTADA_OFERTA_MATERIA
  self.caso.update_attribute(:tipo_status_caso_id, completado_no_aceptado)
  SeguimientoCaso.agregar_seguimiento_caso(:caso_id => self.caso[:id],
                                           :tipo_status_caso_id => completado_no_aceptado,
                                           :usuario_cedula => opciones[:usuario_cedula])
end
```

Figura 3.14: Métodos de cambio de estado para Plan Docente

- **Pruebas** La prueba realizada se basó en ofertar materias por parte del grupo docente, lo cual genera un evento que notifica al Coordinador del Postgrado correspondiente qué docentes de la planta desean dictar materias. Tras dichas postulaciones el Coordinador procedera a ejecutar la programación docente.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Plan Docente	Los profesores se postulan para dictar materias, una vez realizada la postulación se crearan casos para el Coordinador del Postgrado correspondiente para que ejecute la Programación Docente.	Se creen los casos para los docentes involucrados en la programación docente.	Efectivamente se crean los casos para los correspondientes coordinadores y al ejecutar programación docente se informan a los docentes cual ha sido el estado en el cual a pasado su solicitud de dictar una materia en el periodo actual.	

3.8. Iteración 5

- Planificación

Iteración 5	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Admisión
Historias de Usuario a Desarrollar	25- Desarrollar el método asociar_caso que permita crear un caso asociado a la admisión que a su vez tiene una relación con el aspirante 26- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso 27- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso
Tiempo Estimado	12 días
Fechas Inicio/Fin	20-07-2009 / 01-08-2009

- Tareas por Historia de Usuario

HU 25.- Desarrollar el método asociar_caso que permita crear un caso asociado a la admisión que a su vez tiene una relación con el aspirante.

- Obtener el asunto relacionado a la admisión y asociarlo al caso.
- Obtener la descripción relacionada a la admisión y asociarlo al caso.
- Establecer la relación de una admisión con el modelo caso, y hacer que al momento de guardar una admisión se cree un caso.

HU 26.- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso

- Obtener y consultar los usuarios involucrados en la admisión.
- Asociar los usuarios como destinatarios del caso, en el modelo caso_usuario, como lo son la secretaria de la coordinación de postgrado, y al coordinador y la secretaria del postgrado que solicita el aspirante.
- Identificar y asociar al caso el usuario que crea la admisión como otro destinatario.

HU 27.- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso

- Identificar el estado inicial del caso como equivalente del estado inicial de la admisión.
- Asociar el historial con el estado inicial de la admisión al caso, y con el usuario que cree el caso.

- **Bitácora de desarrollo** La bitácora de desarrollo generaliza las historias de usuario para la iteración 5, y es detallada a continuación.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	20/07/2009	23/07/2009	3	3
2	Análisis y diseño de la implementación	-	23/07/2009	26/07/2009	3	3
3	Desarrollo e integración de componentes	-	26/07/2009	28/07/2009	2	2
4	Desarrollo de la interfaz	-	28/07/2009	30/07/2009	2	2
4	Realización de pruebas	-	30/07/2009	01/08/2009	2	2

- **Codificación** En la siguiente figura se muestra el extracto de código que permite crear un caso una vez que el aspirante es registrado para un postgrado.

```

def asociar_caso
  carrera = Carrera.find_by_id(carrera_id)
  arreglo_destinatarios = []
  organizacion = carrera.organizacion
  if carrera.tipo_grado_carrera[:id].eql?(TipoGradoCarrera::POSTGRADO)
    personal_administrativo = PersonalAdministrativo.first(:conditions =>
      ["organizacion_id = ? AND tipo_rol_id = ?",
       Organizacion::COORDINACION_POSTGRADO, TipoRol::OFICINISTA_1],
      :joins => :personal_administrativo_en_organizaciones)
    arreglo_destinatarios << personal_administrativo[:cedula]
  end
  unless arreglo_destinatarios.include?(organizacion.departamento[:docente_cedula_jefe])
    arreglo_destinatarios << organizacion.departamento[:docente_cedula_jefe]
  end
  caso = Caso.guardar({:asunto => "Nuevo Aspirante "+aspirante[:nombre].capitalize+" "+
    aspirante[:apellido].capitalize,
    :descripcion => aspirante[:nombre].capitalize+" "+
    aspirante[:apellido].capitalize+
    " cedula "+ aspirante_cedula.to_s+
    ", se ha registrado para "+carrera[:nombre],
    :organizacion_id => organizacion[:id],
    :categoria_id => id,
    :categoria_type => Admision.to_s,
    :fecha_creacion => Time.now.to_formatted_s(:db),
    :fecha_actualizacion => Time.now.to_formatted_s(:db),
    :usuario_cedula => aspirante_cedula}, {:categoria => "Admision"})
  if caso[:exito]
    SeguimientoCaso.agregar_seguimiento_caso(:caso_id => caso[:retorno][:id],
      :tipo_status_caso_id => TipoStatusCaso::PENDIENTE_ADMISION,
      :usuario_cedula => aspirante_cedula)
    caso[:retorno].agregar_destinatario(:arreglo_destinatario => arreglo_destinatarios)
  end
end
end

```

Figura 3.15: Método asociar_caso

- **Diseño** En la figura 3.16 se muestran las clases involucradas en la solución a las historias de usuario del módulo Admisión

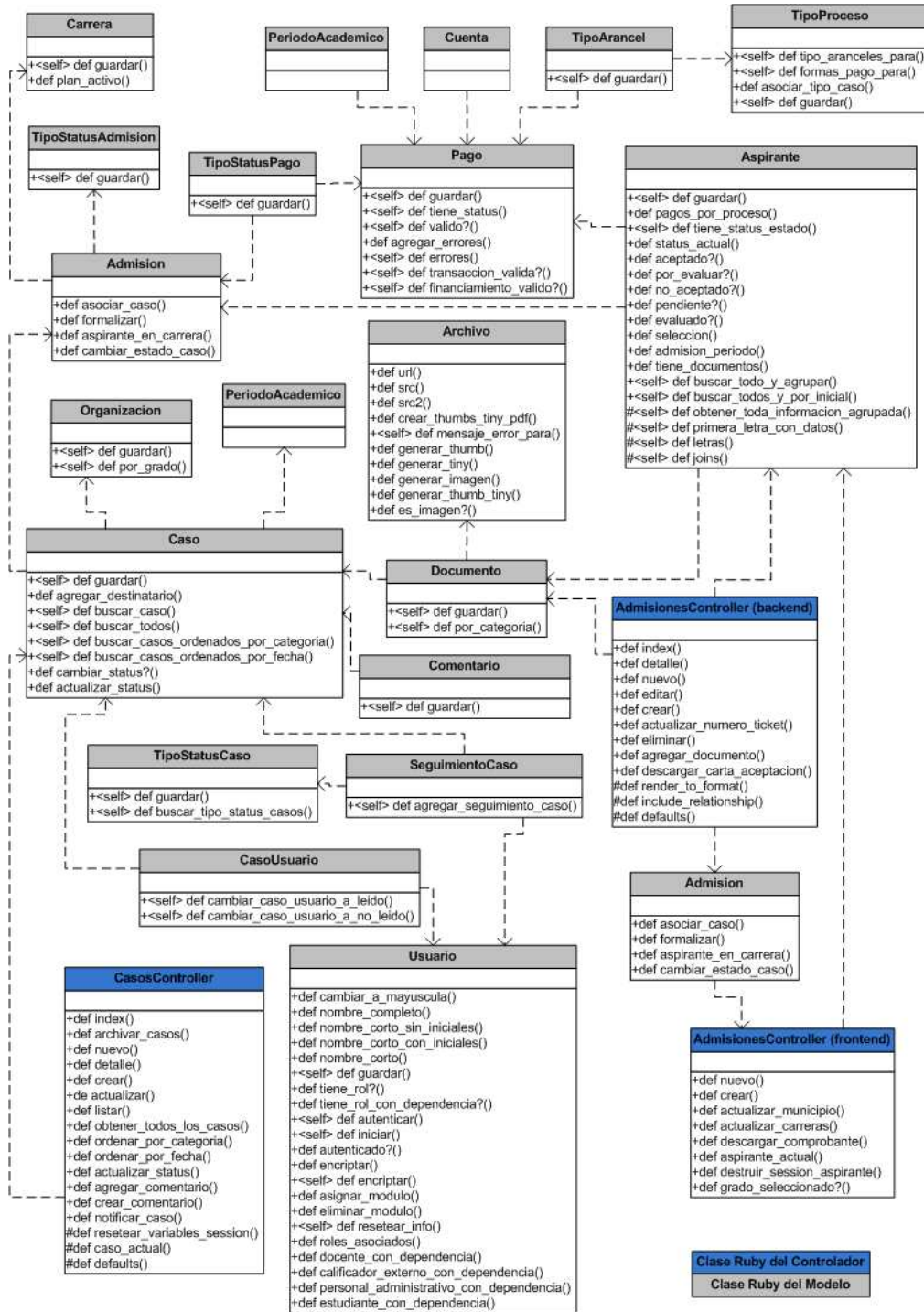


Figura 3.16: Clases de implementación Iteración 5. Módulo Admisión.

- Pruebas

Las pruebas realizadas para este modulo corresponden con la postulación por parte de un aspirante que quiere realizar estudios de postgrado.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Admisión	Se postula un aspirante al postgrado, para ello introduce datos personales e información referente al postgrado que quiere cursar.	Se crea un caso dirigido a la coordinación de postgrado.	Efectivamente se crea el caso dirigido a la coordinación de postgrado con estado "pendiente"	

3.9. Iteración 6

- Planificación

Iteración 6	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Selección
Historias de Usuario a Desarrollar	28- Desarrollar método cambiar_estado_caso que permita modificar el estado del caso referente a la selección, según el paso y el perfil donde se encuentre. 29- Modificación del método tiene_documento para cambiar el estado a “por evaluar”. 30- Modificación del método seleccion para cambiar el estado a “aceptado” o “no aceptado”. 31- Modificación del método formalizar_admision para formalizar después de ser aceptado.
Tiempo Estimado	6 días
Fechas Inicio/Fin	02-08-2009 / 08-08-2009

- Tareas por Historia de Usuario

HU 28.- Desarrollar método cambiar_estado_caso que permita modificar el estado del caso referente a la selección, según el paso y el perfil donde se encuentre.

- Obtener el estado al cual debe pasar el caso y el usuario que genera ese cambio de estado.
- Actualizar el estado del caso.
- Agregar un registro al seguimiento del caso.

HU 29.- Modificación del método tiene_documento para cambiar el estado a “por evaluar”.

- Cuando se agrega un documento al aspirante, cambiar el estado del caso a “por evaluar”.

HU 30.- Modificación del método seleccion para cambiar el estado a “aceptado” o “no aceptado”.

- Cuando el coordinador de área evalúe puede pasar dos estados: “aceptado” o “no aceptado”. Actualizar dichos estado en el caso.

HU 31.- Modificación del método formalizar_admision para formalizar después de ser aceptado.

- Si es aceptado, al momento en que la secretaria de coordinación de Postgrado formaliza la inscripción, cambiar el estado del caso a “formalizada”.

- **Diseño** En la figura 3.17 se muestran las clases involucradas en la solución a las historias de usuario del módulo Selección

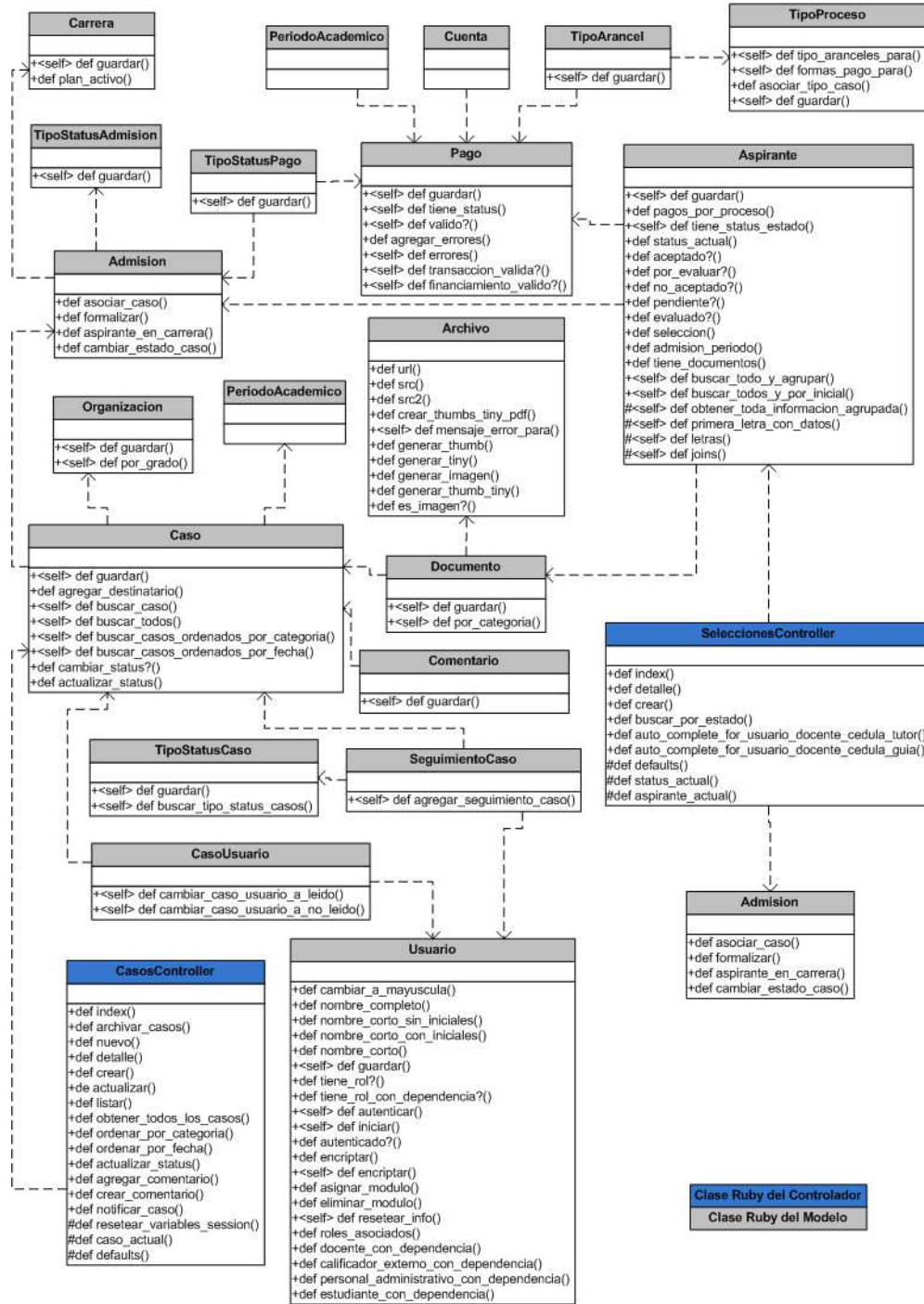


Figura 3.17: Clases de implementación Iteración 6. Módulo Selección.

- Bitácora de desarrollo

La bitácora de desarrollo utilizada para el modulo de Selección fueron tomadas en cuenta en conjunto de las historias de usuario, generalizando así en la iteración lo realizado desde el inicio hasta la culminación de la misma.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	02/08/2009	04/08/2009	2	2
2	Montaje de las plantillas(CSS + HTML)	-	04/08/2009	05/08/2009	1	1
3	Análisis y diseño de la implementación	-	05/08/2009	06/08/2009	1	1
4	Desarrollo de la interfaz	-	06/08/2009	07/08/2009	1	1
5	Realización de pruebas	-	07/08/2009	08/08/2009	1	1

- **Codificación** En la siguiente figura se muestra el extracto de código que realiza la selección de un aspirante y el cambio de estado del caso asociado.

```
def seleccion(attr,opciones={})
  opciones[:periodo_academico] ||= ConestBase.periodo_academico_actual
  admision = self.admisiones.find_by_periodo_academico_id(opciones[:periodo_academico])
  if attr[:tipo_status_admision_id]
    admision.tipo_status_admision = TipoStatusAdmision.find_by_id(attr[:tipo_status_admision_id])
    admision.docente_guia = Docente.find_by_cedula(opciones[:docente_cedula_guia])
    admision.docente_tutor = Docente.find_by_cedula(opciones[:docente_cedula_tutor])
  end
  admision[:observaciones] = attr[:observaciones]
  exito = admision.valid?
  if exito
    admision.save
    begin
      Correo::deliver_enviar_correo_aspirante_resultado(self,opciones) if attr[:tipo_status_admision_id]
    rescue
      parate
    end
    tipo_status_caso = case attr[:tipo_status_admision_id]
      when "A"
        TipoStatusCaso::ACEPTADA_ADMISION
      when "NA"
        TipoStatusCaso::NO_ACEPTADA_ADMISION
      end
    admision.cambiar_estado_caso(tipo_status_caso,opciones[:usuario_actual])
  end
  {:exito => exito, :retorno => [admision]}
end
```

Figura 3.18: Método seleccion

En la siguiente figura se muestra el extracto de código que realiza el cambio de estado de la postulación de un aspirante.

```
def cambiar_estado_caso(id_status_caso, usuario_actual)
  por_evaluar_caso = TipoStatusCaso.find(id_status_caso)
  self.caso.update_attribute(:tipo_status_caso_id, por_evaluar_caso)
  SeguimientoCaso.agregar_seguimiento_caso(:caso_id => self.caso[:id],
                                           :tipo_status_caso_id => por_evaluar_caso,
                                           :usuario_cedula => usuario_actual)
end
```

Figura 3.19: Método `cambiar_estado_caso`

- Pruebas

Las pruebas realizadas para este modulo corresponden con el cambio de estado de una postulación por parte de un aspirante, debe seguir un flujo que pasa por varias personas encargadas de la parte académica del postgrado.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Selección	Cuando la coordinación de postgrado anexe los documentos el caso debe cambiar de estado pendiente a por evaluar.	Debe cambiar de estado el caso y ahora debe ir dirigido al comité académico del postgrado para que realice la correspondiente evaluación.	El caso cambia de estado de pendiente a por evaluar.	
2	Selección	Se genera la constancia de que el aspirante fue evaluado por el comité académico	El estado del caso cambia de estado por evaluar a aceptado la admisión y debe ser notificado el aspirante	El caso cambió de estado y fue creado una cuenta para el aspirante que ahora pasa a ser usuario del sistema y se le crea un caso notificandole que fue aceptado en el postgrado.	
3	Selección	Generar la carta de aceptación y formalizar la selección del aspirante que ahora es estudiante	El estado del caso cambia de estado aceptado a formalizada la admisión y debe ser notificado el aspirante	El caso cambió de estado y fue notificado el estudiante y el comité académico del postgrado solicitado.	

3.10. Iteración 7

- Planificación

Iteración 7	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Evaluación
Historias de Usuario a Desarrollar	<p>32- Desarrollar método que permita guardar una evaluación asociada a una materia y sus diferentes tipos de evaluación</p> <p>33- Desarrollar el método asociar_caso que permita crear un caso relacionado a una evaluación, como una nota informativa</p> <p>34- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso</p> <p>35- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso</p> <p>36- Modificar método crear_actualizar para que se cree un nuevo caso al cambiar las evaluaciones de una materia</p>
Tiempo Estimado	16 días
Fechas Inicio/Fin	09-08-2009 / 25-08-2009

- Tareas por Historia de Usuario

HU 32.- Desarrollar método que permita guardar una evaluación asociada a una materia y sus diferentes tipos de evaluación.

- Crear la relación con Oferta Académica para identificar a que materia de que periodo académico se le hace la evaluación
- Crear la relación con Evaluacion Oferta Academica para asociar todas las evaluaciones con sus tipos y con el porcentaje que pueda tener. (Parcial 1, quiz 1, etc.).
- Guardar una evaluación con los parámetros necesario para lograr la relación con los otros modelos mencionados en los ítem anteriores.

HU 33.- Crear el método asociar_caso que cree un caso asociado a una evaluación.

- Obtener el asunto relacionado a la evaluación y asociarlo al caso.
- Obtener la descripción relacionada a la evaluación y asociarlo al caso.
- Establecer la relación de una evaluación con el modelo caso, y hacer que al momento de guardar una evaluación se cree un caso.

HU 34.- Aplicar el método agregar_destinatario que determine a quienes debe llegar ese caso

- Obtener y consultar los usuarios involucrados en la evaluación.

- Asociar los usuarios como destinatarios del caso en el modelo caso_usuario, como lo son los estudiantes inscritos para la materia a la que se le crea la evaluación y el profesor coordinador de dicha materia, .
- Identificar y asociar al caso el usuario que crea la evaluación como otro destinatario.

HU 35.- Aplicar el método agregar_seguimiento_caso que permita llevar el historial del caso

- Identificar el estado inicial del caso como equivalente del estado inicial de la evaluación.
- Asociar el historial con el estado inicial de la evaluación al caso, y con el usuario que cree el caso.

HU 36.- Modificar método crear_actualizar para que se cree un nuevo caso al cambiar las evaluaciones de una materia.

- Identificar la evaluación a cambiar con sus relaciones y actualizar los valores modificados por el usuario.
- Crear un nuevo caso para la evaluación editada.

- **Bitácora de desarrollo** La bitácora de desarrollo para la iteración de calificación toma en cuenta las historias de usuario planteadas, y éstas se generalizan a continuación.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	09/08/2009	13/08/2009	4	4
2	Montaje de las plantillas(CSS + HTML)	-	13/08/2009	16/08/2009	3	3
3	Análisis y diseño de la implementación	-	16/08/2009	20/08/2009	4	4
4	Desarrollo de la interfaz	-	20/08/2009	23/08/2009	3	3
5	Realización de pruebas	-	23/08/2009	25/08/2009	2	2

- **Diseño** En la figura 3.20 se muestran las clases involucradas en la solución a las historias de usuario del módulo Evaluacion

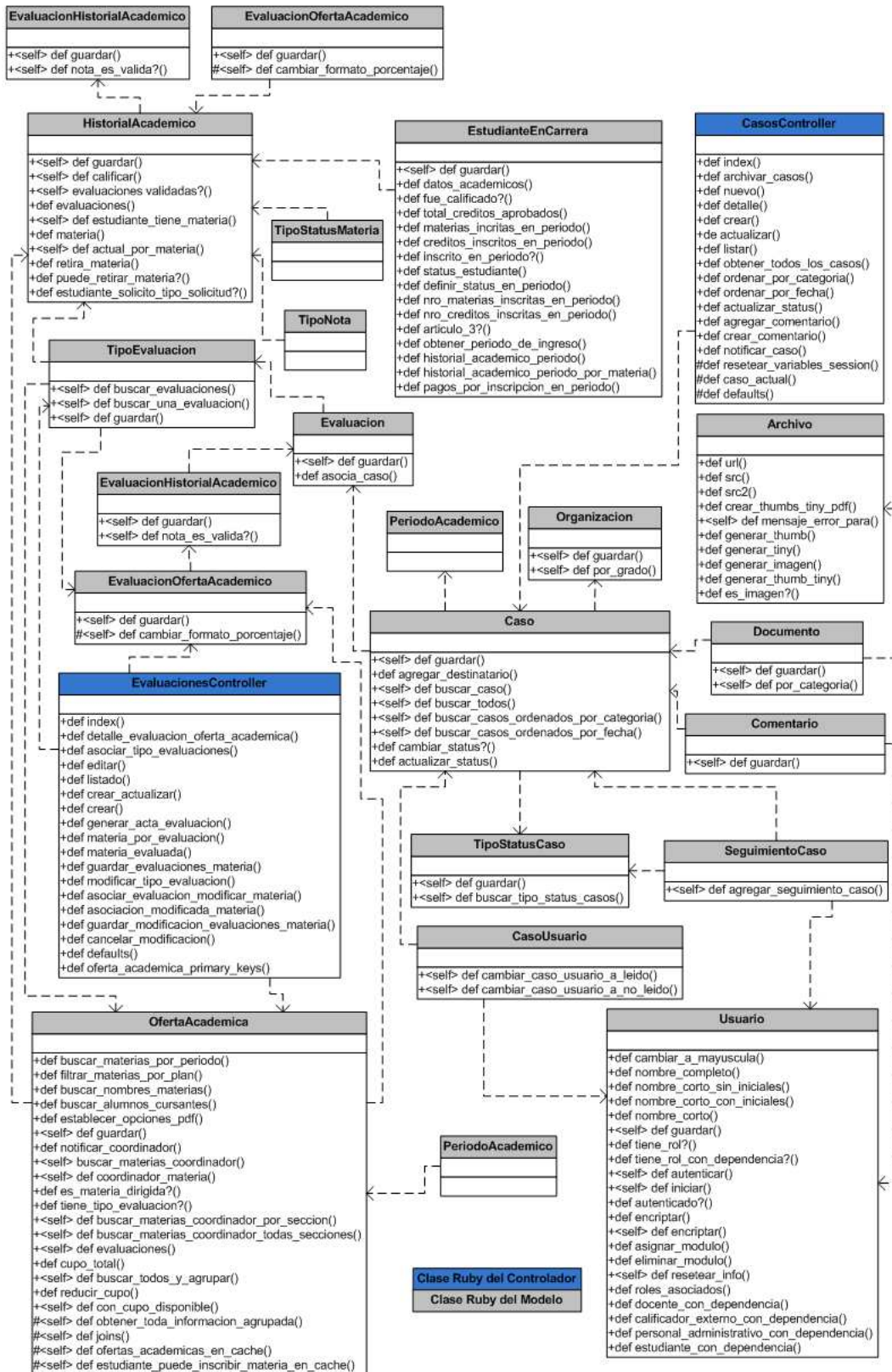


Figura 3.20: Clases de implementación Iteración 7. Módulo Evaluacion.

- **Codificación** En la siguiente figura se muestra el extracto de código que permite crear un caso una vez que el docente cree una evaluación para una materia.

```

def self.guardar(attr, opciones={})
  exito = false
  evaluacion = new(:materia_codigo => attr[:materia_codigo],
                  :periodo_academico_id => attr[:periodo_academico],
                  :nombre_seccion => attr[:nombre_seccion],
                  :fecha_creacion => Time.now,
                  :fecha_actualizacion => Time.now)
  if evaluacion.save
    evaluacion.asociar_caso(attr)
    exito = true
  end
  return {:exito => exito, :retorno => evaluacion}
end

def asociar_caso(opciones)
  arreglo_destinatarios = HistorialAcademico.find(:all,
    :conditions => ["periodo_academico_id = ? AND
    materia_codigo = ? AND nombre_seccion = ? AND
    tipo_status_materia_id = ?",
    opciones[:periodo_academico],
    opciones[:materia_codigo],
    opciones[:nombre_seccion], "SC"]).collect{ |ha| ha[:estudiante_cedula] }
  unless arreglo_destinatarios.include?(opciones[:docente_cedula])
    arreglo_destinatarios << opciones[:docente_cedula]
  end
  caso = Caso.guardar({:asunto => "Nota informativa",
    :descripcion => "asignadas evaluaciones y porcentaje para la materia "+
    opciones[:materia_codigo],
    :organizacion_id => opciones[:organizacion_id],
    :categoria_id => id,
    :categoria_type => Evaluacion.to_s,
    :fecha_creacion => Time.now,
    :fecha_actualizacion => Time.now,
    :usuario_cedula => opciones[:docente_cedula]}, {:categoria => "Evaluacion"})
  if caso[:exito]
    SeguimientoCaso.agregar_seguimiento_caso(:caso_id => caso[:retorno][:id],
    :tipo_status_caso_id => TipoStatusCaso::PENDIENTE_EVALUACION,
    :usuario_cedula => opciones[:docente_cedula])
    caso[:retorno].agregar_destinatario(:arreglo_destinatario => arreglo_destinatarios)
  end
end
end

```

Figura 3.21: Método asociar_caso para una evaluación

- Pruebas

Las pruebas fueron realizadas con el perfil de varios coordinadores al azar y que estos crearon evaluaciones para la materia que dictan.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Evaluación	El coordinador de la materia procede a crear las evaluaciones con los porcentajes y las fechas.	Una vez que el profesor crea y acepta las evaluaciones se crea un caso dirigido a los estudiantes cursantes de la materia y al profesor que la dicta.	Se creó el caso dirigido a cada estudiante que está cursando la materia y al profesor que la dicta.	

3.11. Iteración 8

- Planificación

Iteración 8	
Descripción	Desarrollo de la adaptación de Caso para el módulo de Calificación
Historias de Usuario a Desarrollar	37- Crear método guardar en el modelo Calificación. Solo cuando es calificación definitiva. 38- Desarrollar el método asociar_caso que permita crear un caso relacionado a una calificación. 39- Aplicar el método agregar_destinatarios que determine a quienes debe llegar ese caso. 40- Aplicar método agregar_seguimiento_caso que permita llevar el historial del caso.
Tiempo Estimado	14 días
Fechas Inicio/Fin	26-08-2009 / 09-09-2009

- Tareas por Historia de Usuario

HU 37.- Crear método guardar en el modelo Calificación. Solo cuando es calificación definitiva.

- Crear la relación con Oferta Académica para identificar a que materia de que periodo académico se le hace la calificación
- Crear la relación con Evaluacion Historial Academico ya que allí queda registrada la calificación.
- Guardar una calificación con los parámetros necesario para lograr la relación con los otros modelos mencionados en los ítem anteriores.

HU 38.- Desarrollar el método asociar_caso que permita crear un caso relacionado a una calificación.

- Obtener el asunto relacionado a la calificación y asociarlo al caso.
- Obtener la descripción relacionada a la calificación y asociarlo al caso.
- Establecer la relación de una calificación con el modelo caso, y hacer que al momento de guardar una calificación se cree un caso.

HU 39.- Aplicar el método agregar_destinatarios que determine a quienes debe llegar ese caso.

- Obtener y consultar los usuarios involucrados en la calificación.
- Asociar los usuarios como destinatarios del caso en el modelo caso_usuario, como lo son los estudiantes cursante de la materia a calificar y al profesor coordinador de la materia.

- Identificar y asociar al caso el usuario que crea la calificación como otro destinatario.

HU 40.- Aplicar método agregar_seguimiento_caso que permita llevar el historial del casos.

- Identificar el estado inicial del caso como equivalente del estado inicial de la calificación.
- Asociar el historial con el estado inicial de la calificación al caso, y con el usuario que cree el caso.

- **Bitácora de desarrollo** La bitácora de desarrollo toma en cuenta las historias de usuario propuesta, y el esquema seguido es el siguiente.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	26/08/2009	29/08/2009	3	3
2	Montaje de las plantillas(CSS + HTML)	-	29/08/2009	01/09/2009	3	3
3	Análisis y diseño de la implementación	-	01/09/2009	04/09/2009	3	3
4	Desarrollo de la interfaz	-	04/09/2009	07/09/2009	3	3
5	Realización de pruebas	-	07/09/2009	09/09/2009	2	2

- **Diseño** En la figura 3.22 se muestran las clases involucradas en la solución a las historias de usuario del módulo Calificación

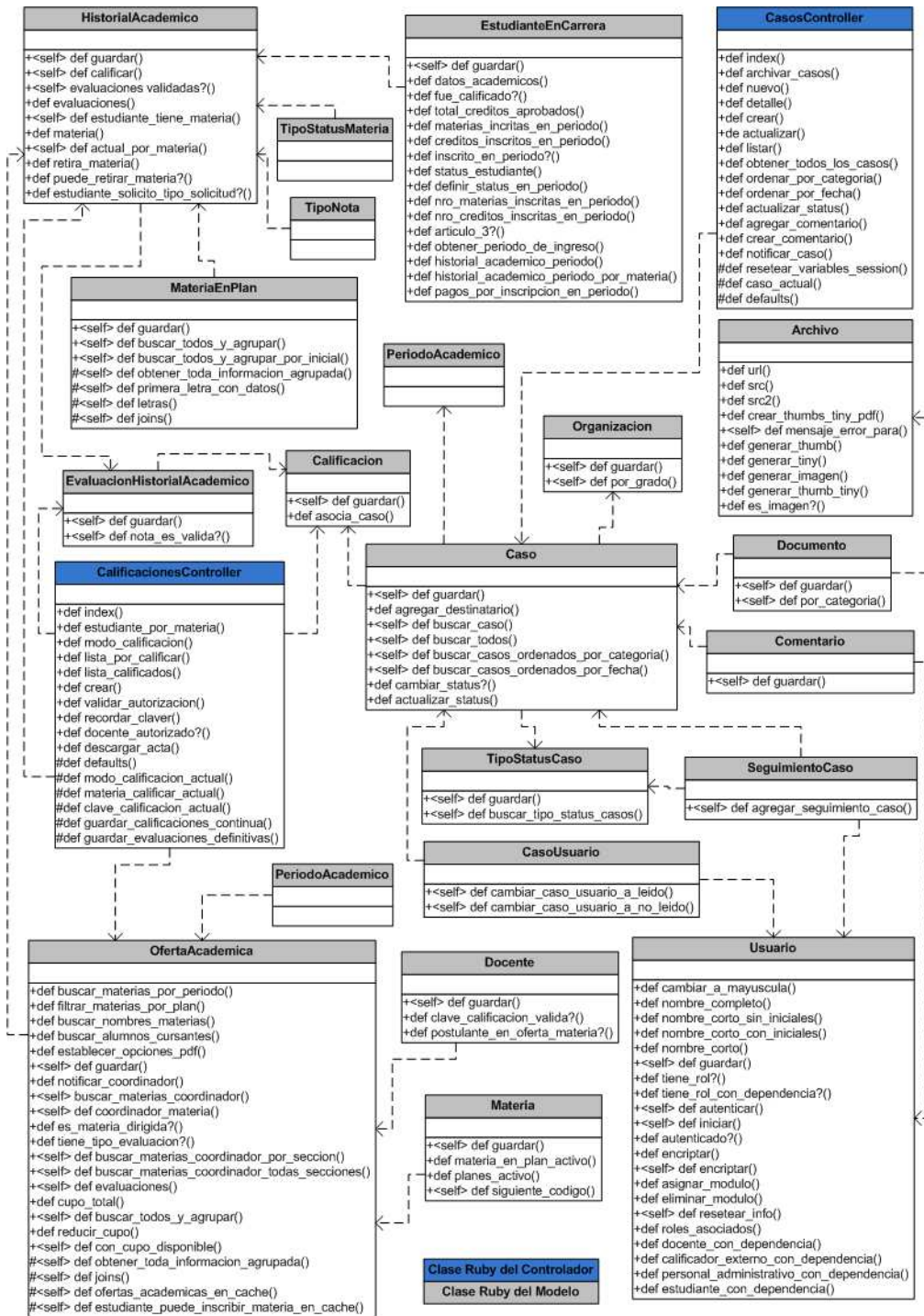


Figura 3.22: Clases de implementación Iteración 8. Módulo Calificación.

- **Codificación** En la siguiente figura se muestra el extracto de código que permite crear un caso una vez que el docente califica una materia de forma definitiva.

```

def self.guardar(opciones)
  exito = false
  calificacion = new(:materia_codigo => opciones[:materia_codigo],
                    :periodo_academico_id => opciones[:periodo_academico],
                    :fecha_creacion => Time.now.to_formatted_s(:db),
                    :fecha_actualizacion => Time.now.to_formatted_s(:db))

  if calificacion.save
    calificacion.asociar_caso(opciones)
    exito = true
  end
  return {:exito => exito, :retorno => calificacion}
end

def asociar_caso(opciones)
  arreglo_destinatarios = opciones[:estudiantes].keys
  unless arreglo_destinatarios.include?(opciones[:usuario_actual])
    arreglo_destinatarios << opciones[:usuario_actual]
  end
  caso = Caso.guardar({:asunto => "Calificacion definitiva materia "+opciones[:materia_codigo],
                    :descripcion => "Calificada la materia
                    <b>#{Materia.find_by_codigo(opciones[:materia_codigo])[:nombre].capitalize}</b>",
                    :organizacion_id => opciones[:organizacion_id],
                    :categoria_id => id,
                    :categoria_type => Calificacion.to_s,
                    :fecha_creacion => Time.now.to_formatted_s(:db),
                    :fecha_actualizacion => Time.now.to_formatted_s(:db),
                    :usuario_cedula => opciones[:usuario_actual][:cedula]}, {:categoria => "Calificacion"})
  if caso[:exito]
    SeguimientoCaso.agregar_seguimiento_caso(:caso_id => caso[:retorno][:id],
      :tipo_status_caso_id => TipoStatusCaso::PENDIENTE_CALIFICACION,
      :usuario_cedula => opciones[:usuario_actual])
    caso[:retorno].agregar_destinatario(:arreglo_destinatario => arreglo_destinatarios)
  end
end

```

Figura 3.23: Método asociar_caso para una calificación

- Pruebas

Las pruebas fueron realizadas con el perfil de varios coordinadores al azar y que estos calificaron a los estudiantes de la materia que dictan.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Calificación	El coordinador de la materia procede a ingresar las notas de los estudiantes en el sistema de forma definitiva, el sistema solicitará una clave previamente enviada por correo al profesor coordinador encargado de calificar a los estudiantes.	Una vez que el profesor califica a los estudiantes y formaliza dicha calificación se crea un caso dirigido a los estudiantes cursantes de la materia y al profesor evaluador.	Se creó el caso dirigido a cada estudiante que están cursando la materia y al profesor calificador.	

3.12. Iteración 9

- Planificación

Iteración 9	
Descripción	Desarrollo de últimas revisiones
Historias de Usuario a Desarrollar	<p>41- Crear la metáfora para identificar que un caso tiene documentos asociados.</p> <p>42- Crear una alerta que notifique cada vez que llegue un nuevo caso o cuando un caso existente sea actualizado.</p> <p>43- Crear nuevas categorías para que muestre el caso según el modulo que represente, tales como Solicitud, Oferta Materia-Planificación Docente, Admisión - Selección, Inscripción, Evaluación y Calificación.</p> <p>44- Crear check-box para seleccionar los casos que se quieran archivar.</p> <p>45- Finalizado el proceso de inscripción notificar al personal administrativo correspondiente de los estudiantes que no se hayan inscrito.</p> <p>46- Crear un módulo activación que permita a la secretaria de Postgrado activar/desactivar los módulos de inscripción y calificación.</p>
Tiempo Estimado	16 días
Fechas Inicio/Fin	10-09-2009 / 26-09-2009

- Tareas por Historia de Usuario

HU 41.- Crear la metáfora para identificar que un caso tiene documentos asociados.

- Para cada caso hacer la búsqueda que indique si tiene documentos.
- Crear la vista que muestre en cada item de la bandeja de casos el icono que indica que un caso tiene documentos.

HU 42.- Crear una alerta que notifique cada vez que llegue un nuevo caso o cuando un caso existente sea actualizado.

- Crear una vista en un pequeño 'divén la parte inferior izquierda que sirva para indicar cuando haya una notificación.
- Consultar la base de datos y verificar si hay un nuevo caso o un caso actualizado para la persona autenticada.
- Realizar llamadas periódica que haga la verificación en la base de datos.

HU 43.- Crear nuevas categorías para que muestre el caso según el módulo que represente, tales como Solicitud, Oferta Materia-Planificación Docente, Admisión - Selección, Inscripción, Evaluación y Calificación.

- Mostrar las categorías en la vista como botones para que el usuario pueda escogerlas.

- Realizar la búsqueda en la base de datos de los casos, según la categoría que seleccione el usuario y según los casos que estén asociados a el usuario autenticado.

HU 44.- Crear check-box para seleccionar los casos que se quieran archivar.

- Se crean en la vista un check-box para cada caso.
- Con el arreglo de casos seleccionados se llama a un método que actualiza en la base de datos el estado de dichos casos para que queden como “guardado”.
- Se coloca en la vista el botón de ‘Todos’ como una opción para visualizar todos los casos, sin importar el estado.

HU 45.- Finalizado el proceso de inscripción notificar al personal administrativo correspondiente, de los estudiantes que no se hayan inscrito.

- Al desactivar el módulo inscripción, crear la consulta que verifique aquellos estudiantes que para el período actual, cumplan con las condiciones para inscribirse y no lo hayan hecho.
- Obtener el asunto relacionado a la inscripción y crear un caso asociado.
- Obtener la descripción relacionada a la inscripción y asociarla al caso.
- Obtener y consultar los usuarios involucrados en la inscripción.
- Asociar los usuarios como destinatarios del caso en el modelo caso_usuario, como lo son la secretaria del postgrado y la secretaria de Coordinación de Postgrado.
- Identificar el estado inicial del caso como equivalente del estado inicial de la inscripción.
- Asociar el historial con el estado inicial de la inscripción al caso, y con el usuario que cree el caso.

HU 46.- Crear un módulo activación que permita a la secretaria de Postgrado activar/desactivar los módulos de inscripción y calificación.

- Crear una interfaz para que se active o desactive los módulos inscripción y/o calificación.
- Al activar el módulo calificación, para todos los profesores coordinadores de materias en el período actual, es generado una clave y ésta es guardada en la base de datos y enviada vía correo electrónico.
- Asignar el módulo activación al rol secretaria de Postgrado.
- Asignar un perfil al módulo administrativo y quitar la opción de borrar un módulo con la X.

- **Diseño** En la figura 3.24 se muestran las clases involucradas en la solución a las historias de usuario del módulo Inscripción

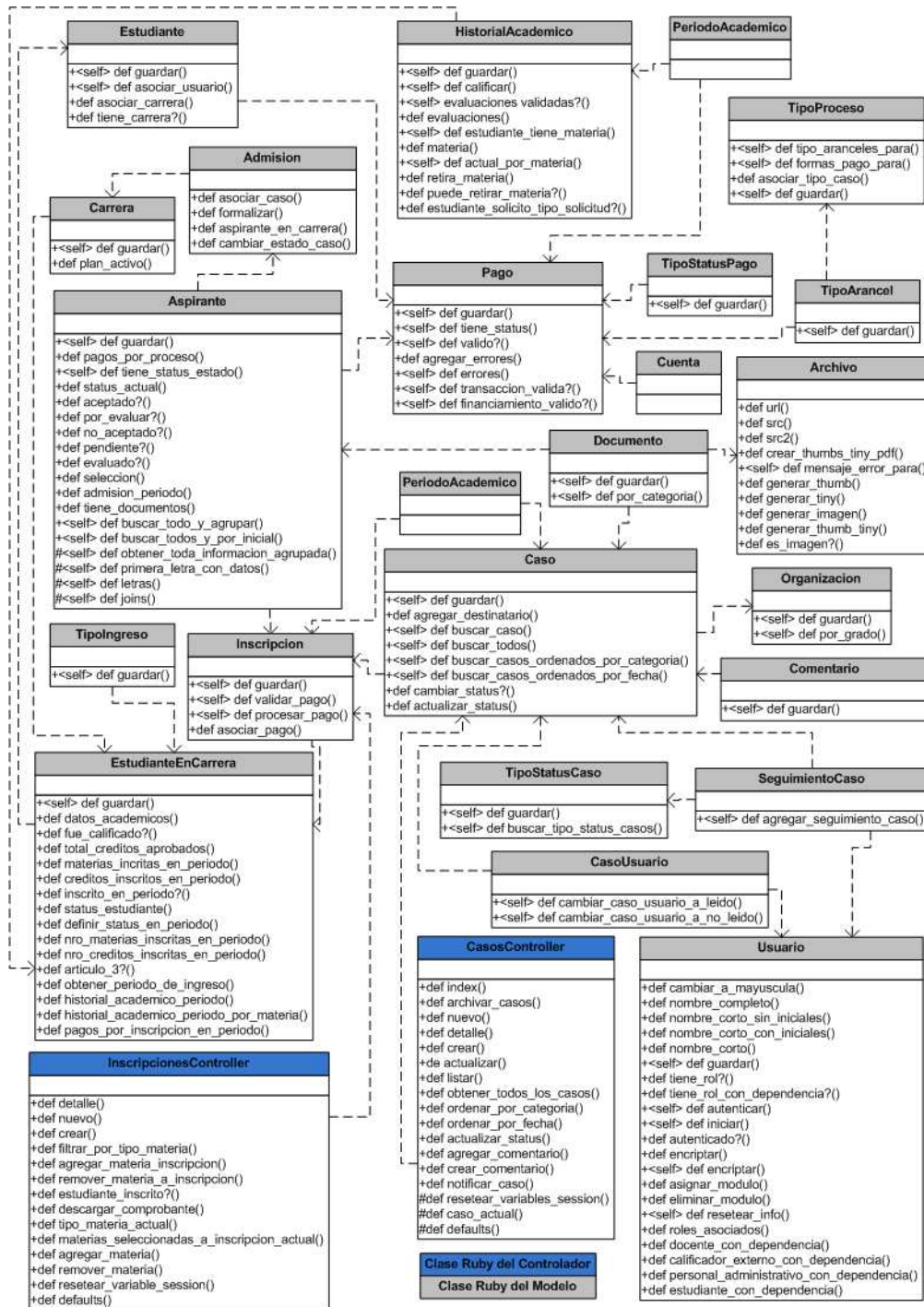


Figura 3.24: Clases de implementación Iteración 9. Módulo Inscripción.

En la figura 3.25 se muestran las clases involucradas en la solución a las historias de usuario del módulo administrativo.

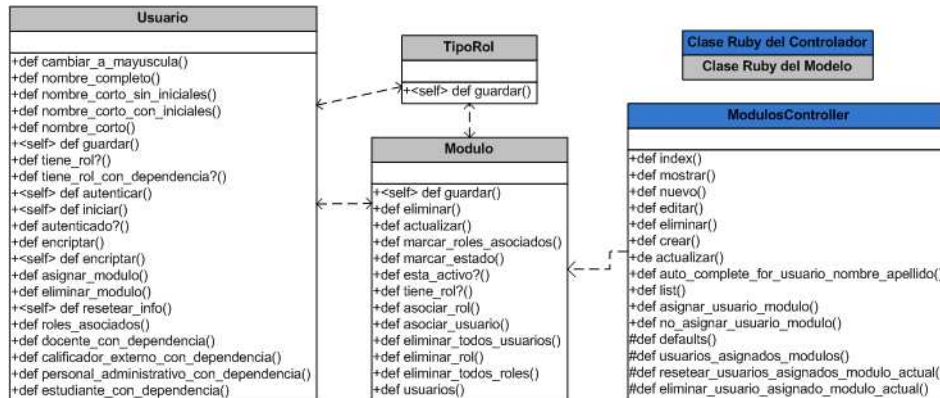


Figura 3.25: Clases de implementación Iteración 9. Módulo Administrativo.

- **Bitácora de desarrollo** La bitácora de desarrollo para todas las iteraciones fueron tomadas en cuenta en conjunto de las historias de usuario, generalizando así en cada iteración lo realizado desde el inicio hasta la culminación de la misma.

Nro	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados	Días realizados
1	Arquitectura	-	10/09/2009	13/09/2009	3	3
2	Análisis y diseño de la implementación	-	13/09/2009	17/09/2009	4	4
3	Desarrollo e integración de componentes	-	17/09/2009	20/09/2009	3	3
4	Desarrollo de la interfaz	-	20/09/2009	23/09/2009	3	3
4	Realización de pruebas	-	23/09/2009	26/09/2009	3	3

- **Codificación** En la siguiente figura se muestra el extracto de código que realizar búsquedas de casos por una categoría asociada.

```

def ordenar_por_categoria
  @casos = Hash.new
  @casos[:data] = Caso.buscar_casos_ordenados_por_categoria(:criterio => params[:criterio],
                                                           :cedula => usuario_actual[:cedula])
  @casos[:categoria] = params[:categoria]
end

```

Figura 3.26: Método por_categoria

En la siguiente figura se muestra el extracto de código que busca los casos nuevos o casos que recientemente acaban de ser actualizados.

```
def notificar_caso
  @resultado = {}
  @resultado[:success] = false

  li = (Time.now-60).to_formatted_s(:db)
  ls = Time.now.to_formatted_s(:db)

  @caso = Caso.find(:first, :conditions => ["usuario_cedula = ? AND
  | fecha_creacion > ? AND
  | fecha_creacion < ?",
  | usuario_actual[:cedula], li, ls])

  unless @caso.nil?
    @resultado[:success] = true
  end
  @resultado
end
```

Figura 3.27: Método notificar_caso

En la siguiente figura se muestra el extracto de código que permite obtener un conjunto de casos y pasarlos al estado de guardado.

```
def index
  @casos = Hash.new
  @casos[:categoria] = "Recibidos"
  @casos[:data] = Caso.buscar_casos(:cedula => usuario_actual[:cedula])
  @casos[:tipo_status_casos] = TipoStatusCaso.buscar_tipo_status_casos(:cedula =>
  | usuario_actual[:cedula])
  resetear_variables_session
end

def archivar_casos
  caso_ids = params[:caso_ids]
  caso_ids.each do |caso_id|
    caso = Caso.find(caso_id)
    categoria = caso[:categoria_type]
    tipo_status_caso = TipoStatusCaso.first(:conditions => ["nombre = ? and aplica_a = ?",
    | "GUARDADO", categoria] )
    caso.update_attribute(:tipo_status_caso_id, tipo_status_caso)
  end
  index
end
```

Figura 3.28: Método archivar_caso

En las siguientes figuras se muestra el extracto de código que obtiene todos aquellos estudiantes que no se inscribieron para el periodo actual, y la creación del caso que notifica al personal administrativo correspondiente.

```

def self.obtener_no_inscritos
  # 'C': Cursante 'N': Nuevo
  @no_inscritos = StatusEstudiantePeriodo.find_by_sql "SELECT * FROM status_estudiante_periodo
  WHERE carrera_id = carrera_actual AND
  ((tipo_status_estudiante_id = 'C' AND
  periodo_academico_id = periodo_academico_anterior) OR
  tipo_status_estudiante_id = 'N') AND estudiante_cedula NOT IN
  SELECT estudiante_cedula FROM inscripcion
  WHERE periodo_academico_id = periodo_academico_actual)"

  tabla = "<table><tr><th width='160px'>Nombre</th>
  <th width='180px'>Cedula</th>
  <th width='60px'>Correo</th></tr>"
  @no_inscritos.each do |est_no_ins|
    no_inscrito = est_no_ins.estudiante_en_carrera.estudiante.usuario
    tabla = tabla+"<tr>
    <td class='codigo' width='155px' align='left'>"+no_inscrito.nombre_corto.titleize+"</td>
    <td width='180px' align='left'>"+no_inscrito.cedula.to_s+"</td>
    <td width='60px' align='center'>"+no_inscrito.correo+"</td></tr>"
  end
  tabla = tabla+"</table>"
end

```

Figura 3.29: Método obtener_no_inscritos

```

def self.asociar_caso(opciones)
  carrera = Carrera.find_by_id(carrera_actual)
  arreglo_destinatarios = []

  if carrera.tipo_grado_carrera[:id].eql?(TipoGradoCarrera::POSTGRADO)
    personal_administrativo = PersonalAdministrativo.first(:conditions =>
      ["organizacion_id = ? AND tipo_rol_id = ?",
      Organizacion::COORDINACION_POSTGRADO,
      TipoRol::OFICINISTA_1], :joins => :personal_administrativo_en_organizaciones)
    arreglo_destinatarios << personal_administrativo[:cedula]
  end

  caso = Caso.guardar({:asunto => "Inscripción",
    :descripcion => "Ha finalizado el proceso de inscripcion para
    el periodo "+opciones[:periodo_academico]+"
    Los siguientes alumnos no se incribieron"+obtener_no_inscritos,
    :organizacion_id => opciones[:organizacion_id],
    :categoria_id => opciones[:periodo_academico],
    :categoria_type => Incripcion.to_s,
    :fecha_creacion => Time.now.to_formatted_s(:db),
    :fecha_actualizacion => Time.now.to_formatted_s(:db),
    :usuario_cedula => opciones[:usuario_cedula]},
    {:categoria => "Inscripcion"})

  if caso[:exito]
    SeguimientoCaso.agregar_seguimiento_caso(:caso_id => caso[:retorno][:id],
      :tipo_status_caso_id => TipoStatusCaso::PENDIENTE_INSCRIPCION,
      :usuario_cedula => opciones[:usuario_cedula])
    caso[:retorno].agregar_destinatario(:arreglo_destinatario => arreglo_destinatarios)
  end
end

```

Figura 3.30: Método Asociar_Caso_Inscripcion

- Pruebas

Las pruebas realizadas para el modulo de revisiones corresponden a inscripciones, notificaciones, guardar u archivar casos, realizar busqueda de casos según una categoría.

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
1	Inscripción	Un usuario con el rol correspondiente para cerrar el proceso de inscripción desactiva el modulo, una vez cerrado el modulo se chequea cuales estudiantes no lograron inscribirse durante dicho periodo.	Que se cree el caso correspondiente dirigido a la secretaria de la Coordinación de Postgrado indicandole cuales estudiantes no se inscribieron durante el proceso de inscripción.	Efectivamente se crea el caso dirigido a la persona esperada.	
2	Notificación	Cada cierto de tiempo dado se realiza una llamada al servidor verificando la existencia de un caso nuevo o de la actualización de algun caso ya existente dirigido a la persona que se encuentre autenticada en sistema para el momento de la llamada.	Tras la creación u actualización de un caso mostrar una notificación indicativa de que se creo u actualizo un caso. Tras la creación u actualización de un caso mostrar una notificación indicativa de que se creo u actualizo un caso.	Efectivamente se logra visualizar la notificación de la actualización o creación del caso.	
3	Guardar u archivar caso.	A los casos se les colocó un checkbox que permite al usuario marcarlo como candidato para pasarlo a estado de guardado de tal manera que no aparezca mas en la bandeja de entrada de casos.	Luego de haber seleccionado uno o un conjunto de casos y al darle al boton Archivar los casos pasan del estado en que se encontraban al estado de guardado y no logran ser visualizados en la bandeja de entrada de los casos.	Efectivamente el caso pasa del estado actual al estado de guardado y no es visualizado en la bandeja de entrada de los casos.	

Nro	Módulo	Descripción del Caso de Prueba	Resultado Esperado	Resultado Obtenido	Motivo de la Falla
4	Mostrar categorías de los casos para la realizar búsquedas según dichos filtros.	Se colocaron filtros en la bandeja de casos para que puedan ser visualizados los casos según una categoría específica.	Una vez presionado el enlace el resultado debe corresponder con el filtro indicado para visualizar los casos.	Efectivamente se obtiene un conjunto de casos asociados a la categoría dada.	
5	Crear metáfora indicando casos con documentos adjuntos.	Se coloca una imagen que indican cuales casos tienen documentos adjuntos.	Que se visualice la imagen solamente en aquellos casos que cuentan con documentos adjuntos.	Efectivamente se visualiza la imagen en los casos que cuentan con documentos adjuntados en el detalle del caso.	

Capítulo 4

Conclusiones

El objetivo del presente Trabajo Especial de Grado es analizar, diseñar y construir un sistema automatizado orientado a la Web, denominado por el grupo de desarrollo en mutuo acuerdo con el cliente como Sistema de Casos, que permita llevar el control de los procesos académicos que se llevan a cabo en la Coordinación de Estudios de Postgrado, logrando mantener un flujo de trabajo de los procesos involucrados de tal manera de mantener informado a los entes involucrados el estado del flujo a través de notificaciones automáticas que informan a los usuarios de los elementos que lo requieren que se le preste atención, contribuyendo en la disminución de los tiempos de respuesta inmersos en la ejecución de los procesos y las actividades relacionadas. Una vez logrado el objetivo general se llega a las siguientes conclusiones:

4.1. Conclusiones

Tras la elaboración del presente Trabajo Especial de Grado, el objetivo general planteado al inicio de la puesta en marcha de este proyecto y planteado en el capítulo I fue alcanzado. Cabe destacar que para el cumplimiento del objetivo planteado fue necesario la constante comunicación por parte del equipo de desarrollo con los integrantes desarrolladores del sistema prototipo, que gestionaría los procesos administrativos que son llevados a cabo entre la Coordinación de Postgrado y los Postgrado que éste coordina.

La implementación de dicho sistema no fue tarea fácil, debido a que de alguna manera dependíamos del sistema prototipo previamente desarrollado y que lleva a cabo la gestión académica de la Coordinación de Postgrado, razón por la cual en muchas ocasiones fue necesario la comunicación con los desarrolladores iniciales para realizar discusiones que logran dar a conocer detalles que llevaron en su fase de análisis para entender el porque de parte de la implementación realizada anteriormente.

La utilización del proceso de desarrollo ágil *Programación Extrema* (XP), permitió que de alguna manera fuese sencilla la implementación del sistema, debido a la práctica que el método de desarrollo utilizado permite, dividiendo problemas complejos en pequeñas partes.

Finalmente la utilización del framework Ruby on Rails y la practicidad con la cual se maneja el lenguaje Ruby, permitieron que la realización del Trabajo Especial de Grado simplificara la solución presentada haciendo uso de las convenciones manejadas por el framework.

4.2. Resultados

Las pruebas funcionales fueron realizadas tras la culminación de cada iteración, conjuntamente con el cliente quién aportó mejoras funcionales para lograr el objetivo general de la aplicación. Como resultados de las pruebas especificadas en cada iteración se logró evidenciar el correcto funcionamiento del sistema de casos.

Además de las pruebas funcionales aplicadas a cada módulo durante la realización y culminación de cada iteración fue realizada una prueba de rendimiento del sistema, el caso de estudio fue el del módulo de solicitudes, específicamente solicitudes estudiantiles donde se ve involucrado el retiro total de semestre.

- Contexto

La herramienta utilizada para la realización de este tipo de pruebas fue *Httpperf*, (<http://www.hpl.hp.com/research/linux/httpperf/>) la cual permite medir el rendimiento del servidor web, en este caso el servidor web utilizado fue Mongrel y la medida utilizada fue milisegundo (ms). A continuación se presenta una breve descripción del caso de estudio seleccionado para realizar la prueba de rendimiento.

Solicitud de Retiro Total del Semestre: Un estudiante que se encuentra en la necesidad de hacer retiro total del semestre, para ello realiza dicha solicitud en el sistema, la solicitud será procesada por el sistema el cual creará un caso a la persona destinada para recibir este tipo de solicitudes la cual procesará una vez que le llegue y percate de la notificación. La prueba de rendimiento consistió en realizar 10 iteraciones de 200 peticiones cada una, de esta manera permitirá evaluar la variación de los datos obtenidos en el tiempo.

- Representación

A continuación se presentan los resultados obtenidos una vez realizada la ejecución anteriormente planteada.

n	Tiempo(ms)	Mínimo(ms)	Máximo(ms)	Mediana(ms)	Desv. Estándar(ms)
1	569,8	446,8	1009,2	544,5	103,8
2	584,6	457,4	952,8	567,5	94,1
3	560,4	450,4	1101,1	545,5	90,2
4	546,9	448,6	987,6	542,5	82,3
5	549,6	461,8	923,9	540,5	80,5
6	560,6	453,4	1010,1	547,5	88,8
7	577	401,5	1066,4	555,5	105,3
8	559,3	425,4	924,1	544,5	80,8
9	553,7	452,5	965,2	539,5	81,4
10	526,7	431,6	716,5	537,5	49,4

Figura 4.1: Resultados de rendimiento Web para el caso de estudio Solicitudes de Retiro Total de Semestre. 100 peticiones por iteración.

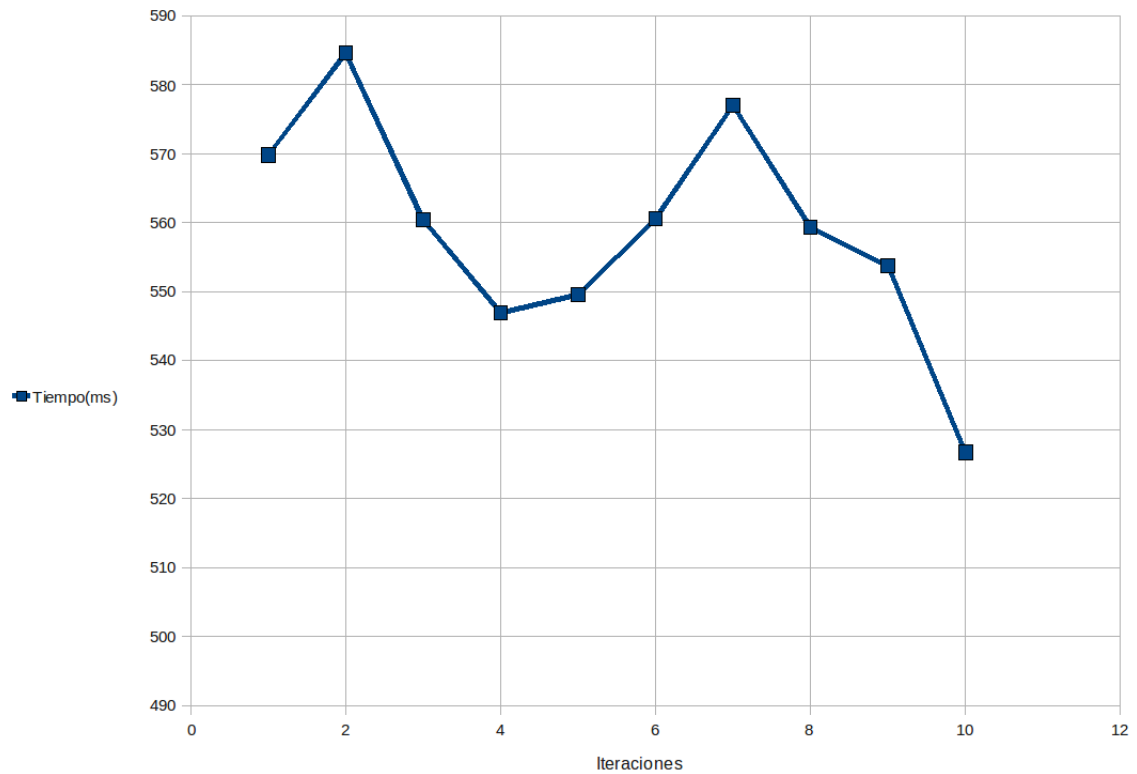


Figura 4.2: Gráfico de resultados de rendimiento Web para el caso de estudio Solicitudes de Retiro Total de Semestre. 100 peticiones por iteración.

- Análisis

Una vez analizado el cuadro y el gráfico, se puede concluir que tras las cantidad de peticiones realizadas al servidor ejecutando un mismo modulo, para el caso de estudio una solicitud de retiro total de semestre, se puede decir que los resultados obtenidos fueron satisfactorios.

4.3. Aportes a la Investigación

Entre los aportes que deja la investigación se tienen los siguientes:

- Se logra completar un primer paso tras la implementación del Sistema de Casos para llevar los procesos académicos de la Coordinación de Postgrados y sus respectivos Postgrados.
- Se deja una base sólida y consolidada la cual será apoyo sustancial para próximos desarrolladores que aporten crecimiento y mejoras al sistema, debido a que fueron seguidos las convenciones que plantea el framework utilizado.

Bibliografía

- [1] Coordinación de Postgrado de la Facultad de Ciencias de la Universidad Central de Venezuela, Manual de Organización, Venezuela, Junio 2005.
- [2] Zambrano, J. *Gestion de los procesos Administrativos de la docencia de la UCV soportado por las TICs. Escuela de Computación. Facultad de Ciencias. Universidad Central de Venezuela. Seminario de la Maestría en Ciencias de la Computación.* Venezuela 2009
- [3] González, Z., *Medición y Visualización de Métricas de Software para Frameworks Web. Caso de Estudio: Ruby on Rails y Frameworks en Java. Escuela de Computación. Facultad de Ciencias. Universidad Central de Venezuela, T.E.G.,* Venezuela 2007.
- [4] Plaza, G. y Ramírez, A., *Desarrollo de un prototipo de aplicación Web para automatizar y optimizar los procesos académicos de la Coordinación de Postgrado de la Facultad de Ciencias de la Universidad Central de Venezuela, T.E.G.,* Venezuela Julio 2009
- [5] Newkirk, J. y Martin, R., *La Programación Extrema en la práctica, Addison-Wesley Iberoamericana Espanya, S.A.,* 2002.
- [6] K. Beck. *Extreme Programming Explained: Embrace Change, Addison-Wesley,* Estados Unidos, Second edition 2005
- [7] Bibeault, B. y Katz, Y., *jQuery in Action, Manning Publications CO. Segunda Edicion.* Estados Unidos, Junio 2009.
- [8] Cueva, J., *Métricas de Usabilidad en la Web España,* 2004.
- [9] Mosberger, D. y Jin, T., *httperf: A Tool for Measuring Web Server Performance* Estados Unidos, Diciembre 1998.
- [10] Haya, M. Franch, X., y Mayo, E., *Uso de los Diagramas de Actividades UML y el Lenguaje i* en el Modelado del Proceso de Implantación del Balanced Scorecard* España, Mayo 2001.

- [11] <http://www.giro.infor.uva.es/oldsite/docproy/wfs/wflowprincipal.htm>, *Flujos de Trabajo*, Dpto. de Informática de la Universidad de Valladolid .