



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

**Prototipo del Sistema de Gestión de Preparadores
de la Escuela de Computación,
módulos Gestión de Retiros y Administrativo**

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
Por las Bachilleres
Aguilar M. Iris H.
Bellet L. Isabelle M.
para optar al título de
Licenciado en Computación

Tutores: Prof. Marcel Castro
Prof. Robinson Rivas

Caracas, Octubre / 2009

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación



ACTA DEL VEREDICTO

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación, para dictaminar sobre el Trabajo Especial de Grado titulado: “Prototipo del Sistema de Gestión de Preparadores de la Escuela de Computación, módulos Gestión de Retiros y Administrativo” y presentado por las Bachilleres Iris Haydeé Aguilar Mora, C.I. V- 15.143.813, e Isabelle Marie Bellet Locker, C.I. V- 15.518.069, para optar al título de Licenciado en Computación, dejan constancia de lo estipulado a continuación:

Leído como fue, dicho trabajo por cada uno de los miembros del Jurado, se fijó el día 29 de Octubre de 2009 a las 9:00 a.m., para que sus autores lo defendieran en forma pública, lo que hicieron en el Centro de Computación de la Facultad de Ciencias, mediante una presentación oral del contenido del Trabajo Especial de Grado, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el Jurado decidió APROBARLO.

En fé de lo cual se levanta la presente Acta, en la Ciudad Universitaria de Caracas a los veintinueve días del mes de octubre del año dos mil nueve.

Firmas del Tutor y Jurados examinadores:

Prof. Marcel Castro
Tutor

Prof. Robinson Rivas
Tutor

Prof. Esmeralda Ramos
Jurado Principal

Prof. Iván Flores
Jurado principal

DEDICATORIA

A Dios Todopoderoso y la Virgen Santísima, por el maravilloso don de la vida y por su incondicional compañía sobre todo en los momentos difíciles.

Mamá y papá, no me equivoco si digo que son los mejores padres del mundo, éste logro se los dedico en agradecimiento por todo su esfuerzo, apoyo y confianza porque siempre, aunque lejos, han estado conmigo. Los quiero mucho.

Iván, más que mi esposo y gran amor, eres mi amigo incondicional, quien me impulsa a seguir luchando por cada uno de mis sueños y sé que para describir todo el apoyo y amor que me das cada día tendría que escribir infinitas líneas, por eso simplemente te digo que éste logro también es tuyo. Te Amo!

Andrés Iván, mi pequeño angelito que con inocencia y ternura llegaste a mi vida colmándola de felicidad y amor, solo te pido perdón por los momentos en que no he podido estar contigo para lograr esta meta. Por todo eso y por las experiencias que hemos vivido y que viviremos juntos te dedico esta tesis.

Carmen, Isabel, Coromoto y Yoleimita, mis hermanas que con su amor y fraternidad complementan parte de mi vida influyendo en el logro de mis metas. Este triunfo va para ustedes.

A **Victoria, Jesús Daniel, Jonathan y Jean Carlos**, mis sobrinitos preciosos con los que he compartido y compartiré muchos momentos agradables llenos de travesuras e inocencia.

Iris Aguilar

Este trabajo de grado que representa un esfuerzo por superarme tanto en mi vida profesional como en la personal, se lo dedico

...a Dios que me da fortaleza espiritual en todo momento

... muy especialmente a mis padres, Gabrielle y Alain, por enseñarme a ser mejor cada día, que hay que esmerarse y sacrificarse, si es necesario, para lograr las metas que nos planteamos. Por su constante apoyo y estímulo

...a la Universidad Central de Venezuela y en especial a la Facultad de Ciencias que me dieron la oportunidad de formar parte de ellas.

Isabelle Bellet

AGRADECIMIENTO

A Dios y la Santísima Virgen, porque me iluminan y acompañan siempre.

A mis padres, porque desde pequeña me enseñaron que el estudio es la mejor herencia que me podían dejar.

A mi esposo Iván, por su amor, apoyo incondicional, constancia y paciencia.

A mi bebé Andrés Iván, quien es el mejor regalo que me ha dado Dios.

A mis hermanas y sobrinitos, por los momentos vividos y los que faltan por compartir.

A mis cuñados, por la ayuda, confianza y el apoyo brindado.

A todos mis **compañeros de estudio** con quienes compartí esta experiencia, y especialmente a quien más que una amiga se ha ido convirtiendo en una hermana, **Isabelle Bellet**, gracias por el apoyo, cariño, comprensión, confianza y todo lo que has hecho porque nuestra amistad crezca cada día más. Mil Gracias!

Al Profesor Marcel, a quien como tutor académico, le agradezco sus enseñanzas, entrega, colaboración y esfuerzo dedicado para que hoy pueda ver realizado uno de mis más grandes sueños; como persona, su comprensión, paciencia, y sobre todo sus buenos y acertados consejos que me ayudaron a crecer tanto personal como profesionalmente.

A todos los profesores, en especial al **Profesor Robinson Rivas**, por sus enseñanzas, apoyo y orientación.

A la U.C.V., especialmente al **personal administrativo** de la Escuela de Computación, por su desinteresada colaboración en el desarrollo de la tesis.

Son muchas las personas especiales a las que me gustaría agradecer su amor, amistad, apoyo, ánimo y compañía en las diferentes etapas de mis estudios. Algunas están aquí conmigo y otras en mis recuerdos y en el corazón. Sin importar en donde estén o si alguna vez llegan a leer estas líneas quiero darles las gracias por formar parte de mí, por las enseñanzas brindadas, experiencias vividas y todas las bendiciones.

Iris Aguilar

A Dios por haberme permitido llegar hasta este punto y haberme dado salud y por iluminarme el camino a seguir.

De igual forma, a mis padres, quienes han sabido formarme con buenos sentimientos, hábitos y valores. Gracias queridos padres.

A mi madre Gabrielle por haberme apoyado en todo momento, por sus consejos, por la motivación constante, pero más que nada, por su amor.

A mi padre Alain por los ejemplos de perseverancia y valores que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mi hermano Eric por acompañarme en esta etapa y para que le sirva de ejemplo y motivación en su vida como estudiante.

Gracias a nuestro tutor Marcel Castro por el apoyo constante que permitió que este trabajo pudiera ser finalizado con éxito, por ayudarnos a mostrar lo mejor de nosotras y por sus consejos más que académicos que crearon enlaces de amistad.

Agradezco a Iris Aguilar por estar mi lado durante estos últimos años de lucha constante, de gratas vivencias, de momentos de éxitos y también de angustias y desesperanza compartidas, y sobre todo por brindarme su amistad.

A todos mis profesores por darme las herramientas para seguir mi camino.

Isabelle Bellet

RESUMEN

El desarrollo del presente trabajo, constituye la construcción de un Prototipo del Sistema de Gestión de Preparadores de la Escuela de Computación, módulos Administrativo y Gestión de Retiros para el mejoramiento de los procesos de dicha gestión, siguiendo las fases de la metodología *OpenUp* produciendo en la primera fase un flujo de trabajo (*Workflow*) que representa las actividades correspondientes a los usuarios actuales. También se llevó a cabo la definición y especificación de los elementos necesarios para la creación de los servicios *Web* que permiten el suministro de los datos para los módulos de Gestión de Retiros y Administrativo empleando la herramienta tecnológica *Ruby on Rails*. Durante la última fase se realizó la simulación de la ejecución de las funcionalidades.

Alcanzando el objetivo principal de este trabajo especial de grado se consiguió como resultado el desarrollo el módulo de Gestión de Retiros, del Sistema de Gestión de preparadores de la Escuela de Computación, en conjunto con el módulo Administrativo del sistema.

Palabras claves: Gestión de preparadores, Gestión por procesos, *Workflow*, servicios *Web*, *OpenUp*, *Ruby on Rails*.

ÍNDICE GENERAL

INTRODUCCIÓN	1
Capítulo 1. Marco Teórico	3
1.1. Gestión por procesos	3
1.2. Workflow	5
1.3. Metodología de desarrollo de software OpenUp	6
1.4. Herramientas de desarrollo	13
1.5. Servicios Web	17
Capítulo 2. Marco Aplicativo.....	22
2.1. Planteamiento del Problema	22
2.2. Antecedentes	23
2.3. Solución Propuesta	24
2.4. Justificación	25
2.5. Objetivo General	26
2.6. Objetivos Específicos.....	26
2.7. Desarrollo de la solución mediante OpenUp.....	26
2.7.1. Fase Inicial	26
2.7.2. Fase de elaboración	37
2.7.3. Fase de construcción	43
2.7.4. Fase de Transición	64
CONCLUSIONES	68
REFERENCIAS.....	70
ANEXOS	72

ÍNDICE DE FIGURAS

Figura 1. El ciclo de Deming	4
Figura 2. El ciclo de vida de OpenUp.....	7
Figura 3. Roles dentro de la metodología OpenUp.....	10
Figura 4. Tareas y artefactos a cargo del rol Analista	10
Figura 5. Tareas y artefactos a cargo del rol Arquitecto.....	11
Figura 6. Tareas y artefactos a cargo del rol Desarrollador	11
Figura 7. Tareas y artefactos a cargo del rol Director del Proyecto	11
Figura 8. Tareas y artefactos a cargo del rol Probador o Tester	12
Figura 9. Arquitectura Básica de Ruby on Rails.....	14
Figura 10. Estructura de una Aplicación Web en Rails	16
Figura 11. Servicio Web.....	17
Figura 12. Petición y respuesta de un servicio Web.	18
Figura 13. Los servicios Web en Funcionamiento.....	18
Figura 14. Interfaz de Servicios Web otros sistemas	19
Figura 15. Estructura de un mensaje SOAP	20
Figura 16. Pila de protocolo de Servicios Web.....	21
Figura 17. Situación actual de la gestión de preparadores de la Escuela de Computación.....	23
Figura 18. Solución Propuesta para la gestión de preparadores de la Escuela de Computación.....	24
Figura 19. Workflow de las actividades para retirar un preparador - Parte I.	30
Figura 20. Workflow de las actividades para retirar un preparador - Parte II.	31
Figura 21. Transición de los estatus de los oficios.....	34
Figura 22. Transición de los estatus de las partidas presupuestarias.....	35
Figura 23. Modelo conceptual del sistema.....	36
Figura 24. Arquitectura General del sistema.....	38
Figura 25. Diagrama de Clases.....	40
Figura 26. Interacción entre los controladores.	41
Figura 27. Lista de preparadores actuales de una materia.	49
Figura 28. Solicitud de un retiro.	49
Figura 29. Lista de solicitudes de trámites.	50
Figura 30. Lista de Oficios por Archivar.	51
Figura 31. Ver Oficios - Lista de solicitudes.	52
Figura 32. Lista de Oficios con Resoluciones.	53
Figura 33. Lista de Oficios listos a ser discutidos en Consejo de Escuela.	54
Figura 34. Completar / Modificar Oficios con Resoluciones.....	55
Figura 35. Lista de oficios por firmar por Director(a).	56
Figura 36. Lista de actividades dado un usuario.....	57
Figura 37. Consultar usuario - Lista de actividades.	58
Figura 38. Consultar usuario - Lista de actividades.	59
Figura 39. Crear o modificar fechas de un periodo.	60
Figura 40. Consultar el log generado por un usuario.	61
Figura 41. Diagrama Interacción entre controladores y servicios Web – Parte I...	62

Figura 42. Diagrama Interacción entre controladores y servicios Web – Parte II..	62
Figura 43. Diagrama Interacción entre controladores y servicios Web – Parte III.	63
Figura 44. Diagrama Interacción entre controladores y servicios Web – Parte IV.	63
Figura 45. Estructura de la aplicación generada usando Ruby on Rails.	64

ÍNDICE DE TABLAS

Tabla 1a. Descripción de los actores del SGP.....	28
Tabla 1b. Descripción de los actores del SGP.....	29
Tabla 2. Especificación del Proceso Retiro.....	32
Tabla 3. Necesidades de los usuarios.	37
Tabla 4a. Plan de Proyecto.....	41
Tabla 4b. Plan de Proyecto.....	42
Tabla 5. Work Item List.	43

ÍNDICE DE ANEXOS

Anexo 1. Plantilla visión del proyecto	73
Anexo 2. Descripción de los símbolos del Workflow	77
Anexo 3. Descripción del servicio Web: crear_oficio_detalle	78
Anexo 4. Descripción del servicio Web: consultar_oficios	79

INTRODUCCIÓN

Para un ejercicio académico cónsono con la misión y visión de la Escuela de Computación de la Universidad Central de Venezuela, los profesores deben contar con personal de apoyo calificado, siendo los auxiliares docentes y preparadores los que permiten afianzar los conocimientos teóricos impartidos mediante clases de práctica.

Los estudiantes que aspiran a un cargo de preparador, deben cumplir con una serie de requisitos que son fundamentales en los procesos de selección y nombramiento, con el fin de garantizar que son el personal indicado para ayudar a otros estudiantes en la comprensión y la puesta en práctica de los conocimientos adquiridos. Finalmente este personal de apoyo pasa por una etapa de retiro voluntario, por culminación de plazo establecido o por alguna razón expresada por el profesor coordinador.

Teniendo presente que es un reto encontrar el personal capacitado que cumpla de manera eficiente sus responsabilidades, y logre satisfacer las necesidades de la Institución, es importante que cada uno de los procesos concernientes a la gestión de preparadores se lleve a cabo siguiendo los reglamentos de la Escuela. Actualmente dichos procesos se ejecutan de forma manual lo que pudiera ocasionar: retardo en los tiempos de respuesta, pérdida de información, error en los datos, falta de información actualizada o histórica, requerida por algunas de las partes involucradas, como los preparadores, personal docente y administrativo.

La Escuela de Computación cuenta con una infraestructura tecnológica capaz de soportar la implantación de un sistema de información, que opere en tiempo real y que sea manejado por los diferentes actores asociados a los procesos de la gestión de preparadores, permitiendo el seguimiento y control de los trámites a través del uso de la Internet, con lo cual se aprovechan las ventajas ofrecidas por las tecnologías de información y comunicaciones (TIC's), logrando así difundir información y facilitar el trabajo interno en la Institución. Es por ello que se consideró pertinente desarrollar un prototipo de un sistema para la gestión de preparadores de la Escuela de Computación (SGP-EC), con miras a satisfacer las necesidades actuales y mejorar los procesos.

El presente Trabajo Especial de Grado (TEG) consistió en el desarrollo de los módulos de Gestión de Retiros de Preparadores y Administrativo del SGP-EC, para controlar de manera efectiva el flujo de información que se genera durante la ejecución de estos procesos. A continuación se indican los capítulos que conforman este documento y que permiten describir el desarrollo de los módulos mencionados:

Capítulo 1. Marco Teórico: Comprende las bases conceptuales que permitieron entender el dominio donde está enmarcado el problema a resolver.

Capítulo 2. Marco Aplicativo: Concentra los aspectos fundamentales para la realización del proyecto de grado. Se presentan el planteamiento del problema, antecedentes, solución propuesta, justificación, objetivo general y objetivos específicos. Adicionalmente, se expone el desarrollo del proyecto siguiendo las fases de la metodología de desarrollo de software *OpenUp*, reflejando en cada una de las fases los artefactos generados para lograr la construcción de los módulos mencionados.

Como parte final se presentan las conclusiones obtenidas y recomendaciones para trabajos futuros.

Capítulo 1. Marco Teórico

En este capítulo se presentan las bases conceptuales sobre Gestión por Procesos, *Workflow*, Metodologías y Herramientas para el Desarrollo de Software (*OpenUp*, *Ruby on Rails*) y *Servicios Web*, consideradas para comprender el contexto en el cual se desarrolló el presente TEG.

1.1. Gestión por procesos

Para introducir el concepto de gestión por procesos, se debe entender la semántica en el contexto empresarial u organizacional de los vocablos “gestión” y “procesos”.

Gestión es sinónimo de administración, dado que ambos términos en su etimología significan acción y efecto de administrar [LD, 2004].

Proceso, según indica la norma ISO 9000, es un “Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados” [Per, 2007]. Las actividades de dicho conjunto se configuran como una secuencia ordenada, repetitiva o cíclica, las cuales generan un producto o resultado cuyo valor es apreciado (estimado) por el cliente o usuario del proceso [Per, 2007].

De lo anterior se infiere que la gestión por procesos consiste en la administración integral de los procesos que se ejecutan en toda organización o empresa, lo cual corresponde con las siguientes definiciones:

- “La gestión por procesos es un esquema que permite organizar los esfuerzos y la utilización de los recursos, para lograr la satisfacción de los entes vinculados a cada uno de los procesos que definen al sistema organizacional o empresarial” [Ben, 2003].
- “Gestión de una organización basada en la comprensión, el análisis, el seguimiento, la medición y la evaluación del desempeño de los procesos y sus interacciones, para conocer cómo se están realizando las actividades y para determinar dónde están las oportunidades de mejora” [Asocalidad].

Por otra parte, la gestión por procesos en las organizaciones es solo un paso ante la consecución del logro de los objetivos que éstas se plantean. Una organización orientada a procesos, que busca obtener beneficios de su ejercicio, requiere del mantenimiento continuo y control de los procesos de negocio, es decir la administración de éstos implica que su ejecución sea eficiente y efectiva [Mue, 2002].

Lo anterior se consigue alineando la gestión de procesos al ciclo de Deming (Ver Figura 1), aplicando las fases que lo componen:

- Planificar (*Plan*): los procesos se identifican, se modelan y se optimizan
- Ejecutar o realizar (*Do*): se implantan los procesos y la estructura de la organización se debe amoldar a los mismos
- Evaluar (*Check*): la efectividad de los procesos implantados en la fase anterior, se debe analizar con base en los datos recopilados y los indicadores de gestión generados por la ejecución y los resultados obtenidos.
- Actuar (*Act*) – se establecen líneas de dirección o toma de decisiones al revisar la evaluación de los procesos, considerando los objetivos estratégicos y operativos de la organización

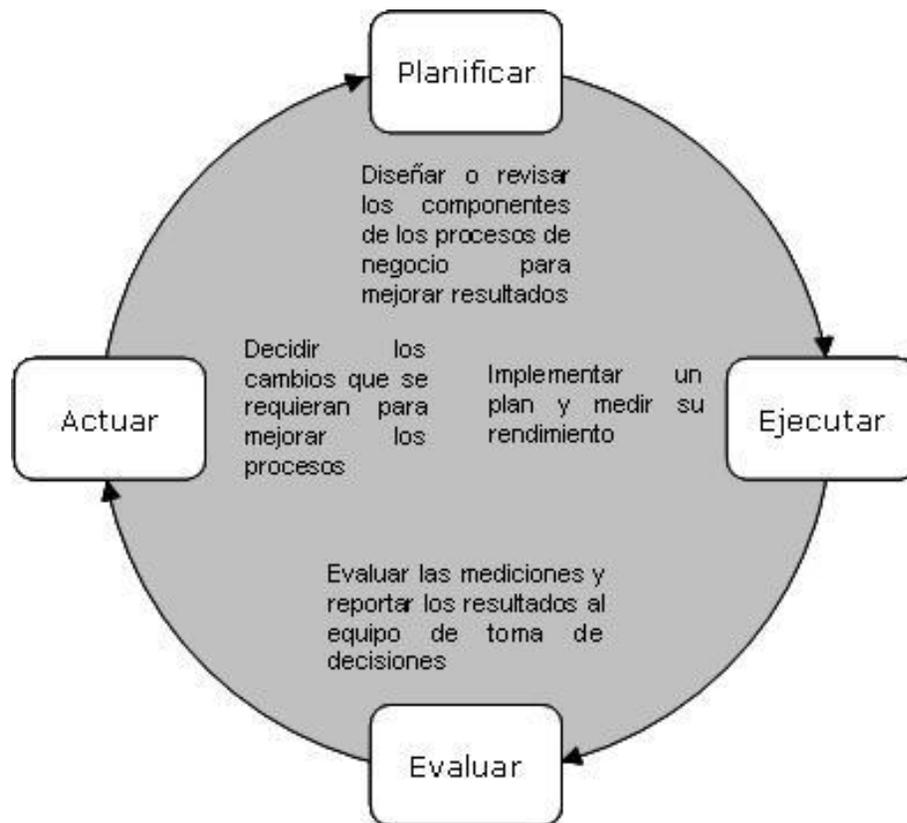


Figura 1. El ciclo de Deming [Mue, 2002].

La importancia de la aplicación de la gestión por procesos dentro de una organización, radica en que permite en una forma estructurada, identificar los actores de cada proceso, conocer sus expectativas, definir objetivos e indicadores (de calidad, costo, productividad, entre otras.), coordinando así las diferentes unidades funcionales que intervienen en los procesos. En este sentido [Ben, 2003] señala que, “Este esquema de gestión requiere que las partes que lo componen se caractericen por crear relaciones coordinadas, para lograr niveles de eficacia y eficiencia en el sistema, que cumplan con los tres elementos básicos de una gestión de calidad: alcanzar los requerimientos de los clientes finales, en los tiempos establecidos y a los costos presupuestados”.

En resumen la gestión por procesos persigue conducir o administrar una organización concentrándose en añadir valor a las entradas y/o salidas de una actividad, con el fin de lograr satisfacer al usuario o cliente final, ya sea interno o externo, buscando generar la calidad necesaria en la gestión empresarial.

1.2. Workflow

Los procesos organizacionales insumen, procesan y generan información necesaria para obtener resultados. Cuando los documentos que soportan esta información son manejados dentro de la organización en un lugar distinto al que los produce, se debe tener en cuenta que éstos pueden ser re-utilizados o sufrir transformaciones en su recorrido antes de llegar a su destino final; y esto es lo que se conoce como flujo de trabajo o *Workflow*. Esta interpretación se basa en [Fer, 2005], al citar a Penadés (1999) cuando éste indica que “en un *Workflow* la información, tareas y documentos pasan de un participante a otro, para que se realicen una serie de acciones de acuerdo con un conjunto de reglas procedimentales”.

Para complementar el concepto anterior de *Workflow*, se presentan las siguientes definiciones:

- “Workflow se refiere al flujo de trabajo a seguir para la consecución de una tarea o trabajo predeterminado. Se define como un sistema de secuencia de tareas de un proceso de negocio. Su definición y control puede ser manual, informatizado o mixto. Organiza y controla tareas, recursos y reglas necesarias para completar el proceso de negocio.” [Pixelware]
- “Una aplicación de *Workflow* automatiza la secuencia de acciones, actividades, o tareas utilizadas para la ejecución del proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo” [Zor, 2007].

- Interpretando lo expuesto en [Kioskea], *Workflow* se refiere a la gestión de procesos de negocios modelada y automatizada, en función de las tareas que deben llevarse a cabo y los roles involucrados.

Las definiciones anteriores hacen diferentes planteamientos que convergen en un mismo punto, y éste consiste en hacer el seguimiento consecutivo a la realización de las tareas, dentro de una organización, mediante el uso de *Workflow*. Existen varias formas de modelar y automatizar el flujo de trabajo, no obstante el objetivo siempre es el mismo: representar y describir las actividades e identificar a los protagonistas involucrados en el desarrollo de los procesos internos de la organización.

Tomando en cuenta lo anterior, la aplicación de un *Workflow* tiene gran importancia, porque busca la automatización parcial o total de los procesos de negocio [Fer, 2005], al generar información útil y necesaria, asociando individuos y grupos de trabajo con tareas estructuradas y no estructuradas, para manejar de manera efectiva los trabajos que se realizan dentro de una organización, haciendo posible la cooperación entre distintos roles.

1.3. Metodología de desarrollo de software *OpenUp*

Dentro de los objetivos de un proceso de desarrollo de *software* está el tratar de cumplir las tareas planteadas de la mejor manera, siendo la calidad del producto el factor principal que se desea lograr. El uso de metodologías apropiadas permite controlar el desarrollo en sus diferentes fases, lo que conlleva a minimizar los errores y optimizar el trabajo en cada iteración.

La metodología *OpenUp* surge de la necesidad de coordinar proyectos pequeños con desarrollos ágiles y a la vez documentados; se incorporan técnicas ágiles ofreciendo como resultado un proceso estructurado, robusto, eficiente, liviano y focalizado en la creación de la documentación justa y necesaria.

OpenUp es un proceso de desarrollo de software de código abierto, diseñado para pequeños equipos organizados quienes quieren tomar una aproximación ágil del desarrollo. *OpenUp* es un proceso iterativo que es mínimo, completo, y extensible. [EPF]

1.3.1. Prácticas a tomar en la metodología

OpenUp está orientado a la realización de reuniones regulares que permiten captar o prevenir posibles riesgos sobre el proyecto y poder así plantear estrategias de mitigación. *OpenUp* recomienda dividir el proyecto en miniproyectos, iteraciones encajadas en tiempo para demostrar valor incremental y obtener retroalimentación temprana y continua.

Muchos elementos necesarios para el desarrollo no se conocen en las fases iniciales, como son las necesidades de los usuarios, los riesgos, comprender las tecnologías del proyecto, o saber cómo trabajar con sus colegas. A medida que se van conociendo estos elementos en su totalidad durante la vida del proyecto se van incorporando, produciendo más iteraciones sobre el mismo. [EPF]

1.3.2. Ciclo de vida del desarrollo de *software*

OpenUp se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición, tal como lo ilustra la Figura 2. Cada fase consiste de una o más iteraciones. El ciclo de vida del desarrollo en *OpenUp* permite evolucionar el producto, durante las diferentes fases en que se elabora el proyecto para obtener retroalimentación continua y mejorada.

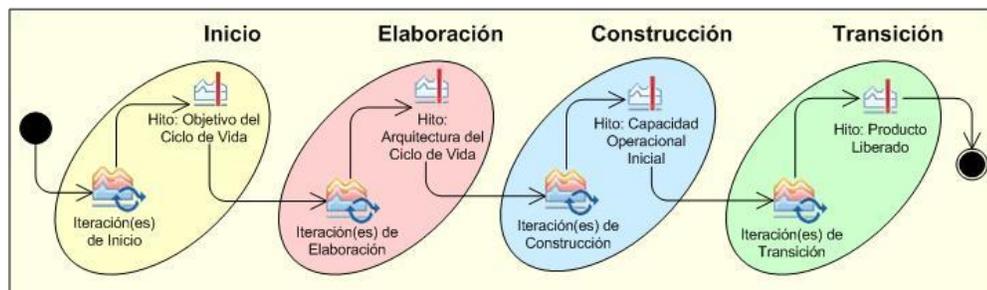


Figura 2. El ciclo de vida de *OpenUp* [EPF].

Fase de Inicio

El propósito en esta fase es lograr comprender el dominio del problema a resolver. Hay cuatro objetivos en esta fase que permiten clarificar el alcance, los objetivos del proyecto y la viabilidad de la solución proyectada:

- Entender lo que se desea construir determinando la visión, el alcance del sistema y sus límites, y se identifica quién está interesado en el sistema y por qué.
- Identificar la funcionalidad clave del sistema, con base en un análisis para conocer cuáles son los requerimientos más críticos.
- Determinar al menos una posible solución. Se identifica al menos una arquitectura candidata y su viabilidad.
- Entender el costo, el plan y los riesgos asociados al proyecto. Para planear el proyecto se provee un mapa a todo el equipo para que conozca la dirección del proyecto. Éste mapa se adapta a los cambios que pueden surgir en el entorno y a la dinámica e interacción que implica el trabajo en equipo. Se evalúan los riesgos,

se determina el tamaño del proyecto y alcance, se definen los objetivos de las iteraciones y se asignan los roles a los miembros del equipo.

Antes de culminar esta fase y pasar a la de elaboración, el equipo de desarrollo de *software* debe responderse de manera afirmativa la siguiente pregunta: ¿Estamos de acuerdo sobre el problema que intentamos resolver? [EPF].

Fase de Elaboración

El propósito de esta fase es crear la arquitectura del sistema y sus componentes. Los objetivos de esta fase son:

- Obtener un entendimiento detallado de los requerimientos que permitan crear un plan más acorde con la realidad del proyecto.
- Diseñar, implementar, validar y establecer la arquitectura.
- Mitigar los riesgos esenciales y producir el plan y la estimación de costos.

Para poder continuar con la fase de construcción luego de esta fase, el equipo de desarrollo de *software* debe responderse de manera afirmativa la siguiente pregunta: ¿Estamos de acuerdo en la vista general de la solución y entendemos los riesgos costos y parámetros de planificación razonablemente bien? [EPF].

Fase de Construcción

El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida en la fase anterior. Sus objetivos son:

- Desarrollar mediante iteraciones un producto completo. Describir los requerimientos que faltan, completar en detalles los diseños, la implementación y la prueba del software.
- Liberar la primera versión operativa del software (beta) del sistema y determinar si los usuarios están listos para que la aplicación sea desplegada.
- Minimizar los costos de desarrollo y alcance, optimizando los recursos.

Al completar esta fase el equipo de desarrollo de *software* debe responderse de manera afirmativa la siguiente pregunta: ¿Estamos de acuerdo en que tenemos un sistema que

cumple con las necesidades claves de los interesados en el desarrollo del *software*? [EPF].

Fase de Transición

El propósito en esta fase es asegurarse que el *software* está listo para su pase a producción. Los objetivos de esta fase son los siguientes:

- La versión beta del *software* permite validar que las expectativas del usuario sean satisfechas, lo cual implica realizar refinamiento del código para depurar errores, y mejorar el desempeño del *software* y su facilidad de uso.
- El desarrollo se considera completo luego de realizar varias pruebas con los usuarios, para poder colocar el producto en producción.
- Optimizar el desempeño en futuros proyectos a través de lecciones aprendidas al documentarlas y mejorar el ambiente de los procesos y las herramientas para el proyecto.

Para efectuar una transición efectiva, el equipo de desarrollo de *software* debe responderse de manera afirmativa la siguiente pregunta: ¿Estamos de acuerdo en que podemos entregar el sistema y finalizar el proyecto? [EPF].

Roles dentro de la metodología *OpenUp*

Se establecen diferentes roles para el desarrollo del proyecto, cada uno de ellos tiene bien establecidas sus actividades a ejecutar. Estos son: Analista, Arquitecto, Desarrollador, Director del Proyecto, Parte Interesada o *Stakeholder*, Probador o *Tester*, (Ver Figura 3) [EPF].



Figura 3. Roles dentro de la metodología *OpenUp* [EPF].

A continuación se describen los roles mencionados anteriormente, según las definiciones de [EPF]:

Analista: Se encarga del levantamiento de información, al interactuar directamente con los clientes y/o usuarios finales involucrados en el proyecto, que permitirá entender el problema a resolver. Su principal responsabilidad consiste en identificar y especificar los requerimientos, capturar y asignar prioridades a los mismos, así como elaborar la visión del proyecto. En la Figura 4 se muestran las tareas y artefactos asociados a este rol.



Figura 4. Tareas y artefactos a cargo del rol Analista [EPF].

Arquitecto: Responsable de diseñar la arquitectura del software, lo cual incluye tomar las decisiones técnicas claves que restringen al diseño total y la implementación del proyecto. La Figura 5 ilustra los artefactos y tareas pertinentes a este rol.



Figura 5. Tareas y artefactos a cargo del rol Arquitecto [EPF].

Desarrollador: Responsable de desarrollar una parte del sistema, incluyendo su diseño para que se ajuste a la arquitectura. También se encarga de la implementación, realizar las pruebas unitarias e integrar los componentes que son parte de la solución. Las tareas y artefactos relacionados al Desarrollador se pueden observar en la Figura 6.



Figura 6. Tareas y artefactos a cargo del rol Desarrollador [EPF].

Director del Proyecto: Lidera la planificación del proyecto, coordina las interacciones con los interesados en el desarrollo del *software*, y mantiene al equipo de desarrollo enfocado en alcanzar los objetivos del proyecto. Las actividades y los artefactos a cargo del Director del Proyecto se ven reflejados en la Figura 7.



Figura 7. Tareas y artefactos a cargo del rol Director del Proyecto [EPF].

Parte interesada o Stakeholder: Representa a grupos de interés cuyas necesidades deben ser satisfechas por el proyecto. Este rol puede ser desempeñado por cualquiera que esté (o potencialmente estará) materialmente afectado por el resultado del proyecto.

Probador o Tester: Responsable de las actividades principales del esfuerzo de las pruebas. Estas actividades incluyen identificar, definir, implementar y dirigir las pruebas necesarias, como también verificar los resultados de las pruebas y analizar los resultados. La Figura 8 muestra los artefactos y tareas que debe llevar a cabo quien desempeñe este rol.



Figura 8. Tareas y artefactos a cargo del rol Probador o Tester [EPF].

Principales artefactos de la metodología *OpenUp*

La metodología cuenta con una serie de plantillas que sirven como modelos para presentar y especificar a los artefactos necesarios en cada una de las fases. Entre los artefactos más representativos se tiene:

- **Documento de Visión:** Este artefacto provee una visión completa del sistema describiendo los requerimientos en un alto nivel y contemplando las restricciones que pudieran existir.
- **Plan de Proyecto:** Este documento informa los objetivos de las iteraciones y sus hitos e indica cómo será organizado el proyecto y quién desempeñará cada rol.
- **Lista de elementos de trabajo o *Work Item List*:** Su objetivo principal es registrar las solicitudes de trabajo que potencialmente deberán realizarse en el marco del proyecto, para que el mismo pueda ser priorizado, se estime el esfuerzo y se controle el progreso.
- **Casos de Uso:** Este artefacto captura la secuencia de acciones que el sistema realiza, brindando un resultado observable a los usuarios que interactúan con él.

- **Plan de Iteración:** Su función es informar los objetivos, las asignaciones de trabajo y los criterios de evaluación para la iteración permitiendo así, evaluar el avance del proyecto.
- **Lista de Riesgos:** Contiene un lista de riesgos conocidos con su estado para realizar un seguimiento de los mismos.
- **Casos de Prueba:** Proveen una forma de capturar las entradas, condiciones y resultados esperados del sistema, así como también, identificar sistemáticamente aspectos del software a probar. Otro objetivo de este artefacto es especificar si los resultados esperados fueron alcanzados basados en la verificación de los requerimientos del sistema.

1.4. Herramientas de desarrollo

Las herramientas de desarrollo utilizadas en el presente TEG son el lenguaje de programación *Ruby* y el marco de trabajo (o *framework*) *Ruby on Rails*, también conocido como *RoR*.

Ruby es un lenguaje de programación interpretado y orientado a objetos, desarrollado en 1993 por Yukihiro Matsumoto y liberado al público en 1995 bajo una licencia de *software* libre [Wil, 2007]. Se caracteriza por poseer una sintaxis bastante clara que facilita la programación, ya que combina de una forma exitosa “la elegancia conceptual de *Smalltalk*, la facilidad de uso y de aprendizaje de *Python* y el pragmatismo de *Perl*” [Hib, 2005].

Aun cuando *Ruby* es orientado a objeto, posee la versatilidad para desarrollar programas bajo el enfoque procedimental o funcional, lo que le permite ser un metalenguaje con el cual se pueden construir lenguajes de dominio específico [FM, 2008], es por ello que han surgido diversos proyectos en ese sentido, siendo *RoR* el más popular de todos [TH, 2006].

RoR es un *framework*, construido con *Ruby*, para el desarrollo de aplicaciones *Web*, cuyo diseño busca facilitar la programación de este tipo de aplicaciones [*Rails*]. Entre las características más relevantes de este *framework* se encuentran [TH, 2006]:

- **Arquitectura MVC:** La estructura de *Rails* está basada en el patrón de diseño Modelo-Vista-Controlador o MVC, por lo que las aplicaciones *Web* elaboradas bajo *RoR* siguen dicho patrón, obteniendo como beneficio independencia entre la lógica del negocio (Modelo), la interfaz del usuario (Vista) y el control de las acciones internas (Controlador), reduciendo así el acoplamiento entre la capa del modelo

del negocio y la de presentación. Esto favorece las labores de depuración y mantenimiento de este tipo de aplicaciones. La Figura 9 ilustra cómo una aplicación *Web* implementada en *RoR* atiende una petición. La petición es suministrada a la aplicación, en el servidor *Web*, mediante el navegador, luego es tomada por el elemento *Dispatcher* quien se encargará de redireccionar la petición al controlador indicado (interacción Vista-Controlador). Determinada la acción a realizar, el *Action Controller* (Controlador) envía hacia el *Active Record* las solicitudes encargadas del procesamiento de datos (interacción Controlador-Modelo), poniéndose en marcha las operaciones de manipulación de datos descritas en la capa Modelo, siendo estas: Creación, Actualización, Eliminación o Búsqueda; operaciones típicas de cualquier sistema interactivo. Luego de realizarse el procesamiento en la Base de Datos, los resultados son enviados nuevamente al *Active Record* quien los devuelve al *Action Controller* para que este envíe la respuesta final (interacción Controlador-Vista), ya sea a través de un correo, haciendo uso del *Action Mailer*, o a través de una vista haciendo uso del *Action View*.

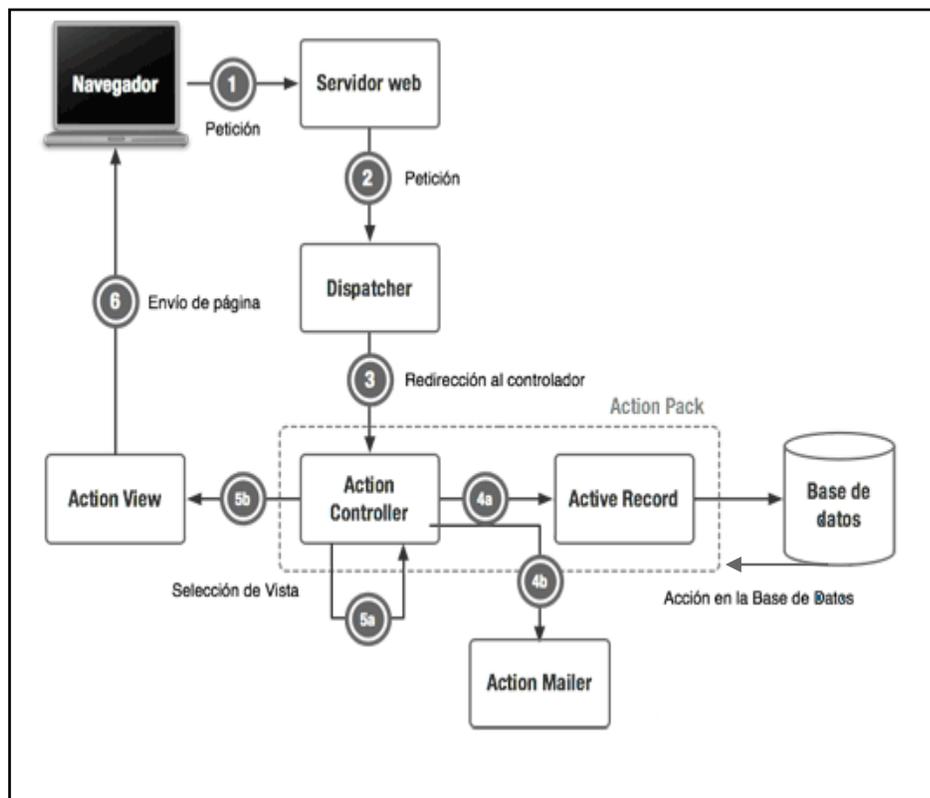


Figura 9. Arquitectura Básica de *Ruby on Rails* [DummyonRails, a].

- **Meta-programación:** Al ser un *framework* elaborado con *Ruby*, hereda de éste su versatilidad permitiendo aplicar técnicas automáticas de generación de códigos de extensión para la aplicación, sin intervención directa del programador.
- **Manejo automático entre elementos de tablas y objetos:** En el manejo del acceso a Bases de Datos Relacionales (BDR), *RoR* introduce la tecnología *Active Record*, la cual se basa en meta-programación para reconocer las columnas de las tablas de una BDR, e incluirlas como atributos en el dominio de objetos de la aplicación.
- **Convención versus Configuración:** Otros *frameworks* para el desarrollo de aplicaciones *Web* obligan al programador a construir archivos de configuración para el mantenimiento operativo de la aplicación, lo cual no aplica en *Rails* ya que provee una notación propia, que sirve de convención, que si es seguida por el desarrollador se pre-establecen los parámetros de configuración de forma transparente.

Rails organiza las aplicaciones creando una estructura fija de carpetas (Ver Figura 10), con el fin de distribuir las funcionalidades en dos áreas principales [Dummyon*Rails*, b]:

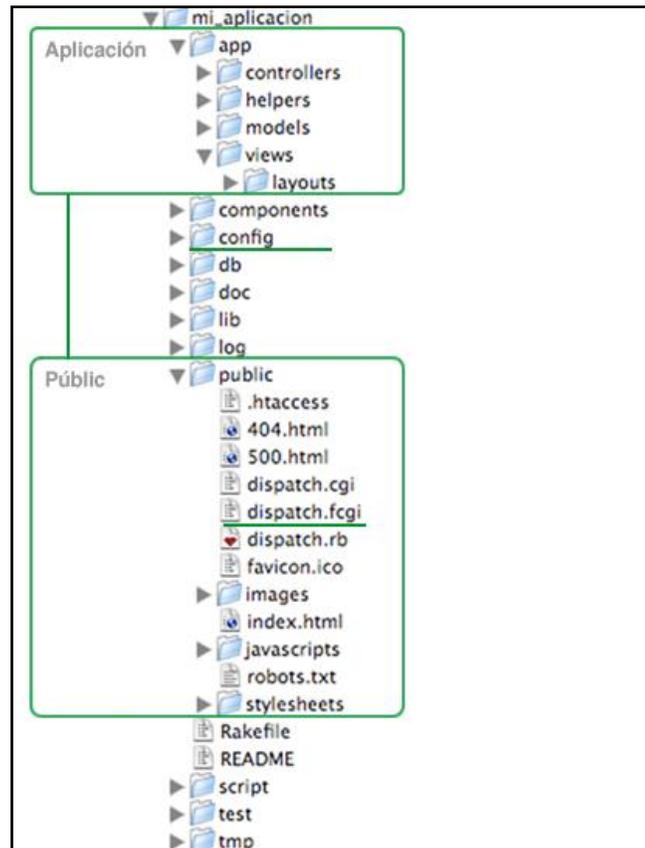


Figura 10. Estructura de una Aplicación Web en Rails [Dummyon Rails, b].

- **Directorio *Public*:** Se aloja en esta carpeta todos los elementos públicos de la aplicación. Por defecto, en ella se incluyen imágenes, hojas de estilo CSS, *Javascript*, etc. También se encuentran en esta carpeta los archivos que representan a los *dispatchers*, (*dispatch.cgi*, *dispatch.fcgi* y *dispatch.rb*), quienes se encargan de re-direccionar las peticiones de los usuarios desde el navegador hacia la aplicación para que los *controllers* se encarguen de procesarlas.
- **Directorio *App*:** Se colocan en esta carpeta los archivos que implementan la codificación de la aplicación siguiendo el patrón MVC, distribuidos en las carpetas que le correspondan de acuerdo al patrón señalado anteriormente (directorios *Models*, *Controllers*, *Views* y *Helpers*).

Adicionalmente a las carpetas mencionadas anteriormente, cabe destacar que en la estructura de directorios de una aplicación Web construida con *RoR*, también se encuentra el directorio *Config*, en el cual se mantienen los archivos de configuración de

las bases de datos y del enrutamiento de las peticiones (archivos que re-direccionan las peticiones cargadas al navegador hacia los controladores).

En conclusión, *Ruby on Rails* es un *framework* que permite de forma sencilla el desarrollo de una aplicación *Web*, es fácil de comprender y utilizar, no requiere de grandes conocimientos para implementar una aplicación, sólo se requieren nociones básicas sobre la programación orientada a objetos y en la medida en que se va interactuando se irán conociendo las bondades que puede ofrecer *Ruby on Rails*. Como un buen *framework*, *Rails* permite trabajar con un orden en donde se percibe la separación de las capas del patrón MVC. *Rails* es flexible, ofrece una estructura ordenada que permite repartir las funcionalidades en diferentes áreas dentro de la estructura ofrecida.

1.5. Servicios *Web*

Existen varias definiciones sobre lo que son los servicios *Web*, entre éstas cabe mencionar la presentada por [Dia]: “Un servicio *Web* es una aplicación de *software* identificada mediante una URI, cuyo interfaz (y uso) es capaz de ser definido, descrito y descubierto mediante artefactos *XML*, y soportar interacciones directas con otras aplicaciones *software* usando mensajes basados en *XML* y protocolos basados en Internet”.

Según la interpretación dada al concepto expuesto en [STK, 2002], un servicio *Web* permite tener acceso al código de una aplicación usando el estándar de Internet (Ver Figura 11). Si una aplicación puede ser alcanzada a través de una red y usando protocolos como *HTTP*, *XML*, *SMTP* entonces ésta es un servicio *Web*.

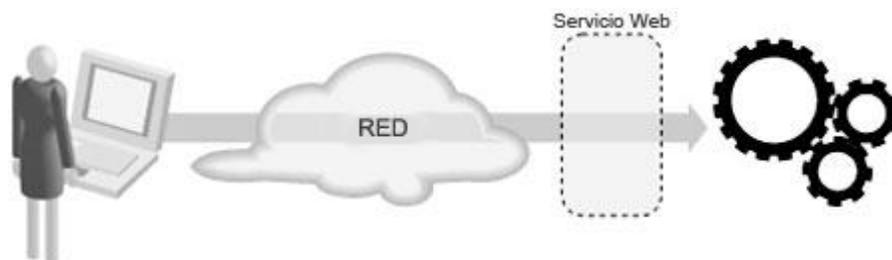


Figura 11. Servicio *Web* [STK, 2002].

Siguiendo con la idea que presentan [STK, 2002], los servicios *Web* son sistemas de mensajería que deben ser capaces de enviar y recibir mensajes usando alguna combinación de los protocolos estándares de Internet. La implementación más común de servicios *Web*, es a través de llamadas a procedimientos que se encuentran disponibles en un servidor. La Figura 12 ilustra el intercambio de los mensajes, el cual se realiza

haciendo una petición a un subprograma, que es el servicio *Web*, con los argumentos necesarios y luego se recibe la respuesta a dicha llamada a través de un mensaje *XML*.



Figura 12. Petición y respuesta de un servicio *Web*. [Cer, 2002].

Es importante recalcar, que este software puede ser alcanzado en Internet por otras aplicaciones. De esta forma, se pueden desarrollar aplicaciones que hagan uso de otras aplicaciones que estén disponibles en Internet interactuando con ellas como se observa en la Figura 13.

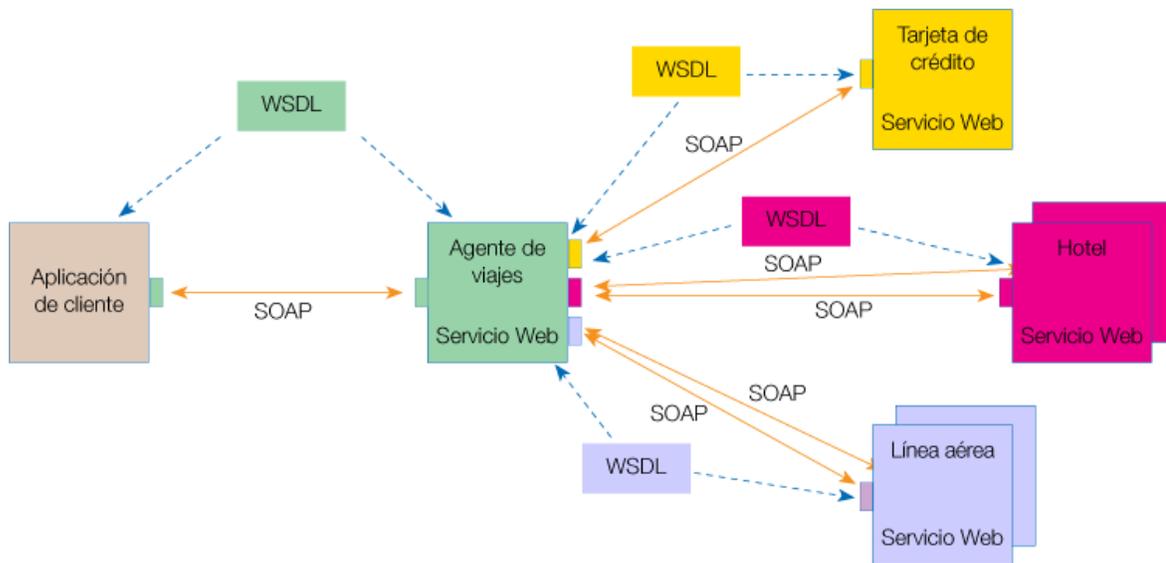


Figura 13. Los servicios *Web* en Funcionamiento [W3C].

Tal como señala [Cer, 2002], otra característica esencial que define a un servicio *Web*, es que éste no se encuentra atado a ningún sistema operativo ni lenguaje de programación. En la Figura 14 se observa esta característica.

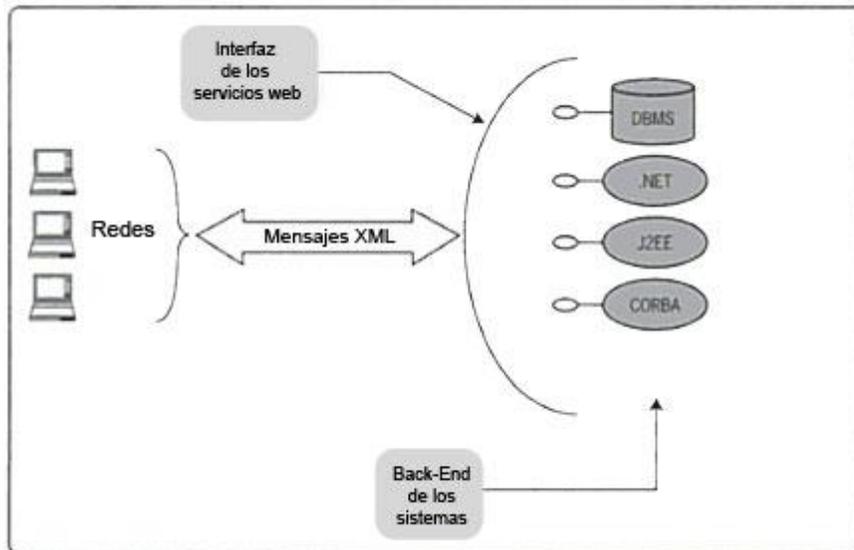


Figura 14. Interfaz de Servicios Web otros sistemas [New, 2002].

1.5.1. Estándares para el desarrollo de servicios Web

[Dia] expresa que las tecnologías necesarias para el funcionamiento de un servicio *Web* son:

- **Formato WSDL** (*Web Service Description Language*).
- **Protocolo SOAP** (*Simple Object Access Protocol*).
- **Protocolo de transporte que por lo general es HTTP**, que se encarga de transportar los mensajes por Internet.

A continuación, se describen algunas de estas tecnologías:

SOAP

[Dia] plantea la siguiente definición: “*SOAP* es el acrónimo de *Simple Object Access Protocol*, es decir, protocolo simple de acceso a objetos. *SOAP* es un protocolo ligero de mensajes *XML* que se usa para codificar la información de los mensajes de petición y respuesta de los servicios *Web* que se envían a través de una red. Los mensajes *SOAP* son independientes de los sistemas operativos y de los protocolos, y pueden ser transportados usando una variedad de protocolos de Internet, incluyendo *SMTP*, y *HTTP*”. En la Figura 15 se ve la estructura del protocolo *SOAP*.

XML es básico para la mayoría de los formatos actuales de empaquetamientos de servicios *Web*, siendo *SOAP* uno de más utilizados construido en *XML*.

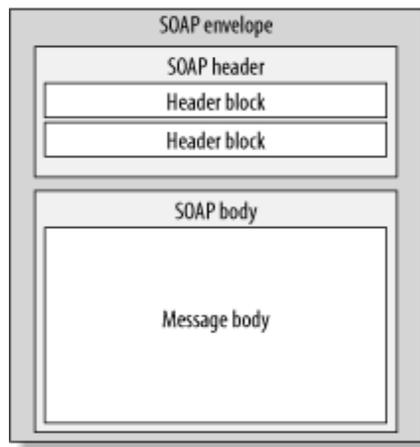


Figura 15. Estructura de un mensaje SOAP [Dia].

WSDL

WSDL es el acrónimo de Lenguaje de Descripción de Servicios *Web* (*Web Services Description Language*), que es un lenguaje *XML* usado para describir la interfaz de un servicio *Web* como un conjunto de métodos capaces de intercambiar mensajes. *WSDL* se considera parte integral de *UDDI*, que es un directorio de servicios *Web* que permite que se listen, busquen y descubran este tipo de software [Dia].

Por último, para que un servicio *Web* pueda ser llevado alrededor de la red por la capa de transporte, debe ser "empaquetado" para que las partes involucradas en el intercambio de mensajes puedan entenderlo. Este proceso también es llamado "serialización". La Figura 16 muestra la pila del protocolo de un servicio *Web*, que está conformada por las tecnologías antes descritas.

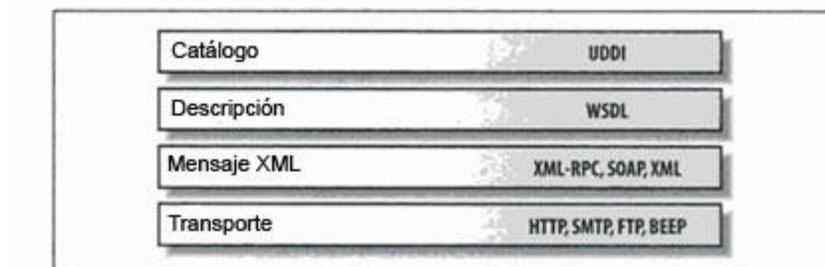


Figura 16. Pila de protocolo de Servicios *Web*. [Cer, 2002].

Capítulo 2. Marco Aplicativo

Éste capítulo comprende el planteamiento del problema, solución propuesta, objetivos, descripción de los actores y requerimientos del producto, así como también la presentación del desarrollo del SGP-EC, específicamente la elaboración de los módulos de Gestión de retiros y administrativo, mediante la aplicación de las fases de la metodología *OpenUp* y el uso de las herramientas tecnológicas.

2.1. Planteamiento del Problema

En la actualidad, la ejecución de los procesos en la Escuela de Computación se realiza de forma manual, lo que pudiera ocasionar, en algunos casos, pérdida de los documentos que se generan durante el desarrollo de cada uno de los procesos inherentes a la gestión de preparadores. Por otra parte, la información tanto histórica como actual, probablemente no se encuentra disponible de forma inmediata al momento de ser solicitada por algunas de las partes involucradas (preparadores, personal docente y administrativo).

Este problema afecta a todos los actores que participan en los procesos de gestión de preparadores y el impacto ocasionado se presenta seguidamente:

- Los resultados asociados a cada trámite depende del tiempo que dedica cada actor a realizar las tareas que le corresponden, por tanto resulta difícil estimar cuánto es el tiempo total en que se pueda atender un proceso de la gestión de preparadores.
- No existe disponibilidad inmediata de la información sobre los trámites.
- La generación de documentos depende de la búsqueda de información dado que esta se realiza manualmente.
- En algunos casos, pudiera ocurrir pérdida de documentos asociados a cada uno de los procesos, dado que el suministro o traspaso de documentos de un actor a otro es en físico.

Lo anteriormente expuesto se ilustra en la Figura 17.

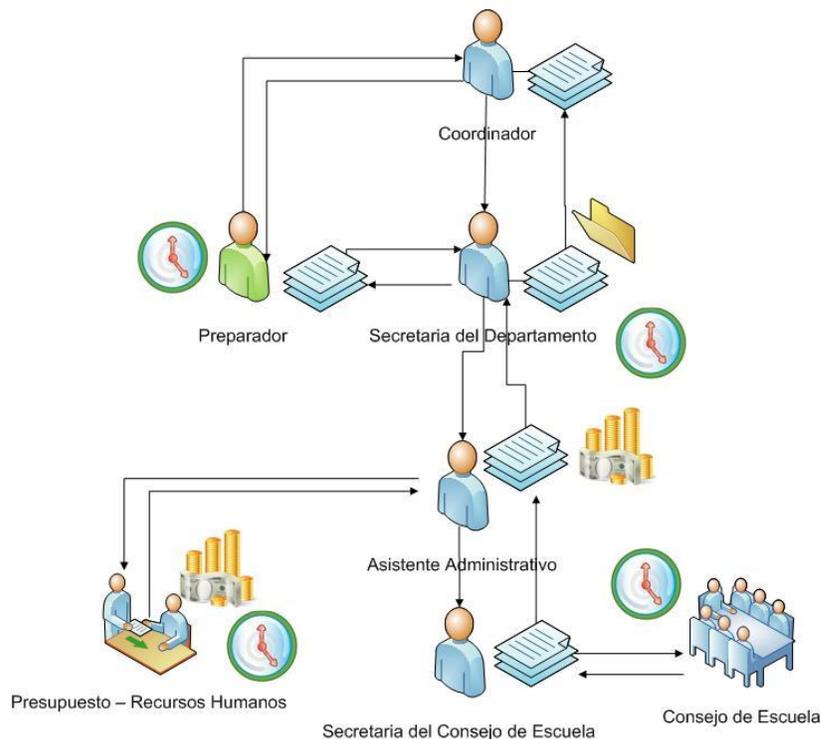


Figura 17. Situación actual de la gestión de preparadores de la Escuela de Computación.

En consecuencia surge la siguiente inquietud:

¿Cómo mejorar los procesos asociados a la gestión de preparadores en la Escuela de Computación, manteniendo el flujo de trabajo actual?

Sería ideal contar con una herramienta que permita gestionar los procesos de forma automatizada sin modificar el flujo de trabajo actual, agilizando los trámites y por ende, minimizando los tiempos de respuesta.

2.2. Antecedentes

Indagando sobre trabajos previos asociados a la gestión de los preparadores en la Facultad de Ciencias, y en específico en la Escuela de Computación, no se encontraron referencias que den indicios de proyectos de sistemas de información que permitan mejorar los procesos de la gestión de preparadores, por lo que el presente TEG se considera pionero en este ámbito, constituyendo un reto importante que implica un análisis y diseño a profundidad desde el estudio de la situación actual, hasta la elaboración del prototipo y específicamente de los módulos requeridos.

2.3. Solución Propuesta

Partiendo de la situación actual sobre la gestión de preparadores existente en la Escuela de Computación y tomando en cuenta el análisis realizado por [AB, 2009], se propuso el desarrollo de un prototipo del sistema de gestión de preparadores, centrándose éste TEG en los módulos de gestión de retiros y la administración del sistema.

La Figura 18 plantea gráficamente la solución propuesta.

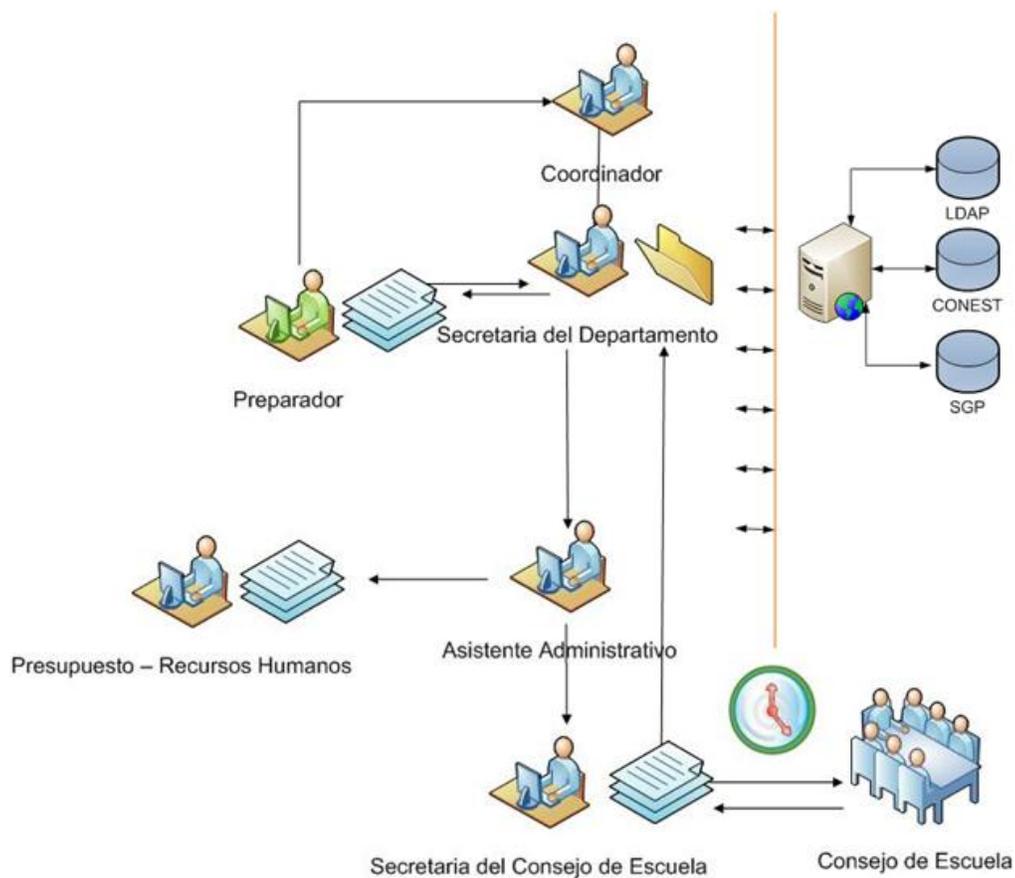


Figura 18. Solución Propuesta para la gestión de preparadores de la Escuela de Computación.

Seguidamente se presenta el beneficio que dicho prototipo dará a los actores involucrados en el proceso:

- Los estudiantes recibirán notificaciones cuando se inicie la gestión de un concurso.

- Los preparadores serán informados vía correo electrónico sobre los resultados de los trámites concernientes al proceso retiro.
- Los coordinadores, director(a) y jefe del departamento podrán tener mejor control y seguimiento de los trámites asociados a la gestión de retiros.
- El personal administrativo podrá optimizar sus labores al minimizar las actividades manuales de la gestión de preparadores.
- La persona encargada de la administración del sistema podrá asignar las tareas a cada uno de los actores asociados a los procesos de gestión de preparadores.

Es importante señalar que los usuarios potenciales del SGP-EC son los actores antes mencionados.

Como parte de la solución propuesta se contempla la descripción de los servicios *Web* porque no se cuenta con el acceso directo a las bases de datos. Sin embargo, se realizó una simulación de acceso a los datos para poder mostrar los resultados a través de la aplicación, dado que estos servicios *Web* no estaban disponibles.

Cabe destacar, que si se logra la puesta en producción del SGP-EC, la capacitación y la concientización de su uso a los usuarios, la Escuela de Computación se beneficiaría de la siguiente manera:

- Se mejoraría el control y la supervisión de los documentos.
- Se contaría con la disponibilidad de la información actualizada en cualquier momento.
- Se facilitaría la generación de constancias a los preparadores y otros documentos como planillas de solicitud de retiro, entre otras.
- Se evitarían posibles errores en datos debido a que éstos provienen directamente de su fuente.
- Se mejoraría la comunicación entre los actores de los procesos.
- Se minimizarían las actividades de manera manual.
- Se estima una mejoría en los tiempos de respuesta en la ejecución de los procesos, porque a través de las notificaciones vía correo electrónico, y de las vistas de consultas de datos de la aplicación, se mantendrán informados y actualizados sobre los estados por los cuales está pasando cualquier trámite que sea de su interés.

2.4. Justificación

La implantación de un sistema de gestión de preparadores en la Escuela de Computación constituye una solución fundamental para lograr que dicha gestión sea más efectiva, al minimizar factores de riesgo tales como costos y tiempo.

Se estima que la contribución que el sistema hará a la Escuela de Computación, será ayudar en el control y seguimiento de los procesos asociados a la gestión de

preparadores, garantizando la efectividad organizacional y el cumplimiento de las normativas y procedimientos establecidos para la gestión de dichos procesos con calidad y excelencia.

2.5. Objetivo General

- Desarrollar los módulos de Gestión de Retiros y Administrativo del sistema de gestión de preparadores de la Escuela de Computación.

2.6. Objetivos Específicos

- Aplicar la metodología *OpenUp* en el desarrollo de los módulos de Gestión de Retiros y Administrativo.
- Ajustar el *Workflow* del proceso de gestión de retiros, para afinar detalles y definir las funcionalidades del módulo Administrativo.
- Definir y especificar los servicios *Web*, que permitirán el suministro de los datos necesarios en los módulos de Gestión de Retiros y Administrativo.
- Construir los módulos de Gestión de Retiros y Administrativo con la herramienta *Ruby on Rails*.
- Desarrollar funcionalidades y vistas comunes para los módulos que conforman al Sistema de Gestión de Preparadores.
- Realizar las pruebas de funcionalidad de los módulos de Gestión de Retiros y Administrativo.

2.7. Desarrollo de la solución mediante *OpenUp*

Tras comparaciones con otras prácticas convencionales de desarrollo de software, se consideró viable la aplicación de la metodología *OpenUp*, porque esta metodología ofrece un manejo apropiado y completo de la gestión de las diferentes áreas del software y del ciclo de vida.

A continuación, se explicarán los pasos que se realizaron en cada fase de la metodología *OpenUp* para el desarrollo de la solución propuesta.

2.7.1. Fase Inicial

En esta fase se definió lo que se iba a construir y se aportaron soluciones previamente contempladas por [AB, 2009].

También se siguió la plantilla correspondiente al artefacto visión del proyecto (ver Anexo 1), respondiendo a las preguntas que ésta plantea. Existen algunos apartados de la dicha plantilla que ya han sido expuestos anteriormente en el presente Capítulo tales como la

“descripción del problema y solución propuesta” que fue presentada en los puntos 2.1 y 2.3 respectivamente, y el objetivo general seguido de los objetivos específicos en los puntos 2.5 y 2.6.

Además se evaluaron las posibles restricciones existentes para el desarrollo del proyecto, sabiendo que se trataba de un proyecto ambicioso que requería del funcionamiento de varios subsistemas con los que tendría que interactuar, tales como el proveedor de los servicios *Web*, que a su vez necesita una base de datos real para llevar a cabo las pruebas.

Debido a que el desarrollo del SGP-EC y sub-sistemas dependientes fue en paralelo, surgió la necesidad de interactuar con los integrantes de otros equipos de desarrollo, realizando reuniones para fijar parámetros de comunicación y de intercambio de datos entre micro-proyectos y equipos de trabajo. Se acordó crear documentos en línea compartidos, en los cuales cada uno de los integrantes pudieron plasmar sus ideas y aportes a la solución integral del proyecto. Se habla de un proyecto integral conformado por varios proyectos desarrollados por diversos equipos de trabajo, encargados de cada uno de los siguientes subsistemas:

- Manejo presupuestario a nivel de la Facultad.
- Mantenimiento e interacción con una base de datos central.
- Gestión de los procesos de concurso y nombramiento de preparadores.
- Gestión de retiros de preparadores, manejo de trámites de los procesos de la gestión de preparadores, así como un módulo administrativo para el sistema.

En cada iteración, realizada semanalmente, se evaluaron los resultados, para poder hacer una proyección de las tareas a realizar en la siguiente iteración, y así poder determinar el éxito o fracaso de la iteración actual, de esta forma se mejoró el proceso de desarrollo y el refinamiento del alcance del proyecto y su duración, causando la restructuración del plan de trabajo.

La identificación de los actores se realizó siguiendo el formato de la plantilla visión del proyecto, asociando los roles de la metodología *OpenUp* a cada uno de esos actores. Los roles de analista, probador, desarrollador y arquitecto fueron asumidos por los integrantes del equipo de proyecto. Los tutores ocuparon el rol de gerente de proyecto y en algunos casos de *stakeholder*. En las Tablas 1a y 1b se hace la descripción y especificación de las responsabilidades dentro del sistema de cada uno de los *stakeholders*.

Nombre	Descripción	Responsabilidades
Director(a)	Es la persona encargada de regir y avalar los procesos académicos y administrativos de la Escuela de Computación, entre estos la gestión de preparadores.	Dar seguimiento a todos los trámites que se generen en la escuela.
Secretaria de Consejo de Escuela	Persona encargada de elaborar las minutas y actas relacionadas con las resoluciones tomadas en las sesiones del Consejo de Escuela. Debe informar y recibir información de los entes relacionados con alguna de sus funciones.	Realiza actividades referentes a los trámites como generar minuta y actualizar los oficios con sus resoluciones de Consejo de Escuela.
Secretaria de Departamento	Es la persona que además de realizar las funciones propias de apoyo al departamento de la Escuela, se encarga de los trámites administrativos de la gestión de preparadores, entre ellos elaborar el oficio, crear y mantener los expedientes de los preparadores actualizados.	Completar y verificar los oficios generados por el sistema. Generar constancia de preparador.
Coordinador	Docente universitario encargado de la coordinación de una materia, cuyas principales funciones relacionadas con la gestión de preparadores son: apertura de concurso para la selección de preparador, aplicación de la evaluación a los aspirantes al cargo de preparador, selección del ganador del concurso, generación del acta con los datos del ganador y el documento con las notas de los concursantes.	Solicitar retiro de preparadores.

Tabla 1a. Descripción de los actores del SGP.

Nombre	Descripción	Responsabilidades
Jefe de Departamento	Docente universitario encargado de solicitar las partidas necesarias, consultar las partidas disponibles, realizar la programación docente, recibir información de los coordinadores relacionada con los procesos de gestión de preparadores.	Dar seguimiento a los trámites generados por el sistema. Notificar la apertura de un concurso.
Preparador	Son estudiantes de pregrado regulares y calificados, que apoyan en el desarrollo de labores docentes, de investigación, administrativas, servicios, deportes, apoyo técnico o cualquier otra equivalente, con una retribución por preparadores.	No aplica.

Tabla 1b. Descripción de los actores del SGP.

Dado que las actividades que conforman el proceso retiro fueron planteadas por [AB, 2009], basándose en un *workflow*, se decidió continuar con el refinamiento del modelado de la secuencia de acciones de igual forma y no como lo propone la metodología *OpenUp* a través de los casos de uso, debido a que se logran los mismos resultados. En el Anexo 2 se describen los elementos que contiene el *workflow* lo que permitirá su mayor comprensión.

Otra actividad que se llevó a cabo en esta fase fue la verificación y refinamiento del flujo de trabajo realizado por [AB, 2009] junto con las partes interesadas, con el fin de definir las funcionalidades a desarrollar. Una vez que el *workflow* fue aceptado y certificado por los usuarios, sólo se realizaron actualizaciones sobre algunas actividades. El *workflow* que se presenta en las Figuras 19 y 20 comprende el flujo de las actividades que conforman la gestión de retiro, desde el punto de vista de la automatización de dicho proceso, comprendiendo consultas y roles asociados a las tareas que se realizan actualmente en combinación con las funcionalidades del sistema a desarrollar.

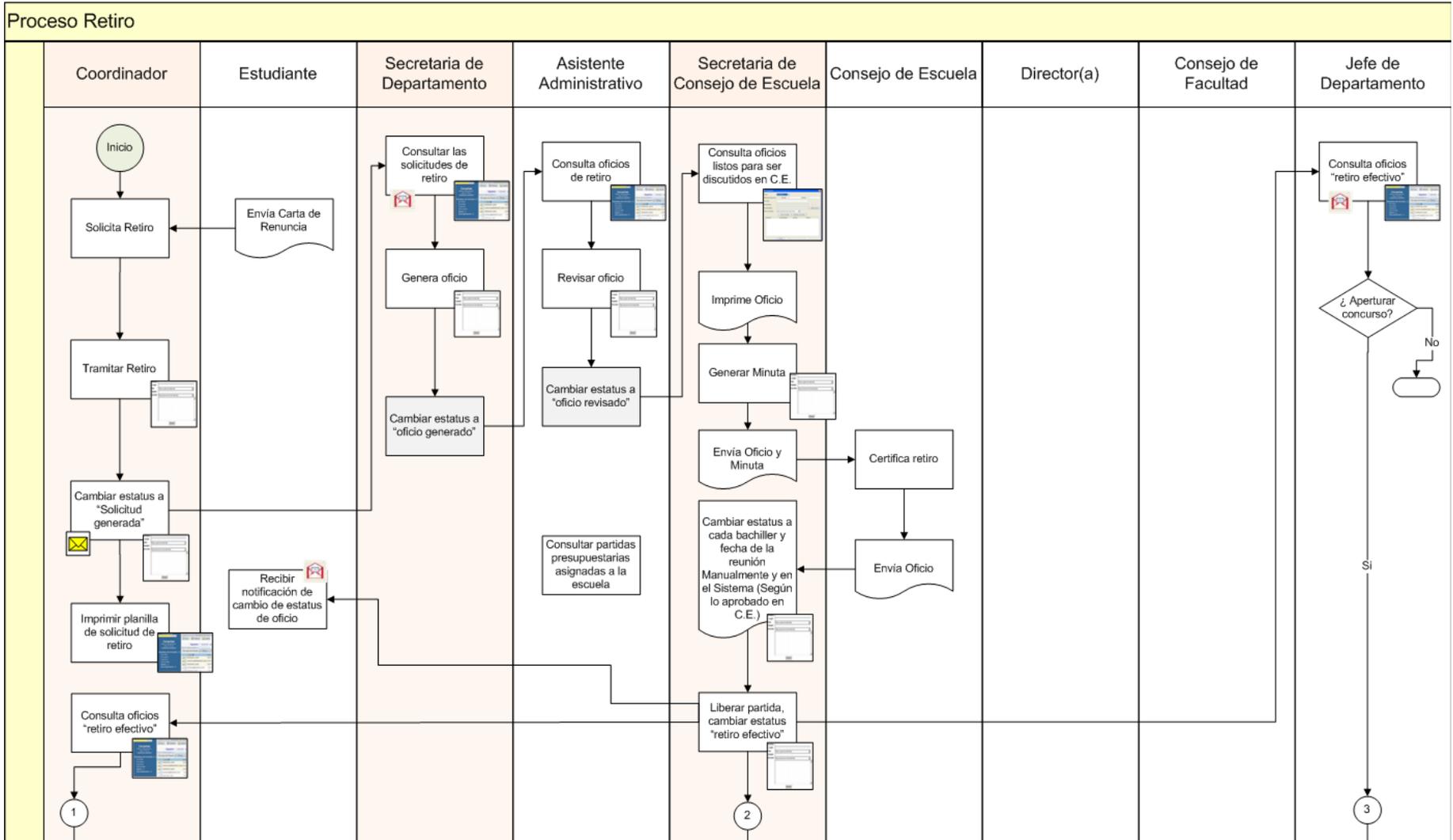


Figura 19. Workflow de las actividades para retirar un preparador - Parte I.

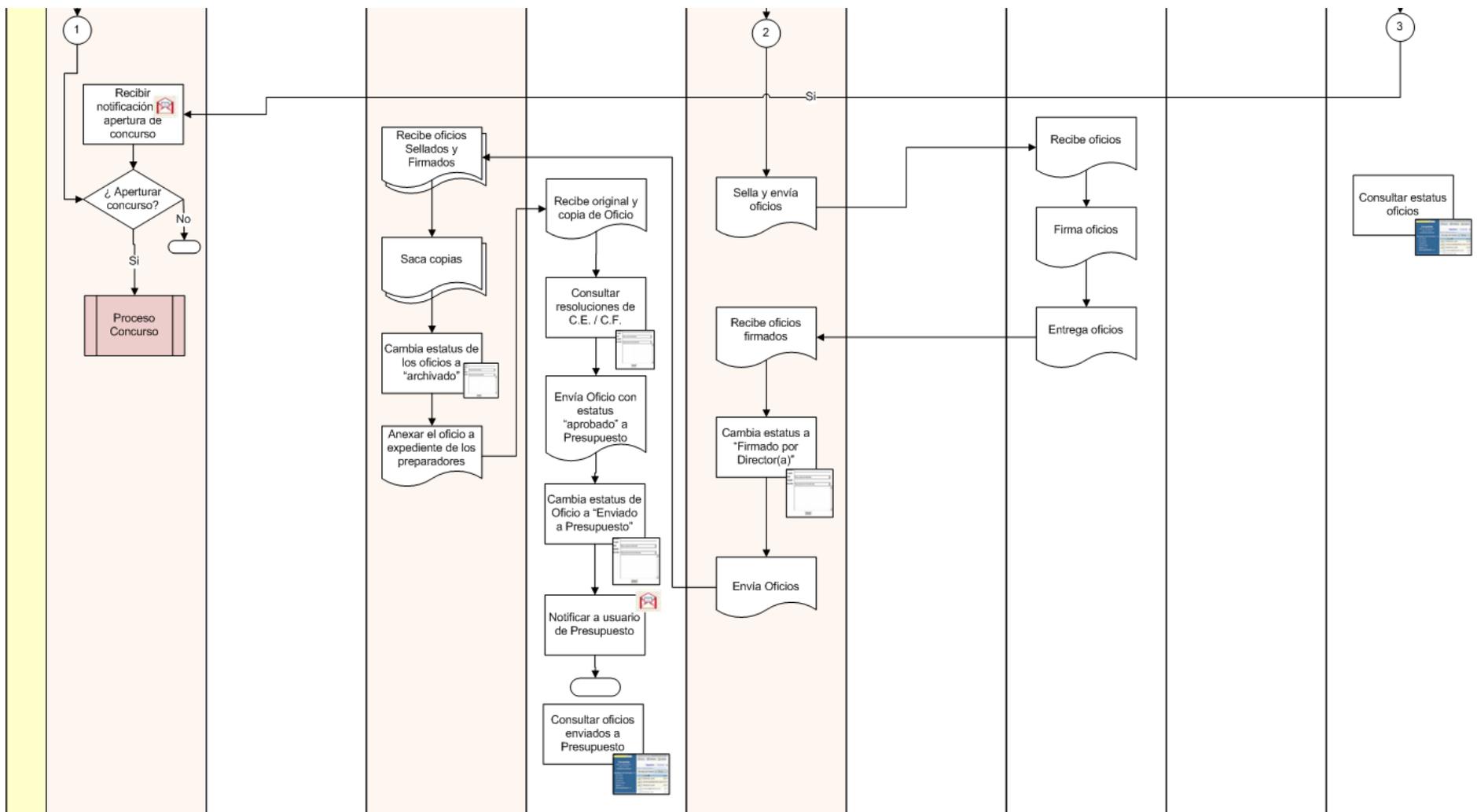


Figura 20. *Workflow* de las actividades para retirar un preparador - Parte II.

La Tabla 2 presenta la especificación del proceso retiro, incluyendo los actores involucrados en este proceso, los flujos principales y alternos, las pre-condiciones y post-condiciones.

Proceso
Retiro
Descripción
Consiste en liberar la partida presupuestaria del preparador que desee retirarse o al cual se le rescinde el cargo de sus actividades.
Actores involucrados
Jefe Departamento Asistente Administrativa Coordinador Estudiante Secretaria del Departamento Secretaria de Consejo de Escuela Directora
Pre-condiciones
Visualizar los preparadores actuales. El estudiante que se desea retirar debe ser preparador actual. El coordinador debe solicitar el retiro de un preparador. En caso que el estudiante sea el que solicite el retiro, debe entregar la carta de renuncia.
Post-condiciones
Se libera la partida presupuestaria asociada al preparador retirado. Generación del oficio de retiro por parte de la Secretaria del Departamento: el oficio es un documento que indica los datos del estudiante que se retira. El retiro se hace efectivo luego de su aprobación por Consejo de Escuela. Según el criterio del Jefe del Departamento, luego de realizarse el retiro se procede a abrir un concurso.
Flujos alternos
En el caso que un preparador fue nombrado “Adhonoram” el proceso de retiro se efectúa de la misma forma que cualquier otro caso, a excepción de que no se libera la partida presupuestaria.

Tabla 2. Especificación del Proceso Retiro.

Cabe destacar que en la gestión de preparadores es necesario el manejo de oficios, los cuales son documentos físicos generados a través del sistema propuesto, pasando por una serie de etapas gestionadas por diferentes usuarios. Estas etapas se le asociaron un conjunto de estatus que serán detallados a continuación en orden lógico de ejecución, y se pueden ver gráficamente en la Figura 21:

- **Oficio nuevo:** Es el primer estatus que puede tener un oficio, se refiere al momento de la creación de éste por primera vez.
- **Oficio generado:** Se refiere al momento en que el oficio pasa a ser revisado y completado por la Secretaria del Departamento.
- **Oficio revisado:** corresponde al visto bueno por parte de la Asistente Administrativo para que el trámite del oficio siga su curso.
- **Oficio Aprobado, Oficio Negado, Oficio Diferido:** Son los estatus que va a tener un oficio luego de su discusión en Consejo de Escuela, estos estatus son asignados por la Secretaria del Consejo de Escuela.
- **Firmado por Director(a):** Denota el estatus de un oficio luego de ser firmado por el(la) director(a).
- **Oficio Archivado:** Indica el estatus de un oficio una vez que es archivado por la Secretaria del Departamento.
- **Enviado a Presupuesto:** Este estatus indica el último paso por el que pasa un oficio de preparadores dentro de la Escuela. Denota el envío a otra unidad administrativa, en este caso, al Departamento de Presupuesto donde se continuará con el trámite respectivo.

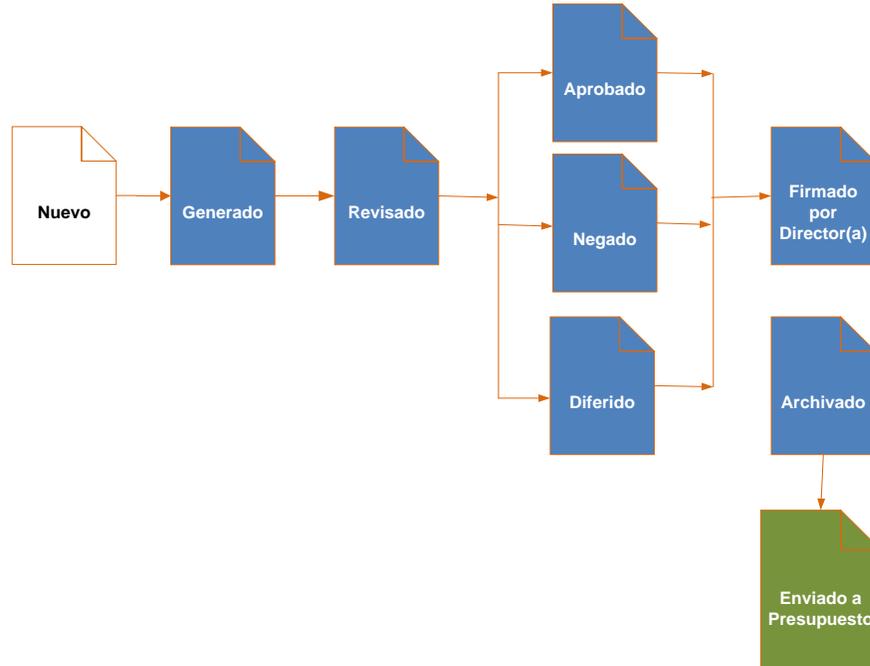


Figura 21. Transición de los estatus de los oficios.

Otro elemento importante son las partidas presupuestarias, las cuales también tienen establecidos estatus, que se definen a continuación:

- **Disponible:** Una partida que se encuentre en este estatus indica que ésta puede ser asignada a cualquier preparador.
- **No Disponible:** Indica que la partida está siendo utilizada es decir está asignada a un preparador, por lo tanto nadie puede ocuparla, hasta que esta sea liberada.
- **Pre-asignable:** Cuando un coordinador de materia solicita el retiro de uno de sus preparadores, el estatus de la partida que tiene asociado ese preparador pasa automáticamente al estatus pre-asignable, es decir, que está próxima a ser liberada, pero por el momento puede ser asignada a otro preparador, con la finalidad de adelantar los trámites de nombramientos y evitar la espera de que el retiro se haga efectivo.
- **No Pre-asignable:** Cuando una partida se encuentra en estatus pre-asignable y es asignada a un preparador, pasa a no pre-asignable. Este estatus es virtual o transitorio, porque una vez aprobado el trámite por Consejo de Escuela la partida pasa a no disponible.

Las posibles transiciones de estatus que puede tener una partida presupuestaria se ven reflejadas en la Figura 22.

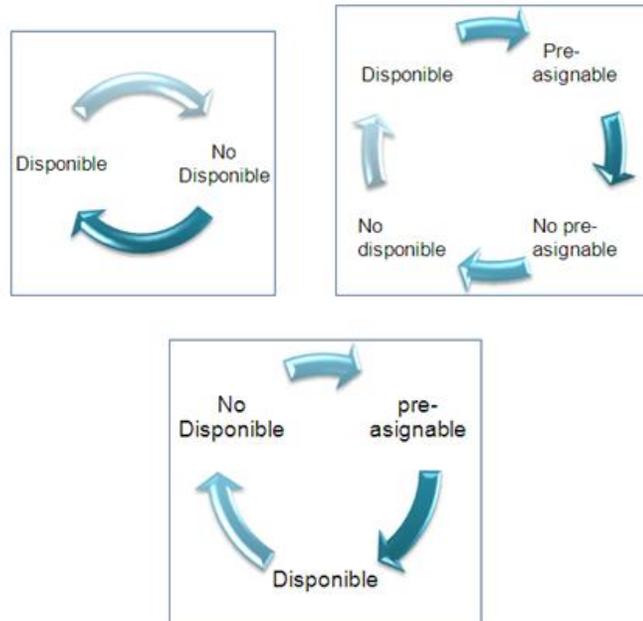


Figura 22. Transición de los estatus de las partidas presupuestarias.

Para responder a las preguntas que propone la plantilla visión del proyecto, en cuanto al “ambiente del usuario”, cada rol descrito en el *workflow* puede ser desempeñado por más de una persona, sin embargo en el caso de la Escuela de Computación un rol es asumido por una sola persona. El tiempo invertido en cada actividad depende de lo que tarda cada actor, por lo esto puede conducir a la pérdida de tiempo y retraso en los procesos.

Existen algunas restricciones tales como la dependencia de la base de datos y de otros sistemas para el funcionamiento de SGP-EC.

Para mejorar la situación actual reflejada en el *workflow* presentado en las Figuras 19 y 20, se propone que la plataforma de la aplicación SGP sea en un entorno *Web*, para facilitar la generación de notificaciones en cada uno de los pasos de los trámites y la interacción con otros sub-sistemas. Esta plataforma esta configurada bajo un ambiente operativo de software libre a nivel de los servidores de aplicaciones y base de datos, la cual está ubicado en el Centro de Computación de la Facultad de Ciencias.

Como se expone en el planteamiento del problema, la gestión de preparadores se realiza de forma manual utilizando herramientas ofimáticas para la creación de los trámites, los datos necesarios son suministrados indirectamente por control de estudios y presupuesto. Por tanto, la solución planteada es sustituir las herramientas ofimáticas por una aplicación *Web* que interactúe y se integre con CONEST (sistema que automatiza la gestión

académica de la Facultad de Ciencias de la Universidad Central de Venezuela) y el sistema de Gestión Presupuestaria.

El modelo conceptual del sistema se muestra en la Figura 23, donde se observa que el SGP está conformado por los módulos de Retiro, Administrativo, Concurso y Nombramiento, que a su vez interactúan con sistemas externos como lo son el Sistema de Gestión Presupuestaria y el Sistema de Manejo de Bases de Datos y Servicios *Web*. Los procesos correspondientes a los módulos Concurso y Nombramiento fueron analizados por [GR, 2009].

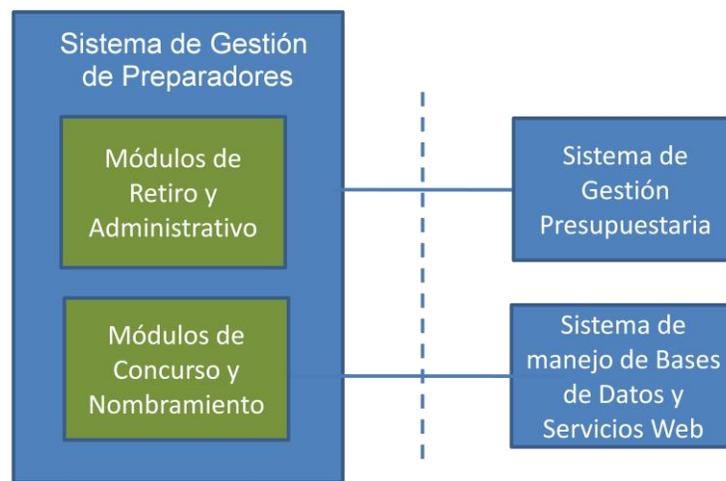


Figura 23. Modelo conceptual del sistema.

El modelo conceptual presentado se limita sólo a lo concerniente a la Escuela de Computación, siendo ajeno al desarrollo de otros sub-sistemas de suma importancia, tales como el sistema que implementa los servicios *Web* para el acceso a las bases de datos y los correspondientes a otras áreas, como el departamento de recursos humanos, departamento de presupuesto y control de estudios.

Siguiendo con la plantilla visión del proyecto se realiza la “descripción del producto” mediante la Tabla 3 que indica las necesidades de los usuarios definiendo las prioridades para el desarrollo y características de los requerimientos.

Necesidad de los usuarios	Prioridad	Características
Mejorar el seguimiento de los trámites	Alta	La gestión de los procesos necesita dar seguimiento a los trámites porque son la base de dicha gestión y así realizar cualquier observación a tiempo.
Manejo automático de las partidas presupuestarias.	Alta	Dado que la escuela cuenta con partidas de preparadores y docentes se hace difícil el manejo de las mismas.
Disponer de información actualizada en cualquier momento.	Alta	Los departamentos y actores vinculados a los procesos de gestión de preparadores están interesados en conocer el estado de los trámites y de las partidas presupuestarias.
Contar con una herramienta que ayude en la generación de constancias, minuta complementaria, oficios.	Media	Actualmente los documentos se generan manualmente pudiendo ocasionar, en algunos casos, errores en los datos.
Recibir notificaciones inmediatas para seguir el curso de los trámites en los procesos de gestión de preparadores	Alta	Con el fin de agilizar los trámites y mejorar los tiempos de respuesta.

Tabla 3. Necesidades de los usuarios.

2.7.2. Fase de elaboración

En la fase de elaboración se definió la arquitectura del sistema. Se obtuvo un entendimiento con mayor nivel de detalle de los requerimientos para poder traducir cada uno de estos en una solución automatizada, y por consiguiente crear un plan de trabajo más preciso.

En esta fase se define cómo es la arquitectura que resuelve los requerimientos, además se analizan las restricciones, así como funcionalidades comunes reflejadas en los requerimientos que puedan ser reutilizadas por los módulos de Gestión de Retiros, Administrativo, Concurso y Nombramiento. En la Figura 24 se muestra gráficamente esta arquitectura.

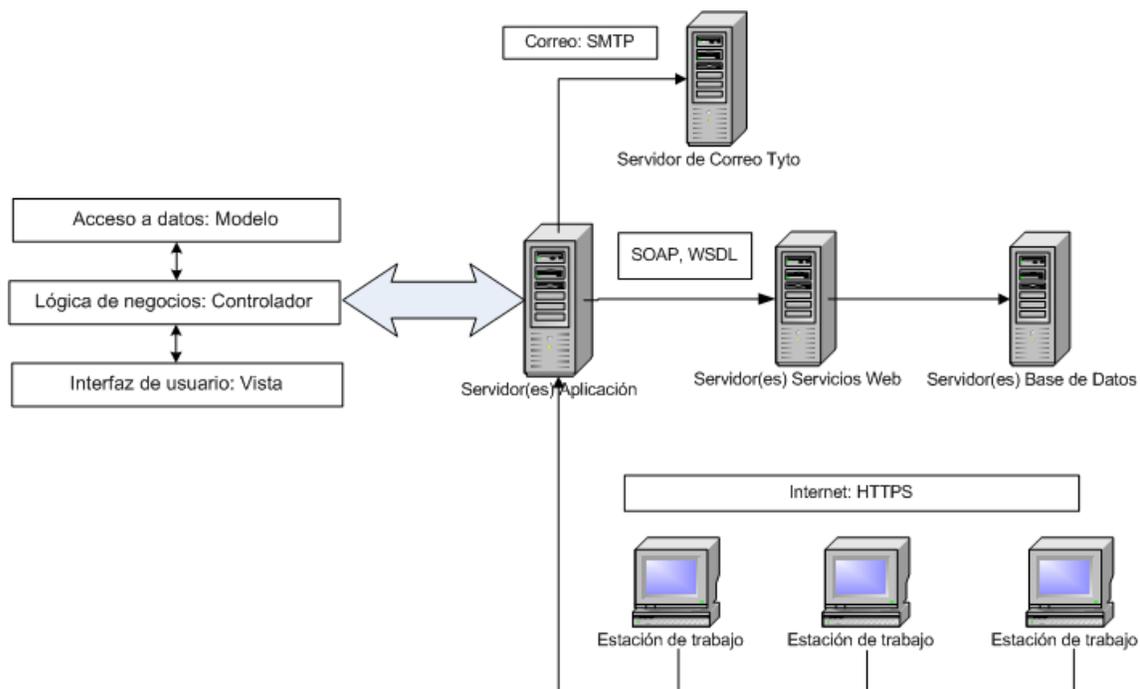


Figura 24. Arquitectura General del sistema.

Se identificaron los elementos de diseño más relevantes, la creación de clases que se correlacionan con los registros provenientes de la base de datos, métodos capaces de procesar la información proveniente de los servicios *Web* y de transformarla en arreglos de objetos de dichas clases.

Esta fase de elaboración constó de la realización de dos actividades:

Actividad 1: Diseñar el esquema de la base de datos

Esta actividad se ejecutó conjuntamente con los equipos de desarrollo de los subsistemas que interactúan con el SGP, para así modelar una única base de datos central que satisfaga los requerimientos. Para este fin, se realizaron reuniones a distancia para determinar los atributos y tablas necesarias.

Actividad 2: Diseñar componentes complejos

Se desarrollaron los elementos específicos para la implementación de la solución como clases y controladores que se ilustran en los apartados Capa Modelo y Capa Controlador de esta fase. Inicialmente se determinaron los componentes que son reusables entre los módulos del sistema de gestión de preparadores, posteriormente se llevó a cabo la escritura del código fuente de los mismos y finalmente la implementación de las reglas del negocio propias de la gestión de retiros de preparadores.

Un punto influyente en el desarrollo del proyecto fue la solicitud de cambios recurrentes que surgieron producto de nuevos requerimientos por parte de los equipos de desarrollo, estos cambios se dieron a nivel de campos nuevos en la base de datos y por lo tanto en la definición de los atributos de salidas de los servicios *Web*. En esta fase, la integración de los cambios entre los diferentes módulos que se desarrollaron en paralelo constituyó una tarea compleja que se dificultó a medida que los códigos fuentes se incrementaban.

Se ejecutaron pruebas para determinar la calidad del producto y se verificó la implementación mediante presentaciones a las partes interesadas, para así poder asegurar el cumplimiento del propósito del sistema, pudiendo, de esta forma, identificar posibles fallas para luego proponer un plan de trabajo con las respectivas soluciones. Cabe destacar que se presentó una limitante para la realización de las pruebas con los servicios *Web*, debido a que sólo se pudieron simular, basándose en los acuerdos y criterios de comunicación e intercambio de datos entre módulos y los sub-sistemas.

➤ **Arquitectura del sistema**

La arquitectura del sistema fue diseñada en capas, usando el patrón Modelo – Vista - Controlador (MVC), con una variante que consta de una capa intermedia entre las capas modelo y controlador que provee las peticiones a los servicios *Web* para la ejecución de las consultas, modificaciones, eliminaciones y/o inserciones en la base de datos central, minimizando así la dependencia de la misma con los demás sistemas que interactúan con ella.

A continuación se presentan las capas que conforman la arquitectura del sistema:

➤ **Capa Modelo**

Esta capa se realizó con el fin de mantener la estructura MVC aunque la aplicación no realiza ninguna conexión a la base de datos, se crearon algunas clases que hacen correlación a los resultados o parámetros de salidas de los servicios *Web*. La Figura 25 ilustra las clases para tal fin. El detalle de estas clases se explica más adelante en el apartado Estructura de la Capa Modelo en la fase de construcción.

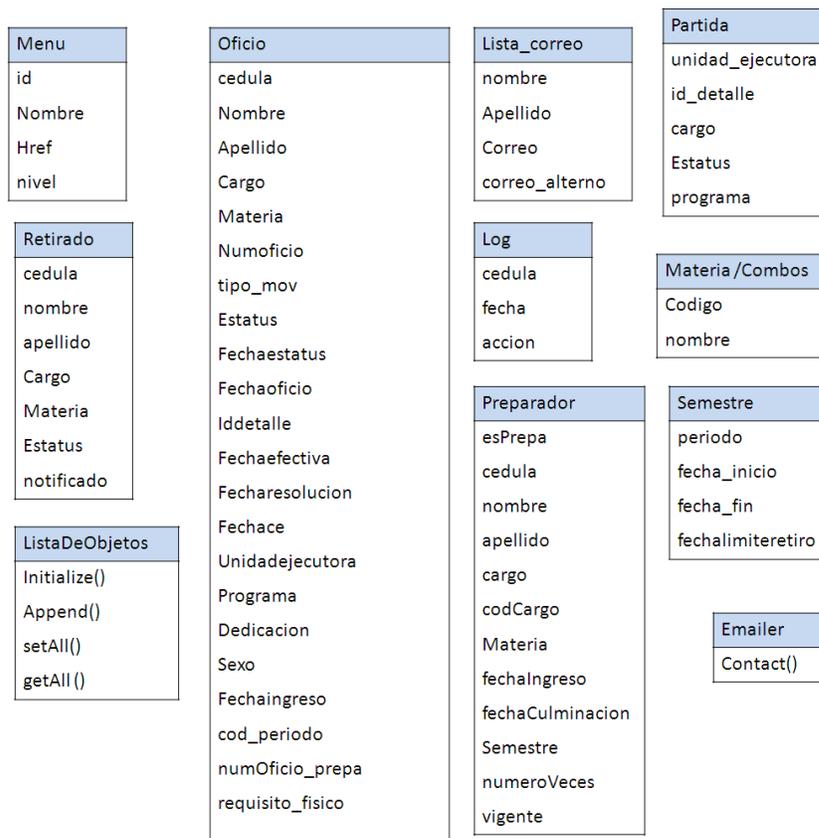


Figura 25. Diagrama de Clases.

➤ **Capa Vista**

Está compuesta por las interfaces o vistas diseñadas con la hoja de estilo suministrada por la Escuela de Computación. Cada vista se comunica con un controlador en particular, el cual se encarga de validar y procesar la petición realizada por el usuario. En el apartado Estructura de la Capa Vista de la fase de construcción se muestran fragmentos de las interfaces principales con sus descripciones y los roles que accederán a ellas.

➤ **Capa Controlador**

Esta capa implementa la lógica del negocio. En la Figura 26 se observan las posibles conexiones que existen entre controladores. La explicación detallada de cada uno de los métodos implementados en los controladores se encuentra en el apartado Estructura de la Capa Controlador de la fase de construcción.

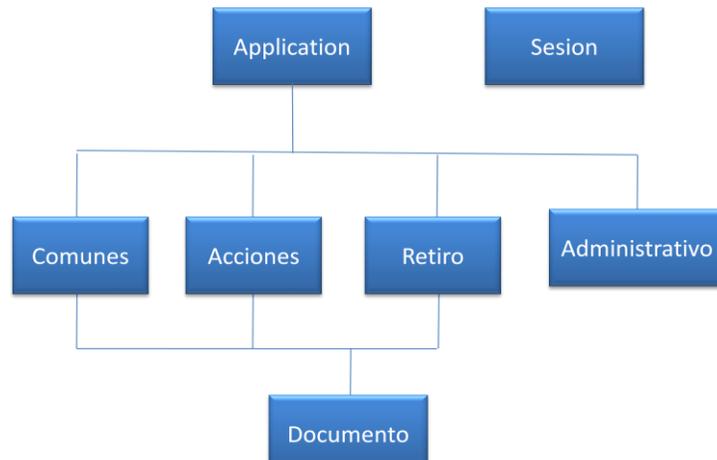


Figura 26. Interacción entre los controladores.

En las Tablas 4a y 4b se presenta el plan de proyecto ejecutado durante el desarrollo del sistema. Este plan de proyecto está basado en la plantilla ofrecida por la metodología *OpenUp* mostrando los objetivos primarios en cada una de las fases con sus respectivas iteraciones así como el tiempo y esfuerzo estimado.

Fase	Iteración	Objetivos primarios	Calendario inicio/Fin	Esfuerzo estimado
Inicio	1	Preparar el ambiente de desarrollo Revisar interfaces diseñadas por [AB, 2009] Definir la estructura de las interfaces del sistema Elaborar el documento visión del proyecto Definir los conceptos manejados en el sistema	25/02/2009 15/03/2009	13 días
Elaboración	1	Elaborar el plan del proyecto Definir la arquitectura del sistema Definir y desarrollar las funcionalidades comunes del sistema Definir los servicios <i>Web</i> necesarios para los módulos de gestión de retiros y administrativo Desarrollar y verificar el componente de autenticación del sistema Desarrollar las interfaces del módulo de gestión de retiros	16/03/2009 05/06/2009	60 días

Tabla 4a. Plan de proyecto.

Fase	Iteración	Objetivos primarios	Calendario inicio/Fin	Esfuerzo estimado
	2	Finalizar el desarrollo de las funcionalidades comunes Verificar los servicios <i>Web</i> definidos con los otros equipos de desarrollo Verificar las interfaces del módulo de gestión de retiros	08/06/2006 19/06/2009	10 días
Construcción	1	Desarrollo de las interfaces comunes del sistema Realizar pruebas en el sistema de parte de los desarrolladores Verificar y validar las interfaces del módulo de gestión de retiros Verificar y validar las interfaces del módulo administrativo	22/06/2006 31/07/2009	30 días
	2	Finalizar el desarrollo de los módulos de gestión de retiros y administrativo Finalizar y verificar las interfaces comunes del sistema	03/08/2009 14/08/2009	10 días
Transición	1	Preparar del ambiente de producción Alojar la aplicación SGP en el servidor Realizar las pruebas generales del sistema Planificar los siguientes pasos de transición	17/08/2009 28/08/2009	10 días

Tabla 4b. Plan de proyecto.

La Tabla 5 corresponde a un ejemplo de un “*work ítem list*” con base al objetivo primario “desarrollo de las interfaces comunes del sistema” especificando sus objetivos secundarios con su prioridad, estado, número de iteraciones, rol a quién se le asigna, esfuerzo estimado y horas trabajadas.

Nombre / Descripción	Prioridad	Estado	Num. Iteraciones	Asignado a	Esfuerzo Estimado (horas)	Horas trabajadas
Trámite / Completar Oficios con Resoluciones / Ver oficio (pdf)	1	Terminado	1	Isabelle Bellet	15	20
Trámite / Actualizar Firma Director / Ver oficio (pdf)	1	Terminado	1	Iris Aguilar	15	20
Trámite / Consultar Oficios por archivar / Ver oficio (pdf)	1	Terminado	1	Iris Aguilar	15	20
Trámite / Notificar resoluciones a presupuesto / Ver oficio (pdf)	1	Terminado	1	Isabelle Bellet	15	20

Tabla 5. Work Item List.

2.7.3. Fase de construcción

Continuando con el desarrollo iterativo de la solución, esta fase se basó en completar y afinar la implementación de los componentes diseñados para satisfacer los requerimientos de los usuarios en el sistema.

➤ Estructura de la Capa Controlador

En la capa controlador se realizaron las validaciones de entrada de datos o selección hecha por los usuarios en una vista, además de contener los métodos que procesan las tareas la gestión de retiros y el módulo administrativo. También se ven reflejadas las funcionalidades que son re-usadas entre controladores de diferentes módulos. Los controladores que se desarrollaron fueron:

Controlador *Application*: Es donde se definen las funcionalidades comunes del sistema de gestión de preparadores:

- **Funciones “Parsear” (analizador sintáctico):** Permite procesar la información de las diferentes respuestas que generan los servicios *Web*.
- **La funcionalidad *Paginate*:** Permite la presentación paginada de los resultados de una consulta o un listado.
- **El método *WSDL*,** en donde se especifica la ruta única en donde se encuentran alojados todos los servicios *Web* a ser consumidos por el sistema.
- **La función *listar_oficios*,** se encarga de retornar una lista de objetos de tipo de dato de la clase *Oficio*, la cual se llamará desde el controlador *Trámites*, donde se muestra el detalle de los oficios. Esta función consulta al servicio *Web* *consultar_oficios*,

pasándole varios parámetros, siendo el estatus del oficio el más importante, ya que a través de este se va desgregando la información de los mismos en diferentes vistas.

Controlador Retiro: Contiene las funciones necesarias para el manejo de las vistas correspondientes al módulo de retiro. Las funciones que se definieron fueron:

- **Funciones para la vista listado de solicitudes de retiro,** se interactúa servicio *Web* que devuelve todos los registro de los oficio de retiro generados.
- **Funciones para la vista Solicitar Retiro,** se interactúa servicio *Web* que registra en la base de datos el oficio de retiro generado, además de realiza las notificaciones a los actores involucrados.
- **Función para la lista de preparadores retirados,** se encarga de obtener registros de la base de datos de los preparadores que tengan asociados oficios de retiros en estatus aprobado.
- **Función para notificación para el coordinador,** su finalidad es enviar un correo electrónico al coordinador al momento que el Jefe del Departamento seleccione desde una vista un cargo vacante de un preparador para indicar al coordinador de esa materia que debe abrir un concurso.
- **Función para “parsear” (analizador sintáctico) los datos de los preparadores retirados,** esta función es invocada desde lista de preparadores retirados, y su finalidad es pasar la información obtenida desde un servicio *Web* a una lista de objetos, que serán desplegados en una interfaz al usuario.

Controlador Sesion: Controla lo referente a las sesiones del sistema, como los datos de autenticación y el menú dinámico por usuario, garantizando de esta forma parte de la seguridad del sistema. Las funciones que se definieron en este controlador fueron:

- **Index:** Controla que ningún usuario pueda entrar a una opción del sistema directamente desde la URL de browser sin anteriormente estar autenticado.
- **Buscar usuario:** Se encarga de buscar en la base de datos, a través de la consulta de un servicio *Web*, si un usuario y clave son validos, de ser así, se llama otra función *find_menu*, que cargará las opciones o actividades que puede ver ese usuario autenticado en una variable sesión.
- **La función parsear_XML_usuario:** Es propio para el caso del servicio *Web* autenticar y se encarga de pasar la respuesta *XML* a un arreglo de tipos de datos simples.

Controlador acciones: Contiene el manejo de las vistas de consultas sobre las tablas preparadores y partidas, en este controlador se puede encontrar las siguientes funcionalidades:

- **lista_preparador_actuales:** Permite consultar la información completa de los preparadores actuales de la escuela.
- **consultar_partidas:** Lista la información de las partidas presupuestarias con el estatus asociado actual, en caso que sea una partida ocupada, se muestra que persona la está utilizando.
- **lista_preparador_historia:** Muestra el detalle completo de toda la trayectoria que ha tenido un estudiante como preparador en la Escuela.
- **lista_preparador_constancia:** Permite generar en formato PDF la información de la trayectoria de un preparador, ya sea que esté desempeñando actualmente el cargo o no.

Controlador comunes: Maneja las funcionalidades de las vistas que son comunes para el módulo de gestión de retiros, concurso y nombramiento. La principal función de este controlador es aplicar la lógica de negocio para el manejo de los oficios de preparadores, es decir, contiene el código necesario para cargar los diferentes estatus de los oficios y modificarlos. Seguidamente se presentan los métodos implementados en este controlador:

- **lista_oficios_con_resoluciones:** Lista los oficios con estatus aprobado por Consejo de Escuela.
- **listar_oficios_archivar:** Lista los oficios que deben ser archivados, es decir que tiene como estatus actual "Firmado por Director".
- **actualizar_firma_director:** Lista los oficios con estatus "Aprobados en Consejo de Escuela" y el usuario puede llevar el control de los mismos, una vez que han sido firmados por el Director de la Escuela.
- **lista_oficios_resoluciones_modificar:** Permite listar los oficios que ya tienen los estatus discutidos en el Consejo de Escuela para que el usuario pueda modificar estos valores, además de la fecha de resoluciones y fecha de Consejo de Escuela.
- **lista_oficios_a_discutir:** Lista los oficios completos y revisados que se encuentran listos para ser discutidos en Consejo de Escuela, además en esta vista, es donde por primera vez se imprime los oficios.
- **lista_oficios_enviados_presupuesto:** Interfaz que permite consultar los oficios enviados al Departamento de Presupuesto.
- **lista_solicitud_retiro:** Lista los oficios que tiene como estatus "oficio nuevo", es decir, que se creó alguna solicitud de retiro, nombramiento, de aumento o disminución de horas; por lo tanto la secretaria del departamento o usuario encargado de realizar esta

acción debe consultar esta vista para completar los datos del oficio para que luego sean revisados por la Asistente Administrativa o usuario con esa actividad asignada.

- **lista_oficios_generados:** Permite cargar en la vista correspondiente los datos referentes a los oficios que ya fueron completados y revisados, es decir, listos para ser discutidos en Consejo de Escuela.
- **filtros:** Es una funcionalidad común o general para diferentes interfaces, particularmente, para las vistas que manejan los oficios y que se encuentran en el controlador comunes. Permite cargar variables correspondientes a la lista de semestre, materias, tipos de movimiento y cargos, para ser mostrados en forma de combos en una vista, para que el usuario pueda seleccionar algunos de estos filtros al momento de realizar una consulta.
- **validar_seleccion_busqueda:** Esta función se encarga de validar los datos introducidos por el usuario en la vista de filtros para consultar los oficios, como es el caso del campo cédula que debe ser numérico o simplemente que realice la consulta seleccionando al menos una opción de las ofrecidas en los filtros.
- **completar_oficio y completar_oficio_accion:** Se encargan de mostrar y registrar cambios del estatus del oficio, fecha efectiva y fecha de Consejo de Escuela.
- **cambiar_estatus:** Es una función general que es ejecutada desde la acción de las vistas en donde el usuario cambia el estatus de algún oficio, llamando a un servicio *Web* que se encarga de actualizar dicho estatus en la base de datos.

Controlador documentos: Este controlador permite a través de sus funciones la generación de documentos, como constancias, oficios y minutas, en formato PDF. A continuación se describen sus funciones:

- **ver_oficio:** Sirve para convertir los datos de un oficio de cualquier tipo de movimiento (retiro, concurso, nombramiento), provenientes de la respuesta de un servicio *Web*, a un archivo con formato PDF, para luego imprimirlo o simplemente visualizarlo.
- **ver_solicitud_retiro:** Se encarga de crear la planilla de solicitud de retiro en formato PDF para que el coordinador la pueda imprimir y firmar.
- **generar_minuta:** Función encargada de crear la minuta complementaria, en formato PDF, con los puntos seleccionados por la Secretaria de Consejo de Escuela para ser discutidos en Consejo de Escuela.

Controlador Administrativo: Es el controlador que maneja toda la información necesaria para las vistas pertenecientes al administrador del sistema. Sus funcionalidades se presentan seguidamente:

- **listado_semestre:** Lista las fechas inicio, fecha fin de cada semestre que ha sido creado desde la puesta en funcionamiento del sistema. Además muestra la fecha límite de retiro para cada periodo creado.
- **crear_modificar_semestre:** Actualiza las fechas asociadas a cada periodo creado: inicio, fin y límite retiro.
- **lista_usuario:** Devuelve la lista de los usuarios del sistema.
- **buscar_usuario:** Valida en la base de datos, a través de un servicio *Web*, si una cédula pertenece a un usuario del sistema.
- **buscar_usuario_actividad:** Esta función permite buscar las actividades asignadas a un usuario válido del sistema, suministrando como entrada su número de cédula.
- **buscar_actividades:** Obtiene todos los registros que se encuentran en la tabla actividades del sistema, para desplegarlo luego en una vista donde el administrador selecciona las actividades que le asignará a un usuario dado.
- **consultar_log:** Consulta en la base de datos la tabla *log* de transacciones dado una cédula de usuario y/o un rango de fecha.

➤ Estructura de la Capa Modelo

Las llamadas a los servicios *Web* se ejecutan desde la capa controlador, seguido de la ejecución de los métodos “*parsear*” (analizador sintáctico) que procesan la información que devuelven los servicios *Web* solicitados, generando las listas de los objetos de las clases, para luego desde las vistas recorrer estos objetos y así mostrar la información obtenida a los usuarios. Las clases creadas para este fin fueron:

- **Clase listaDeObjetos:** Esta clase es general y se usa para cargar una lista de objetos de cualquier clase en la capa controlador, y desde las diferentes vistas se recorre este objeto mostrando los valores de cada atributo.
- **Clase Menu:** Se usa para cargar las actividades asociadas al usuario que se autenticó en el sistema. Contiene los mismos atributos de la tabla actividades de la base de datos.
- **Clase Materia/Combos:** Es una clase que contiene dos atributos necesarios para listar catálogos como son la materia, cargos, tipo movimiento o cualquier otro que se use para cargar los combos en las vistas; los atributos de esta clase son: código y descripción.
- **Clase Preparador:** Esta clase permite obtener la lista de preparadores de la Escuela, según los criterios seleccionados por el usuario en los filtros y que han sido pasados como parámetros al servicio *Web* correspondiente. Los atributos de esta clase son: cedula, nombre, apellido, cargo, codCargo, materia, fechaIngreso, fechaCulminacion, semestre, esPrepa, numeroVeces, vigente.
- **Clase Partida:** Permite cargar las partidas presupuestarias.
- **Clase PartidaDetalle:** Su función es permitir la carga de las partidas presupuestarias con su detalle.

- **Clase Semestre:** Esta clase fue creada para establecer correlación con la tabla semestre de la base de datos central de la gestión de preparadores.
- **Clase Log:** Sirve para almacenar en una variable los registros obtenidos de consultar las actividades realizadas por algún usuario del sistema, esta consulta es llevada a cabo por el administrador.
- **Clase Lista_correo:** Es una clase que permite guardar valores referentes a la dirección de correos electrónicos de los usuarios del sistema.
- **Clase Retirado:** Permite almacenar los datos provenientes de un servicio *Web*, encargado de traer información referente a los preparadores que tienen alguna solicitud de retiro. Los atributos correspondientes a esta clase son: cedula, nombre, apellido, materia, cargo, estatus, notificado.
- **Clase Emailer:** Esta clase sirve para el envío de correo electrónico y es llamada desde los controladores de la siguiente forma: *contact(escuela, recipient, message, sent_at)*, en donde el atributo *escuela* se refiere al nombre de la escuela perteneciente al usuario autenticado, *recipient* es la lista de correos, *message* es el cuerpo del mensaje que se desea enviar, *sent_at* la fecha del envío. En la capa vista es donde se encuentra el formato del correo que se va enviar, específicamente en *contact.html*.
- **Clase Oficio:** Es una de las clases más importantes del sistema, porque a través de ésta se carga toda la información referente al oficio, que es el documento principal en cualquier trámite de la gestión de preparadores.

➤ Estructura de la Capa Vista

Continuando con la presentación de la estructura general del sistema, se describen las interfaces implementadas:

Usuario: Coordinador

Nombre: Solicitar retiro - Lista de preparadores actuales

Descripción: Muestra una lista de los preparadores actuales (Ver Figura 27), de los cuales puede seleccionar uno de ellos para realizar la solicitud de retiro del cargo que ocupa. En el momento en que se selecciona un preparador se cargarán, en otra interfaz, los siguientes datos del mismo: cédula de Identidad, nombres y apellidos y el cargo que desempeña. Posteriormente, el coordinador debe llenar un formulario (Ver Figura 28) exponiendo la justificación del retiro, ya sea porque es solicitado por el coordinador o por renuncia expresa por parte del preparador, y finalmente procede a enviar la solicitud.

FAC C I E N C I A S
UCCV
ESCUELA Computación

Gestión de Preparadores
 UCV · Ciencias · Intranet · Contacto · Mapa · Créd
 Usuario: Angel Diaz (coordinador) | Cerrar sesion

Domingo, 11 de Octubre de 2009

Concurso ▼ Nombramiento ▼ Retiro ▼ Trámite ▼ Más acciones ▼

Periodo actual: I-2009
 Materia: Sistemas Operativos

Solicitar retiro - Lista de preparadores actuales

Cédula	Nombre	Cargo	Fecha ingreso	Retirar
14520369	Juan Manuel Cañas	Preparador I	01-01-2008	
14520369	Juan Manuel Cañas	Preparador I	24-05-2008	
12345678	Ana Fernández	Preparador I	24-05-2008	

< Anterior 1 2 Siguiente >

Mostrando los resultados 1 - 3 de 5 en total

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
 © 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)
 Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.

Figura 27. Lista de preparadores actuales de una materia.

FAC C I E N C I A S
UCCV
ESCUELA Computación

Gestión de Preparadores
 UCV · Ciencias · Intranet · Contacto · Mapa · Créd
 Usuario: Angel Diaz (coordinador) | Cerrar sesion

Domingo, 11 de Octubre de 2009

Concurso ▼ Nombramiento ▼ Retiro ▼ Trámite ▼ Más acciones ▼

Periodo actual: I-2009
 Materia: Sistemas Operativos

Solicitar retiro

Cedula: 14520369

Nombre y Apellidos: Juan Manuel Cañas

Cargo: Preparador I

Justificación:

Carta de renuncia: Si No

← Regresar

* Campos Obligatorios

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
 © 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)

Figura 28. Solicitud de un retiro.

Usuario: Secretaria del Departamento

Nombre: Consultar solicitudes - Lista de solicitudes de Trámites

Descripción: La Secretaria del Departamento dispone de esta funcionalidad para visualizar el listado de los preparadores que han realizado alguna solicitud de retiro. Además, en esta interfaz podrá asignar a cada oficio un número que lo represente, de igual forma, internamente cada oficio maneja un único número asociado. Luego de completar esta información, la secretaria podrá imprimir el oficio completo. Si cualquier oficio presenta una incoherencia o error en los datos, la secretaria del departamento debe hacer la observación pertinente llenando el campo observación presentado en una nueva vista, esta información es almacenada en la base de datos y notificada al administrador del sistema. Como último paso procede a marcar los oficios verificados para que sigan el flujo de trabajo. (Ver Figura 29).

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: Jazmin Diaz (sec_dpto) | Cerrar sesion

Concurso ▾ Nombramiento ▾ Retiro ▾ Trámite ▾ Más acciones ▾

Periodo actual: I-2009 Domingo, 11 de Octubre de 2009

Consultar solicitudes - Lista de solicitudes de Trámites

- Oficios con Estatus: Solicitud generada - Búsqueda Seleccionada: II-2008 /

Cédula	Nombre	Cargo	Tipo Mov.	Materia	Obser.	Nº Oficio	Actualizar	Generar Oficio	Verificado
12345678	Felipe A. Diaz P.	Preparador I	Nombramiento	Sistemas Operativos		03.13.2009-039			<input type="checkbox"/>
12345678	Felipe A. Diaz P.	Preparador I	Retiro	Sistemas Operativos		03.13.2009-040			<input type="checkbox"/>
18765432	María V. Castro L.	Preparador II	Nombramiento	Sistemas Operativos		33453			<input type="checkbox"/>

« Anterior 1 2 Siguiente »

Mostrando los resultados 1 - 3 de 5 en total

Nueva Búsqueda

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
© 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)
Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.

Figura 29. Lista de solicitudes de trámites.

Usuario: Secretaria del Departamento

Nombre: Consultar Oficios por archivar - Lista de Oficios por Archivar

Descripción: Esta interfaz muestra los oficios aprobados por Consejo de Escuela, firmados por la directora y por tanto, listos para ser archivados en los expedientes de los preparadores. Una vez que estos documentos han sido archivados, la secretaria del departamento selecciona los oficios en esta vista para marcarlos con el estatus de “archivados” como se observa en la Figura 30.

The screenshot displays the 'Gestión de Preparadores' web application. At the top left, there are logos for 'FACULTAD DE CIENCIAS UCV' and 'ESCUELA DE COMPUTACIÓN'. The main title 'Gestión de Preparadores' is on the right. Below the title, there are navigation links: 'UCV · Ciencias · Intranet · Contacto · Mapa · Créd', and a user session bar: 'Usuario: Jazmin Diaz (sec_dpto) | Cerrar sesion'. A menu bar contains 'Concurso', 'Nombramiento', 'Retiro', 'Trámite', and 'Más acciones'. The current date is 'Domingo, 11 de Octubre de 2009'. The main content area shows 'Periodo actual: I-2009' and a link to 'Consultar Oficios por archivar - Lista de Oficios por Archivar'. Below this, it indicates '- Oficios con Estatus: Firmado por Director -' and 'Búsqueda Seleccionada: I-2008 /'. A table lists three offices with columns for 'Cédula', 'Nombre y Apellidos', 'Cargo', 'Materia', 'Nro.Oficio', 'Asunto', 'Ver Oficio', and 'Archivado'. The table shows three rows of data. Below the table are navigation controls: '< Anterior', '1', '2', 'Siguiente >', and 'Mostrando los resultados 1 - 3 de 5 en total'. There is a 'Notificar' button and a search icon labeled 'Nueva Búsqueda'. At the bottom, there is a footer with contact information for the 'Sistema de Gestión Preparaduría'.

Cédula	Nombre y Apellidos	Cargo	Materia	Nro.Oficio	Asunto	Ver Oficio	Archivado
12345678	Felipe A. Diaz P.	Preparador I	Sistemas Operativos	03.13.2009-039	Nombramiento		<input type="checkbox"/>
12345678	Felipe A. Diaz P.	Preparador I	Sistemas Operativos	03.13.2009-040	Retiro		<input type="checkbox"/>
18765432	María V. Castro L.	Preparador II	Sistemas Operativos	33453	Nombramiento		<input type="checkbox"/>

Figura 30. Lista de Oficios por Archivar.

Usuario: Asistente Administrativo

Nombre: Consultar Solicitudes / Ver Oficios - Lista de solicitudes

Descripción: La persona que ejerce el rol de asistente administrativo podrá llevar a cabo, a través de esta interfaz ilustrada en la Figura 31, una de sus funciones importantes en el proceso retiro, que consiste en verificar las solicitudes de retiro para posteriormente enviar el oficio a la secretaria del Consejo de Escuela.

Si cualquier oficio presenta una incoherencia o error en los datos, quien desempeñe el rol de asistente administrativo debe hacer la observación pertinente llenando el campo observación presentado en una nueva vista, esta información es almacenada en la base de datos y notificada al administrador del sistema

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: Mariela Jimenez (asist_adm) | Cerrar sesion

Concurso ▼ Nombramiento ▼ Retiro ▼ Trámite ▼ Más acciones ▼

Periodo actual: I-2009 Domingo, 11 de Octubre de 2009

[Consultar Solicitudes / Ver Oficios - Lista de solicitudes](#)

- Oficios con Estatus: Oficio Generado -

Búsqueda Seleccionada: I-2008 /

Cédula	Nombre	Cargo	Tipo Mov.	Materia	Unidad Ejecutora	Programa	IDDetalle	Obser.	Ver Oficio	Verificado
12345678	Felipe A. Diaz P.	Preparador I	Nombramiento	Sistemas Operativos	0313000000	0313000000	13622			<input type="checkbox"/>
12345678	Felipe A. Diaz P.	Preparador I	Retiro	Sistemas Operativos	0313000000	0313000000	13622			<input type="checkbox"/>
18765432	María V. Castro L.	Preparador II	Nombramiento	Sistemas Operativos	0313000000	0313000000	13623			<input type="checkbox"/>

« Anterior | 1 | 2 | Siguiente »

Mostrando los resultados 1 - 3 de 5 en total

Nueva Búsqueda

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
© 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)

Figura 31. Ver Oficios - Lista de solicitudes.

Usuario: Asistente Administrativo

Nombre: Notificar Resoluciones a Presupuesto - Lista de Oficios con Resoluciones.

Descripción: Como se observa en la Figura 32, esta vista permite cumplir con la actividad que culmina el flujo de trabajo de un oficio dentro de la Escuela, dado que a través de la misma se notifica a presupuesto sobre los oficios que han sido debidamente procesados en las instancias de la Escuela involucradas en la gestión de preparadores.

The screenshot displays a web application interface for 'Gestión de Preparadores'. At the top left, there are logos for 'FACULTAD DE CIENCIAS' and 'ESCUELA DE COMPUTACIÓN'. The main title 'Gestión de Preparadores' is prominently displayed. Below the title, there are navigation links: 'UCV · Ciencias · Intranet · Contacto · Mapa · Créd' and a user session indicator: 'Usuario: Mariela Jimenez (asist_admin) | Cerrar sesion'. A horizontal menu contains dropdowns for 'Concurso', 'Nombramiento', 'Retiro', 'Trámite', and 'Más acciones'. The current date is 'Domingo, 11 de Octubre de 2009' and the current period is 'Periodo actual: I-2009'. The main content area is titled 'Notificar Resoluciones a Presupuesto - Lista de Oficios con Resoluciones' and shows a sub-header '- Oficios con Estatus: Aprobado en Consejo de Escuela -'. A search filter 'Búsqueda Seleccionada: I-2008 /' is present. A table lists three entries with columns: 'Cédula', 'Nombre y Apellidos', 'Cargo', 'Materia', 'Nro.Oficio', 'Asunto', 'Ver Oficio', and 'Enviado'. Below the table are navigation controls: '< Anterior', '1', '2', 'Siguiente >', and 'Mostrando los resultados 1 - 3 de 5 en total'. A 'Notificar' button is located below the table. At the bottom right, there is a 'Nueva Búsqueda' search icon. The footer contains contact information for the 'Sistema de Gestión Preparaduría' at the 'Escuela de Computación'.

Cédula	Nombre y Apellidos	Cargo	Materia	Nro.Oficio	Asunto	Ver Oficio	Enviado
12345678	Felipe A. Diaz P.	Preparador I	Sistemas Operativos	03.13.2009-039	Nombramiento		<input type="checkbox"/>
12345678	Felipe A. Diaz P.	Preparador I	Sistemas Operativos	03.13.2009-040	Retiro		<input type="checkbox"/>
18765432	María V. Castro L.	Preparador II	Sistemas Operativos	33453	Nombramiento		<input type="checkbox"/>

Figura 32. Lista de Oficios con Resoluciones.

Usuario: Secretaria del Consejo de Escuela

Nombre: Lista de Oficios listos a ser discutidos en Consejo de Escuela

Descripción: En esta vista se muestran los datos de los oficios verificados por la Secretaria del Departamento y la Asistente Administrativo, lo que indica que pueden ser llevados a discusión en el Consejo de Escuela (Ver Figura 33). La secretaria del Consejo de Escuela debe imprimir en este momento los oficios que llevará a discusión y colocará como puntos en la minuta.

Previamente para poder obtener la lista de oficios, el usuario debe ingresar algunos datos en los filtros ofrecidos: número de oficio, cédula del preparador, período, tipo de movimiento, materia y/o cargo, siendo el período el único criterio obligatorio a seleccionar, los demás filtros sirven para el refinamiento de la consulta.

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: Janeth Perez (sec_cons_esc) | Cerrar sesion

Concurso ▼ Nombramiento ▼ Retiro ▼ Trámite ▼ Más acciones ▼

Periodo actual: I-2009 Domingo, 11 de Octubre de 2009

Lista de Oficios listos a ser discutidos en Consejo de Escuela

- Oficios con Estatus: Revisado -

Búsqueda Seleccionada: I-2008 /

Nº Oficio	Asunto	Cargo	Cédula	Nombre	Materia	Imprimir Oficio	Seleccionar
03.13.2009-039	Nombramiento	Preparador I	12345678	Felipe A. Diaz P.	Sistemas Operativos		<input type="checkbox"/>
03.13.2009-040	Retiro	Preparador I	12345678	Felipe A. Diaz P.	Sistemas Operativos		<input type="checkbox"/>
33453	Nombramiento	Preparador II	18765432	María V. Castro L.	Sistemas Operativos		<input type="checkbox"/>
33455	Retiro	Preparador I	12876453	Adriana E. Muñoz R.	Bases de Datos		<input type="checkbox"/>
33454	Retiro	Preparador I	18123678	Luis A. Diaz G.	Bases de Datos		<input type="checkbox"/>

Generar Minuta

Nueva Búsqueda

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
© 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)
Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.

Figura 33. Lista de Oficios listos a ser discutidos en Consejo de Escuela.

Usuario: Secretaria del Consejo de Escuela

Nombre: Completar / Modificar Oficios con Resoluciones

Descripción: Los oficios con estatus “listos a ser discutidos en Consejo de Escuela” son los presentados en esta interfaz, para que al momento de concluida la reunión en Consejo de Escuela, la Secretaria del Consejo de Escuela proceda a seleccionar cada oficio discutido y completar los siguientes datos: fecha efectiva del oficio, estatus del oficio (aprobado, negado, diferido) y la fecha de la reunión en Consejo de Escuela. Esta vista es una de las más importantes del SGP, porque se actualizan no solo los estatus de los oficios sino también los estatus de las partidas asociadas a los oficios. En el caso de un oficio con un tipo de movimiento “retiro”, si la resolución del mismo fue aprobada, la partida asociada pasa al estatus “disponible” si no ha sido tomada por otro proceso. En caso que el estatus de la partida se encuentre como “pre-asignable”, es decir, que está siendo utilizada, y el proceso no puede liberar completamente la partida, simplemente permanece en su estado actual. Esta interfaz permite modificar los datos insertados, en caso de posibles cambios o errores en los mismos. Previamente para poder obtener la lista de oficios, se debe ingresar algunos datos en el filtro solicitado: número de oficio, cédula del preparador, período, tipo de movimiento, materia y/o cargo. (Ver Figura 34).

FACULTAD DE CIENCIAS

ESCUELA de Computación

Gestión de Preparadores

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: Janeth Perez (sec_cons_esc) | Cerrar sesion

Concurso | Nombramiento | Retiro | Trámite | Más acciones

Periodo actual: I-2009

Domingo, 11 de Octubre de 2009

Completar / Modificar Datos del Oficio

Nº Oficio:	03.13.2009-039
Cédula:	12345678
Nombre:	Felipe A. Diaz P.
Cargo:	Preparador I
Materia:	Sistemas Operativos
Fecha Efectiva:	10-10-2009 *
Estatus:	Aprobado *
Fecha C.E.:	10-10-2009 *

Guardar Cancelar

* Campos Obligatorios

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
© 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)
Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.

Figura 34. Completar / Modificar Oficios con Resoluciones.

Usuario: Secretaria del Consejo de Escuela

Nombre: Lista de oficios por firmar por Director(a)

Descripción: Una vez que los oficios de gestión de preparadores generados en la Escuela hayan sido discutidos en Consejo de Escuela y tengan asignados los estatus correspondientes a la decisión tomada (aprobado, diferido, negado), la secretaria del Consejo de Escuela procede a entregar dichos oficios físicos a la directora(a) de la Escuela para obtener su firma, una vez firmados, se debe entrar al sistema para cambiar el estatus de estos oficios a "Firmado por Director(a)" (Ver Figura 35). Previamente para poder obtener la lista de oficios, el usuario debe ingresar algunos datos en el filtro solicitado: número de oficio, cédula del preparador, período, tipo de movimiento, materia y/o cargo. Cabe destacar, que en cualquier fase por la que pasa el oficio, el usuario siempre tendrá la posibilidad de consultar el oficio original en formato PDF e imprimirlo.

FACULTAD DE CIENCIAS

UCV

ESCUELA DE COMPUTACIÓN

Gestión de Preparadores

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: Janeth Perez (sec_cons_esc) | Cerrar sesion

Concurso | Nombramiento | Retiro | Trámite | Más acciones

Periodo actual: I-2009

Domingo, 11 de Octubre de 2009

Lista de oficios por firmar por Director(a)

- Oficios con Estatus: Aprobados en Consejo de Escuela -

Búsqueda Seleccionada: I-2008 /

Cédula	Nombre y Apellidos	Cargo	Materia	Nro.Oficio	Asunto	Ver Oficio	Firmado
12345678	Felipe A. Diaz P.	Preparador I	Sistemas Operativos	03.13.2009-039	Nombramiento		<input type="checkbox"/>
12345678	Felipe A. Diaz P.	Preparador I	Sistemas Operativos	03.13.2009-040	Retiro		<input type="checkbox"/>
18765432	María V. Castro L.	Preparador II	Sistemas Operativos	33453	Nombramiento		<input type="checkbox"/>

< Anterior 1 2 Siguiente >

Mostrando los resultados 1 - 3 de 5 en total

Notificar

Nueva Búsqueda

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
© 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)
Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.

Figura 35. Lista de oficios por firmar por Director(a).

Usuario: Administrador

Nombre: Consultar usuario - Lista de actividades – Modificar actividades

Descripción: Permite asignar las actividades a cada uno de los usuarios del sistema, tomando en cuenta las tareas definidas para simular el *workflow* de la gestión de preparadores de una Escuela. Inicialmente el administrador debe buscar un usuario a través de su cédula para luego asignarle las actividades que va a desempeñar. (Ver Figura 36).

The screenshot shows a web application interface for 'Computación'. At the top, there is a navigation bar with links for 'UCV · Ciencias · Intranet · Contacto · Mapa · Créd' and a user profile for 'Usuario: Luisa Jimenez (admin) | Cerrar sesion'. Below this, there are dropdown menus for 'Usuarios', 'Semestres', 'Log Transacciones', and 'Enviar correo'. The current date is 'Jueves, 24 de Septiembre de 2009' and the current period is 'Periodo actual: I-2009'. The main content area is titled 'Consultar usuario - Lista de actividades' and displays user information: 'Cedula: 10520140', 'Nombre y Apellido: Janeth Jimenez', and 'Cargo: Coordinador'. A table lists activities with columns for 'Módulo', 'Actividad', and 'Seleccione'. The activities are grouped into sections: 'Concurso', 'Nombramiento', 'Retiro', 'Trámite', and 'Más acciones'. A 'Guardar' button is located below the table. At the bottom, there is a footer with contact information for the 'Sistema de Gestión Preparaduría' at the 'Escuela de Computación'.

Módulo	Actividad	Seleccione
Concurso	Abrir Concurso	<input checked="" type="checkbox"/>
Nombramiento	Consultar preparadores con código asignado	<input checked="" type="checkbox"/>
Retiro	Consultar Solicitudes de Retiro	<input checked="" type="checkbox"/>
Trámite	Consultar Solicitudes / Generar Oficios	<input type="checkbox"/>
	Consultar Oficios / Detalle	<input checked="" type="checkbox"/>
Más acciones	Consultar Oficios por archivar	<input type="checkbox"/>
	Generar Constancia de Preparador	<input type="checkbox"/>
	Consultar historial de Preparador	<input type="checkbox"/>

Figura 36. Lista de actividades dado un usuario.

Usuario: Administrador

Nombre: Consultar usuario - Lista de actividades

Descripción: A través de esta interfaz el administrador observa las actividades que tiene asignadas un usuario, es una vista informativa tal como se ve en la Figura 37.

The screenshot displays the 'Gestión de Preparadores' web application. The header includes the logo for 'FACULTAD DE CIENCIAS' and 'ESCUELA DE COMPUTACIÓN' on the left, and the title 'Gestión de Preparadores' on the right. A navigation menu contains links for 'UCV · Ciencias · Intranet · Contacto · Mapa · Créd'. The user is identified as 'José Rodríguez (admin)'. Below the header, there are dropdown menus for 'Usuarios', 'Semestres', 'Log Transacciones', and 'Enviar correo'. The current period is 'I-2009' and the date is 'Domingo, 11 de Octubre de 2009'. The main content area is titled 'Consultar usuario - Lista de actividades' and shows details for user 'Janeth Jimenez' (Cedula: 15143888, Cargo: Coordinador). A table lists activities for different modules:

Módulo	Actividad
Concurso	Abrir Concurso
Nombramiento	Consultar preparadores con código asignado
Retiro	
Trámite	Consultar Solicitudes / Generar Oficios Consultar Oficios / Detalle
Más acciones	

At the bottom, a footer contains contact information for the 'Sistema de Gestión Preparaduría' at the 'Escuela de Computación' of the 'Universidad Central de Venezuela'.

Figura 37. Consultar usuario - Lista de actividades.

Usuario: Administrador

Nombre: Consultar usuario - Lista de actividades

Descripción: Esta interfaz ilustrada en la Figura 38, permite al administrador del sistema asignar o actualizar las actividades de un usuario.

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: José Rodríguez (admin) | Cerrar sesión

Usuarios ▾ Semestres ▾ Log Transacciones ▾ Enviar correo ▾

Periodo actual: I-2009

Domingo, 11 de Octubre de 2009

Consultar usuario - Lista de actividades

Cedula: 14556799
Nombre y Apellido: Janeth Jimenez
Cargo: Coordinador

Módulo	Actividad	Seleccione
Concurso		
	Abrir Concurso	<input checked="" type="checkbox"/>
Nombramiento		
	Consultar preparadores con código asignado	<input checked="" type="checkbox"/>
Retiro		
	Consultar Solicitudes de Retiro	<input checked="" type="checkbox"/>
Trámite		
	Consultar Solicitudes / Generar Oficios	<input checked="" type="checkbox"/>
	Consultar Oficios / Detalle	<input checked="" type="checkbox"/>
Más acciones		
	Consultar Oficios por archivar	<input type="checkbox"/>
	Generar Constancia de Preparador	<input type="checkbox"/>
	Consultar historial de Preparador	<input type="checkbox"/>

Guardar

Figura 38. Consultar usuario - Lista de actividades.

Usuario: Administrador

Nombre: Lista semestres

Descripción: A través de esta interfaz el administrador realiza la carga de la información referente a los periodos académicos, la cual es vital para el funcionamiento del SGP, indicando las fechas de inicio y fin de cada semestre y fecha límite de retiro, estas fechas se validan en diferentes puntos del sistema (Ver Figura 39).

The screenshot displays the 'Gestión de Preparadores' web application interface. At the top left, there are logos for 'FACULTAD DE CIENCIAS UCV' and 'ESCUELA DE COMPUTACIÓN'. The main title 'Gestión de Preparadores' is on the right. Below the title, there are navigation links: 'UCV · Ciencias · Intranet · Contacto · Mapa · Créd' and a user session bar: 'Usuario: José Rodríguez (admin) | Cerrar sesión'. A menu bar contains 'Usuarios', 'Semestres', 'Log Transacciones', and 'Enviar correo'. The current date is 'Domingo, 11 de Octubre de 2009'. The main content area shows 'Periodo actual: I-2009' and a 'Modificar Semestre' link. A form titled 'Modificar Semestre' contains the following fields: 'Semestre:' with a dropdown set to '2009'; 'Fecha inicio:' with a date field '06-10-2009'; 'Fecha fin:' with a date field '10-2-2010'; and 'Fecha Límite Retiro:' with a date field '22-10-2008'. Below these fields are 'Guardar' and 'Cancelar' buttons. A 'Regresar' button with a left-pointing arrow is located to the right of the form. At the bottom, a footer contains contact information: 'Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela © 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax) Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.'

Figura 39. Crear o modificar fechas de un periodo.

Usuario: Administrador

Nombre: Consultar Log

Descripción: Esta funcionalidad permite al administrador llevar un monitoreo de los usuarios y las actividades que han realizado dentro del sistema en un periodo determinado. Puede refinar la consulta por cédula y/o por un período. (Ver Figura 40).

UCV · Ciencias · Intranet · Contacto · Mapa · Créd

Usuario: José Rodríguez (admin) | Cerrar sesion

Usuarios ▾ Semestres ▾ Log Transacciones ▾ Enviar correo ▾

Periodo actual: I-2009

Domingo, 11 de Octubre de 2009

Consultar Log

Cédula	Nombre y Apellido	Fecha y Hora	Acción
15347272	Iris Aguilar	2008-05-12 8:50 am	autenticar
12345119	Isabelle Bellet	2008-05-12 9:01 am	generar_constancia_preparador
14532617	Ana Hernández	2008-05-12 3:50 pm	solicitar_retiro

« Anterior 1 2 Siguiente »

Mostrando los resultados 1 - 3 de 4 en total

Sistema de Gestión Preparaduría · Escuela de Computación · Facultad de Ciencias · Universidad Central de Venezuela
© 2007-2008 · (58 212) 605.1132 / 1258 / 1042 (fax)
Av. Los Ilustres, Los Chaguaramos, Caracas. ZP 1040.

Figura 40. Consultar el log generado por un usuario.

➤ Estructura de la Capa Servicios Web

Como se describió en la Capa Modelo fue necesaria la correlación o mapeo del resultado de las llamadas a los servicios Web con las clases creadas para tal fin. En las Figuras 41, 42, 43 y 44 se observa la interacción entre los controladores y los servicios Web. En los Anexos 3 y 4 se presenta la especificación de algunos de los servicios Web.

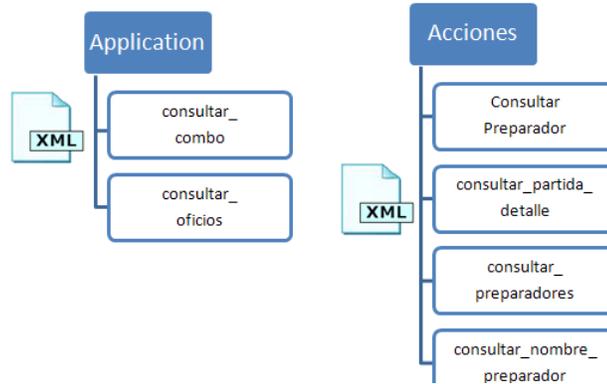


Figura 41. Diagrama Interacción entre controladores y servicios Web – Parte I.

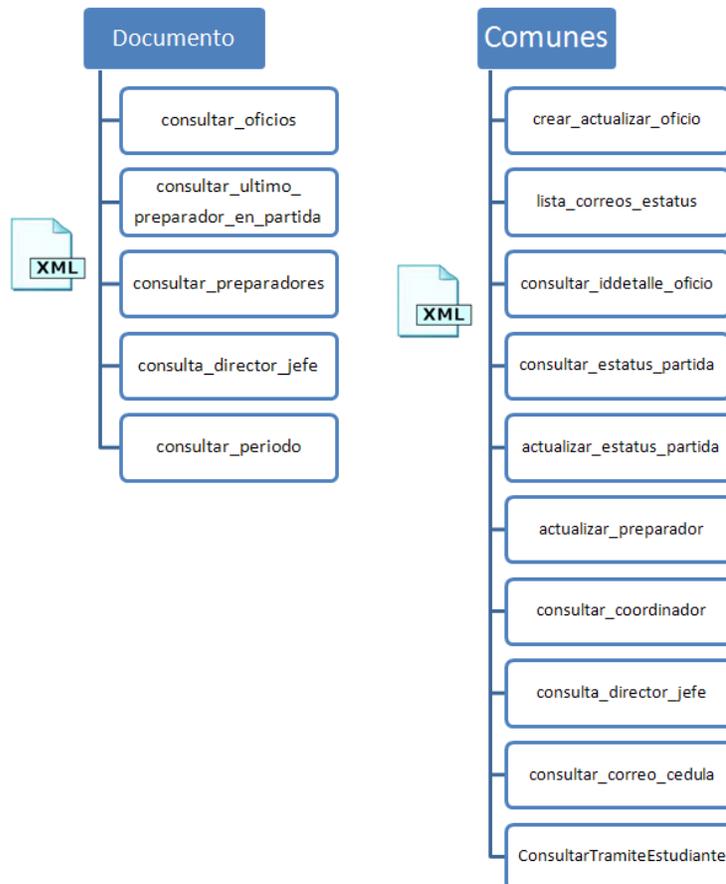


Figura 42. Diagrama Interacción entre controladores y servicios Web – Parte II.

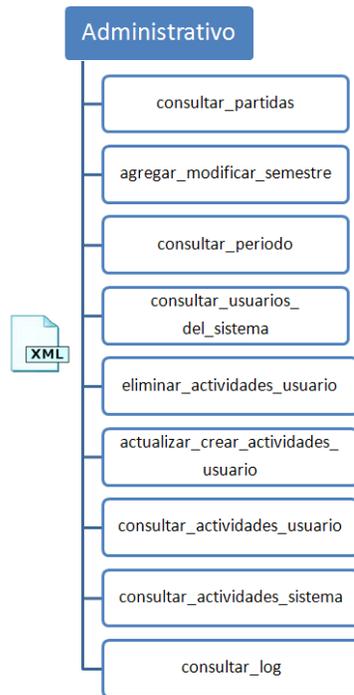


Figura 43. Diagrama Interacción entre controladores y servicios Web – Parte III.

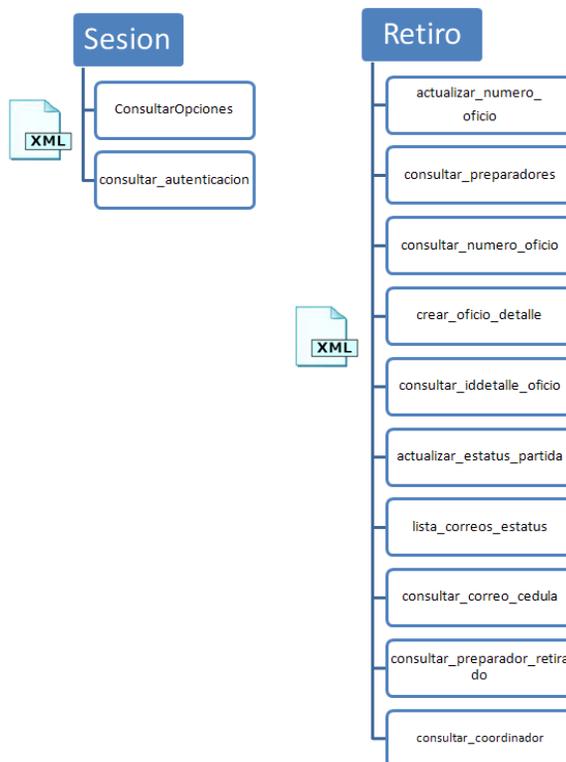


Figura 44. Diagrama Interacción entre controladores y servicios Web – Parte IV.

La estructura del sistema representada bajo *Ruby on Rails* se observa en la Figura 45; en la carpeta “app” es donde se concentra el código fuente, las otras son de soporte para el proyecto.

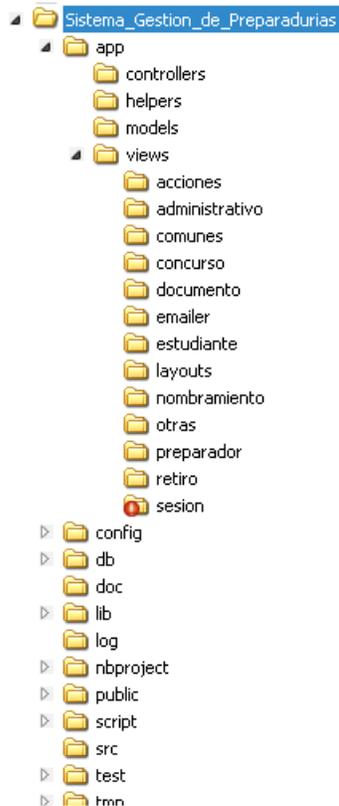


Figura 45. Estructura de la aplicación generada usando *Ruby on Rails*.

2.7.4. Fase de Transición

El sistema se desarrolló de tal manera que la fase de transición tenga el menor impacto posible. Esta fase no se alcanzó completamente, debido a que el sistema necesita contar con el desarrollo completo de otros sub-sistemas que serán mencionados más adelante. Sin embargo, el SGP-EC ha sido alojado en un servidor de la Institución ubicado en el Centro de Computación de la Facultad de Ciencias, habiendo previamente preparado el ambiente de producción para el cual fue necesario cumplir con los siguientes requerimientos: instalación de *Ruby* 1.8.6 y *Rails* 2.3.2, así como de las gemas para el envío de correo electrónico: *tmail*; para la funcionalidad de paginación en las vistas:

will_paginate; para la creación de archivos en formato PDF: *pdf-write*. En la Figura 24 se observa la configuración de los servidores necesarios para el funcionamiento del sistema.

Adicionalmente, como parte del cumplimiento de esta fase, se realizó la simulación del sistema haciendo las pruebas respectivas que elaboradas y ejecutadas por el equipo desarrollador.

➤ **Interacción con otros sub-sistemas**

Las interacciones entre los módulos de Gestión de Retiros, Concursos y Nombramientos se refleja en dos momentos, el primero es cuando un estudiante es nombrado preparador, a partir de ese momento se podría comenzar a tramitar el retiro de dicho preparador, el segundo caso, ocurre cuando un preparador es efectivamente retirado, quedando cargos vacantes para alguna materia, pudiéndose activar nuevamente el ciclo de concurso. Las interacciones entre el subsistema de Gestión de Retiros de preparadores y los subsistemas de Gestión de Concurso y Nombramiento, Sistema de Gestión Presupuestaria y el Sistema de manejo de Base de Datos y servicios *Web* se pueden ver en la Figura 23.

La siguiente interacción que puede ocurrir es entre el módulo de Gestión de Retiro y el sub-sistema de Gestión Presupuestaria, cuando un coordinador solicita un retiro de un preparador, lo que conlleva a que la partida asignada al mismo se debe cambiar de estatus y notificar al personal de presupuesto que se va a generar un movimiento o trámite con dicha partida; la otra interacción con este sub-sistema, es cuando un retiro de un preparador se hace efectivo, dado que se libera la partida presupuestaria que tenía asignada y de igual forma se notifica al personal de presupuesto sobre el trámite correspondiente a dicha partida.

➤ **Interacción con la base de datos**

Se tiene una interacción considerada vital y es con el sistema que desarrolla los servicios *Web* para las consultas y modificaciones de la base de datos, las cuales se llevan a cabo cada vez que se realiza cualquier acción o consulta desde en las vistas del sistema.

Existen ventajas y desventajas de trabajar bajo este esquema, la ventaja es que la aplicación se hace totalmente independiente de la base de datos, de su ubicación o del manejador que se esté usando; pero existe una posible desventaja producto de la dependencia de las respuestas de los servicios *Web* y es el impacto que tenga en el rendimiento de la aplicación.

La intención de éste sistema es acceder a los información contenida en las bases de datos existentes en la Facultad de Ciencias: LDAP (base de datos que contiene los datos

del personal administrativos, profesores y estudiantes de la Facultad de Ciencia y es administrada por el Centro de Computación), CONEST, Preparadores, Presupuesto, a través de la invocación o consumo de funciones o procedimientos alojados en algún servidor, que puede ser o no el mismo de la aplicación.

➤ Aspectos para la implantación

Debido a que este trabajo de grado no abarca la puesta en producción de los módulos de Gestión de Retiros y Administrativo, se explicará a continuación los pasos necesarios para llevar a cabo dicha acción.

Primero se debe contar con el desarrollo de todos los servicios *Web* solicitados y con las pruebas previas de acceso a estos servicios *Web* desde el SGP-EC. Para realizar estas pruebas es necesario conocer la ruta real donde se alojan los servicios *Web*, que se debe especificar en el método *WSDL* del controlador *Application*.

Actualmente el envío de correos que se realiza desde la aplicación se hace mediante una cuenta de un correo *gmail*: `sistemapreparaduria@gmail.com`, lo ideal sería usar una cuenta propia del servidor institucional, por ejemplo: `sistemapreparaduria@tyto.ciens.ucv.ve`, para lo cual, hace falta realizar las pruebas necesarias de envío de correos a través de este servidor.

Es importante recalcar, que el sistema debe contar con una serie de datos que deben estar previamente cargados en la base de datos para su correcto funcionamiento. Las tablas que deben poseer datos en el arranque del sistema son:

- **cargo:** Cuyos valores iniciales deben ser al menos los siguientes y en el mismo orden, para conservar el identificador: coordinador, empleado, estudiante, director, jefe de departamento, jefe de centro, secretaria del departamento, asistente administrativa, secretaria del consejo de escuela, preparador I, preparador II, *adhonorem1*, *adhonorem2*, auxiliar docente.
- **partida:** Son las partidas presupuestarias que tiene asignada la Escuela en el momento que arranque el sistema.
- **estatus_partida:** Sus valores iniciales son: Disponible, No Disponible, Pre-asignable, No Pre-asignable.
- **estatusOficio:** Oficio nuevo, Oficio generado, Oficio revisado, Oficio Aprobado, Oficio Negado, Oficio Diferido, Firmado por Director(a), Oficio Archivado, Enviado a Presupuesto.
- **programacionDocente:** Debe estar cargada la programación docente del semestre que va a iniciar cuando el sistema sea puesto en producción para poder realizar las aperturas de concursos.

- **tipoMovimiento:** nombramiento, retiro, aumento de horas, disminución de horas.
- **materia:** Toda la información de las materias de la Escuela, tanto básicas como electivas.
- **estudiante:** Lista de todos los estudiantes de la Escuela, éstos ya se encuentran cargados en la base de datos de Control de Estudios.
- **accionPreparaduria:** Se refiere a las acciones que un usuario puede realizar en el sistema, con el fin de llevar un control a través del *log* sobre qué actividades ha realizado un usuario en un tiempo determinado.
- **actividad:** Es muy importante que esta tabla se encuentre cargada, para que el administrador pueda asignar las actividades a cada usuario del sistema.
- **departamento:** Son los departamento(s) de las escuelas.
- **escuela:** Las escuelas de la Facultad de Ciencias
- **oficio_escuela:** Esta tabla lleva el control de los números de oficios que genera automáticamente el sistema por escuela, se debe inicializar en el número del primer oficio que se genere con el sistema.

CONCLUSIONES

Al finalizar este TEG se pudo obtener importantes conclusiones, una de ellas y tal vez la más importante es que para la realización de un sistema de información se necesita disponer de metodologías que permitan la adaptación del equipo de trabajo a las situaciones que se presenten durante el desarrollo debido a los factores que intervienen como el tiempo disponible, equipo de trabajo, los límites y el alcance del proyecto, entre otros.

Con la metodología de desarrollo de software aplicada, *OpenUp*, se obtuvieron los resultados esperados sin interferir en el desarrollo del proyecto y se mejoró el desempeño de los integrantes del equipo, porque permitió tener siempre presente los requerimientos funcionales a implementar y de este modo contar con una mejor organización, cumpliéndose así uno de los objetivos específicos que era aplicar dicha metodología en la implementación de los módulos de Gestión de Retiros y Administrativo.

Otro de los objetivos planteados fue ajustar el *workflow* del proceso de gestión de retiros, para afinar detalles y definir las funcionalidades del módulo Administrativo, el cual se logró mediante un análisis minucioso de cada actividad en las fases de inicio y elaboración de la metodología.

La limitante principal que surgió en las fases, fue el no contar con el desarrollo previo de los subsistemas necesarios para poner en funcionamiento los módulos de Gestión de Retiros y Administrativo, razón por la cual al momento de realizar las pruebas de funcionalidad en conjunto con los usuarios, no se pudieron llevar a cabo y en lugar de éstas se realizó una simulación. Por lo tanto, se recomienda realizar los pasos necesarios para ejecutar la fase de transición explicada en el Capítulo 3 del presente trabajo, logrando de esta manera, el propósito para el que se construyeron éstos módulos.

La uso del *workflow* para la especificación de los procesos resultó muy apropiada, pues facilita el entendimiento completo del flujo de trabajo frente a cualquier tipo de usuario y permite ver la secuencia desde el inicio hasta el final de un proceso sin importar el usuario o rol que ejecuten las tareas necesarias durante dicho proceso.

La utilización de *Ruby on Rails* como herramienta tecnológica en la construcción de los módulos de Gestión de Retiros y Administrativo fue una ventaja porque además de ser muy intuitiva, el equipo contaba con experiencia previa en el uso de la misma.

Con el cumplimiento de los objetivos específicos, se consiguió desarrollar el módulo de Gestión de Retiros, del SGP-EC, en conjunto con el módulo Administrativo del sistema, alcanzando el objetivo principal de este trabajo especial de grado. Adicionalmente se desarrollaron funcionalidades comunes para los módulos del SGP-EC, siendo estas:

generación de PDF, envío de correo, paginación de registros en las vistas, generación de constancias, consultas de datos históricos de preparadores y las interfaces que permiten dar seguimiento a los trámites.

Además, es recomendable reforzar el envío, almacenamiento de datos encriptados antes de la puesta en producción, así como la elaboración del manual de usuario para complementar y dar soporte a la capacitación de los usuarios.

Es importante mencionar que a futuro se puede aprovechar este TEG como modelo para aplicarlo en las diferentes escuelas de la Facultad de Ciencias y a su vez en las facultades de la Universidad Central de Venezuela. Aunque el contexto de este trabajo fue la Escuela de Computación, se tomaron en cuenta muchos factores para que otros trabajos puedan aprovechar las funcionalidades desarrolladas para implantarlo en otras escuelas.

Se recomienda hacer pruebas exhaustivas junto con los usuarios para poder asegurar más aún un producto de calidad antes de ofrecerlo como una herramienta de uso cotidiano en la escuela.

Se podría considerar aplicar un motor de *workflow* porque resultaría conveniente al tratar de resolver posibles cambios en la secuencia de ejecución de los procesos, dado que el SGP-EC se realizó basado en un modelo *workflow* estático.

REFERENCIAS

- [AB, 2009] Aguilar Iris, Bellet Isabelle. **Prototipo para la automatización de la Gestión de Preparadores de la Escuela de Computación. Módulos Retiro y Administrativo**. Seminario. Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, (2009)
- [Asocalidad] Asocalidad. Disponible en:
[HTTP://www.asocalidad.org.ve/diccionario-g.htm#gestionprocesos](http://www.asocalidad.org.ve/diccionario-g.htm#gestionprocesos)
- [Ben, 2003] Benavides Luis, J. **Gestión por Procesos**. Disponible en:
[HTTP://www.calidadlatina.com/pub/036-JUL-03.pdf](http://www.calidadlatina.com/pub/036-JUL-03.pdf)
- [Cer, 2002] Cerami Ethan. **Web Services Essential**. O'Reilly Media, Inc., (2002)
- [Dia] Díaz T Manuel D. **Web Services. Introducción y Escenarios para su Uso**. Disponible en:
[HTTP://www.moisesdaniel.com/es/wri/wsepsu.htm](http://www.moisesdaniel.com/es/wri/wsepsu.htm)
- [DummyonRails, a] **Arquitectura de Rails**. Disponible en:
[HTTP://dummyonRails.lacotelera.net/post/2007/06/08/arquitectura-Rails](http://dummyonRails.lacotelera.net/post/2007/06/08/arquitectura-Rails)
- [DummyonRails, b] **Estructura de una aplicación Rails**. Disponible en:
[HTTP://dummyonRails.lacotelera.net/post/2007/06/08/estructura-una-aplicacion-Rails](http://dummyonRails.lacotelera.net/post/2007/06/08/estructura-una-aplicacion-Rails)
- [EPF] *Eclipse Process Framework Project*. **OpenUp**. Disponible en:
[HTTP://epf.eclipse.org/wikis/OpenUp/index.htm](http://epf.eclipse.org/wikis/OpenUp/index.htm)
- [Fer, 2005] Fernández L. Javier D. **Sistemas organizacionales. Teoría y práctica**. Editorial Universidad Cooperativa de Colombia (Educc), Bogotá, (2005).
- [FM, 2008] Flanagan David, Matsumoto Yukihiro. **The Ruby programming language**. O'Reilly Media, Inc., (2008).
- [Hib, 2005] Hibbs Cliff. **Rolling with Ruby on Rails**. Disponible en:
[HTTP://oreilly.com/Ruby/archive/Rails.html](http://oreilly.com/Ruby/archive/Rails.html)
- [GR, 2009] Gouveia Rosa, Ruiz Isaura. **Prototipo para la automatización de la Gestión de Preparadores de la Escuela de Computación. Módulos Concurso y Nombramiento**. Seminario. Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, (2009)

- [Kioskea] Kioskea. Disponible en:
[HTTP://es.kioskea.net/contents/entreprise/workflow.php3](http://es.kioskea.net/contents/entreprise/workflow.php3)
- [LD, 2004] Laborda C. Leopoldo, De Zuani Elio. **Fundamentos de Gestión Empresarial**. Valletta Ediciones SRL, Buenos Aires, (2004).
- [Mue, 2004] Muehlen Michael. **Workflow - based Process Controlling**. Logos, (2004).
- [New, 2002] Newcomer Eric. **Understanding Web services: XML, WSDL, SOAP, and UDDI**. Pearson Education, Inc., (2002).
- [Per, 2007] Pérez F de V. José. A. **Gestión por procesos**. ESIC Editorial, Madrid, (2007).
- [Pixelware] Pixelware. Disponible en:
[HTTP://www.pixelware.com/workflow-flujo-trabajo.htm](http://www.pixelware.com/workflow-flujo-trabajo.htm)
- [Rails] Rails Project. Disponible en:
[HTTP://guides.RubyonRails.org/getting_started.html](http://guides.RubyonRails.org/getting_started.html)
- [STK, 2002] Snell James, Tidwell Doug, Kulchenko Pavel. **Programming Web services with SOAP**. O'Reilly Media, Inc., (2002).
- [TH, 2006] Tate Bruce, Hibbs Curt. **Ruby on Rails: up and running**. O'Reilly Media, Inc., (2006)
- [Wil, 2007] Williams Justin. **Rails solutions: Ruby on Rails made easy**. Friends for ED, (2007).
- [W3C] World Wide Web consortium - W3C. Disponible en:
[HTTP://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb#uso](http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb#uso)
- [Zor, 2007] Zorrilla C. Unai. **Modelando procesos de negocio con Workflow Foundation**. Krasis Press, (2007).

ANEXOS

Anexo 1. Plantilla visión del proyecto

<Project Name> Vision

Nota de uso: Existen guías en esta plantilla que aparecen en un estilo llamado InfoBlue. Este estilo tiene un atributo de fuente oculta que permite hacerla visible o ocultarla en esta plantilla. Use el menu Herramientas → Opciones → Vista → Texto Oculto checkbox para activar esta opción. Una opción similar existe para la impresión Herramientas → Opciones → Imprimir

1. Introducción

2. Descripción del Problema y Solución propuesta

2.1 Enunciado del Problema

[Provea un enunciado en el que se resume el problema a ser resuelto por este proyecto. Puede usarse el siguiente formato:]

El problema es	<i>[describa el problema]</i>
A quien afecta	<i>[los stakeholders afectados por el problema]</i>
Cuál es su impacto	<i>[cual es el impacto del problema?]</i>
Actualmente cómo se atiende	<i>[tratamiento actual del problema?]</i>
la solución satisfactoria sería	<i>[listar los beneficios de la solución satisfactoria]</i>

2.2 Descripción solución propuesta

[Proveer un enunciado que resuma, en un alto nivel, la solución propuesta. El siguiente formato puede ser usado:]

Beneficiarios	<i>[objetivo del cliente]</i>
Usuarios poenciales	<i>[A quién está dirigido la solución]</i>

Resultados previstos	<i>[Descripción de la solución]</i>
Ventajas	<i>[declaración de las principales ventajas]</i>

3. Objetivos

[Objetivos general y específicos del proyecto]

3.1 Objetivo General

3.2 Objetivos Especificos

4. Descripción de los Actores (Stakeholders)

4.1 Resumen de los Actores

Name	Description	Responsibilities
<i>[Nombre del tipo de stakeholder.]</i>	<i>[Breve descripción del stakeholder.]</i>	<i>[Resumir las responsabilidades claves del stakeholder con respecto al sistema que esta siendo desarrollado, es decir, sus intereses como stakeholder. Por ejemplo, Este stakeholder asegurará que el sistema sea mantenible, asegurará que habrá una demanda del mercado para las características del producto, monitoreará el progreso del proyecto, aprobará el financiamiento, etc.]</i>

4.2 Ambiente de usuario

[Detalle el ambiente de trabajo del usuario final. Aquí algunas sugerencias:

¿Número de personas involucradas en completar la tarea? ¿Es esta un cambio?

¿Cuan largo es el ciclo de la tarea? ¿Cantidad de tiempo invertido en cada actividad? ¿Es esta un cambio?

¿Alguna restricción ambiental: móvil, al aire libre, en vuelo, etc.?

¿Cuál sistema operativo es usado actualmente? ¿Cuál es la plataforma futura?

¿Qué otra aplicación está en uso? ¿La aplicación necesita integrarse con esta?

Aquí pueden incluirse extractos del Modelo de Negocio para esbozar actividades y funciones involucradas.]

5. Descripción del producto

5.1 Características y necesidades

[Evitar el diseño. Tratar de mantener una descripción a nivel general. Centrarse en las capacidades necesarias y por qué (no a nivel técnico) estas deben ser implementadas. Capturar las prioridades de los stakeholders y planificar los release de cada característica.]

Necesidad	Prioridad	Características	Planificación del release

6. Otros requerimientos del producto

[En un alto nivel, listas de estándares aplicables, hardware, o requerimientos de plataforma; requerimientos de desempeño; y requerimientos ambientales.

Definir los rangos de calidad para rendimiento, robustez, tolerancia a fallas, usabilidad y características similares que no son capturadas en el conjunto de características.

*Tenga en cuenta las restricciones de diseño, restricciones externas, hipótesis y otras dependencias que, si cambian, alterarán el documento **Visión**. Por ejemplo, una hipótesis puede indicar que un sistema operativo específico está disponible para el hardware diseñado para el producto. Si el sistema operativo no está disponible, el documento **Visión** tendrá que cambiar.*

Definir los requisitos de documentación específicos, incluyendo manuales de usuario, ayudas en línea, instalación, etiquetamiento y empaquetamiento.

Definir la prioridad de estos otros requerimientos del producto. Incluir, si se considera oportuno, atributos tales como estabilidad, esfuerzo y riesgos.]

Requerimiento	Prioridad	Planificación del release

Anexo 2. Descripción de los símbolos del *Workflow*

Símbolo	Descripción
	Representa los documentos físicos que se manejan en el proceso, tales como: oficio de retiro y carta de renuncia.
	Representa las actividades que ejecuta el sistema siendo activadas por un rol.
	Representa las actividades que ejecuta el sistema automáticamente en algún punto del proceso.
	Representa el envío un correo electrónico con información pertinente al evento que active por un rol.
	Representa la recepción de un correo electrónico con información pertinente al evento activado por otro rol.
	Representa el intercambio de información con sistemas externos, a través de mensajes XML haciendo uso de servicios web.
	Representa una interfaz de consulta de información.
	Representa un formulario para la carga de información.
	Representa la continuidad del flujo de actividades, utilizando números consecutivos.
	Representa el inicio de un proceso.
 	Representa el fin de un proceso. Representa una caja negra indicando que lo se ejecuta es transparente al proceso en curso.

Anexo 3. Descripción del servicio *Web*: crear_oficio_detalle

Service Contract Definition					
Service Name		crear_oficio_detalle			
No. Service		ws-00003			
Service Description		Crea oficio			
Service Type		inserción			
Tablas		oficiodetalle			
Input Data Structure					
S.No.	Detail	Type	Default Value	Validation or Business Rules	Description
Nombre					
	cod_escuela	int		no puede ser null	
	cedula	int		no puede ser null	log
	numoficioprepa	string		no puede ser null	
	idestatus	int		no puede ser null	
	fechaestatus	date		null	
Output Data Structure					
S.No.	Detail	Type	Default Value	Validation or Business Rules	Description
Nombre					
1	acuse	int	1	inserción satisfactoria	
2					

Anexo 4. Descripción del servicio Web: consultar_oficios

Service Contract Definition					
Service Name		consultar_oficios			
No. Service		comun_ws-000020			
Service Description		Devuelve toda la información de uno o varios oficios filtrados por los parámetros de entrada.			
Service Type		consulta			
Tablas		oficio	estudiante		
Input Data Structure					
o.	Detail	Type	Value	Rules	Description
	Nombre				
	cod_escuela	int		no puede ser vacío	
	cedula	int		no puede ser vacío	log
1	cedula	int		Al menos un parámetro debe ser distinto de null	
2	numoficioprepa	string			
	idestatus	int			= 456 sede retornar todos los registros que correspondan a los estatus 4, 5 y 6
	periodo	string			siempre es el periodo actual
	cargo	string			
Output Data Structure					
o.	Detail	Type	Value	Rules	Description
	Nombre				
1	numoficio	string			
2	estatus	string			
3	cedula	int		es la cedula del preparador	
4	iddetalle	int			
5	tipomov	int			1: retiro
6	fechaefectiva	date			formato: dd/mm/aaaa
7	fechaoficio	date			formato: dd/mm/aaaa
8	fecharesolucion	date			formato: dd/mm/aaaa
9	fechace	date			formato: dd/mm/aaaa
10	nombre	string		es el nombre del preparador	
11	apellido	string		es el apellido del preparador	
12	fechaestatus	date			formato: dd/mm/aaaa
13	unidadejecutora	string			
14	programa	string			
15	materia	string		nombre completo de la materia	
16	cargo	string		nombre del cargo de ese	
17	dedicacion	string			ejemplo: "seis horas"
18	sexo	char			
19	fechaingreso	date		donde preparador.fechaingreso	
	numOficioprepa	string			
	director	string			
	jefe_dpto	string			
	periodo	string			
	requisito_fisico	boolean		valores posibles: "true", "false"	