



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Diseño e Implementación de un sistema para la gestión de viáticos

presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres:

César Mora
C.I.: 18.467.695

Viviana Delgado
C.I.: 20.049.256

para optar al título de Licenciado en Computación

Tutores: Prof. Eugenio Scalise
Profa. Adriana Liendo

Caracas, Julio 2013

Resumen

El presente Trabajo Especial de Grado se basó en el desarrollo de una aplicación web para la automatización los procesos y trámites de la solicitud de viáticos que realizan los empleados de la Fundación Venezolana de Investigaciones Sismológicas (Funvisis), con el fin de reducir costos de tiempo, papel y recursos humanos para que puedan ser utilizados para otra actividad. El proyecto estuvo enmarcado dentro de la metodología de la programación extrema (XP). Adicionalmente, para el desarrollo de la aplicación se utilizaron diversas tecnologías como el *framework* Ruby on Rails y el sistema manejador de base de datos PostgreSQL. Como resultado se obtuvo un sistema que facilita la automatización de estos trámites, cumpliendo las normas y reglamentos que rigen estos procesos, alcanzando los beneficios de la simplificación del trabajo que debe realizar el personal operativo de la institución, la mejora en el seguimiento de las solicitudes y los tiempos de respuesta por parte de los departamentos involucrados.

Palabras clave: Fundación Venezolana de Investigaciones Sismológicas (Funvisis), Programación Extrema (XP), Ruby on Rails (RoR), PostgreSQL.

Tabla de Contenido

ÍNDICE DE FIGURAS	4
ÍNDICE DE TABLAS.....	5
INTRODUCCIÓN	7
CAPÍTULO I: FUNDACIÓN VENEZOLANA DE INVESTIGACIONES SISMOLÓGICAS ..	9
1.1 DESCRIPCIÓN ORGANIZACIONAL	10
1.2 REGLAMENTO DE VIÁTICOS	13
1.3 GESTIÓN DE VIÁTICOS	20
CAPÍTULO II: MARCO TEÓRICO Y TECNOLÓGICO.....	25
2.1 PROGRAMACIÓN EXTREMA (XP)	25
2.2 RUBY ON RAILS.....	32
2.3 POSTGRESQL	34
CAPÍTULO III: MARCO APLICATIVO	36
3.1 DEFINICIÓN DEL PROBLEMA.....	36
3.2 OBJETIVO GENERAL	37
3.3 OBJETIVOS ESPECÍFICOS	37
3.4 ANTECEDENTES	38
3.5 PLANIFICACIÓN DEL PROYECTO	40
3.6 PLAN DE ENTREGA	48
3.6.1 ITERACIÓN I – INGRESO Y ADMINISTRACIÓN.....	49
3.6.2 ITERACIÓN II - SOLICITUD DE VIÁTICOS	63
3.6.3 ITERACIÓN III - RENDICIÓN DE CUENTAS.....	74
3.6.4 ITERACIÓN IV - APROBACIONES ITERACIÓN	85
CONCLUSIONES Y RECOMENDACIONES	93
REFERENCIAS	95
ANEXOS	97

Índice de Figuras

Figura 1.1 Estructura organizacional de Funvisis.....	13
Figura 1.2 Diagrama de actividades del proceso actual de solicitud de viáticos.....	23
Figura 1.3 Diagrama de actividades del proceso actual de rendición de cuentas.....	24
Figura 2.1 Evolución de los largos ciclos de desarrollo en cascada (a) a ciclos iterativos más cortos (b) y a la mezcla que hace XP.....	25
Figura 2.2 Paradigma de la arquitectura Modelo Vista Controlador.....	33
Figura 2.3 Componentes importantes en un sistema PostgreSQL.....	35
Figura 3.1 Diagrama de actividades de la propuesta de automatización de solicitud de viáticos.....	41
Figura 3.2 Diagrama de actividades de la propuesta de automatización de rendición de cuentas.....	42
Figura 3.3 Diagrama de la base de datos en la primera iteración.....	54
Figura 3.4 Sistema de Gestión de Memos.....	57
Figura 3.5 Pantalla para la selección del módulo.....	58
Figura 3.8 Código del ApplicationHelper.....	59
Figura 3.7 Lista de países.....	60
Figura 3.8 Vista de nuevo imprevisto.....	61
Figura 3.9 Vista de de la edición de un proyecto.....	62
Figura 3.10 Vista de una nueva acción.....	62
Figura 3.11 Diagrama de la base de datos en la segunda iteración.....	66
Figura 3.14 Código ActiveRecord de viáticos.....	67
Figura 3.15 Código de los pasos para el formulario de viáticos.....	68
Figura 3.14 Campos dinámicos para la generación de imprevistos.....	68
Figura 3.17 Código donde se crea un pdf con su código de barra.....	69
Figura 3.16 Vista de la solicitud de viáticos general.....	71
Figura 3.17 Vista de la solicitud de viáticos para el interior del país.....	72
Figura 3.18 Vista para la confirmación de los datos de la solicitud de viáticos.....	73
Figura 3.19 Diagrama de la base de datos en la tercera iteración.....	78
Figura 3.22 Código del método searchviatico.....	80
Figura 3.21 Vista de relación de gastos por día.....	81
Figura 3.22 Vista de Nueva rendición.....	83
Figura 3.23 Vista de la rendición de cuentas con la información indicada por el usuario..	84
Figura 3.26 Código para el rechazo de las solicitudes.....	89
Figura 3.25 Lista de solicitudes de viáticos pendientes.....	90
Figura 3.26 Lista de solicitudes de rendiciones de cuentas pendientes.....	90
Figura 3.27 Vista de solicitud de viáticos para la asignación de la imputación.....	91
Figura 3.28 Vista de aprobación para una rendición de cuentas.....	92

Índice de Tablas

Tabla 1.1 Niveles de Autorización de las Actividades dentro del país.....	15
Tabla 1.2 Viáticos diarios dentro del país.	16
Tabla 1.3 Gastos generados por traslados urbanos, suburbanos e interurbanos.	17
Tabla 1.4 Niveles de Autorización de las Actividades fuera del país.	18
Tabla 3.1 Historia de usuario: Autenticación de usuario	44
Tabla 3.2 Historia de usuario: Administración del sistema.....	44
Tabla 3.3 Historia de usuario: Solicitud de viáticos para el interior del país.....	45
Tabla 3.4 Historia de usuario: Solicitud de viáticos para el exterior del país.....	45
Tabla 3.5 Historia de usuario: Solicitud de imprevistos.....	46
Tabla 3.6 Historia de usuario: Rendiciones de cuenta.....	46
Tabla 3.7 Historia de usuario: Aprobación de solicitud.	47
Tabla 3.8 Historia de usuario: Aprobación de rendiciones.....	47
Tabla 3.9 Planificación de entrega del proyecto.	48
Tabla 3.10 Historia de usuario: Registro de administradores del sistema.....	49
Tabla 3.11 Historia de usuario: Administración de las U.T asignadas para las solicitudes.	50
Tabla 3.12 Historia de usuario: Registro de imprevistos	50
Tabla 3.13 Historia de usuario: Registro de proyectos.	50
Tabla 3.14 Historia de usuario: Registro de acciones.....	50
Tabla 3.15 Historia de usuario: Registro de países.	51
Tabla 3.16 Tarea: Conexión de la base de datos y el sistema.....	51
Tabla 3.17 Tarea: Creación de la base de datos para la administración del sistema.....	52
Tabla 3.18 Tarea: Mensajes de validación para la administración del sistema.	52
Tabla 3.19 Tarea: Creación de formularios para la administración del sistema.	52
Tabla 3.20 Tarea: Formularios para la actualización de la administración del sistema....	53
Tabla 3.21 Tarea: Formularios para la actualización de la administración del sistema....	53
Tabla 3.22 Tarea: Listado de atributos para la administración del sistema.....	53
Tabla 3.23 Historia de usuario: Solicitud de viáticos general.	64
Tabla 3.24 Tarea: Creación de las tablas para la solicitud de viáticos.	64
Tabla 3.25 Tarea: Mensajes de validación para la solicitud de viáticos.	64
Tabla 3.26 Tarea: Creación de formularios para la solicitud de viáticos.	65
Tabla 3.27 Tarea: Listado de solicitudes.	65
Tabla 3.28 Tarea: Creación de status de solicitudes.	65
Tabla 3.29 Tarea: Creación del módulo de pasos para la solicitud de viáticos.	65
Tabla 3.30 Historia de usuario: Rendiciones.	74
Tabla 3.31 Historia de usuario: Relaciones interiores.....	75
Tabla 3.32 Historia de usuario: Relaciones exteriores.....	75
Tabla 3.33 Historia de usuario: Relaciones generales.....	76
Tabla 3.34 Tarea: Creación de las tablas para la rendición de cuentas.....	76
Tabla 3.35 Tarea: Mensajes de validación para la rendición de cuentas.	76
Tabla 3.36 Tarea: Creación de formularios para la rendición de cuentas.	77

Tabla 3.37 Tarea: Listado de rendición de cuentas.	77
Tabla 3.38 Tarea: Creación de status de rendiciones de cuentas.	77
Tabla 3.39 Tarea: Creación del módulo de pasos para la rendición de cuentas.	78
Tabla 3.40 Tarea: Modificación de la base de datos para las aprobaciones.	85
Tabla 3.41 Tarea: Validación de usuario para aprobar en más de un nivel de aprobación.	86
Tabla 3.42 Tarea: Listado de solicitudes.	86
Tabla 3.43 Tarea: Listado de rendición de cuentas.	86
Tabla 3.44 Niveles de aprobación de las solicitudes.	88
Tabla 3.45 Niveles de aprobación de las rendiciones.	88

Introducción

En el último siglo el desarrollo industrial y las diversas exploraciones para crear nuevos avances desde el punto de vista tecnológico han ocasionado una aceleración en los métodos de producción, generando así, hoy en día la necesidad de automatización dentro de las instituciones, empresas e industrias, entendiendo a ésta como aquél sistema en el cual se transfieren diversas tareas y actividades que por lo general son desempeñadas por personas, a una serie o conjunto de elementos tecnológicos como dispositivos electrónicos y computacionales, teniendo como principales objetivos: mejorar la productividad haciéndola de manera más rápida, reducir los costos de producción, mejorar la calidad de las condiciones de trabajo y la simplificación de las diversas operaciones que se deban realizar.

Este Trabajo Especial de Grado surge debido a la necesidad que posee la Fundación Venezolana de Investigaciones Sismológicas (Funvisis) en automatizar su proceso de solicitud de viáticos, siendo éstos el conjunto de provisiones o dinero que se les entrega a los trabajadores para la realizar una actividad encomendada por la fundación dentro o fuera de la República Bolivariana de Venezuela. El proceso de solicitud de viáticos consiste en diferentes trámites a realizar, los cuales son realizados con gran frecuencia por los empleados de la institución e involucra un manejo de grandes volúmenes de papeles (documentos) asociados a la solicitud, para lo cual se desarrolla un sistema capaz de gestionar cada una de las solicitudes y permita realizar las aprobaciones por las personas encargadas, organizando y agilizando el proceso administrativo.

Para el desarrollo del presente trabajo, se dispuso de tres (3) capítulos. El primero describe a Funvisis, su historia, los objetivos y las funciones que desarrolla, además de su estructura organizacional y del reglamento por el cual se rige la institución para el manejo de los viáticos, con el propósito de brindar al

lector un conocimiento sobre la institución para la comprensión del proceso de gestión de viáticos.

En el segundo capítulo, se describe el método ágil Programación Extrema (XP) utilizado para el desarrollo de la aplicación. De igual manera, se mencionan y describen brevemente las herramientas tecnológicas Ruby on Rails y el manejador de base de datos PostgreSQL utilizados en el transcurso del diseño y desarrollo de la aplicación.

En el tercer capítulo se presenta el problema a resolver, y los objetivos que debe cumplir la aplicación para alcanzar la automatización en el sistema de solicitud de viáticos, describiendo detalladamente la configuración de la metodología XP y la documentación del proceso para el desarrollo del software. Por último se presentan las conclusiones y recomendaciones para la adecuada utilización del sistema diseñado.

Capítulo I: Fundación Venezolana de Investigaciones Sismológicas

Luego del terremoto que tuvo lugar en Caracas- Venezuela el 29 de Julio de 1967 que alcanzó una magnitud de 6.7 en la escala de Richster y un balance de 235 muertos, aproximadamente 2000 heridos y pérdidas materiales que aproximan los 10 millones de dólares, despertó el interés por parte del gobierno nacional en crear organismos o instituciones que se dedicaran a la investigación de estos fenómenos y de evaluar la vulnerabilidad sísmica que pueda existir en el territorio nacional, para crear medidas capaces de reducir los riesgos y capacitar a la población ante un evento sísmico que pudiera ocurrir. Para ello se crea la Fundación Venezolana de Investigaciones Sísmicas (Funvisis), mediante el decreto N° 1053, publicado en Gaceta Oficial N° 29864 de fecha de 27 de julio de 1972.

Esta fundación además de las investigaciones, es el ente encargado de operar y mantener la Red Sismológica y la Red Acelerográfica nacional, y se encarga a su vez de divulgar el conocimiento relacionado con las técnicas de prevención mediante el programa Aula Sísmica, que promueve la formación de personal especializado en el área sismológica, además de impartir conocimiento mediante talleres que se realizan en instituciones educativas, en donde el personal docente, directivo y la comunidad educativa en general quede capacitado para elaborar planes de emergencia y evacuación por medio de simulacros en los planteles educativos.

1.1 Descripción organizacional

Funvisis cuenta con distintos departamentos [1] encargados de llevar a cabo diversas funciones, como se indica a continuación:

- **Consejo Directivo:** Le corresponde la dirección y administración de la Fundación. Dicta las normas generales para el funcionamiento de la misma; acepta las contribuciones y aportes hechos a la Fundación para el cumplimiento de sus objetivos; conoce y aprueba el presupuesto anual; dirige, por intermedio de la Presidencia, las relaciones de la Fundación; rinde informe y cuenta al Ministerio del Poder Popular para Ciencia, Tecnología e Innovación.

Está constituido por el Presidente de la Fundación y por representantes de las siguientes organizaciones:

- Ministerio del Poder Popular para Ciencia, Tecnología e Innovación.
 - Ministerio del Poder Popular para la Educación.
 - Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT).
 - Dirección Nacional de Protección Civil y Administración de Desastres del Ministerio del Poder Popular para Relaciones Interiores y Justicia.
-
- **Auditoría Interna:** Planifica, coordina, ejecuta y vela por el establecimiento del sistema de control interno; auditar el proceso dando prioridad a las áreas críticas dirigidas a mejorar la efectividad y eficiencia de la gestión administrativa de la Fundación. Está integrada por:
 - Coordinación de Control Posterior.
 - Coordinación de Determinación de Responsabilidades.

- **Presidencia:** Le corresponde, entre otras atribuciones, la representación jurídica del organismo conforme a las leyes y estatutos. Su titular preside el Consejo Directivo y la Junta Administradora; dirige y controla las gestiones del Director Técnico y del Director de Administración y Servicios; suscribe en nombre y representación de la Fundación todo tipo de documentos públicos y privados.
- **Consultoría Jurídica:** Es la unidad de apoyo de la Fundación y su función principal es asesorar a las máximas autoridades y a las unidades que la conforman en materia jurídica, para garantizar la legalidad de sus actuaciones.
- **Dirección de Administración y Servicios:** La Dirección de Administración y Servicios, tiene como misión la gerencia, administración y control de todos los recursos financieros y tecnológicos de la Fundación, así como la ejecución de todos los procesos relacionados con el reclutamiento, selección, remuneración y capacitación del personal que en ella presta servicios, de tal forma que alineado con la misión y objetivos generales de esta Fundación, la ejecución de todos los procesos claves de negocio pueda ser llevada a cabo en forma satisfactoria. De la dirección Administrativa y servicios dependen:
 - El departamento RHH
 - El departamento de servicios administrativos
 - El departamento de servicios generales.
- **Dirección de Planificación y Presupuesto:** es la unidad encargada de formular, dirigir y coordinar el desarrollo de planes tanto operativos como estratégicos, el presupuesto de la fundación y llevar el control de la gestión interna de los proyectos que en ella se ejecutan.

- **Dirección Técnica:** Planifica, dirige y controla las actividades que integran los programas relacionados con la ejecución de investigaciones relacionadas con la evaluación de la amenaza sísmica. De la Dirección Técnica dependen varios departamentos:
 - Departamento de Informática,
 - Departamento de Geofísica,
 - Departamento de Instalación Electrónica
 - La Unidad de Geomática
 - Departamento de Ciencias de la Tierra, éste a su vez posee una dependencia: La Unidad de Geotécnica.
 - Departamento de Documentación e Información, éste posee el Museo Sismológico.
 - Departamento de Ingeniería Sísmica.

Las actividades que podrán realizar estos distintos niveles de jerarquía se verán más adelante en el Capítulo III donde se indicará el rol de cada usuario que ingrese en el sistema, así como las distintas actividades que podrá realizar de una manera clara y rápida.

La estructura organizacional de Funvisis se observa en la Figura 1.1.

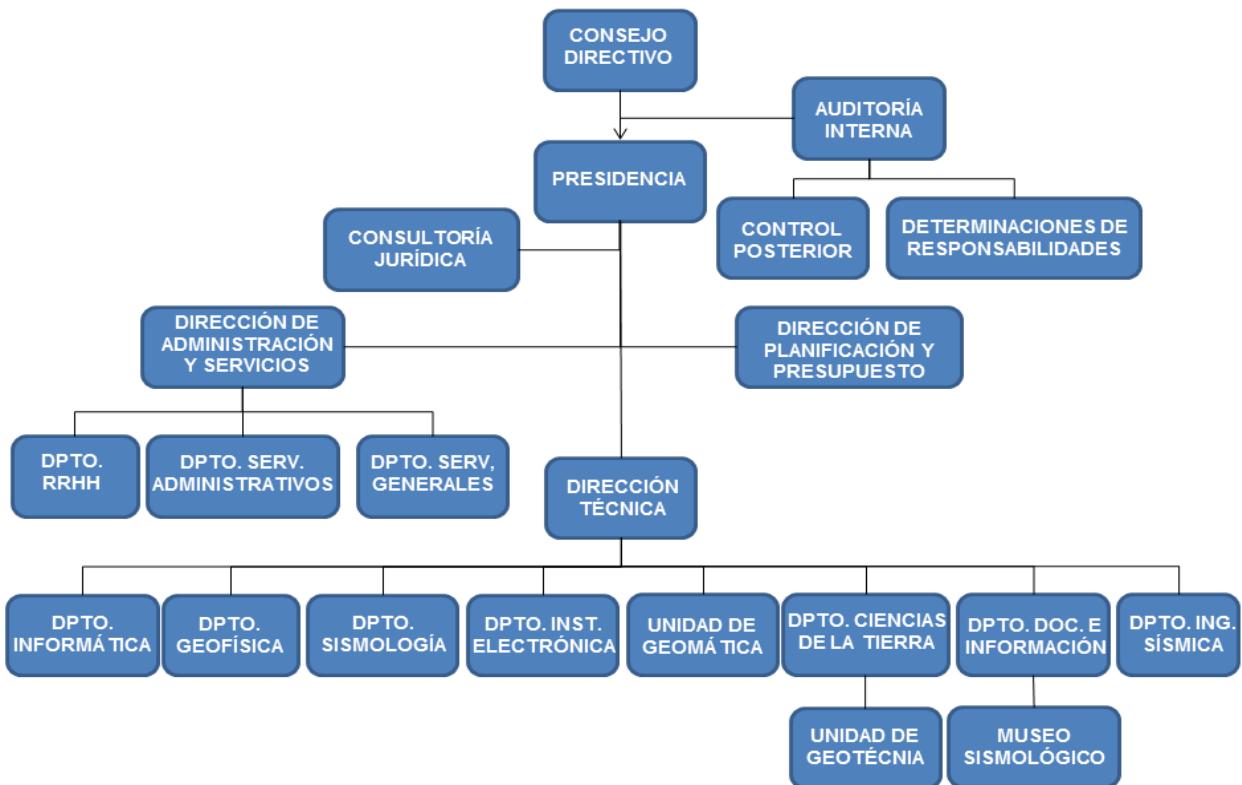


Figura 1.1 Estructura organizacional de Funvisis.

1.2 *Reglamento de viáticos*

Funvisis cuenta con un reglamento (Anexo 1) donde se establecen las normas y estandarizan las actividades del procedimiento de asignación y control de viáticos para el personal de la fundación dentro y fuera del país, derivados de funciones o tareas oficiales a desempeñar.

Para entender todo lo que este proceso conlleva, cabe resaltar los siguientes conceptos del reglamento:

- Actividades Administrativas y Técnicas: Son las realizadas por el personal de la fundación, derivadas del mantenimiento de bienes y equipos de la organización, coordinación y administración de los recursos

presupuestarios, financieros, y de personal de servicios y apoyo jurídico; las cuales tienen por objetivo brindar soporte a todas las actividades de la fundación.

- b) Actividades de Investigación: Son las relacionadas con programas, planes y proyectos de investigación determinados por la fundación, para el cumplimiento de sus objetivos, así como las derivadas de convenios o acuerdos de cooperaciones interinstitucionales o internacionales.
- c) Distrito Metropolitano de Caracas: Es la unidad político territorial de la ciudad de Caracas, goza de personalidad jurídica y autonomía dentro de los límites de la Constitución y la ley. El Distrito Metropolitano de Caracas, está constituido por: El Municipio Libertador del Distrito Capital, los Municipios Sucre, Baruta, Chacao y el Hatillo del Estado Miranda.
- d) Gastos de Alimentación: Son aquéllos que se generan de alimentos y bebidas mientras se realizan las actividades encomendadas por la fundación.
- e) Gastos de Alojamiento: Son aquéllos que se generan por la pernocta en el lugar y durante el tiempo que dure la actividad encomendada por la fundación.
- f) Gastos Complementarios: Son los asociados al desempeño de la actividad a realizar, tales como: gasolina, lubricantes, peajes, impuestos de salida y transporte de bulto o equipajes adicionales siempre y cuando estén asociados a la actividad a realizar.
- g) Gastos Imprevistos: Son gastos no estipulados dentro de las actividades encomendadas por la fundación, los cuales podrán ser pagados siempre

que estén justificados y se hayan generado durante el desarrollo de las actividades encomendadas por la fundación.

- h) Gastos de Traslado: Son aquéllos en que incurre la fundación por la movilización de sus trabajadores, de sus bienes y equipos, mientras dure la ejecución de las actividades encomendadas por la fundación.
- i) Rutas de Transporte: Las rutas de transporte terrestre público en la República Bolivariana de Venezuela, se clasifican en urbanas, suburbanas e interurbanas, de conformidad con la Ley de Tránsito y Transporte Terrestre.
- j) Viáticos: Son las sumas de dinero que se pagan cuando el (la) trabajador (a) desempeña alguna actividad encomendada por la fundación dentro o fuera de la República Bolivariana de Venezuela. Los viáticos serán destinados exclusivamente para el cumplimiento de las actividades relacionadas con ésta.

La solicitud de viáticos puede variar dependiendo de si las actividades son dentro o fuera del país. Si las actividades son dentro de la República Bolivariana de Venezuela se debe tener en cuenta la Tabla 1.1 donde se especifica el nivel de autorización necesario para la aprobación de la solicitud dependiendo del cargo que ejerce el empleado dentro de la institución.

Cargo	Nivel de Autorización
Director (a) Consejo Directivo y personas naturales (sin relación de dependencia).	Presidente (a)

Tabla 1.1 Niveles de Autorización de las Actividades dentro del país.¹

¹ Artículo 6, Reglamento de viáticos y pasajes, Funvisis, Caracas, Marzo 2013. 3p.

Cargo	Nivel de Autorización
Presidente (a)	Presidente (a)
Directores (as) y Consultor (a) Jurídico (a).	Presidente (a)
Jefes de Departamentos.	Director (a) – Presidente.
Trabajadores (as) fijos y contratados por contrato de trabajo.	Jefes de Departamento, avalados por los Directores (dependiendo la adscripción del trabajador(a)) y Directores y aprobados por Presidente.
Director (a) Consejo Directivo y personas naturales (sin relación de dependencia).	Presidente (a)

Tabla 1.1 (cont.) Niveles de Autorización de las Actividades dentro del país.

También hay que tener en cuenta los montos que son asignados al momento de la solicitud de viáticos, si los viáticos son solicitados para una actividad dentro del país, se calculará en base a la unidad tributaria del año fiscal anterior a la vigente, y se ha establecido como se muestra en la Tabla 1.2.

Niveles	Concepto	Viáticos Diarios
Presidente, Directores (as) y Consultor (a) Jurídico (a) Personal, Profesional, Técnico, Administrativo, Coordinadores y Jefes de Departamento.	Alojamiento	7 UT
	Alimentación	4 UT

Tabla 1.2 Viáticos diarios dentro del país.¹

Cuando se realiza una solicitud de viáticos existen ocasiones donde Funvisis no suministra el transporte, ni los gastos que se generan por los traslados

¹ Artículo 7, Reglamento de viáticos y pasajes, Funvisis, Caracas, Marzo 2013. 4p.

urbanos, suburbanos e interurbanos, por lo que durante la ejecución de las actividades encomendadas los viáticos se pagarán conforme a la Tabla 1.3.

Niveles	Concepto
Gasto de Estacionamiento	Contra Factura
Traslado – Residencia – Aeropuerto y viceversa (Máximo)	4 UT (Ida y vuelta)
Traslado de Residencia – Sede del Terminal de Autobuses y viceversa (Máximo)	1.25 UT
Traslado ida y regreso a la sede de Funvisis, dentro del Distrito Metropolitano de Caracas	1.5 UT
Valor del impuesto de salida y de la tasa aeroportuaria	Lo establecido en los Aeropuertos Nacionales.

Tabla 1.3 Gastos generados por traslados urbanos, suburbanos e interurbanos.¹

Al solicitar viáticos para realizar actividades fuera del Distrito Metropolitano de Caracas, donde sea un viaje ida y vuelta y sea suministrado el transporte, sólo se pagará el gasto de alimentación equivalente a un monto de uno punto cinco unidades tributarias (1.5 U.T), previa presentación de las facturas correspondientes. Si al realizar las actividades encomendadas el solicitante de viáticos tiene gastos imprevistos los mismos podrán ser pagados siempre y cuando ocurran durante la ejecución de las actividades y se presenten las facturas correspondientes; también existe el caso que el gasto del solicitante de viáticos al realizar sus actividades fuera menor al límite máximo establecido, por lo que el trabajador debe devolver la cantidad restante a Funvisis o puede suceder que la actividad a realizar se interrumpe antes del tiempo previsto, por lo que el

¹ Artículo 8, Reglamento de viáticos y pasajes, Funvisis, Caracas, Marzo 2013. 4p.

empleado de Funvisis deberá informar en un plazo de cinco (5) días hábiles siguientes a la fecha del hecho y proceder al reintegro del monto total o el monto restante de los viáticos recibidos, según sea el caso.

La solicitud de viáticos para realizar actividades fuera de la República Bolivariana de Venezuela tiene los niveles de autorización mostrados en la Tabla 1.4 para su aprobación. Es importante que la solicitud de viáticos al exterior del país sea tramitada ante la autoridad competente por lo menos con un (1) mes de antelación y en caso de emergencia, el (la) Presidente (a) podrá aprobar el viaje.

Cargo	Nivel de Autorización
Presidente (a).	Ministro (a) de Adscripción
Director (a) del Consejo Directivo.	Presidente (a). Ministro (a) de Adscripción.
Directores (as) y Consultor (a) Jurídico(a) de la Fundación.	Presidente (a). Consejo Directivo – Ministro.
Jefes de Departamentos.	Director (a) – Presidente, Consejo Directivo – Ministro.
Trabajadores (as) independientemente del cargo y personas naturales (sin relación de dependencia).	Director (a) – Presidente, Consejo Directivo – Ministro.

Tabla 1.4 Niveles de Autorización de las Actividades fuera del país.¹

1.3 Rendición de cuentas

El proceso de solicitud de viáticos, no sólo conlleva solicitarlos sino también la rendición de cuentas que se debe realizar luego de culminar la actividad realizada por la que se solicitó los viáticos. La rendición de cuentas se deberá realizar en un lapso no mayor de cinco (5) días hábiles siguientes a la culminación de la misión; de no cumplir con esto y acumular dos (2) relaciones de viáticos, no

¹ Artículo 13, Reglamento de viáticos y pasajes, Funvisis, Caracas, Marzo 2013. 5p.

le serán autorizadas las nuevas salidas hasta tanto no presente las rendiciones de cuentas correspondientes. La rendición de cuenta es obligatoria y el empleado deberá realizar lo siguiente:

- a) La persona que realizó la actividad encomendada por la Fundación deberá presentar ante el supervisor inmediato el respectivo informe y la rendición de cuenta, en los formatos previstos para ello.
- b) La persona que realizó la actividad encomendada por la Fundación deberá anexar al informe las facturas o comprobantes originales para justificar los gastos, los cuales deben estar sin enmiendas, tachaduras ni borrones. Las facturas deberán cumplir con lo establecido en el Servicio Nacional Integrado de Administración Aduanera y Tributaria (SENIAT). Así como se le adjuntará los pases a bordo en caso de transporte aéreo o boletos en caso de transporte terrestre.
- c) Para el caso en que la movilización se realice en un transporte aéreo constitucional o charter y éste haya sido provisto por otra institución, únicamente deberán hacer constar esta particularidad en el informe del cumplimiento de la misión. En este caso, se procederá conforme lo establecido en el Capítulo II de este reglamento.
- d) Los informes deberán imprimirse en dos (2) ejemplares, uno reposará en la Dirección de adscripción de la persona que realizó la actividad encomendada por la fundación y el otro será enviado a la Dirección de Administración y Servicios. Dichos informes y la rendición deberán estar firmados en cada hoja tanto por la persona que realizó la actividad encomendada por la fundación, como por el Supervisor.

1.4 Gestión de viáticos

El resultado de analizar el reglamento y la forma cómo se gestiona los viáticos en Funvisis se concluye que los viáticos se manejan por proyectos, los mismos tienen asignados por partidas una cierta cantidad de dinero para los viáticos y el proceso actual para solicitar viáticos es el siguiente:

- 1) La secretaría con toda la información requerida para la solicitud de viáticos genera un documento para ser enviado a los distintos departamentos.
- 2) La solicitud de viáticos es enviada al jefe del departamento y/o coordinador de la acción para que el mismo firme el documento y verifique la información.
- 3) El documento posteriormente se lleva a Dirección Técnica y al coordinador de la sub acción para que sea revisado y firmado.
- 4) Al obtener las firmas necesarias para la solicitud de los viáticos el documento es llevado a Planificación y Presupuesto, donde se determinará si existe o no el dinero suficiente para otorgar los viáticos y hacer las acciones correspondientes ya sea para su aprobación o negación.
- 5) El departamento de Planificación y Presupuesto firma la solicitud y genera una certificación de compromiso presupuestario, este documento es llevado a Servicios de Administración y a la Dirección de Administración y Servicios para ser firmado y generen el cheque correspondiente.
- 6) Al culminar todos los pasos anteriores el Presidente de la Fundación coloca la última firma al documento para aprobar así los viáticos solicitados.

Es así como se maneja actualmente la solicitud de viáticos, teniendo en cuenta que el manejo de documentos se hace a través de secretarías quienes son las que llevan una hoja de control (Anexo 2) para determinar por cuál departamento va la solicitud de viáticos.

La solicitud de viáticos también conlleva otras actividades como la de solicitar anticipos, los cuales son actividades que se sabe de antemano que se realizarán en el lugar destino y su ejecución es un gasto, si realizando la actividad indicada en la solicitud de viáticos se realiza alguna actividad que genera un gasto al empleado el mismo podrá gastar de su dinero y luego será devuelto por Funvisis, de igual modo si el empleado deja de realizar alguna actividad y debe rendir cuentas a favor de Funvisis para devolver algún dinero. Este proceso es conocido como rendición de cuentas el cual debe realizarse al culminar la actividad indicada en la solicitud de viáticos. El proceso de rendición de cuentas es similar al proceso de solicitud de viático y se diferencia en lo siguiente:

- 1) El empleado debe llenar un documento de evaluación de trabajo de campo, en donde especifica todas las actividades realizadas por día si así se requiere y dárselo a la secretaría de su departamento.
- 2) El empleado debe anexar al documento de evaluación de trabajo de campo todas las facturas que posea.
- 3) La secretaría del departamento debe llenar el documento de relación de gastos de viajes para verificar la exactitud de los montos o existencia de una devolución de dinero a favor de Funvisis o del empleado, siendo alguno de los dos (2) casos, se debe anexar un memo donde se indicará el dinero sobrante y a favor de quién corresponda.

- 4) Los documentos son llevados a presupuesto para que realice las actividades correspondientes según sea el caso: las cuentas sean exactas, las cuentas sean a favor aumentando el presupuesto del proyecto o en contra de Funvisis disminuyendo el presupuesto disponible.
- 5) Al igual que la solicitud de viáticos la rendición pasa a Servicios Administrativos y al Director de Administración y Servicios para ser generado o no el cheque según corresponda.

Los procesos expuestos anteriormente para la gestión de viáticos en Funvisis se pueden observar en los diagramas de flujos de las figuras 1.2 y 1.3.

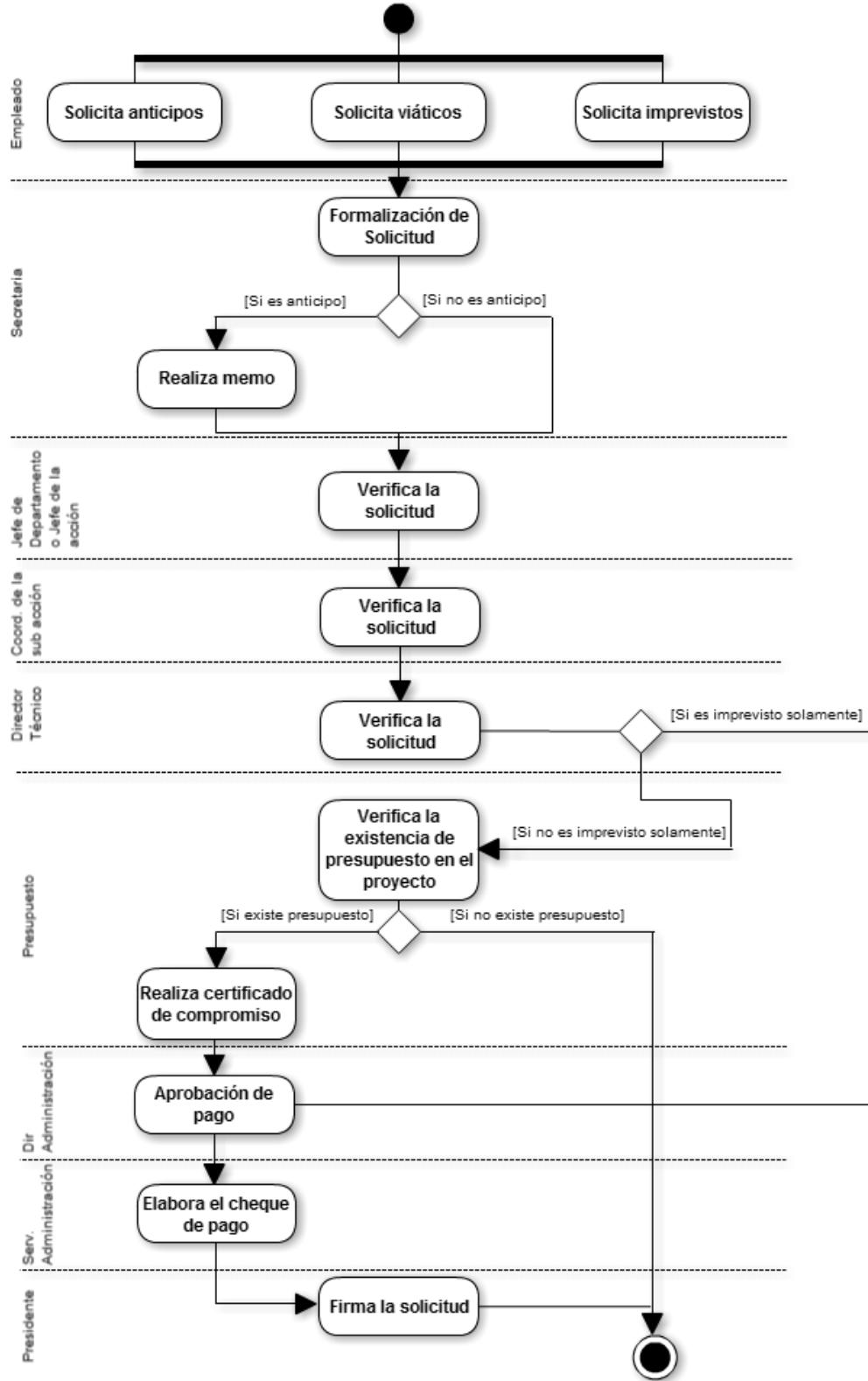


Figura 1.2 Diagrama de actividades del proceso actual de solicitud de viáticos.

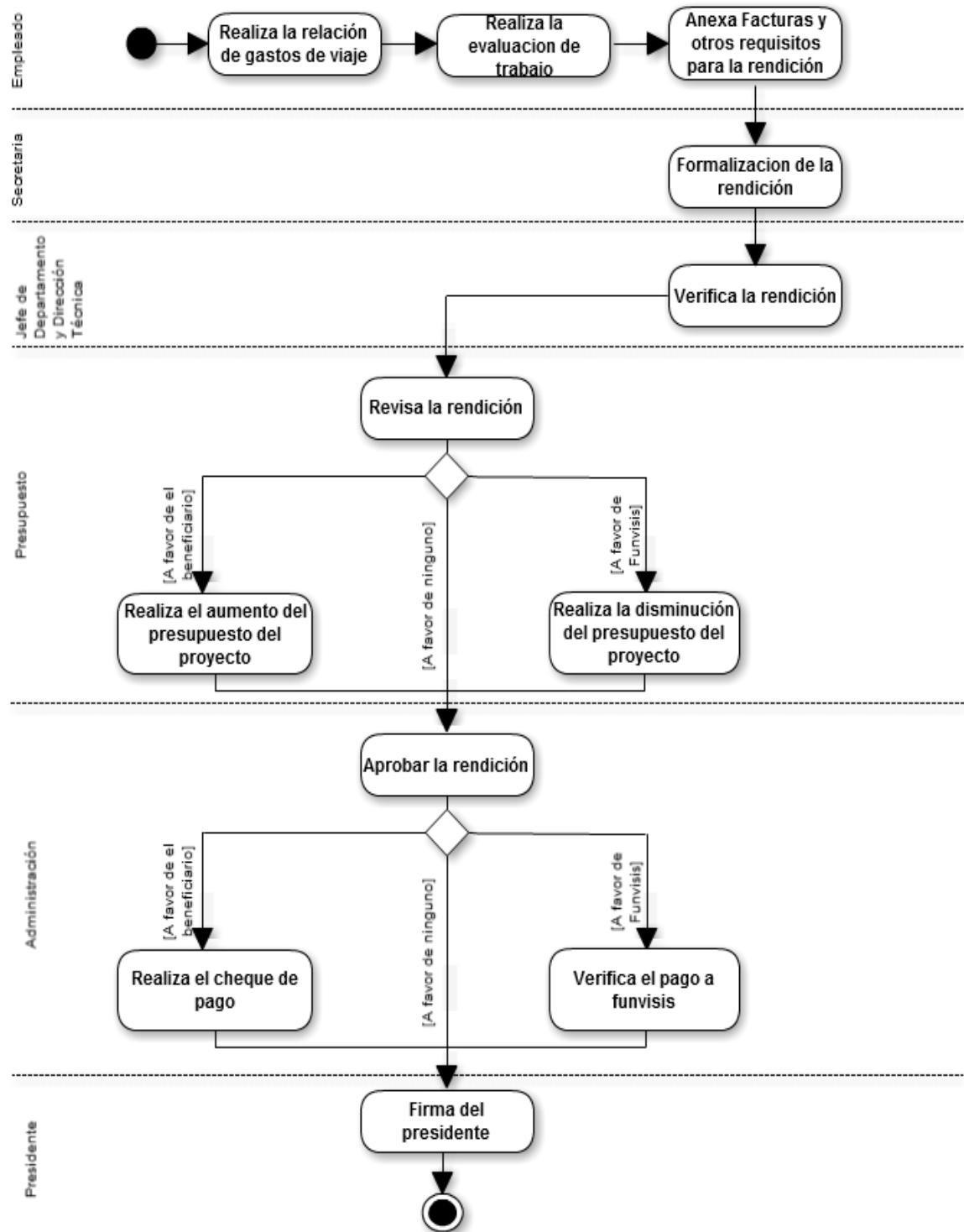


Figura 1.3 Diagrama de actividades del proceso actual de rendición de cuentas.

Capítulo II: Marco Teórico y Tecnológico

En este capítulo se desarrollan las bases teóricas que fundamentan el presente trabajo especial de grado, se describe la metodología ágil programación extrema (XP) utilizada para el desarrollo de software, así como las tecnologías Ruby on Rails como *framework* de aplicación web escrito en el lenguaje de programación Ruby y PostgreSQL como sistema de gestión de base de datos relacional orientada a objetos.

2.1 Programación Extrema (XP)

XP es la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. De la mano de Kent Beck, XP ha conformado un extenso grupo de seguidores en todo el mundo, disparando una gran cantidad de libros a los que dio comienzo el mismo Beck [2] en *Extreme Programming Explained*. Inclusive la editorial Addison-Wesley ha creado una serie de libros denominada *The XP Series*. En la figura 2.2 se puede observar la evolución de ciclos de desarrollo hasta llegar a XP [3].

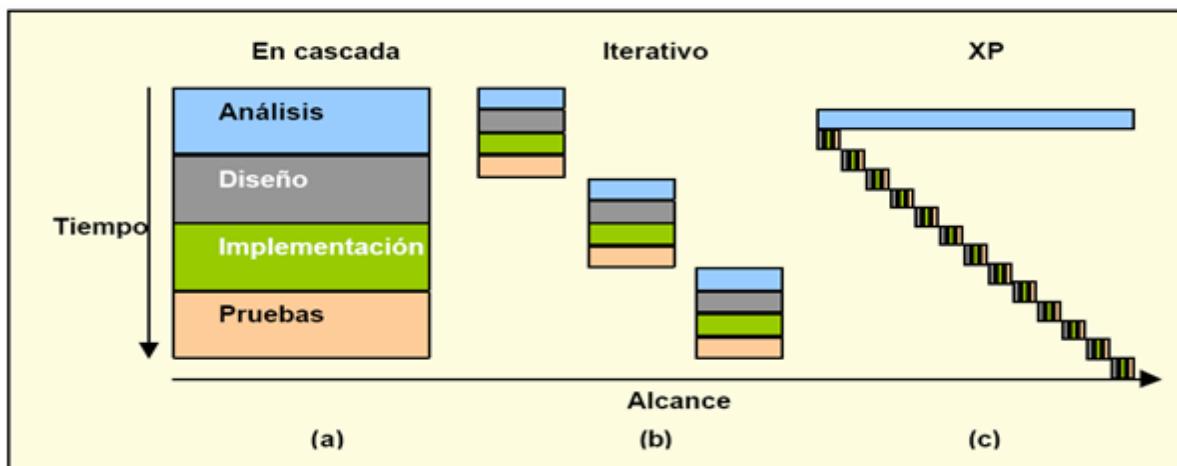


Figura 2.1 Evolución de los largos ciclos de desarrollo en cascada (a) a ciclos iterativos más cortos (b) y a la mezcla que hace XP.

XP [3] es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

La metodología XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas [4].

Historias de Usuario:

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Beck en su libro presenta un ejemplo de ficha (*customer story* y *task card*) en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado cosas por terminar y comentarios. A efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). Las historias de usuario son descompuestas en tareas de

programación (*task card*) y asignadas a los programadores para ser implementadas durante las iteraciones.

Roles XP:

Los roles de acuerdo con la propuesta original de Beck son:

- **Programador:** El programador escribe las pruebas unitarias y produce el código del sistema.
- **Cliente:** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad historias de usuario y decide cuáles a las se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- **Encargado de pruebas (Tester):** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker):** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- **Entrenador (Coach):** Es responsable de proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

- **Gestor (Big boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Proceso XP

El ciclo de vida ideal de un proyecto utilizando XP, se puede separar en las siguientes fases:

- **Fase de exploración:** Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son tentativas, y podrían variar cuando se analicen más en detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado.
- **Fase de planificación:** La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas.
- **Fase de iteraciones:** Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con

el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

- **Fase de puesta en producción:** Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.
- **Fase de mantenimiento:** Requiere de un mayor esfuerzo para satisfacer también las tareas del cliente. Así, la velocidad del desarrollo puede desacelerar después de que el sistema esté en la producción. La fase de mantenimiento puede requerir la incorporación de nueva gente y cambiar la estructura del equipo.
- **Fase de muerte:** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega

del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

- **El juego de la planificación.** Hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Entregas pequeñas.** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de 3 meses.
- **Metáfora.** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- **Diseño simple.** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

- **Pruebas.** La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- **Refactorización (Refactoring).** Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- **Programación en parejas.** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores, etc.).
- **Propiedad colectiva del código.** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- **Integración continua.** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- **40 horas por semana.** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- **Cliente presente.** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del

proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

- **Estándares de programación.** XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

2.2 Ruby on Rails

Ruby [5] es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro Matz Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Ruby es un lenguaje de programación interpretado y su implementación oficial es distribuida bajo una licencia de software libre.

Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC)

mostrado en la Figura 2.2. Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby [5]

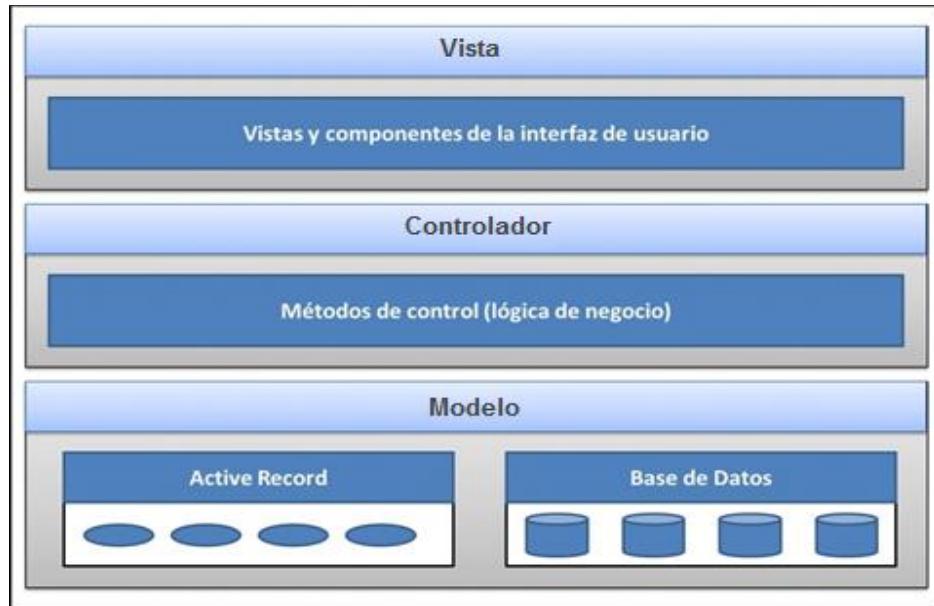


Figura 2.2 Paradigma de la arquitectura Modelo Vista Controlador.

El principio MVC divide el trabajo de una aplicación en tres subsistemas independientes pero estrechamente relacionados. Estos en Ruby on Rails son los siguientes:

- **Modelo.** Es una colección de clases implementada en la biblioteca del framework como “ActiveRecord”, consiste en las clases que representan a las tablas de la base de datos, además de brindar la interfaz para su manipulación.
- **Vista.** Esta clase (ActionView) está implementada en la biblioteca del framework, aunque en la actualidad existen muchas maneras de gestionar

las vistas en RoR se utiliza por defecto Ruby Empotrado (ERb) para definir las plantillas de presentación de los datos y consiste en códigos HTML con código Ruby, siguiendo una sintaxis muy similar a JavaServer Pages (JSP).

- **Controlador.** Esta clase se encuentra implementada en la biblioteca del framework como ActionController, las cuales responden a la interacción del usuario e invocan a la lógica de la aplicación, que a su vez manipula los datos de las clases del Modelo y muestra los resultados usando las Vistas.

Funvisis requiere que sus proyectos sean aplicaciones innovadoras que vayan de la mano con la actualidad tecnológica y no tengan que invertir grandes cantidades de recursos económicos, por lo que utilizan Ruby on Rails como *framework* de código abierto, para el desarrollo de sus aplicaciones Web en cumplimiento al decreto N° 3390¹.

2.3 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando[6]. En la siguiente figura se ilustra de manera general los componentes más importantes en un sistema PostgreSQL.

¹ Gaceta Oficial de la República Bolivariana de Venezuela, N° 38.095 de fecha, 28 de Diciembre de 2004, Decreto N° 3390, Artículo 1°

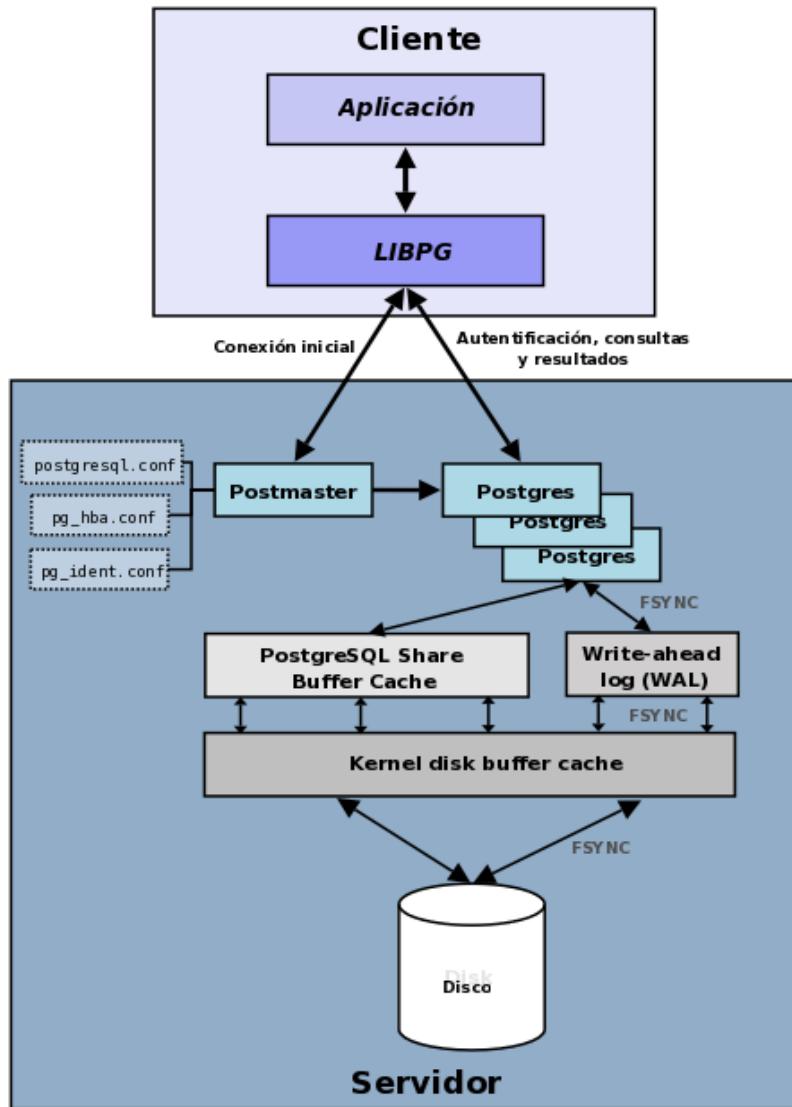


Figura 2.3 Componentes importantes en un sistema PostgreSQL.

PostgreSQL se distribuye bajo licencia BSD, lo que permite su uso, redistribución, modificación con la única restricción de mantener el Copyright del software a sus autores, en concreto el PostgreSQL Global Development Group y la Universidad de California. Es por ello que gracias a todas las características de este sistema de gestión de base de datos Funvisis hace uso de él, siendo el manejador estándar de base de datos para las nuevas aplicaciones que se desarrollen e implementen en la fundación.

Capítulo III: Marco Aplicativo

A continuación se identifica el problema de la Fundación Venezolana de Investigaciones Sismológicas (Funvisis) en su proceso de gestión de viáticos expuesto en el capítulo I del presente trabajo. Además, se describe el objetivo general y los objetivos específicos que dan solución a la problemática y se presenta la aplicación del método ágil Programación Extrema (XP) aplicada para el desarrollo de la aplicación.

3.1 Definición del problema

Funvisis lleva a cabo diversas actividades para realizar todas sus funciones eficientemente, una de ellas es el proceso para la solicitud de viáticos, el cual incluye el proceso de rendición de cuentas como se describió en la sección 1.2. Estos procesos son efectuados de forma manual, donde las secretarías de cada departamento son las encargadas de realizar solicitud de viáticos para el empleado e incluir los documentos que sean necesarios, llevando un control del estado de la solicitud entre los distintos departamentos en los cuales deba ser procesada.

Por tal motivo, se plantea el desarrollo de una aplicación que permita agilizar el proceso, disminuir el número de formularios impresos y reducir el tiempo que tarda una solicitud de viáticos en ser aprobada o rechazada según sea el caso, además de contribuir con los procesos asociados, como la rendición de cuentas.

3.2 Objetivo General

Desarrollar una aplicación web para la automatización de la solicitud de viáticos que realizan los empleados, con el fin de reducir costos de tiempo, papel y recursos humanos para que puedan ser utilizados para otra actividad.

3.3 Objetivos específicos

Los objetivos específicos para dar cumplimiento a los objetivos generales son:

- Obtener información sobre trabajos similares que se hayan realizado para Funvisis u otras empresas para el proceso de solicitud de viáticos.
- Determinar el flujo de actividades y tareas del proceso de solicitud de viáticos.
- Proponer mejoras que se realizarán al flujo actual de solicitud de viáticos y todos sus procesos involucrados.
- Aplicar un método ágil para el desarrollo de la aplicación.

3.4 Antecedentes

En la actualidad algunas organizaciones del Estado ya han automatizado el proceso de solicitud de viáticos y los procesos que el mismo conlleva. Para el desarrollo de este sistema es importante estudiar las aplicaciones ya creadas que cumplen un conjunto de normas y funciones que pueden servir de guía en el desarrollo de la misma y brindarle a Funvisis una solución que mejore los procesos actuales involucrados en la solicitud de viáticos.

INAVI (Instituto Nacional de la Vivienda) utiliza un sistema que sólo le permite crear solicitudes de viáticos, sin embargo, las mismas no tienen seguimiento a través de la aplicación y son impresas para enviarlas a los sitios correspondientes para que la solicitud de viáticos fuese aprobada o rechazada.

La empresa CANTV utiliza el sistema SAP (Sistemas Aplicaciones y Productos) para gestionar todas sus operaciones administrativas, este sistema abarca la solicitudes de viáticos donde el trabajador accede a Autogestión en Línea, selecciona la opción anticipo de viáticos e introduce los datos solicitados: destino, hora de salida, fecha de inicio, fecha de finalización y hora de salida, inmediatamente debe enviar vía electrónica dicha solicitud para que su supervisor apruebe el mismo y la Unidad de Nómina pueda cargar la información y depositar 10 días hábiles después de dicha solicitud. Al igual que sucede en Funvisis, una vez que la actividad para la cual fue solicitado el anticipo se haya realizado, el trabajador debe proceder a liquidar dentro de los 30 días posteriores ingresando en el sistema y cargando la información asociada al viático, teniendo que anexar a la planilla las facturas y soportes en físico para que los documentos sean revisados y almacenados según corresponda; de no hacerlo la empresa procede a descontarlo de la quincena siguiente a la fecha de vencimiento. [7]

En el Ministerio del Poder Popular para la Cultura se emplea un sistema para la solicitud de viáticos en donde el empleado llena una planilla y el departamento de nómina es el encargado de buscar en la sección de viáticos y verificar la solicitud realizada por el empleado, al igual que el proceso de Funvisis se ven involucrados varios niveles de aprobación como el departamento de presupuesto y administración en donde cada departamento tiene una función asociada y fundamental para la aprobación de la solicitud de viáticos. [8]

Luego de estudiar estas propuestas y aplicaciones que funcionan a diario en las organizaciones ya mencionadas, vemos lo complejo que puede ser cada proceso según sean las necesidades de cada organización y según como se maneje el proceso de solicitud de viáticos y sus asociados. En el caso de Funvisis hay que tener en cuenta al momento de desarrollar el sistema que se pueda realizar una actividad con carácter urgente y luego se realice una rendición de cuenta, por lo que se planteó como solución que el empleado realice la actividad, luego haga una solicitud de viáticos y finalmente realice la rendición de cuentas, obteniendo un manejo correcto de cada solicitud con su respectiva rendición.

3.5 Planificación del proyecto

El resultado del estudio y análisis realizado en el seminario culminamos seleccionando XP como método de desarrollo de software para llevar a cabo esta aplicación web, debido a su gran variedad de ventajas basada esencialmente en la simplicidad, agilidad y adaptación al cambio.

En esta fase se representaron en diagramas de flujos los procesos actuales de solicitud de viáticos y los diversos procesos que resultaron de la automatización, mediante una aplicación web para dicha solicitud, los cuales se pueden observar en las Figuras 3.1 y 3.2.

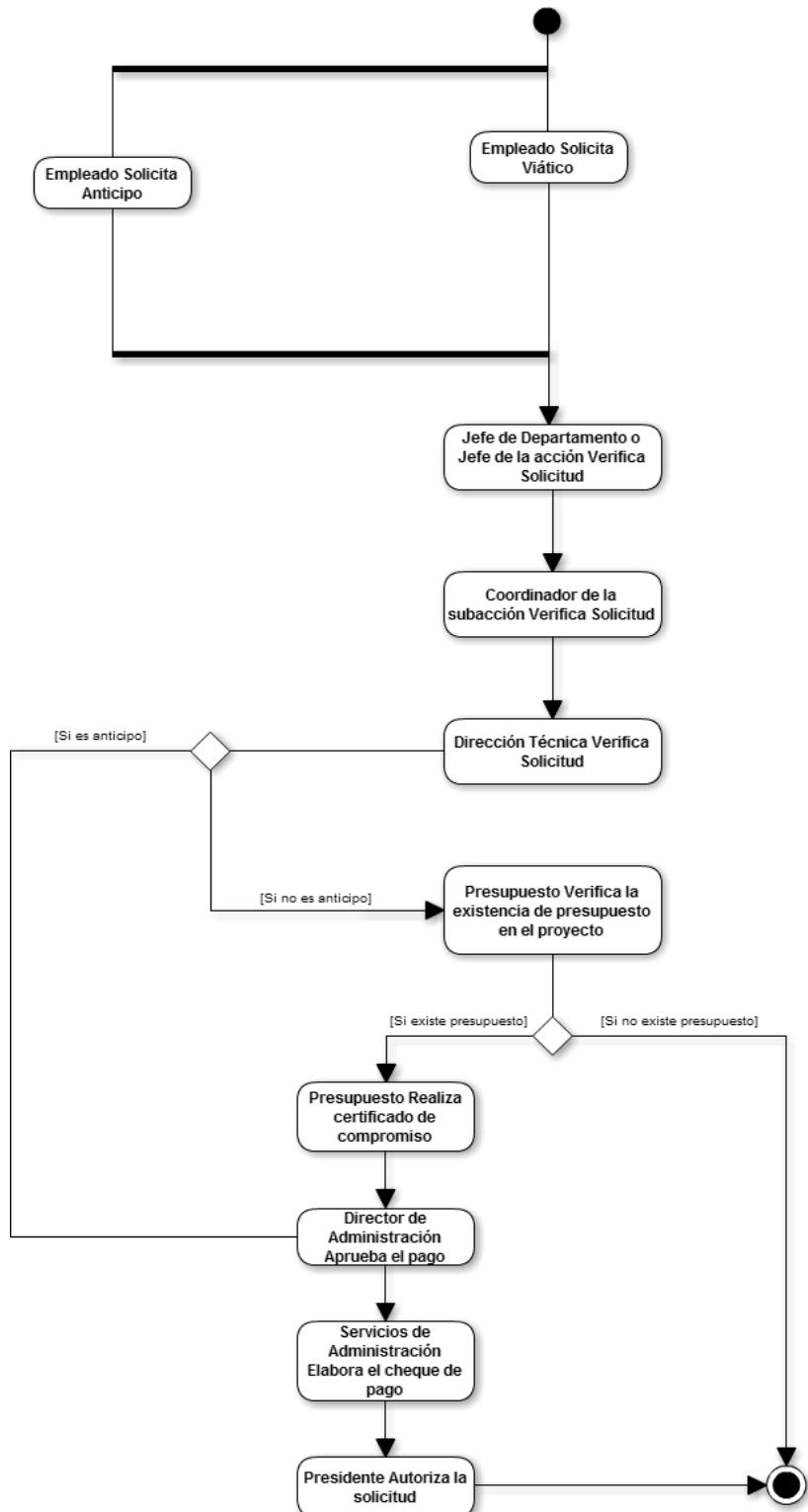


Figura 3.1 Diagrama de actividades de la propuesta de automatización de solicitud de viáticos.

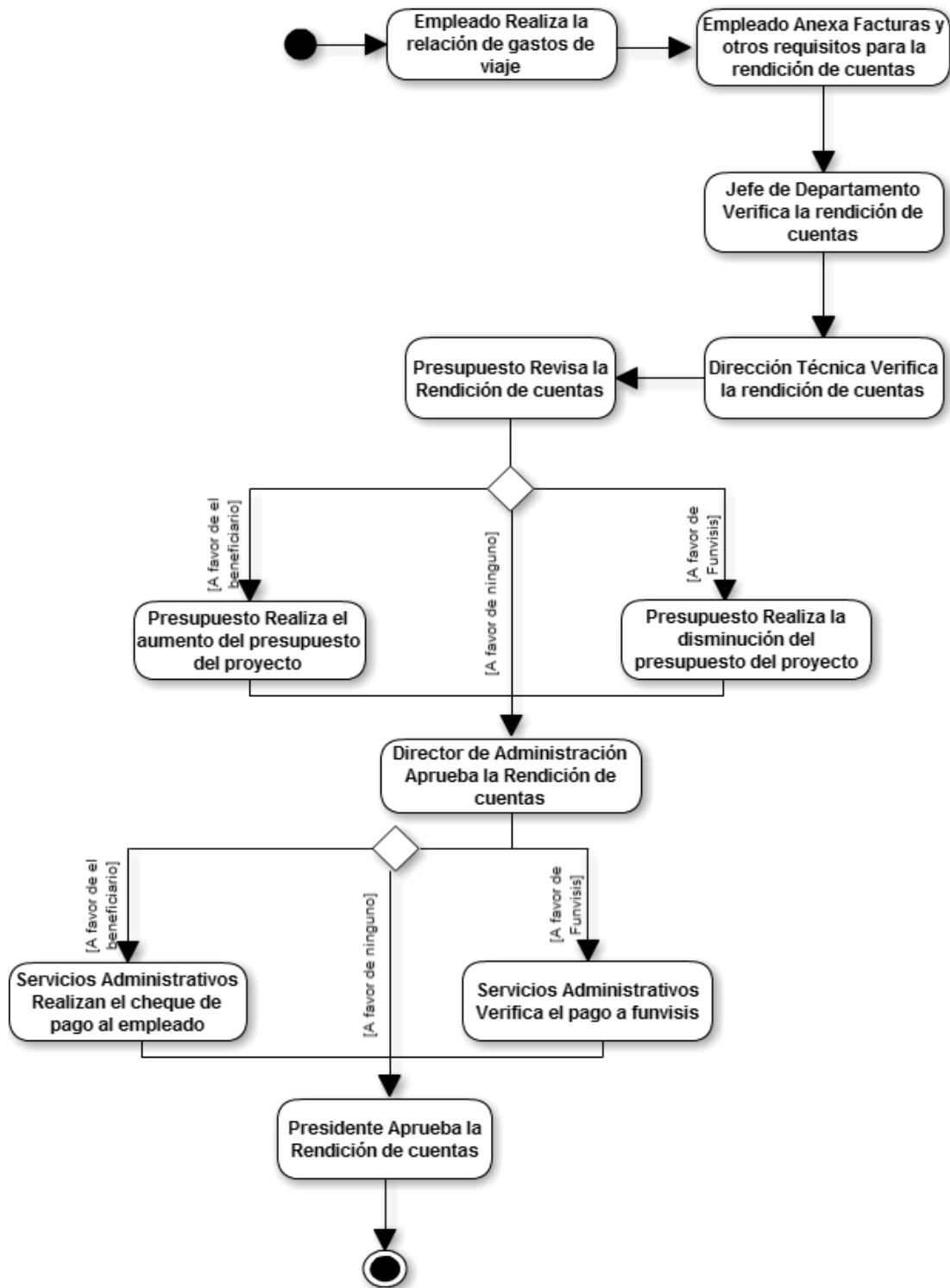


Figura 3.2 Diagrama de actividades de la propuesta de automatización de rendición de cuentas.

El método de desarrollo de software XP para la creación de una aplicación sugiere el uso de historias de usuario para la recolección de datos y mantener un orden en el desarrollo del sistema, teniendo el rol de usuario un papel importante en el sistema, por lo que para efectos de esta aplicación se establecen tres roles:

- Empleado: Es el usuario capaz de solicitar viáticos, solicitar imprevistos y asociar una rendición de cuenta a un viático dado.
- Administrador: Es el usuario que modificará o almacenará información de vital importancia para la aplicación como lo es el valor del dólar utilizado para la solicitud de viáticos al exterior, el valor de la unidad tributaria utilizada para solicitar en la solicitud de viáticos, entre otros.
- Encargado: Usuario responsable de la aprobación o rechazo de las solicitudes realizadas por el empleado.

Al tener conocimiento de los distintos roles del sistema es posible establecer historias de usuario, que se elaboraron en las reuniones con el cliente, y así para recolectar los requerimientos funcionales del sistema, destacando que un usuario del sistema puede manejar uno o más roles dentro del sistema. El desarrollo de la aplicación se dividió en 5 (cinco) módulos, cada módulo corresponde a una agrupación lógica de funcionalidades y a su vez a un conjunto de historias de usuario tal como se presenta en las tablas 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 y 3.8.

Módulo Nº 1: Ingreso al Sistema

Historia de Usuario	
Número: 01	Nombre de historia: Autenticación de usuario
Usuario: Administradores, empleados y encargados	
Riesgo en desarrollo: Bajo	Iteración asignada: 1
Descripción: El usuario se autenticará en el sistema y tendrá el acceso a las distintas funciones de acuerdo a su nivel de permisología.	
Observaciones: Se debe verificar la existencia del usuario en el sistema y el cargo que posee para determinar las funcionalidades a las que pueda acceder ya sea administrador, encargado o empleado. Solo desplegando las opciones a las que tenga acceso.	

Tabla 3.1 Historia de usuario: Autenticación de usuario

Módulo Nº 2: Administración

Historia de Usuario	
Número: 02	Nombre de historia: Administración del sistema
Usuario: Administrador	
Riesgo en desarrollo: Medio	Iteración asignada: 1
Descripción: El usuario administrador podrá crear o editar países usados en las solicitudes al exterior, indicará la unidad tributaria actual, la cantidad de unidades tributarias designada para viajes al interior y/o exterior de Caracas, también podrá guardar en el sistema las acciones y proyectos necesarios en las solicitudes, así como los imprevistos y sus montos asociados indicando si los mismos serán editables o no al momento de realizar una solicitud.	
Observaciones: Esta parte del sistema será solamente para configurar campos necesarios para la generación de las solicitudes de viáticos e imprevistos así como de las rendiciones de cuenta.	

Tabla 3.2 Historia de usuario: Administración del sistema.

Módulo Nº 3: Solicitudes

Historia de Usuario	
Número: 03	Nombre de historia: Solicitud de viáticos para el interior del país
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 2
<p>Descripción: El usuario de tipo empleado podrá crear una solicitud de viáticos para el interior del país y los imprevistos asociados a la misma.</p> <p>Observaciones: En esta parte del sistema el usuario solicitará sus viáticos hacia el interior del país a través de un formulario por pasos, donde primero indica la información de los viáticos en general y luego indica la información referente al viático hacia el interior del país, posteriormente saltará la sección de viáticos para el exterior y confirmará la solicitud.</p>	

Tabla 3.3 Historia de usuario: Solicitud de viáticos para el interior del país

Historia de Usuario	
Número: 04	Nombre de historia: Solicitud de viáticos para el exterior del país
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 2
<p>Descripción: El usuario de tipo empleado podrá crear una solicitud de viáticos para el exterior del país y los imprevistos asociados a la misma.</p> <p>Observaciones: En esta parte del sistema el usuario solicitará sus viáticos a través de un formulario por pasos, donde primero indicara la información de los viáticos en general, posteriormente podrá decidir si desea realizar una solicitud para el interior del país o saltar al siguiente paso para realizar la solicitud al exterior con los imprevistos que estén asociado para su confirmación.</p>	

Tabla 3.4 Historia de usuario: Solicitud de viáticos para el exterior del país

Historia de Usuario	
Número: 05	Nombre de historia: Solicitud de imprevistos
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 2
Descripción: El usuario podrá solicitar imprevistos que no estén relacionados a una solicitud de viáticos hacia el interior o exterior del país.	
Observaciones: El usuario podrá modificar el monto del imprevisto que posea la configuración de editable desde el módulo de administración, además solo permitirá realizar esta solicitud en los casos de que no se hayan realizado ninguna solicitud de viáticos hacia el interior o exterior del país en los pasos previos.	

Tabla 3.5 Historia de usuario: Solicitud de imprevistos.

Módulo Nº 4: Rendiciones de cuenta

Historia de Usuario	
Número: 06	Nombre de historia: Rendiciones de cuenta
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 3
Descripción: El sistema permitirá al usuario realizar la rendición de cuentas de las actividades desarrolladas o solicitudes de viáticos aprobadas, indicando la información solicitada por el sistema para la formalización de la solicitud y adjuntando el resumen de las actividades correspondientes.	
Observaciones: El sistema realizará un esquema similar a la solicitudes de viáticos y se implementara un formulario por pasos donde la primera sección será la información general, la rendición para las solicitudes al interior, exterior e imprevistos según corresponda.	

Tabla 3.6 Historia de usuario: Rendiciones de cuenta.

Módulo Nº 5: Aprobaciones

Historia de Usuario	
Número: 07	Nombre de historia: Aprobación de solicitud.
Usuario: Encargado	
Riesgo en desarrollo: Medio	Iteración asignada: 4
Descripción: Los usuarios de tipo encargado podrán ver las solicitudes que posean los estados: aprobadas, rechazadas y pendientes en sus respectivas secciones. Además de permitir la visualización de la información suministrada para su aceptación o negación.	
Observaciones: Si el estado de aprobación de la solicitud se encuentra en Presupuesto, ellos podrán indicar la imputación de la solicitud.	

Tabla 3.7 Historia de usuario: Aprobación de solicitud.

Historia de Usuario	
Número: 08	Nombre de historia: Aprobación de rendiciones.
Usuario: Encargado	
Riesgo en desarrollo: Medio	Iteración asignada: 4
Descripción: Los usuarios de tipo encargado podrán ver las rendiciones que posean los estados: aprobadas, rechazadas y pendientes en sus respectivas secciones. Además de permitir la visualización de la información suministrada para su aceptación o negación.	
Observaciones: Las rendiciones no podrán ser automatizadas en su totalidad por el sistema, motivado a la necesidad del anexo físico de todas las facturas correspondientes a las rendiciones de cuenta, pero proporcionara el seguimiento del trámite.	

Tabla 3.8 Historia de usuario: Aprobación de rendiciones.

3.6 Plan de entrega

Se acordó un tiempo de entrega de cinco (5) a seis (6) meses. En base a esto se realizó la siguiente planificación (ver tabla 3.9):

Iteración	Nº de historia	Historia de usuario	Inicio	Fin
1	1	Autenticación de usuario	04/11/2012	07/11/2012
	2	Administración del sistema	08/11/2012	13/11/2012
2	3	Solicitud de viáticos para el interior.	14/11/2012	28/11/2012
	4	Solicitud de viáticos para el exterior.	04/12/2012	11/11/2012
	5	Solicitud de imprevistos.	11/12/2012	18/12/2012
3	6	Rendiciones de cuenta.	14/01/2013	21/01/2013
4	7	Aprobación de solicitud.	21/01/2013	31/01/2013
	8	Aprobación de rendiciones.	31/01/2013	07/02/2013

Tabla 3.9 Planificación de entrega del proyecto.

Dado el número de programadores en el equipo de desarrollo, se decidió no programar rotaciones durante el desarrollo de las tareas. Ambos programadores trabajaron de manera conjunta. Durante esta iteración se llevaron a cabo reuniones de manera frecuente (nueve o diez), para discutir las tareas que se desarrollarían, considerando los posibles problemas que pudieran suscitar y sus posibles soluciones. Teniendo estas reuniones una duración entre quince (15) y veinte (20) minutos.

A continuación se describen las iteraciones.

3.6.1 Iteración I – Ingreso y administración

Esta iteración trata del desarrollo del módulo para el ingreso al sistema por parte del usuario con su respectiva verificación de rol y desarrollar las tareas que un empleado de Funvisis con rol administrador está capacitado para hacer. A continuación se lista la documentación de la planificación, diseño, codificación y pruebas que tuvieron lugar durante la primera iteración del proyecto.

3.6.1.1 Planificación

En esta fase se especificará con detalle las historias de usuario involucradas, el plan de entrega, el plan de esta iteración y las reuniones realizadas para esta iteración 1.

➤ Historias de usuario

Las historias de usuario involucradas en esta iteración son las especificadas en la Tabla 3.9 de Autenticación de usuario y de Administración del sistema, siendo la historia de usuario número 2 (ver tabla 3.2) seccionada en ocho (8) historias de usuario debido a su complejidad, las cuales se mostrarán a continuación en las tablas 3.10, 3.11, 3.12, 3.13, 3.14 y 3.15

Historia de Usuario	
Número: 2.1	Nombre de historia: Registro de administradores del sistema
Usuario: Administrador	
Riesgo en desarrollo: Medio	Iteración asignada: 1
Descripción: El sistema debe permitir el registro de los administradores que serán los responsables de las configuraciones.	

Tabla 3.10 Historia de usuario: Registro de administradores del sistema.

Historia de Usuario	
Número: 2.2	Nombre de historia: Administración de las U.T (Unidades Tributarias) asignadas para las solicitudes.
Usuario: Administrador	
Riesgo en desarrollo: Alto	Iteración asignada: 1
Descripción: El sistema debe permitir el registro y la administración de las U.T asignadas para las solicitudes de viáticos dentro del territorio nacional.	

Tabla 3.11 Historia de usuario: Administración de las U.T asignadas para las solicitudes.

Historia de Usuario	
Número: 2.3	Nombre de historia: Registro de imprevistos
Usuario: Administrador	
Riesgo en desarrollo: Medio	Iteración asignada: 1
Descripción: El sistema debe permitir el registro de la descripción de los imprevistos más utilizados además asignarle el monto que se tiene estipulado para dichas solicitudes e indicar si su monto puede ser editable o no.	

Tabla 3.12 Historia de usuario: Registro de imprevistos

Historia de Usuario	
Número: 2.4	Nombre de historia: Registro de proyectos
Usuario: Administrador	
Riesgo en desarrollo: Medio	Iteración asignada: 1
Descripción: El sistema debe permitir el registro de los proyectos al sistema e indicarles la fecha desde el cual se encontrará activo.	

Tabla 3.13 Historia de usuario: Registro de proyectos.

Historia de Usuario	
Número: 2.5	Nombre de historia: Registro de acciones
Usuario: Administrador	
Riesgo en desarrollo: Medio	Iteración asignada: 1
Descripción: El sistema debe permitir el registro de las acciones al sistema y debe poder indicar la fecha desde la cual se encontrará activo.	

Tabla 3.14 Historia de usuario: Registro de acciones.

Historia de Usuario	
Número: 2.6	Nombre de historia: Registro de países
Usuario: Administrador	
Riesgo en desarrollo: Medio	Iteración asignada: 1
Descripción: El sistema debe permitir el registro de los países a los cuales se podrán realizar solicitudes de viáticos al exterior, además de indicar la U.T. asignada a los destinos.	

Tabla 3.15 Historia de usuario: Registro de países.

➤ *Plan de iteración*

A continuación se muestran el conjunto de tareas a completar durante esta iteración para el desarrollo del módulo de ingreso y administración del sistema, en las tablas 3.16, 3.17, 3.18, 3.19, 3.20, 3.21 y 3.22.

Tarea	
Número: 1	Número de historia: 1
Nombre tarea: Conexión de la base de datos y el sistema	
Tipo de tarea: Desarrollo	
Descripción: Para que el usuario ingrese al sistema de solicitud de viáticos es necesario establecer la conexión entre el sistema creado y la base de datos que posee Funvisis de los usuarios.	

Tabla 3.16 Tarea: Conexión de la base de datos y el sistema.

Tarea	
Número: 2	Número de historia: 1 y 2
Nombre tarea: Creación de la base de datos para la administración del sistema	
Tipo de tarea: Administración	
Descripción: Se crea las tablas en la base de datos necesarias para la creación de proyectos, acciones, imprevistos, países, manejo de sesiones, unidades tributarias y administradores.	

Tabla 3.17 Tarea: Creación de la base de datos para la administración del sistema.

Tarea	
Número: 3	Número de historia: 1 y 2
Nombre tarea: Mensajes de validación para la administración del sistema	
Tipo de tarea: Desarrollo	
Descripción: Se crean mensajes para la validación de las variables que se ingresan al sistema.	

Tabla 3.18 Tarea: Mensajes de validación para la administración del sistema.

Tarea	
Número: 4	Número de historia: 2
Nombre tarea: Creación de formularios para la administración del sistema	
Tipo de tarea: Desarrollo	
Descripción: Se crean los formularios correspondientes para la administración del sistema.	

Tabla 3.19 Tarea: Creación de formularios para la administración del sistema.

Tarea	
Número: 5	Número de historia: 2
Nombre tarea: Formularios para la actualización de la administración del sistema	
Tipo de tarea: Desarrollo	
Descripción: Se crean los formularios correspondientes para actualizar la información referente a la administración del sistema.	

Tabla 3.20 Tarea: Formularios para la actualización de la administración del sistema.

Tarea	
Número: 6	Número de historia: 2
Nombre tarea: Formularios para la actualización de la administración del sistema	
Tipo de tarea: Desarrollo	
Descripción: Se crean los formularios correspondientes para actualizar la información referente a la administración del sistema.	

Tabla 3.21 Tarea: Formularios para la actualización de la administración del sistema.

Tarea	
Número: 7	Número de historia: 2
Nombre tarea: Listado de atributos para la administración del sistema	
Tipo de tarea: Desarrollo	
Descripción: Se crean todos los listados y se implementan métodos de búsqueda para algunos campos como lo son administradores y países.	

Tabla 3.22 Tarea: Listado de atributos para la administración del sistema.

3.6.1.2 Diseño

En esta fase se realizó el diseño de la base de datos se puede observar en la Figura 3.5.

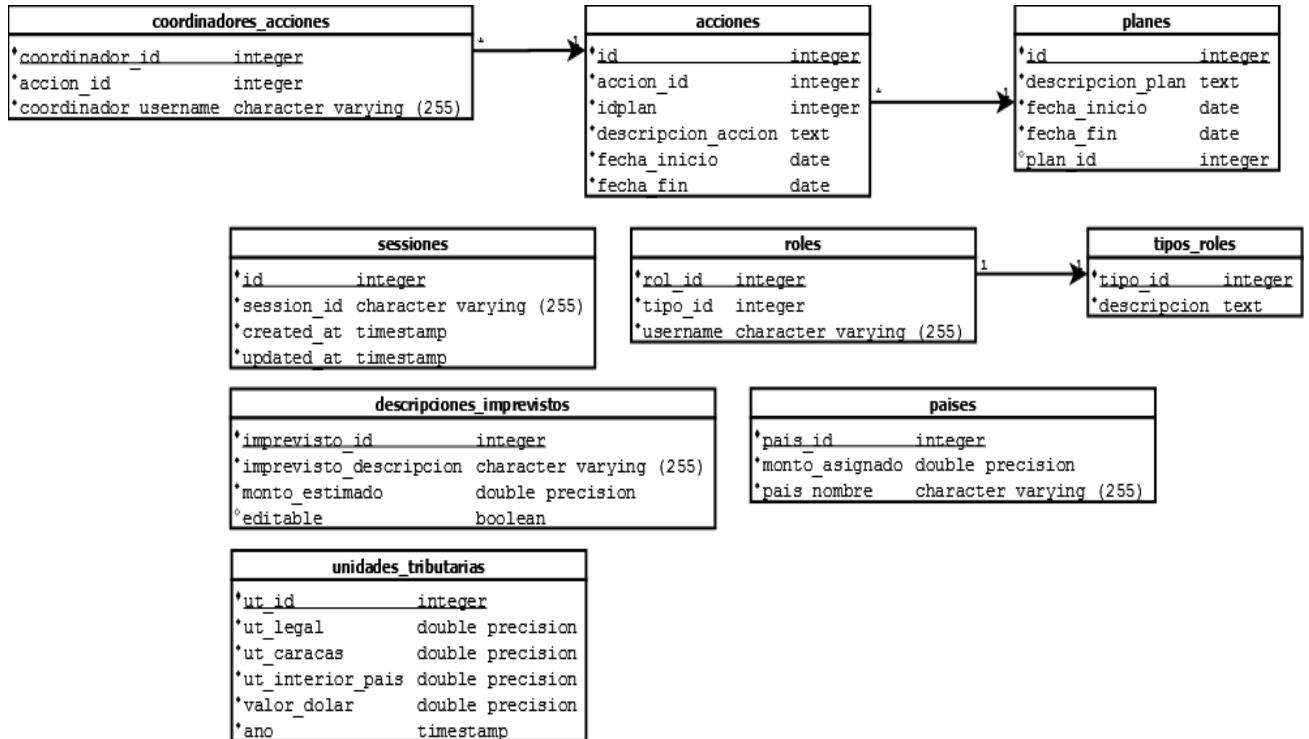


Figura 3.3 Diagrama de la base de datos en la primera iteración.

En este modelo de la base de datos se observan las tablas y sus atributos, representando las claves primarias mediante el subrayado e indicando si el atributo puede estar vacío o no, a través de un círculo vacío o lleno respectivamente. Este nomenclatura será utilizara en las próximas iteraciones para el diseño de las base de datos.

En el presente diseño se consideró la existencia de la tabla `proyectos` dentro del esquema `sisevaluacion` en la base de datos de la fundación, la cual posee la información referente a las sub acciones. Por tal motivo, se implementó la tabla `planes` destinada al almacenamiento de la información de los proyectos

que son realizados por la institución, además estos están conformados por un conjunto de acciones las cuales son almacenados en la tabla `acciones` y cuyos responsables están contenidos en la tabla `coordinadores_accion`. Obteniendo así los elementos del esquema actual de trabajo en Funvisis, en el cual un proyecto posee varias acciones y estas a su vez están constituidas por múltiple sub acciones.

Motivado al desarrollo de la historia de usuario de administración, se consideró necesario almacenar la información que cambia muy poco al transcurrir el tiempo, pero que son de vital importancia para la solicitud de viáticos y rendición de cuentas, es por esto que se generan las tablas `paises`, `unidades_tributarias`, `descripciones_imprevistos`. En la tabla `paises`, se almacena los países con el respectivo monto en dólares que se le será otorgado por día al empleado que realice una solicitud de viáticos. En la tabla `unidades_tributarias` se guarda el valor de la unidad tributaria y la cantidad que se le puede otorgar por día al empleado que solicita viáticos y también se almacena el valor del dólar actual. En la tabla `descripciones_imprevistos` se encuentra la información referente a los imprevistos más conocidos así como el valor que al mismo se le otorga indicando también si el precio del mismo se podrá o no editar por el empleado que esté realizando la solicitud de imprevisto.

Para el desarrollo de la autenticación de usuario, se crearon las tablas `roles` y `tipos_roles`, donde en una se almacena la relación existente entre los distintos usuarios y el rol almacenado en el sistema y en la otra se indica la descripción de cada tipo rol respectivamente. Además, para el manejo de las sesiones y su información dentro del sistema se creó la tabla `sesiones`.

Es importante destacar que para el desarrollo de este modelo no se tomó en cuenta la implementación de la tabla `usuarios`, puesto que es implementada

actualmente en Funvisis en su base de datos en el esquema público y el sistema es capaz de establecer una conexión con dicha base de datos, de la misma manera sucede con la tabla de las sub acciones siendo la misma almacenada en un esquema independiente.

3.6.1.3 Codificación

Previo a la etapa de codificación se acordó la utilización de nombres de variables informativos (sin importar su longitud) y separados por “_”. Además la aplicación se desarrolló utilizando el patrón de arquitectura de software MVC, ya que el mismo es implementado en RoR (Ruby on Rails) como se describió en el Capítulo II. Las consultas a las base de datos que puedan retornar grandes volúmenes de información deben ser paginadas, por tal motivo se utilizó la gema “will_paginate”¹.

Para la elaboración de los estilos y diseños de las páginas se utilizó como base y referencia el de la Figura 3.6 perteneciente al Sistema de Gestión de Memos de Funvisis con sus ajustes y mejoras al nuevo sistema en desarrollo.

¹ Will Paginate, API que sirve para realizar paginación de consultas con Active Record, DataMapper Y Sequel. recuperado el 16 de Mayo del 2013, de http://rubygems.org/gems/will_paginate

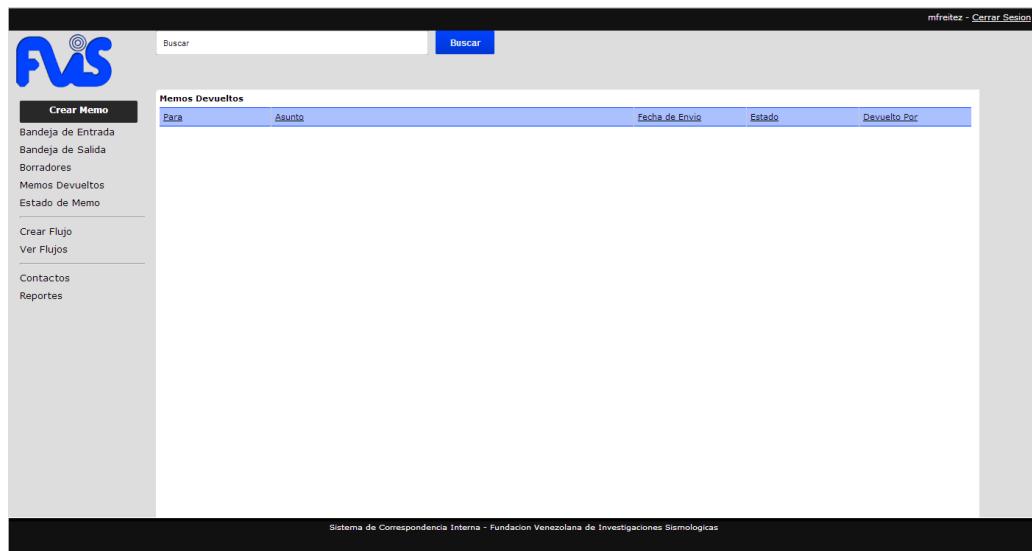


Figura 3.4 Sistema de Gestión de Memos.

Se realizó una agrupación de las funciones del sistema en tres secciones: administración, encargado y empleado. Cada una de estas secciones estará disponible para los usuarios dependiendo del rol que desempeñe en Funvisis y podrá ser seleccionado posterior al ingreso del sistema como se muestra en la Figura 3.7. La sección desarrollada durante esta iteración fue la de administración.



Figura 3.5 Pantalla para la selección del módulo.

Para el manejo de la administración de los datos de las U.T. (Unidades Tributarias) se acordó la restricción de no permitir la eliminación ni edición de este elemento si existe una solicitud que hizo uso de esta información, con la finalidad de mantener un registro histórico de los valores utilizados al momento de realizar una solicitud.

Para la presentación de las listas de datos, se añadieron enlaces a las cabeceras de cada columna que son generados en el *helper* de la Figura 3.8, que permiten la organización del contenido de las listas por el valor de la columna seleccionada y en el caso de los países se incluyó un buscador como muestra la Figura 3.9 por el valor del nombre para agilizar su búsqueda.

```

module ApplicationHelper

def sortablepais(column,title)
  css_class= column==sort_columnpais ? "current #{sort_directionpais}" : nil
  direction= column==sort_columnpais && sort_directionpais=="asc"? "desc" : "asc"
  link_to title, params.merge(:sort=> column, :direction=> direction), {:class=>
    css_class}
end
def sortablerol(column,title)
  css_class= column==sort_columnrol ? "current #{sort_directionrol}" : nil
  direction= column==sort_columnrol && sort_directionrol=="asc" ? "desc" : "asc"
  link_to title, params.merge(:sort=> column, :direction=> direction), {:class =>
    css_class}
end
def sortableut(column,title)
  css_class = column==sort_columnnut ? "current #{sort_directionut}" : nil
  direction = column==sort_columnnut && sort_directionut=="asc" ? "desc" : "asc"
  link_to title, params.merge(:sort => column, :direction => direction), {:class =>
    css_class}
end
def sortableaccion(column,title)
  css_class = column==sort_columnaccion ? "current #{sort_directionaccion}" : nil
  direction = column==sort_columnaccion && sort_directionaccion=="asc"? "desc": "asc"
  link_to title, {:sort => column, :direction => direction}, {:class => css_class}
end
def sortableplan(column,title)
  css_class= column==sort_columnplan ? "current #{sort_directionplan}" : nil
  direction= column==sort_columnplan && sort_directionplan == "asc" ? "desc" : "asc"
  link_to title, {:sort => column, :direction => direction}, {:class => css_class}
end
def sortablerendicion(column,title)
  css_class=column==sort_columnrendicion?"current #{sort_directionrendicion}" : nil
  direction=column==sort_columnrendicion&&sort_directionrendicion=="asc"? "desc": "asc"
  link_to title, params.merge(:sort => column, :direction => direction), {:class =>
    css_class}
end
...

```

Figura 3.6 Código del ApplicationHelper

Paises

Nombre ▲	Monto en Bs.
<u>Nigeria</u>	137.0 Bs
<u>Nipal</u>	77.0 Bs
<u>Noruega</u>	193.0 Bs
<u>Nueva Caledonia</u>	85.0 Bs
<u>Nuevas Hebridas</u>	70.0 Bs
<u>Ohan</u>	170.0 Bs
<u>Pakistan</u>	84.0 Bs
<u>Panama</u>	156.0 Bs
<u>Papua-Nva. Guinea</u>	190.0 Bs
<u>Paraguay</u>	140.0 Bs
<u>Peru</u>	140.0 Bs
<u>Polonia</u>	193.0 Bs
<u>Portugal</u>	193.0 Bs
<u>Puerto Rico</u>	115.0 Bs
<u>Qatar</u>	148.0 Bs

← Anterior 1 2 3 4 5 6 7 8 9 10 11 12 Siguiente →

Figura 3.7 Lista de países.

Para la elaboración de la función de autocompletado a los campos de búsqueda, se utilizó jQuery. Para evitar realizar consultas al servidor por cada letra introducida por el usuario se activa un temporizador que permitirá saber cuándo el usuario terminó de escribir su consulta y la activación de la función será a partir del tercer carácter.

En total se crearon veintinueve (29) vistas, dos (2) controladores, un (1) *helper* y quince (15) modelos para dar cumplimiento con las historias de usuarios planificadas en esta iteración.

3.6.1.4 Pruebas

Durante esta iteración las pruebas de validación fueron aplicadas de forma manual, verificando el correcto funcionamiento del sistema de autenticación y de administración, así como el conjunto de formularios y vistas que posee esta sección del sistema.

Las pruebas fueron realizadas por los desarrolladores y Adriana Liendo (el cliente), en donde se sugirió que los campos que fueran de cantidades numéricas se bloquearan las teclas que no fueran numéricas, esta funcionalidad fue realizada con el Plug-in “Jquery–Numeric”¹ desarrollado por Sam Collett. En las siguientes figuras se muestran algunas capturas de pantallas de la interfaz gráficas y algunos valores de las pruebas realizadas.

Nuevo Imprevisto

The screenshot shows a web-based form titled "Nuevo Imprevisto". At the top, there is a red warning box containing a warning icon and the text "Los siguientes errores evitan que este imprevisto sea creado". Below this, there are several input fields:

- A large text area labeled "Descripción del imprevisto *:" with a red border. Below it, an error message says "Este campo no puede estar en blanco".
- A text input field labeled "Monto estimado Bs. *:" with a red border. Below it, an error message says "Este campo no puede estar en blanco".
- A checkbox labeled "Editable" with a small square icon next to it.
- A note "(*) campos obligatorios" indicating mandatory fields.

At the bottom right of the form is a "Crear" (Create) button.

Figura 3.8 Vista de nuevo imprevisto.

¹ Jquery – Numeric, Plug-in desarrollado por Sam Collet para permitir solo la escritura de números en los cuadro de texto, recuperado el 16 de Mayo del 2013, de <http://www.texotela.co.uk>

Editar Proyecto

Identificador del proyecto *:

Descripción *:

Prevención y Mitigación de Riesgos
ante Eventos Adversos

Fecha inicio *:

Fecha fin *:

(*) campos obligatorios

Figura 3.9 Vista de la edición de un proyecto.

Nueva Acción

Proyecto al que pertenece *:

Identificador de la acción *:

Descripción acción *:

Fecha inicio *:

Fecha fin *:

(*) campos obligatorios

Figura 3.10 Vista de una nueva acción.

3.6.2 Iteración II - Solicitud de viáticos

El objetivo de esta iteración era desarrollar un formulario para la solicitud de viáticos tanto para el exterior como el interior del país, así como también para la solicitud de imprevistos necesarios bien sea correspondiente a un viático o para ejecutar alguna actividad en específico para la fundación. A continuación se lista la documentación de la planificación, diseño, codificación y pruebas que tuvieron lugar durante la segunda iteración del proyecto.

3.6.2.1 Planificación

En esta fase del desarrollo se especificarán las historias de usuario involucradas, el plan de entrega, el plan de esta iteración y las reuniones realizadas para esta iteración.

➤ Historias de usuario

Las historias de usuario involucradas en esta segunda iteración son las especificadas en las tablas de solicitud de viáticos para el interior del país, solicitud de viáticos para el exterior del país y la tabla de solicitud de imprevistos, siendo la historia de usuario número 3 (ver tabla 3.3) dividida en 1 (una) historia de usuario, la cual se muestra en la Tabla 3.23.

Historia de Usuario	
Número: 3.1	Nombre de historia: Solicitud de viáticos general
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 2
Descripción: los usuarios con acceso al módulo de empleado podrán a través del primer paso del formulario el ingresar la información general de la solicitud de viáticos, es decir información general y compartida entre cualquier solicitud de viáticos ya sea para al interior o exterior del país e imprevistos.	

Tabla 3.23 Historia de usuario: Solicitud de viáticos general.

➤ Plan de iteración

A continuación se muestran las tareas a completar durante esta iteración para el desarrollo del módulo de solicitudes, en las tablas 3.24, 3.25, 3.26, 3.27, 3.28 y 3.29.

Tarea	
Número: 1	Número de historia: 3, 4 y 5
Nombre tarea: Creación de las tablas para la solicitud de viáticos.	
Tipo de tarea: Administración	
Descripción: Se crea las tablas en la base de datos necesarias para la creación de viáticos interiores, viáticos exteriores e imprevistos.	

Tabla 3.24 Tarea: Creación de las tablas para la solicitud de viáticos.

Tarea	
Número: 2	Número de historia: 3, 4 y 5
Nombre tarea: Mensajes de validación para la solicitud de viáticos	
Tipo de tarea: Desarrollo	
Descripción: Se crean mensajes para la validación de las variables que se ingresan al sistema.	

Tabla 3.25 Tarea: Mensajes de validación para la solicitud de viáticos.

Tarea
Número: 3 Número de historia: 3, 4 y 5
Nombre tarea: Creación de formularios para la solicitud de viáticos
Tipo de tarea: Desarrollo
Descripción: Se crean los formularios correspondientes para la solicitud de viáticos.

Tabla 3.26 Tarea: Creación de formularios para la solicitud de viáticos.

Tarea
Número: 4 Número de historia: 3, 4 y 5
Nombre tarea: Listado de solicitudes
Tipo de tarea: Desarrollo
Descripción: Se crean todos los listados de solicitudes tanto aprobadas, rechazadas y pendientes.

Tabla 3.27 Tarea: Listado de solicitudes.

Tarea
Número: 5 Número de historia: 3, 4 y 5
Nombre tarea: Creación de status de solicitudes
Tipo de tarea: Desarrollo
Descripción: Se asocia a la creación de solicitud de viáticos un status para poder indicarle al usuario por cual nivel de aprobación se encuentra la solicitud.

Tabla 3.28 Tarea: Creación de status de solicitudes.

Tarea
Número: 6 Número de historia: 3, 4 y 5
Nombre tarea: Creación del módulo de pasos para la solicitud de viáticos
Tipo de tarea: Desarrollo
Descripción: El sistema será capaz de mostrar los formularios correspondientes a la solicitud de viáticos e imprevistos a través de pasos, para que en una misma solicitud indique viáticos al interior, exterior e imprevistos.

Tabla 3.29 Tarea: Creación del módulo de pasos para la solicitud de viáticos.

3.6.2.2 Diseño

En esta fase se agregó al diseño de la base de datos se puede observar en la Figura 3.13.

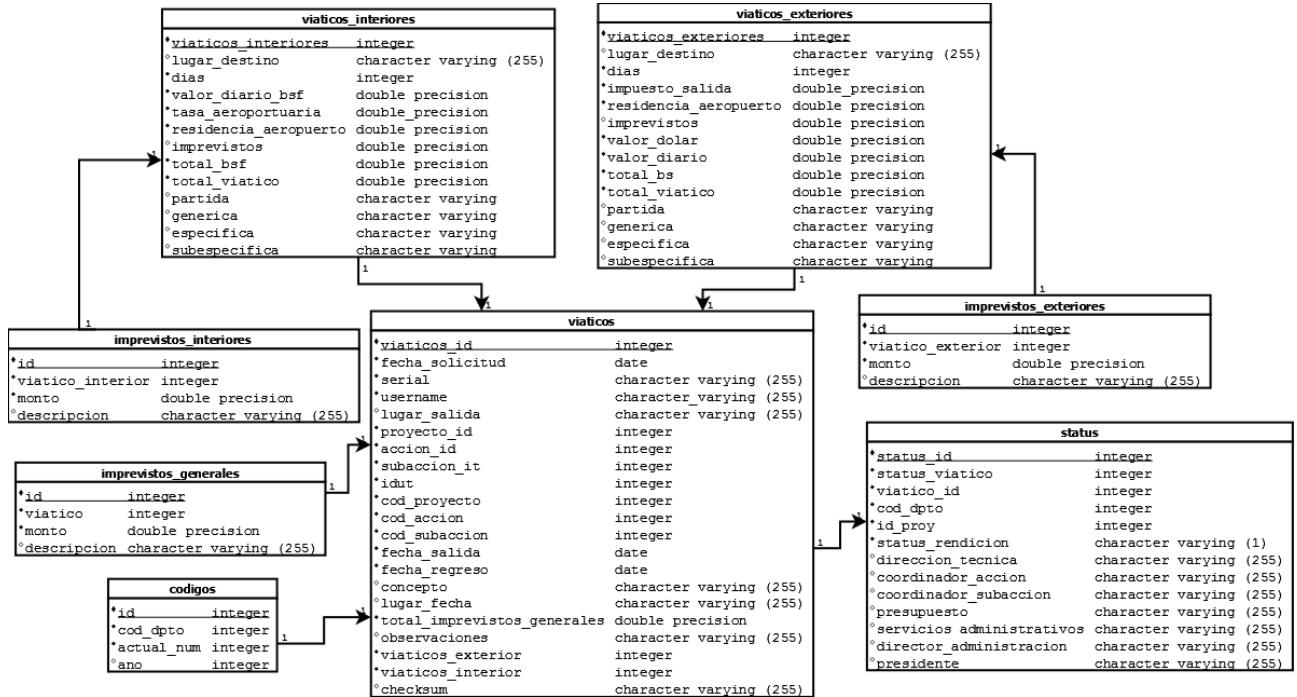


Figura 3.11 Diagrama de la base de datos en la segunda iteración.

Para esta iteración se trabajó en la implementación de las tablas de viáticos e imprevistos divididas en las tablas generales, interiores y exteriores, para poder mantener la información clasificada para cada caso y así acceder rápidamente a ella. También se observa la tabla codigos destinada a almacenar la información referente a los seriales que le será asignado a cada solicitud, dependiendo del año en curso y el departamento al que pertenece la solicitud.

Es importante destacar que para el desarrollo de este modelo se empezó a tomar en cuenta la información que era necesaria para la aprobación de los viáticos y así tener claro qué se debe almacenar y en qué sitio para mantener la información sin redundancia.

3.6.2.3 Codificación

Para la implementación de las distintas solicitudes en esta iteración, se acordó desarrollar una única solicitud que englobara todas las posibles opciones a través de un formulario por pasos, durante el cual los usuarios suministrarían la información necesaria a la solicitud deseada. Cada uno de los pasos existentes en el formulario corresponde a un tipo de solicitud que el empleado puede realizar incluyendo un último paso de confirmación de la solicitud.

Para llevar el control de los pasos del formulario y de las solicitudes de viáticos se implementó una solución mostrada en la Figura 3.14, destacando las funciones `current_step`, `steps`, `next_steps` y `previous_step` de la Figura 3.15 que permiten el control de la lógica de los pasos dentro del formulario.

```
class Viatico < ActiveRecord::Base

  self.table_name = 'sisviaticos.viaticos'
  self.primary_key = :viaticos_id
  attr_accessible :subaccion_id,:imprevistos_general_attributes,:serial,
    :total_imprevistos_generales,:viaticos_id,:concepto,
    :accion_id,:fecha_Regreso,:fecha_Salida,
    :fecha_solicitud,:lugar_fecha,:lugar_salida,
    :observaciones,:proyecto_id,:username,
    :viaticos_interior,:viaticos_exterior,:cod_proyecto,
    :cod_subaccion,:cod_accion,:idut

  has_many :imprevistos_general, :foreign_key => "viatico"

  accepts_nested_attributes_for :imprevistos_general,
    :reject_if=> lambda{|a| a[:descripcion].blank?},
    :allow_destroy=> true

  belongs_to :usuario, :foreign_key => "username"
  belongs_to :plan, :foreign_key => "proyecto_id"
  belongs_to :accion, :foreign_key => "accion_id"
  belongs_to :status, :foreign_key => "viatico_id"
  belongs_to :proyecto, :foreign_key => "subaccion_id"
  belongs_to :viatico_interior, :foreign_key => "viaticos_interior"
```

Figura 3.12 Código ActiveRecord de viáticos

```

def current_step
    @current_step || steps.first
end

def steps
    %w[solicitud interior exterior imprevisto confirmacion]
end

def next_step
    self.current_step = steps[steps.index(current_step)+1]
end

def previous_step
    if !self.first_step?
        self.current_step = steps[steps.index(current_step)-1]
    end

```

Figura 3.13 Código de los pasos para el formulario de viáticos.

Además, en la implementación de las secciones de imprevistos en las distintas solicitudes se utilizó jQuery para la creación de campos de formularios dinámicos como se muestra en la Figura 3.16 con las restricciones que se establecieron en las historias de usuario.

Imprevistos para el Interior

Descripción	Monto (Bs.)	
Corte de Maleza	250	Eliminar
Seleccione un Imprevisto		Eliminar

[Agregar un Imprevisto Nuevo](#)

Figura 3.14 Campos dinámicos para la generación de imprevistos.

Para el seguimiento y control de la aprobación de las solicitudes se implementó un esquema en el que cada solicitud tendrá una referencia a la tabla `status` la cual permitirá obtener información sobre el nivel de aprobación actual y las personas que han aprobado o rechazado la solicitud.

En la presentación de múltiples solicitudes al usuario se mantuvo el mismo esquema y estilo de las listas que se desarrolló durante la primera iteración. La

generación de las solicitudes en formato PDF (Anexo 3), fue elaborado a través del Plug-in wicked_pdf¹ cuya función principal es la generación de un archivo en formato PDF a partir de una vista en HTML facilitando el desarrollo y la gema barby² para la generación de la imagen del código de barras. Además, estas funciones fueron englobadas en la biblioteca de la Figura 3.17 para que fuesen usadas en cualquier de las secciones del sistema que así lo requiera.

```
module Generacionpdf

def solicitudpdf

require 'barby'
require 'barby/barcode/code_128'
require 'barby/outputter/png_outputter'

@barcode = @variableshash = @hashsolicitud = nil

@viatico=Viatico.find_by_viaticos_id(params[:id]) if params[:id]!=nil
if @viatico!=nil
  @usuario= @viatico.usuario
  @interior= @viatico.viatico_interior if @viatico.viaticos_interior!=nil
  @exterior= @viatico.viatico_exterior if @viatico.viaticos_exterior!=nil
```

Figura 3.15 Código donde se crea un pdf con su código de barra.

En total se crearon diecinueve (19) vistas, un (1) controlador y siete (7) modelos para cumplir con las historias de usuarios estipulados en esta iteración.

¹ Wicked_pdf, Plug-in para Ruby on Rails que permite la generacion de PDF a partir de una página en lenguaje HTML ,recuperado el 16 de Mayo de 2013, de http://rubygems.org/gems/wicked_pdf

² Barby, gema de Ruby que permite la generación de código de barras, recuperado el 16 de Mayo de 2013, de <http://rubygems.org/gems/barby>

3.6.2.4 Prueba

Durante esta iteración las pruebas de validación se realizaron con el usuario igual a la primera iteración y su ejecución fue realizada por el mismo equipo de trabajo anterior. De estas pruebas surgió la solicitud de colocar el monto de traslado al aeropuerto de forma fija ya que Funvisis para esta solicitud posee un monto establecido y la posibilidad de colocar días fraccionados sin superar el límite de días comprendido entre las fechas suministradas. En las figuras 3.16, 3.17 y 3.18 se observan algunas capturas de pantalla de la interfaz gráficas y algunos valores de las pruebas realizadas.

Nueva Solicitud



Datos Personales:

Nombre: Cesar
Apellido: Mora
Cédula: 18467695
Cargo: Tesista

Datos de la solicitud:

Lugar salida *:

Fecha salida *:

Fecha de regreso *:

Concepto *:

Trabajo especial de grado

Proyecto *:

Prevención y Mitigación de Riesgos ante Eventos Adversos ▼

Acción *:

Formación para la Apropriación Social del Conocimiento en Prevención ante Eventos Adversos ▼

Sub acción *:

Microzonificación sísmica para la planificación urbana y la gestión local de riesgo ▼

Observaciones:

(*) campos obligatorios

Continuar

Figura 3.16 Vista de la solicitud de viáticos general.

Nueva Solicitud



Datos Personales:

Nombre: Cesar

Apellido: Mora

Cédula: 18467695

Cargo: Tesista

Viáticos para el Interior

Lugar de destino *:

Portuguesa ▼

Número de días *:

2.0 ▼

Tasa aeropuertaria Bs.:

Solicitud de traslado de residencia / aeropuerto y viceversa

Imprevistos para el Interior

Descripción	Monto (Bs.)	
Corte de maleza ▼	200	Eliminar
Ropa y Equipamiento ▼	250	Eliminar

[Agregar un Imprevisto Nuevo](#)

(*) campos obligatorios

[Continuar](#) [Cancelar](#) [Omitir](#) [Volver](#)

Figura 3.17 Vista de la solicitud de viáticos para el interior del país.

Nueva Solicitud



Datos Personales:

Nombre: Cesar

Apellido: Mora

Cédula: 18467695

Cargo: Tesista

Datos de la Solicitud

Lugar de Salida: Caracas

Fecha Salida: 07-03-2013

Fecha de Regreso: 03-04-2013

Concepto: Trabajo especial de grado

Proyecto: Prevención y Mitigación de Riesgos ante Eventos Adversos

Acción: Formación para la Apropiaciación Social del Conocimiento en Prevención ante Eventos Adversos

Sub-Acción: Microzonificación sísmica para la planificación urbana y la gestión local de riesgo de las ciudades en los ejes de desarrollo: caso piloto Guarenas y Guatire. CUBA-Venezuela, FONDEN.

Cálculo de Viáticos al Interior

Lugar de Destino: Portuguesa

Números de Días: 2.0

Valor Diario en Bs: 77.0 Bs

Total en Bs: 154.0 Bs

Tasa Aeropuertaria: 0 Bs

Traslado de Residencia / Aeropuerto y Viceversa: 0.0 Bs

Imprevistos Interior

Descripción	Monto
Corte de Maleza	250.0 Bs
Ropa	100.0 Bs
Total	350.0 Bs

Total Viático Interior: 504.0 Bs

Finalizar **Cancelar** **Volver**

Figura 3.18 Vista para la confirmación de los datos de la solicitud de viáticos.

3.6.3 Iteración III - Rendición de cuentas

Esta iteración consiste en desarrollar un formulario para la realización de rendición de cuentas luego de finalizar la actividad encomendada por la fundación. A continuación se expone la planificación, diseño, codificación y pruebas que tuvieron lugar durante la tercera iteración del proyecto.

4.1.1.1 Planificación

En esta fase de desarrollo se especifican las historias de usuario involucradas, el plan de entrega, el plan de esta iteración y las reuniones realizadas para esta tercera iteración.

➤ Historias de usuario

La historia de usuario involucrada en esta iteración es la especificada en la Tabla 3.6 rendiciones de cuenta, siendo dividida en 4 historias de usuario adicionales, las cuales se muestran en las tablas 3.30, 3.31, 3.32 y 3.33.

Historia de Usuario	
Número: 6.1	Nombre de historia: Rendiciones
Usuario: Empleado	
Riesgo en desarrollo: Medio	Iteración asignada: 3
Descripción: El usuario de tipo empleado indica la información general de la rendición de cuentas.	
Observaciones: El usuario ingresará la información que se tiene en cuenta tanto si la solicitud fue para el interior o exterior del país.	

Tabla 3.30 Historia de usuario: Rendiciones.

Historia de Usuario	
Número: 6.2	Nombre de historia: Relaciones Interiores
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 3
<p>Descripción: El usuario ingresa los gastos realizados en alojamiento, alimentos, combustible y otros gastos realizados durante la actividad encomendada por la fundación.</p> <p>Observaciones: Estas relaciones de gastos son los realizados en una solicitud en el interior del país.</p>	

Tabla 3.31 Historia de usuario: Relaciones interiores.

Historia de Usuario	
Número: 6.3	Nombre de historia: Relaciones Exteriores
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 3
<p>Descripción: El usuario ingresa los gastos realizados en alojamiento, alimentos, combustible y otros gastos realizados durante la actividad encomendada por la fundación.</p> <p>Observaciones: Estas relaciones de gastos son los realizados en una solicitud de viatico al exterior del país.</p>	

Tabla 3.32 Historia de usuario: Relaciones exteriores.

Historia de Usuario	
Número: 6.3	Nombre de historia: Relaciones Generales
Usuario: Empleado	
Riesgo en desarrollo: Alto	Iteración asignada: 3
Descripción: El usuario ingresa los gastos realizados en alojamiento, alimentos, combustible y otros gastos que surgieron durante la actividad encomendada por la fundación.	
Observaciones: Estas relaciones de gastos son los realizados en una solicitud de imprevistos.	

Tabla 3.33 Historia de usuario: Relaciones generales.

➤ Plan de iteración

A continuación se muestran las tareas a completar durante esta iteración para el desarrollo del módulo de rendición de cuentas, en las tablas 3.34, 3.35, 3.36, 3.37, 3.38 y 3.39.

Tarea	
Número: 1	Número de historia: 6
Nombre tarea: Creación de las tablas para la rendición de cuentas	
Tipo de tarea: Administración	
Descripción: Se crea las tablas en la base de datos necesarias para la creación de rendiciones de cuentas y sus relaciones de gastos asociadas.	

Tabla 3.34 Tarea: Creación de las tablas para la rendición de cuentas.

Tarea	
Número: 2	Número de historia: 6
Nombre tarea: Mensajes de validación para las rendiciones de cuentas	
Tipo de tarea: Desarrollo	
Descripción: Se crean mensajes para la validación de las variables que se ingresan al sistema.	

Tabla 3.35 Tarea: Mensajes de validación para la rendición de cuentas.

Tarea	
Número: 3	Número de historia: 6
Nombre tarea: Creación de formularios para la rendición de cuentas	
Tipo de tarea: Desarrollo	
Descripción: Se crean los formularios correspondientes para realizar las rendiciones de cuentas.	

Tabla 3.36 Tarea: Creación de formularios para la rendición de cuentas.

Tarea	
Número: 4	Número de historia: 6
Nombre tarea: Listado de rendición de cuentas	
Tipo de tarea: Desarrollo	
Descripción: Se crean todos los listados de rendición de cuentas aprobadas, rechazadas y pendientes.	

Tabla 3.37 Tarea: Listado de rendición de cuentas.

Tarea	
Número: 5	Número de historia: 6
Nombre tarea: Creación de status de rendiciones de cuentas	
Tipo de tarea: Desarrollo	
Descripción: Se asocia a la creación de la rendición de cuentas un status para poder indicarle al usuario en qué nivel de aprobación va la rendición.	

Tabla 3.38 Tarea: Creación de status de rendiciones de cuentas.

Tarea	
Número: 6	Número de historia: 6
Nombre tarea: Creación del módulo de pasos para la rendición de cuentas	
Tipo de tarea: Desarrollo	
Descripción: El sistema será capaz de mostrar los formularios correspondientes a la rendición de cuentas a través de pasos, para que el usuario indique la información general de la rendición y luego realice la relación de gastos asociados a la solicitud de viáticos solicitada.	

Tabla 3.39 Tarea: Creación del módulo de pasos para la rendición de cuentas.

4.1.1.2 Diseño

En esta fase se agregó al diseño de la base de datos se puede observar en la Figura 3.13 para el manejo del proceso de rendición de cuentas.

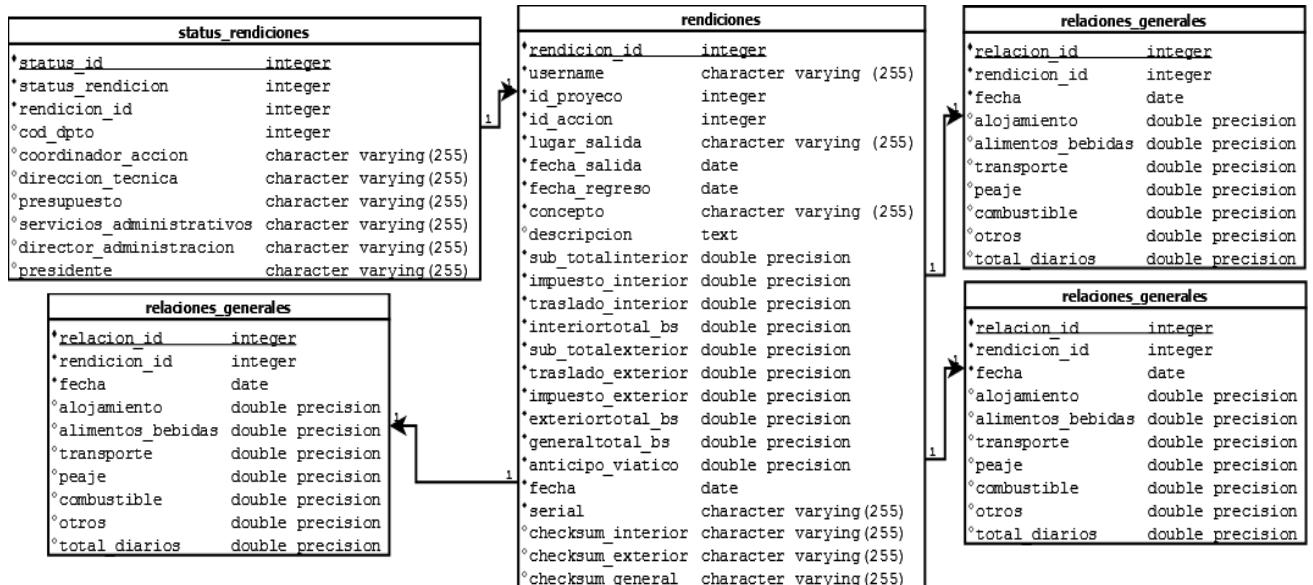


Figura 3.19 Diagrama de la base de datos en la tercera iteración.

En este diagrama se almacenan los datos generales que se presentan en cualquier rendición de cuentas, ya sea del exterior, interior o una rendición de

cuentas de imprevistos. Las relaciones de gastos como el alojamiento, el pasaje, alimentación u otros, serán almacenadas en su respectiva tabla donde se exprese si pertenece a una solicitud de viáticos en el interior, exterior o a unos imprevistos.

Es importante destacar que el desarrollo de la tabla `status_rendiciones` se enmarcó en un esquema similar a las aprobaciones de viáticos creadas en la segunda iteración.

4.1.1.3 Codificación

Para la implementación de las rendiciones de cuentas, se usó un esquema similar a la segunda iteración, con una única rendición de cuenta para diversos tipos de solicitudes, se incluye la generación de una rendición a solicitudes de viáticos existente en el sistema, cuyos datos los serán cargados a través de una función jQuery que realizará una consulta al servidor por un método GET invocando a la función `searchviatico` de la Figura 3.22, el cual retorna los datos de la solicitud de viáticos seleccionada.

```

def searchviatico #Función para cargar datos de una solicitud en rendición
    viatico = nil
    if params[:term]
        viatico = Viatico.find_by_serial(params[:term])
    end

    list = []
    if viatico!=nil and viatico.username == session[:user_id]
        plan= Plan.find_by_id(viatico.proyecto_id) if viatico.proyecto_id!=nil
        accion= Accion.find_by_id(viatico.accion_id) if viatico.accion_id!=nil

        if session[:rendicion_params] != nil
            session_fecha_salida=session[:rendicion_params]["fecha_salida"]
            session_fecha_regreso=session[:rendicion_params]["fecha_regreso"]
        end
        fecha_salida=(session_fecha_salida)?
            Date.parse(session_fecha_salida) : viatico.fecha_salida
        fecha_regreso=(session_fecha_regreso)?
            Date.parse(session_fecha_regreso):viatico.fecha_regreso

    list = [
        serial: viatico.serial,
        fecha: viatico.fecha_solicitud,
        lugar_salida: viatico.lugar_salida,
        fecha_salida: viatico.fecha_salida.strftime("%d-%m-%Y"),
        fecha_regreso: viatico.fecha_regreso.strftime("%d-%m-%Y"),
        proyecto: viatico.proyecto_id,
        proyecto_n: plan.descripcion_plan,
        accion: viatico.accion_id,
        accion_n: accion.descripcion_accion,
        concepto: viatico.concepto
    ]
end

    render json: list
end

```

Figura 3.20 Código del método searchviatico.

Para realizar la rendiciones y solicitar la información por día, se usó un formulario similar a la sección de imprevistos durante el proceso de las solicitudes, pero ahora solicitando la información de las rendiciones como se muestra en la Figura 3.23 en los cuales los días ya están pre-cargados por el sistema.

Nueva Rendición



Datos Personales:

Nombre: Cesar
Apellido: Mora
Cédula: 18467695
Cargo: Tesista

Relacion de gastos para Rendiciones O Imprevistos:

Fecha	Alojamiento	Alimentos y Bebidas	Gastos de Transporte	Gastos de Peaje	Gastos de Combustible	Otros Gastos	Total Diario (Bs.)	
03-03-2013							0	Eliminar
04-03-2013							0	Eliminar
05-03-2013							0	Eliminar

[Agregar un Nuevo Campo](#)

[Continuar](#) [Cancelar](#) [Volver](#)

Figura 3.21 Vista de relación de gastos por día.

Para la generación de las rendiciones en formato de PDF (Anexo 4), se hizo uso de las mismas gemas wicked_pdf y barby. Además se implementó el modelo similar al desarrollado en las solicitudes para el seguimiento y control de la aprobación de las rendiciones. Adicionalmente, se hizo uso de la gema carrierwave¹ encargada de gestionar y administrar la subida de los resumen de actividades al servidor.

En total se crearon siete (7) vistas, un (1) controlador y cinco (5) modelos para dar cumplimiento con las historias de usuarios pautadas en esta iteración.

¹ CarrierWave, gema que provee una manera simple y flexible para la carga de archivos de una aplicación en Ruby, recuperada el 16 de Mayo de 2013, de <http://rubygems.org/gems/carrierwave>

4.1.1.4 Pruebas

Durante esta iteración las pruebas se realizaron igual a las iteraciones anteriores y su ejecución fue realizada por el mismo equipo de trabajo anterior. Al ejecutar estas pruebas se registraron errores en la presentación de los formularios, estos fueron corregidos de forma inmediata. En las figuras que se muestran en esta sección se presentan algunas capturas de pantalla de la interfaz gráfica y algunos parámetros de las pruebas realizadas.

Nueva Rendición



Datos Personales:

Nombre: Cesar

Apellido: Mora

Cédula: 18467695

Cargo: Tesista

Datos de la rendición:

Solicitud *:

Seleccione una Solicitud ▼

- Seleccione una Solicitud
- CDT-0004/2013
- CDT-0002/2013**

Fecha salida *:

Fecha de regreso *:

Concepto *:

Proyecto *:

Seleccione un Proyecto ▼

Acción: *:

Seleccione una Acción ▼

Descripción

(*) campos obligatorios

Figura 3.22 Vista de Nueva rendición.

Rendición



La rendicion fue creada de forma exitosa

Datos Personales:

Nombre: Cesar

Apellido: Mora

Cédula: 18467695

Cargo: Tesista

Datos de la rendición:

Lugar de Salida: Revisando

Fecha Salida: 03-03-2013

Fecha de Regreso: 05-03-2013

Concepto: Revisando

Proyecto: Prevención y Mitigación de Riesgos ante Eventos Adversos

Accion: Formación para la Apropiación Social del Conocimiento en Prevención ante Eventos Adversos

Descripción del Gasto General:

Fecha	Alojamiento	Alimentos y Bebidas	Gastos de Transporte	Gastos de Peaje	Gastos de Combustible	Otros Gastos	Total Diarios
2013-03-03	50.0 Bs						50.0 Bs
2013-03-04		30.0 Bs					30.0 Bs
2013-03-05			60.0 Bs				60.0 Bs
							Sub Total 140.0 Bs

TOTAL: 140.0 Bs

Total en bs de viaticos Aprobados: 300.0 Bs

Diferencia: 160.0 Bs A favor de Funvisis

Resumen de actividades

[Generar PDF](#)

Figura 3.23 Vista de la rendición de cuentas con la información indicada por el usuario.

4.1.2 Iteración IV - Aprobaciones

Esta iteración consiste en el desarrollo del módulo para las aprobaciones de las solicitudes y rendiciones de cuentas. A continuación se lista la documentación de la planificación, diseño, codificación y pruebas que tuvieron lugar durante la cuarta iteración del proyecto.

4.1.2.1 Planificación

En esta fase del desarrollo se especifican con detalle las historias de usuario involucradas, el plan de entrega, el plan de esta iteración y las reuniones que se realizaron para esta cuarta iteración.

➤ Historias de usuario

Las historias de usuario involucradas en esta iteración son las especificadas en las tablas 3.7 y 3.8.

➤ Plan de iteración

A continuación se muestran las tareas a completar durante esta iteración en las tablas 3.40, 3.41, 3.42 y 3.43.

Tarea
Número: 1 Número de historia: 7 y 8
Nombre tarea: Modificación de la base de datos para las aprobaciones
Tipo de tarea: Administración
Descripción: Se modifica las tablas de la base de datos donde se almacena la información necesaria para registrar la aprobación o rechazo de la solicitud, quitando o agregando datos que sean necesarios.

Tabla 3.40 Tarea: Modificación de la base de datos para las aprobaciones.

Tarea	
Número: 2	Número de historia: 7 y 8
Nombre tarea: Validación de usuario para aprobar en más de un nivel de aprobación.	
Tipo de tarea: Desarrollo	
Descripción: Desarrollar la lógica del sistema en donde el usuario involucrado en más de un nivel de aprobación sea capaz aprobar de una vez los distintos niveles de la solicitud o rendición de cuentas y no deba volver a hacer el proceso por cada uno de los niveles requeridos.	

Tabla 3.41 Tarea: Validación de usuario para aprobar en más de un nivel de aprobación.

Tarea	
Número: 3	Número de historia: 7 y 8
Nombre tarea: Listado de solicitudes	
Tipo de tarea: Desarrollo	
Descripción: Se crean todos los listados de las solicitudes de viáticos que el usuario encargado debe aprobar, las que ya aprobó y las que han rechazado.	

Tabla 3.42 Tarea: Listado de solicitudes.

Tarea	
Número: 4	Número de historia: 6
Nombre tarea: Listado de rendición de cuentas	
Tipo de tarea: Desarrollo	
Descripción: Se crean todos los listados de rendición de cuentas tanto aprobadas, rechazadas y pendientes.	

Tabla 3.43 Tarea: Listado de rendición de cuentas.

4.1.2.2 Diseño

En cuanto al diseño de la base de datos, las tablas donde se trabaja el estatus de las solicitudes y rendiciones de cuentas fueron creadas en la segunda y tercera iteración, ya que era importante tener una idea de la información que se debería almacenar al respecto del estado que tenía cada una de estas. El esquema general de la base de datos se puede observar en el anexo 5.

En la codificación será mostrado y tratado el valor que tiene cada solicitud al ser aprobada o rechazada según el modelo de las tablas con las que se trabajó, en esta iteración utilizaron las tablas `status_rendiciones` y `status` donde en una se almacena el estado de las rendiciones de cuentas y en la otra el estado de las solicitudes de viáticos o imprevistos respectivamente.

4.1.2.3 Codificación

Para la implementación de las aprobaciones se utilizaron las tablas y modelos de `status` y `status_rendicion` desarrolladas durante las últimas dos iteraciones, además de las funciones en el modelo de usuario que retornaran los niveles de aprobación para las solicitudes y rendiciones en los cuales el usuario estaría autorizado, como también los proyectos, acciones y sub-acciones en los cuales el usuario es responsable.

Además, en esta iteración se acordó definir los valores para el campo `status` en los modelos, los cuales son establecidos en la Tabla 3.44 y la Tabla 3.45.

Nombre	Aprobado	Rechazado
Anulado por el usuario	0	0
Jefe de Departamento	1	-1
Dir. Técnica	2	-2
Jefe de la acción	3	-3
Dir. Planificación y Presupuesto	4	-4
Dir. Administración	5	-5
Serv. Administración	6	-6
Presidencia	7	-7

Tabla 3.44 Niveles de aprobación de las solicitudes.

Nombre	Aprobado	Rechazado
Jefe de Departamento	1	-1
Dir. Técnica	2	-2
Dir. Planificación y Presupuesto	3	-3
Dir. Administración	4	-4
Serv. Administración	5	-5
Presidencia	6	-6

Tabla 3.45 Niveles de aprobación de las rendiciones.

Para ambos casos la lógica implementada consistió en que cada solicitud aumenta su nivel de aprobación hasta alcanzar el final de la tabla, que concluiría en la aprobación por todos los departamentos de la fundación y en caso que algunos de los departamentos la rechazara, el valor asignado sería el valor negativo al nivel en que se encuentre la solicitud. En la Figura 3.26 se visualiza una muestra de la lógica implementada para el rechazo de las solicitudes.

```

def atributosrechazar(nivel,username)

niveles=nivel.to_f
case niveles
  when 7
    return {:presidente=> username , :status_viatico=> (-1)*niveles }
  when 6
    return {:servicios_administrativos=>username,:status_viatico=>(-1)*niveles}
  when 5
    return {:director_administracion=>username,:status_viatico =>(-1)*niveles}
  when 4
    return {:presupuesto=> username,:status_viatico=>(-1)*niveles}
  when 3
    return {:coordinador_subaccion=>username,:status_viatico=>(-1)*niveles}
  when 2
    return {:direccion_tecnica =>username,:status_viatico=>(-1)*niveles}
  when 1
    return {:coordinador_accion=>username,:status_viatico=>(-1)*niveles}
  else
    return nil
  end

```

Figura 3.24 Código para el rechazo de las solicitudes.

Para el caso de las solicitudes, cuando llega al nivel de aprobación de la dirección de planificación y presupuesto las solicitudes de viáticos para el interior o para el exterior debían ser imputadas, por tal motivo se desarrolló un formulario que permitiría realizar dichas imputaciones al momento de realizar la aprobación.

Durante el desarrollo de esta iteración, el cliente presentó la modificación en el logotipo de la fundación, por tal situación se decidió crear un mecanismo en el módulo de administración que permita cambiar el logotipo de la fundación, así como el membrete que están incluidos en los PDF. Para esta función adicional se usó nuevamente la gema carriewave para la carga de las nuevas imágenes al servidor.

En total se crearon un total de doce (12) vistas, 1 controlador y tres (3) modelos para cumplir con las historias de usuarios estipulados en esta iteración.

4.1.2.4 Pruebas

Durante esta iteración las pruebas de validación se realizaron igual a la primera iteración y su ejecución fue realizada por el mismo equipo de trabajo anterior. De estas pruebas surgió la solicitud de la modificación del mecanismo de aprobación para los jefes de departamentos, dirección técnica y jefe de sub-acción, para que al realizar una solicitud de viáticos no tuvieran que ser aprobadas por ellos mismos. En las siguientes figuras 3.25, 3.26, 3.27 y 3.28 se muestran algunas capturas de pantallas de la interfaz gráficas y algunos valores de pruebas realizadas.

Lista de Solicitudes de Viaticos Pendientes

Lista de Solicitudes de Viaticos Pendientes					
ID	Número Solicitud	Fecha Solicitud	Nivel de Aprobación	Proyecto	Departamento
47	CDT-0005/2013	11-04-2013	Jefe del Departamento	Proyecto de investigación aplicada a la gestión integral de riesgo en cuencas urbanas en el periodo 2007-2009. Misión ciencia 1, FONACIT 2007000939	Ciencias de la Tierra

Figura 3.25 Lista de solicitudes de viáticos pendientes.

ID	Número Solicitud	Fecha Rendición	Nivel de Aprobación	Departamento
17	CDT-0003/2013	08-05-2013	Jefe del Departamento	Ciencias de la Tierra

Figura 3.26 Lista de solicitudes de rendiciones de cuentas pendientes.

Viatico Pendiente

Datos Personales:

Nombre: Cesar
Apellido: Mora
Cédula: 18467695
Cargo: Tesista

Datos de la Solicitud

Fecha Solicitud: 2013-04-02
Serial: INF-0003/2013
Lugar de Salida: caracas
Fecha Salida: 2012-01-01
Fecha de Regreso: 2012-01-01
Concepto: asdas
Proyecto: Prevención y Mitigación de Riesgos ante Eventos Adversos
Accion: Formación para la Apropación Social del Conocimiento en Prevención ante Eventos Adversos
Observaciones: asdasd

Cálculo de Viáticos al Interior

Lugar de Destino: Anzoátegui
Número de Dias: 2.0
Valor Diario en Bs: 98.1 Bs
Total en Bs: 196.2 Bs
Tasa Aeropuertaria: 0 Bs
Traslado de Residencia / Aeropuerto y Viceversa: 0 Bs
Total Viático Interior: 196.2 Bs

Imputación

Partida	Específica	Genérica	Sub-Específica

Imputación Presupuestaria para la Solicitud al Interior

Seleccione una Imputación ▼

Figura 3.27 Vista de solicitud de viáticos para la asignación de la imputación.

Rendicion

Datos Personales:

Nombre: Yoly
Apellido: Diaz
Cédula: 13692722
Cargo: Técnico Electrónico II

Datos de la Rendición:

Lugar de Salida: caracas
Fecha Salida: 01-05-2013
Fecha de Regreso: 05-05-2013
Concepto: Mantenimiento de equipos
Proyecto: Prevención y Mitigación de Riesgos ante Eventos Adversos
Accion: Formación para la Apropiación Social del Conocimiento en Prevención ante Eventos Adversos

Descripción del Gasto Interior:

Fecha	Alojamiento	Alimentos y Bebidas	Gastos de Transporte	Gastos de Peaje	Gastos de Combustible	Otros Gastos	Total Diarios
2013-05-01	100.0 Bs						100.0 Bs
2013-05-02		120.0 Bs					120.0 Bs
2013-05-03			130.0 Bs				130.0 Bs
							Sub Total 350.0 Bs

Valor del Impuesto de Salida del País: 101.0 Bs

Traslado de Residencia / Aeropuerto y Viceversa: 100.0 Bs

TOTAL: 551.0 Bs

Total en bs de viáticos Aprobados para el Interior: Bs 490.5 Bs

Diferencia: 60.5 Bs A favor del Empleado

Resumen de actividades

Figura 3.28 Vista de aprobación para una rendición de cuentas.

Conclusiones y Recomendaciones

En este trabajo se automatizó el proceso de gestión de viáticos de Funvisis por medio del diseño y desarrollo de una aplicación web. Para el cumplimiento de este objetivo, en principio fue necesario obtener información sobre las metodologías existentes, con el fin de seleccionar una que se adaptará a las necesidades y posteriormente hacer un levantamiento de los requisitos fundamentales, para ello se investigó las aplicaciones afines dentro de otras empresas e incluso se conoció que en Funvisis ya existía una aplicación incompleta desarrollada en PHP. La información recolectada permitió determinar el flujo de actividades y tareas del proceso actual, para así proponer las modificaciones al flujo de trabajo de los procesos involucrados en la gestión de viáticos en Funvisis.

Para realizar la aplicación Web, se recurrió al método de desarrollo XP, y con esto se obtuvo como ventaja la adaptabilidad del sistema a las necesidades de Funvisis, ya que en cada reunión que se llevó a cabo se pudo identificar nuevos requerimientos o detalles que al principio cuando se realizó el levantamiento de la información no fueron tomados en consideración, permitiendo así la realización de las modificaciones pertinentes en base a las correcciones, corroborando entonces la practicidad y dinamismo de la utilización del método XP ante el cambio de los requisitos para el desarrollo de la aplicación. Como principal desventaja la necesidad de contar con la disponibilidad del usuario final, dado que en Funvisis las personas responsables e involucradas en el proceso no siempre estaban disponibles, ocasionando así retrasos en el desarrollo de la aplicación.

El uso del framework Ruby on Rails, provee un excelente medio de persistencia de datos que permitió a los desarrolladores centrar su atención en la implementación de la lógica del sistema, además de ofrecer herramientas (gemas)

tales como scaffold¹ que incrementa notablemente los niveles de producción, al contar con un alto porcentaje de código funcional auto generado y de fácil modificación.

El conjunto de pruebas realizadas durante la aplicación de la metodología XP, aseguraron que el sistema en desarrollo cumplía con los requisitos solicitados por el usuario y comprobar su correcto funcionamiento. Estas pruebas realizadas al final de cada módulo permitieron realizar correcciones recomendadas por el usuario y así reducir las ocurrencias de errores al finalizar el desarrollo de la aplicación.

Durante el desarrollo del sistema se presentaron diversos inconvenientes en el manejo de las solicitudes de viáticos, debido a la ausencia de las personas encargadas de aprobar o denegar la solicitud de viáticos y los trámites asociados a la misma; por lo que se planteó como solución la designación por parte del presidente de la institución a una nueva persona encargada y cuyo cambio en sistema sería ejecutado por el departamento de recursos humanos. Además, se indicó la limitante de las rendiciones de cuentas que será usado como sistema de monitoreo para estos trámites, motivado al requisito indispensables del anexo de las facturas originales a la solicitud realizada.

Se recomienda para complementar del sistema, el desarrollo e inclusión de firmas electrónicas, para que con esto quede constancia de quienes aprobaron o rechazaron las solicitudes sean las personas designadas en llevar a cabo dichas acciones. Además de contemplar un periodo de prueba en donde se mantenga el uso del documento en físico generado por el sistema que permita verificación de la información, y así ir adiestrando a los usuarios en el nuevo flujo de trabajo para la gestión de viáticos y el resto de procesos implementados en la aplicación web.

¹ Scaffold es un script de generación de código Ruby on Rails con funcionalidad CRUD presente en el modelo, vista y controlador de la aplicación

Referencias

- [1] Funvisis, Estructura Organizativa, Recuperado el 15 de Mayo de 2012, de http://www.funvisis.gob.ve/estructura_org.php
- [2] Beck, K. Extreme Programming Explained. Embrace Change., Pearson Education, 1999. Traducido al español como: Una explicación de la programación extrema. Aceptar el cambio., Addison Wesley, 2000.
- [3] Begoña C., TALISMAN: Desarrollo ágil de Software con Arquitecturas Dirigidas por Modelos, Tesis Doctoral, Universidad de Oviedo, 2007. Recuperado el 15 de Mayo de 2012, de <http://es.scribd.com/doc/71854769/TalismanCrisPelayo>.
- [4] Canós J., Letelier P., Penadés C., Metodologías Ágiles en el desarrollo de software, Trabajo de Investigación, Universidad Politécnica de Valencia, 2007. Recuperado el 15 de Mayo de 2012, de <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
- [5] Director de Ruby en español. Acerca de Ruby. Recuperado el 26 de marzo de 2013, de <http://www.ruby-lang.org/es/about/>
- [6] Rafael Méndez. Sobre PostgreSQL. Recuperado el 26 de marzo de 2013, de http://www.postgresql.org.es/sobre_postgresql
- [7] Gerente de CANTV. Solicitud de viáticos. Recuperado el 28 de marzo de 2013, de <http://www.geocities.ws/mogala03/ayb/T1.html>

[8] Ministerio del Poder Popular para la Cultura. Proceso de solicitud de viáticos.

Recuperado el 28 de marzo de 2013, de

<http://colabora.softwarelibre.gob.ve/home/sigesp/MANUAL%20DE%20VIATIC>

O%20FUNDACION%20CENTRO%20NACIONAL%20DE%20FOTOGRAFIA.p

df

Anexos

Anexo I

Reglamento de viáticos y pasajes

Anexo II

Planilla de control y seguimiento de la correspondencia

Anexo III

Planilla de solicitud de viáticos

Anexo IV

Planilla de relación de gastos de viaje

Anexo V

Esquema de la Base de datos