Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL DE ACCESO Y GESTIÓN DE DOCUMENTOS DIGITALES USANDO HUELLA DACTILAR

Trabajo Especial de Grado presentado ante la Ilustre

Universidad Central de Venezuela

Por las Bachilleres

Alfaro B. Diana C. y Martínez R. Rut

Para optar al título de Licenciado en Computación

Tutores: Prof. Carlos Acosta Y Profa. Ana Morales

Caracas, Noviembre 2013

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

ACTA DEL VEREDICTO

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado presentado por las bachilleres **Diana Carolina Alfaro Bazán**, cédula de identidad número **V-18.030.633** y **Rut Martínez Ramírez**, cédula identidad número **V-20.049.137**, bajo el título "**Desarrollo de una aplicación web para el control de acceso y gestión de documentos digitales usando huella dactilar**", a los fines de cumplir con el requisito legal para optar al título de Licenciada en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los miembros del Jurado, se fijó el día 5 de noviembre de 2013, a las 8:00 am, para que sus autoras lo defendieran en forma pública, en la Sala de Reuniones del Centro de Computación, lo cual se realizó mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas a los cinco días del mes de noviembre del año 2013, dejándose también constancia de que, conforme a lo dispuesto en la normativa jurídica vigente, actuaron como Coordinadores del Jurado el Tutor Académico, Profesor Carlos Acosta y la Tutora Académica, Profesora Ana Morales.

_____	_____
Prof. Carlos Acosta	**Profa. Ana Morales**
Tutor Académico	**Tutora Académica**
_____	_____
Prof. Robinson Rivas	**Profa. Adriana Liendo**
Jurado Principal	**Jurado Principal**

RESUMEN

El presente trabajo especial de grado se basó en el desarrollo de una aplicación web para el control de acceso físico y la gestión de documentos utilizando certificados digitales y la huella dactilar como característica biométrica para la firma digital de documentos en formato PDF. El proyecto estuvo enmarcado en una adaptación de la metodología de la programación extrema (metodología XP). Adicionalmente para el desarrollo de la aplicación se utilizaron diversas tecnologías como Java, el sistema manejador de base de datos PostgreSQL y el Kit de Desarrollo de Software (SDK) Verifinger de Neurotechnology. Como resultado se obtuvo un sistema que permite controlar el acceso lógico y físico además de facilitar la gestión de documentos. **Palabras claves:** *Certificado Digital, Firma Digital, XP, Verifinger, Neurotechnology.*

DEDICATORIA

Este Trabajo Especial de Grado se lo dedico a mis padres que han esperado tanto este momento, quienes me han dado su apoyo incondicional y porque gracias a ellos soy la persona que soy y he llegado a donde estoy. Jamás los defraudaré. Los quiero,

Diana Alfaro

DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional.

Agradezco también a mi Padre Julio Martínez y Madre Justina Ramírez por ser el apoyo en mi carrera, en mis logros y por su gran amor incondicional. También a mis hermanas Ester Martínez y Verónica Martínez por sus apoyos y sus consejos me ha ayudado a afrontar los retos que se me han presentado a lo largo de mi vida. A mis dos hermosos sobrinos Jonatán Elías Pinzón Martínez y Melco David Pinzón Martínez por quererme, pronto los veré en la UCV. A mí querido cuñado Yobanny Pinzón Serrano por su ayuda y consejo. A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

A mi entrenador, amigo y familia Alfredo Hernández por su apoyo incondicional en el transcurso de mi carrera universitaria, por compartir momentos de alegría, tristeza y demostrarme que siempre podré contar con él.

Rut Martínez

AGRADECIMIENTOS

Ante todo a Dios por siempre darme fuerzas y aliento para seguir adelante y nunca desfallecer.

A mi mamá María Bernarda Bazán Bianchi y mi papá Felipe Humberto Alfaro Rubiños por estar siempre al pendiente de mí, por escucharme, por darme sus palabras de apoyo y por soportarme en esos días y noches estresantes.

A mis amigos y demás compañeros por compartir gratos momentos durante esta hermosa carrera.

A mis amigos veganos Amarilis Obelmejia y Andrés Parra por estar siempre en cada momento en que necesitaba más ánimo, muchísimas gracias.

A la magna e ilustre Universidad Central de Venezuela, por brindarme una educación de excelencia a nivel académico y personal. A mí querida Facultad de Ciencias, a la Escuela de Computación, a todos los profesores, profesoras y preparadores con los que tuve la oportunidad y el honor de compartir y quienes contribuyeron en mi formación. A mis tutores el profesor Carlos Acosta y la profesora Ana Morales por su paciencia y disposición en todo momento, a la profesora Adriana Liendo por su interés en el tema desarrollado. Igualmente al profesor Robinson Rivas por estar siempre pendiente del desarrollo de este Trabajo Especial de Grado y sus palabras de aliento en todo momento.

Al Centro de Computación y a todos mis compañeros que laboran allí por brindarme el tiempo y permitirme utilizar las instalaciones y los recursos tecnológicos para llevar a cabo y culminar este Trabajo Especial de Grado. Lugar donde he tenido la oportunidad de crecer personal y profesionalmente y al cual estoy muy agradecida.

A mis amigos y compañeros de Corporación Perzona, también muchísimas gracias, porque a pesar del poco tiempo siempre han estado al pendiente de mí, del desarrollo de este trabajo y me han apoyado en todo lo que he necesitado.

A todas las personas que estuvieron involucradas de una u otro manera en este momento tan importante en mi carrera y con las que compartí lindos momentos, gracias por su tiempo y disposición.

Gracias a todos,

Diana Alfaro

AGRADECIMIENTOS

Agradezco a Dios por darme la oportunidad de finalizar mi carrera profesional y por el apoyo que me brindo durante estos 5 años. No tengo apalabras para agradecer todo lo que me has dado.

Agradezco a mi hermosa familia por su comprensión, amor y paciencia. Los quiero mucho y gracias por estar siempre.

Agradezco a mis compañeros y profesores de la escuela de computación (UCV) por el apoyo que me han brindado durante la carrera.

Agradezco al club de atletismo UCV por permitir representar a mi universidad en diversos eventos deportivos y por el apoyo brindado por mis compañeros atletas.

Finalmente a Profesor Carlos Acosta, Profesora Ana morales y Compañera Diana Alfaro porque cada una con sus valiosas aportaciones hicieron posible este trabajo de grado.

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este trabajo de grado.

Gracias a todos.

Rut Martínez

TABLA DE CONTENIDO

LISTAS DE FIGURAS

LISTAS DE TABLAS

INTRODUCCIÓN

Tradicionalmente, las organizaciones hacen uso de contraseñas o de tarjetas ID para permitir el acceso a lugares restringidos. Sin embargo, en estos sistemas la seguridad puede ser fácilmente violada cuando se divulga la contraseña a un usuario no autorizado o cuando un impostor roba la tarjeta. Los sistemas biométricos son un medio adecuado que cumplen con un mayor grado de seguridad. Las técnicas biométricas usan características o comportamientos fisiológicos propios de cada individuo para identificarlo. En especial, la identificación por medio de huellas dactilares es una de las formas más representativas de la utilización de la biometría y más factibles económicamente con respecto a otras técnicas biométricas.

Así mismo, el manejo de documentos en las organizaciones es muy común. Generalmente están respaldados por una o varias firmas de los entes involucrados. Siendo, todavía la firma manuscrita la forma más utilizada y "confiable" para relacionar un documento con una persona en particular, de manera legal. Sin embargo, este método tiene de diversas imperfecciones, entre ellas la posibilidad de falsificación y las dificultades en el proceso de verificación de la firma.

La firma en sí, involucra dos acciones: la acción de firmar y la acción de verificación de la firma. La acción de firmar, en el caso de la firma manuscrita, consiste en que una persona deje su firma por escrito; mientras que la acción de verificación es un proceso más complejo debido a que se requiere en algunos casos la utilización de tecnología altamente sofisticada y siempre con probabilidad de error. Otra limitación que se presenta durante el manejo de documentos firmados es la necesidad de contar con la presencia física y simultánea de las personas involucradas, lo cual se convierte en proceso lento y costoso en una organización.

Una solución interesante a estos problemas puede ser a partir del uso de una nueva tecnología capaz de reemplazar la firma manuscrita y es denominada firma digital. Tiene claras ventajas respecto a la firma manuscrita, razón por la cual muchos países hacen los esfuerzos necesarios para implantarla en sus naciones, promulgando leyes y promoviendo su uso.

La Fundación Venezolana de Investigaciones Sismológicas (FUNVISIS) es una institución que promueve de forma permanente investigaciones y estudios especializados en sismología, ciencias geológicas e ingeniería sísmica, con el propósito de contribuir a la reducción de la vulnerabilidad en el país.

FUNVISIS cuenta actualmente con un sistema de control de acceso a espacios físicos, que registra las entradas y salidas de cada empleado, igualmente el sistema permite acceder a ciertas áreas restringidas. El sistema posee funcionalidades tales como: generación de reportes, autenticación unívoca del empleado para el control de acceso y aunque es una buena solución, presenta el inconveniente con los equipos biométricos (lectores de huellas dactilares) que fueron instalados en FUNVISIS los cuales no proveen ningún SDK (Kit de Desarrollo de Software o *Software Development Kit*) genérico que permita en un futuro realizar modificaciones o ajustes de acuerdo a las necesidades de la institución. Aunado a ello, FUNVISIS como muchas organizaciones maneja grandes volúmenes de documentos en físico (papel), que en la mayoría de las ocasiones deben ser manejados por varias personas en los distintos departamentos con los que cuenta la institución. Documentos que deben ser firmados por las partes correspondientes.

Para ello se planteó como objetivo principal de este Trabajo Especial de Grado el desarrollo de una aplicación web para el control de acceso, gestión y autenticación de documentos digitales usando huella la dactilar.

Si bien es cierto que ya se están desarrollando aplicaciones independientes que permiten el control de acceso con huella dactilar y aplicaciones para firmar documentos digitalmente, el objetivo es utilizarla huella dactilar para el control de acceso físico, el acceso a la aplicación web y para la gestión de documentos digitales, integrados en una única aplicación web para optimizar su uso.

El presente documento de Trabajo Especial de Grado se encuentra estructurado de la siguiente manera:

- Capítulo 1: se describe el planteamiento del problema donde se presenta el objetivo general y los objetivos específicos de este Trabajo Especial de Grado y su justificación.
- Capítulo 2: expone el marco teórico donde se muestran las bases en las que se fundamenta el desarrollo de este trabajo. Se definen los conceptos básicos, funciones y características de la biometría por huella dactilar y los conceptos básicos, funciones y características de la firma digital. Además, se describe las aplicaciones web cliente-servidor, su funcionamiento, arquitectura, características principales y tecnologías usadas para su desarrollo.
- Capítulo 3: se describe como se realizó la adaptación de la metodología de la Programación Extrema (XP) para el desarrollo del sistema planteado y se explica en detalle las diferentes etapas del proceso de implementación mediante la metodología XP. Se incluye las pruebas de aceptación llevadas a cabo junto con el análisis de los resultados obtenidos.

Finalmente se presentan las conclusiones, referencias bibliográficas y anexos utilizados durante el desarrollo de este Trabajo Especial de Grado.

CAPITULO 1. PLANTEAMIENTO DEL PROBLEMA

En este capítulo se describe el contexto y la definición del problema, luego se expone el objetivo general, los objetivos específicos, la metodología para el desarrollo de la solución, las tecnologías y plataformas de esta investigación, así como la justificación de la realización del presente Trabajo Especial de Grado.

1.1. Contexto del problema

Actualmente se utilizan los documentos en formato físico (papel), como uno de los medios para difundir la información dentro y fuera de las organizaciones. Este proceso de difusión y manipulación de documentos, puede ocasionar problemas de seguridad tales como: violaciones o fallas en la autenticación, confidencialidad e integridad de los documentos físicos, traspapeleo (dada la manipulación de los documentos dentro de las organizaciones, lo que ocasiona extravíos), difícil seguimiento (cada documento difundido dentro de las organizaciones, conlleva a un rastreo para verificar que la información llegue a su destino), el manejo de grandes volúmenes de papel almacenado (al almacenar o archivar los documentos físicos, ocupan espacio dentro de las organizaciones), gastos innecesarios (espacios físicos, papel, tinta...) dentro de las organizaciones. Son sólo algunas de las problemáticas que presentan las organizaciones durante la gestión de documentos.

Otro punto importante es el control de acceso a instalaciones físicas, que constituye uno de los servicios indispensables de seguridad dentro de las organizaciones, existiendo una amplia variedad de formas y métodos para implementarlo, como es el uso de equipos biométricos (entre ellos los lectores de huella dactilar) para restringir el acceso a dichos espacios físicos.

Dentro de las organizaciones que presentan los problemas antes expuestos, se encuentra la Fundación Venezolana de Investigaciones Sismológicas (FUNVISIS), la cual es una institución que promueve de forma permanente investigaciones y estudios especializados en sismología, ciencias geológicas e ingeniería sísmica, con el propósito de contribuir a la reducción de la vulnerabilidad en el país.

FUNVISIS cuenta actualmente con diferentes departamentos, donde cada departamento tiene funcionalidades diferentes, por ende necesitan comunicarse entre ellos. La figura 1 muestra la estructura formal de FUNVISIS.

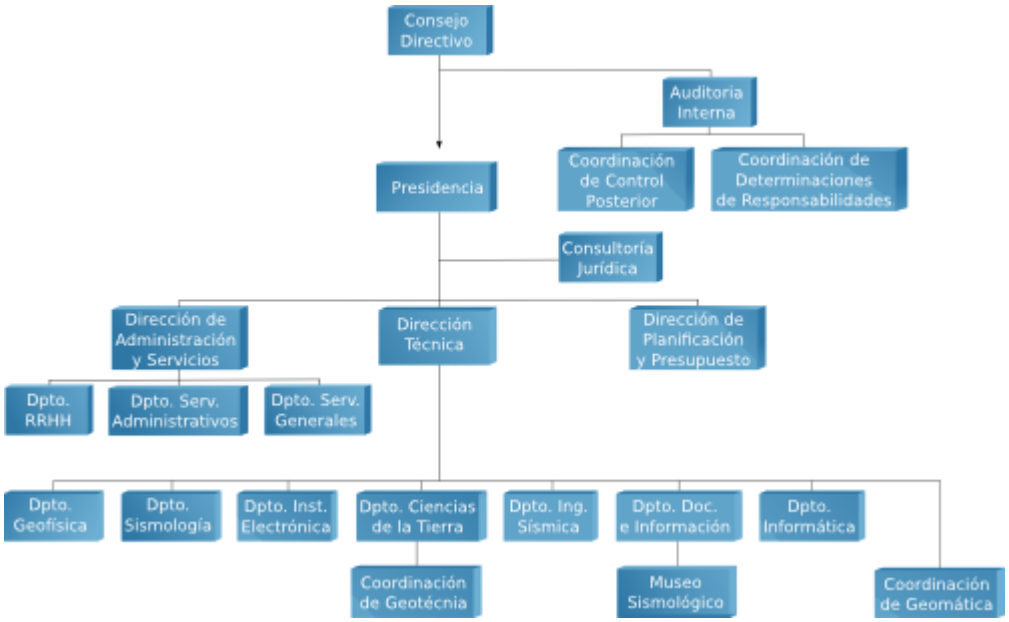


Figura 1: Estructura organizacional de FUNVISIS

En la siguiente sección se presentará un panorama sobre la problemática actual que presenta FUNVISIS con respecto al control de acceso físico y la gestión de documentos dentro de la institución.

1.2. Definición del problema

FUNVISIS realiza una serie de funciones las cuales deben ser ejecutadas eficientemente, entre ellas se encuentra la gestión de documentos dentro de la organización. Este proceso de gestión de documentos es llevado a cabo muchas veces de forma manual, dirigiéndose la persona responsable al departamento correspondiente, llevando un control del estado de la solicitud entre los distintos departamentos en los cuales debe ser procesado el documento.

Ejemplos de este proceso de gestión de documentos se muestran con el flujo de tres tipos de documentos en FUNVISIS: viáticos (Ver Figura 2), memos dirigidos a Recursos Humanos (Ver Figura 3) y rendición (Ver Figura 4):

Figura 2: Descripción del flujo para viáticos

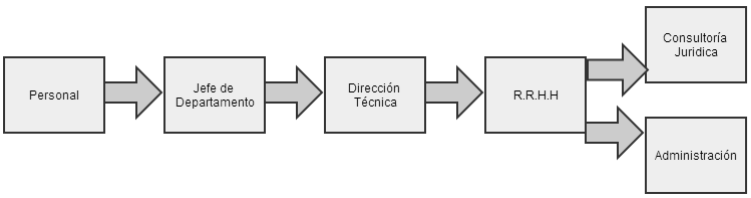


Figura 3: Descripción del flujo para memos dirigidos a R.R.H.H

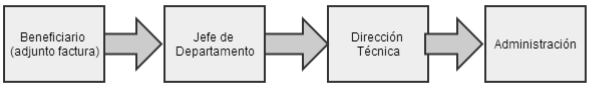


Figura 4: Descripción del flujo para rendición

No sólo la manipulación de documentos es un punto crítico, el acceso a ciertos perímetros o instalaciones (por ejemplo, sala de servidores), el acceso virtual a ciertas aplicaciones o información debe ser controlado, de lo contrario representa un alto riesgo de seguridad.

Es por ello que se plantea el desarrollo de una aplicación web que, primero: centralice y gestione la información y manipulación de los documentos digitales y digitalizados, garantizando la autenticidad a través de la firma digital (previa verificación de la huella digital capturada por un lector biométrico) que permite: el envío y recepción de documentos digitales y digitalizados, el seguimiento y estatus del documento enviado, la creación de flujos de trabajo y tipos de documentos.

Segundo, el control del acceso físico que realice una determinada persona en FUNVISIS a través del lector de huellas, con la posibilidad de restringir a donde puede acceder y el respectivo registro de quienes, a donde y cuando intentaron o accedieron a una ubicación física.

En las secciones siguientes se presenta la solución implementada basada en la tecnología biométrica basada en la huella dactilar y el uso de firmas digitales, con el soporte de la tecnología de firma digital para la gestión de los documentos.

1.3. Antecedentes

A continuación se presentan varios casos de estudio en los que se han llevado a cabo la implementación de sistemas de control de acceso, así como de uso de firmas digitales.

- **Firma digital en historia clínica**

En Argentina se usa la firma digital para la Historia Médica Digital (HMD) en Internet, que brinda accesibilidad controlada de los datos clínicos. Esta herramienta permite la encriptación de los datos para proteger la información médica, asegurándose la privacidad de los datos de los pacientes y que los mismos sólo sean leídos por las personas autorizadas. [BIOCOM, 2012].

En Venezuela la empresa Servicios Informáticos Remoto (SIR), diseñó un sistema manejador de flujos de documentos diseñado en aplicaciones web dirigido principalmente a instituciones prestadoras de servicio de salud y afines. Donde todos los documentos inherentes al expediente médico (historia médica) y administrativo del paciente son introducidos al sistema, bien por vía de scanners o auto escaneo de formatos. También los documentos electrónicos son firmados digitalmente, cumplen con un proceso de validación de reconocimiento de la firma. [SIR, 2001]

- **Control de accesos biométrico para geriátricos**

En España algunos centros geriátricos cuentan con habitaciones para los residentes, apartamentos y espacios restringidos, tanto para la administración y personal de asistencia como espacios comunes. Solamente las personas autorizadas a entrar en cada uno de estos espacios deben poder entrar en ellos, por lo tanto es necesario realizar un control de acceso. En estos centros, se escogió la tecnología biométrica de reconocimiento de huella digital para identificar las personas e implementar el control de acceso. La comodidad de uso, fiabilidad y seguridad fueron los motivos por los que se eligió la opción biométrica. [Novenca Security Systems, 2012]

- **Firma Digital para BlackBerry PlayBook - Movistar**

En Venezuela a través del dispositivo BlackBerry PlayBook se puede firmar cualquier documento a distancia con todas las garantías, sin necesidad de la instalación de ningún elemento hardware, ni cualquier elemento adicional en el dispositivo. La aplicación instalada en su BlackBerry PlayBook le permite seleccionar el documento y firmarlo fácilmente.

La firma digital es el equivalente electrónico a un documento de identidad. Permite identificarnos, firmar y cifrar electrónicamente documentos y mensajes. Se trata de una herramienta indispensable para las empresas a la hora de tramitar documentación en soporte electrónico. Internalia es una empresa certificada como partner de BlackBerry a nivel internacional [Blackberryvzla, 2012].

- **Firma electrónica con ClickSign 10.5**

ClickSign 10.5 es un programa orientado a la firma electrónica de documentos mediante contraseña (pin) y la versión gratuita únicamente permite procesar archivos en formato PDF. Otro aspecto positivo es que está disponible tanto para Windows como para Linux. [Isigma Asesoría Tecnológica, 2010].

- **Firma electrónica con JSIGNPDF 1.3**

JSIGNPDF 1.3 es herramienta que firmar digitalmente documentos PDF, mediante una contraseña (pin) y proporciona un régimen de software libre. En caso de que optes por no utilizar el instalador, previamente deberás agregar a tu sistema la versión más reciente de Java Runtime Environment. Su gran inconveniente es que únicamente permite validar documentos en formato PDF [Josef Cacek, 2013].

- **Firma electrónica con Acrobat**

Por otro lado Adobe tiene al Acrobat, software que permite crear documentos PDF y aplicar codificaciones, permisos y firmas electrónicas a los archivos. Los documentos PDF pueden visualizarse con el software gratuito Adobe Reader y con esta herramienta se permite además validar firmas electrónicas y comprobar la certificación de documentos. Además, mediante el Adobe Live Cycle Reader

Extensions, los usuarios Adobe Reader podrán firmar electrónicamente los archivos PDF. [Adobe, 2012].

- **Control de Acceso en FUNVISIS**

En FUNVISIS utilizan un tipo de equipo biométrico denominado lector de huella dactilar: Biotrack (bioclock plus, biodoor y bioaccessip) para el control de acceso de áreas restringidas, controlar entradas y salidas de los empleados de FUNVISIS, éste equipo biométrico ofrece funcionalidades básicas (registrar, autenticar, generación de reportes, etc), estos equipos funcionan bajo el sistema operativo Windows y bajo el esquema de licencia propietaria. [BioTrackCorp, 2012]

1.4. Justificación

La seguridad provista por medios informáticos, es vista hoy en día como una necesidad vital para las organizaciones, dado el valor subyacente de sus activos de información que muchas veces superan en gran medida el valor de sus activos físicos. Por otro parte, los documentos digitales y digitalizados son un mecanismo más sencillo, eficiente y rentable para almacenar, administrar, consultar grandes volúmenes de información y garantizar su conservación en óptimas condiciones, obteniendo mayor seguridad para que la información sea recuperada aún en caso de siniestros. Este proceso de digitalización puede implementarse en todo tipo de organizaciones (sector público o privado), que tengan información que les sea conveniente acceder de manera más ágil en un formato digital.

Se debe tomar en cuenta que todo documento digital y digitalizado dentro de una organización debe ser firmado, para garantizar la validez e integridad de dicho documento. Además el uso de firma digital en la autenticación de los documentos

juega un papel importante, lo que ha permitido aumentar los niveles de seguridad de muchos sistemas hoy en día.

Otro punto importante es garantizar la seguridad de un determinado espacio físico dentro de una organización, una forma es mediante sistemas biométricos basados en la huella dactilar. Esta tecnología es más segura y cómoda que los sistemas tradicionales basados en claves o tarjetas de acceso. El acceso a través de huella dactilar a un PC o a una sala restringida no depende de algo que sepa (claves) o que tenga (tarjetas de acceso) el usuario, que puede ser fácilmente robado o clonado, sino que depende de la característica biométrica única del usuario. Disminuyendo el riesgo de robo, pérdida o clonación de los elementos necesarios para acceder a un espacio o sistema protegido biométricamente sea considerablemente menor que en el caso tradicional (claves o tarjetas de acceso), y en caso de alguna eventualidad identificar el causante y lugar.

En virtud de esto, el sistema desarrollado para FUNVISIS, es un sistema que ofrece un grado de seguridad robusto, multiplataforma y provee centralización de información, que integra el control de acceso así como la manipulación y gestión de documentos digitales y digitalizados en un solo sistema. Utilizando para ello la huella dactilar como mecanismo de identificación del individuo. Dicho sistema soporta un conjunto de equipos biométricos para la detección de la huella dactilar, de esta forma se desliga la dependencia del sistema con un determinado equipo biométrico.

1.5. Objetivo General

Para el desarrollo de este Trabajo Especial de Grado se estableció el siguiente objetivo general:

Desarrollar una aplicación web para el control de acceso a espacios físicos, así como la gestión y firma digital de documentos utilizando la integración de la tecnología de huella dactilar y los certificados digitales.

1.6. Objetivos Específicos

Como objetivos específicos se fijaron los siguientes:

- Diseñar e implementar la firma de documentos PDF a partir del certificado digital.
- Implementar la gestión de documento con el certificado digital, dentro de FUNVISIS.
- Controlar y gestionar el control de acceso a espacios físicos y lógicos mediante huella dactilar como característica biométrica.
- Diseñar e implementar el módulo de gestión de control de acceso a espacios físicos basado en la captura, registro e identificación de la huella dactilar como característica biométrica.
- Diseñar e implementar el módulo de consultas y reportes, para el control de acceso a espacios físicos.
- Defina una política de seguridad en el sistema, que garantice personas no autorizadas intervengan en el sistema web con fines malignos.
- Implementar unas medidas de seguridad para la manipulación de datos de cliente-servidor.

1.7. Descripción de la solución

La solución se basa en el desarrollo de una aplicación web, para la gestión y autenticación de los documentos digitales y digitalizados, así como el control de acceso físico y lógico.

Para ello se diseñó e implementó los siguientes módulos en el sistema web:

- **Módulo de monitoreo y control de acceso**

Mediante un sistema biométrico basado en huella dactilar, se controla el acceso a determinados espacios físicos y solamente personal autorizado podrá ingresar, esto se realiza mediante una previa verificación de la huella digital ingresada en la base de datos. Además con la huella dactilar se tiene acceso lógico a la aplicación.

Para el desarrollo de éste módulo se utilizó un SDK basado en huella dactilar, el SDK utilizado es el Verifinger de Neurotechnology el cual es software propietario, debido a que en la investigación previa realizada se determino que los sistemas desarrollados bajo la filosofía de SL para el reconocimiento biométrico basados en huella dactilar no son lo suficientemente estables y confiables. Como es el caso de libfprint [freedesktop, 2013], que a pesar de estar desarrollado de bajo la filosofía de SL tiene aún muchas fallas, lo cual no es una opción cuando se trata de seleccionar un software para la seguridad de un sistema. Cabe destacar que el SDK utilizado en el desarrollo de este Trabajo Especial de Grado se adapta al estándar BIOAPI el cual permite que el software de múltiples proveedores de sistemas biométricos pueda ser integrado en conjunto entre uno o más sistemas que utilizan datos biométricos dentro de un sistema [BIOAPI, 2013].

El proceso de control de acceso se realiza de la siguiente manera:

1. El usuario debe autenticarse por medio de su huella dactilar.

2. Se deberá tener el lector conectado al equipo donde se realice la autenticación.

3. La página web carga de forma automática el applet de Java encargado de la toma de la entrada/salida de la ubicación física, el cual muestra la fecha y la hora actual.

El applet de control de acceso tiene las siguientes características:

- ✓ Acceso lógico
- ✓ Registro de entrada
- ✓ Registro de salida
- ✓ Acceso denegado

- **Módulo de gestión de documentos digitales y digitalizados**

Se garantiza la validez de un documento (digital y digitalizado) mediante la firma digital, así como la autenticación, confidencialidad, integridad y disponibilidad de dicho documento.

Para la integración entre la huella digital y el certificado digital para lograr la firma digital de un documento. Se plantearon tres posibles soluciones que se describen a continuación:

1. Añadir el cifrado de la huella digital (hash) de quien va a firmar el documento en algunos de los campos de extensión que provee el formato X.509 V3 referente a los certificados digitales [William Stallings, 2004].
2. Cifrar el documento a ser enviado con el cifrado de la huella digital (hash) de quien va a firmar el documento.
3. Utilizar la huella digital de la persona que va a firmar como un mecanismo de identificación |antes de proceder a firmar digitalmente el documento.

Las soluciones propuesta en los puntos 1 y 2 expuestos anteriormente aunque son posibles soluciones para la integración entre la huella digital y el certificado digital, presentan ciertas desventajas para su implementación en el sistema a desarrollar.

En el caso del punto 1, la modificación de los campos del certificado digital no es una solución viable dado que los certificados digitales que serán utilizados en el sistema son proporcionados por una Autoridad Certificadora la cual tiene delimitado los campos a solicitar para la emisión del certificado digital y entre esos campos no se encuentra actualmente la adición de una huella digital al certificado.

La solución presentada en el punto 2, hace uso de la huella tanto para cifrar el documento como para descifrarlo (cifrado simétrico) lo que ocasionaría la necesidad de transmitir los datos de la huella que se usó para cifrar el documento a la (s) persona (s) que necesitan descifrar el documento, representando un riesgo de seguridad el mecanismo que se adopte para la transmisión de los datos de la huella a dicha (s) persona (s). Además de tornarse un proceso tedioso para los documentos en el que estén involucrados distintos destinatarios o en el caso en el que el documento deba seguir un flujo de personas, donde cada una de las personas tendría que tener "acceso" a los datos de la huella del responsable de cifrar el documento para poder acceder a su contenido.

Por las razones antes expuestas, la solución planteada en el punto 3 como el más viable para la integración entre la huella digital y el certificado digital. Para ello, la firma digital es obtenida a través del certificado digital del usuario, al momento de que un usuario quiera firmar digitalmente un documento le será solicitado la clave de su certificado así como su huella dactilar, de ser ambos datos correctos, es decir, que correspondan los datos al usuario que va a firmar el documento, entonces se procederá a firmar digitalmente el documento solicitado. El documento previamente

firmado (digitalmente) es enviado de un departamento a otro siguiendo un flujo asignado anteriormente, de esta manera se puede consultar el estado de dicho documento si requiere de aprobación o solamente comprobar que la información llegó a destino.

- **Módulo de consultas y reportes**

En este módulo, es posible la consulta de información de una determinada persona con respecto a los documentos (digitales y digitalizados) que ha firmado. También es posible visualizar un historial o bitácora (fecha/hora, lugar, usuario) sobre el ingreso del personal a un determinado espacio físico restringido dentro de FUNVISIS.

La aplicación consta de los siguientes roles de usuario: usuario básico, administrador general, administrador de documentos y administrador de usuarios.

a) Usuario Básico

Los usuarios básicos con esta categoría pueden a través de la aplicación web realizar las siguientes actividades:

- Autenticación de usuarios: por medio de su huella digital, un nombre de usuario y una clave el usuario podrá ingresar al sistema.
- Perfil: muestra los datos personales y laborales del usuario.
- Actualizar datos:
    - Nombre, en caso de que en el momento del registro haya existido un error.
    - Fecha de Nacimiento
    - Número de teléfono

- Extensión
- Sexo
- Cambiar clave: el usuario podrá cambiar la clave siempre y cuando cumpla con las siguientes características:
  - Mínimo 8 caracteres
  - Por lo menos un número (0 - 9)
  - Por lo menos una letra mayúscula (A-Z)
  - Por lo menos una letra minúscula (a-z)

- Salir o Cerrar Sesión: esta sección del sistema permite al usuario finalizar la sesión de trabajo.
- Acceso a un módulo para la gestión y control de documentos digitales y digitalizados, donde podrá realizar envíos de un documento con firma digital, debido a que existen varios remitentes y varía dependiendo de la categoría del documento, se puede escoger un flujo de remitente ya preestablecido o generar uno propio, adaptándose a las necesidades de los usuarios, también visualiza los documentos recibidos, enviados y digitalizados.

Cada documento enviado tendrá un seguimiento, como se ilustra en la siguiente figura:

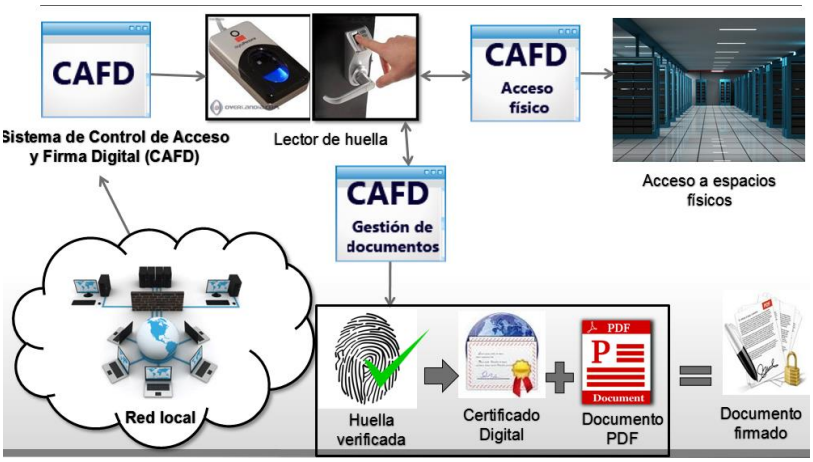


Figura 1: Diagrama de descripción de la solución (Manipulación Documentos)

b) Administrador General

El administrador general con esta categoría puede realizar las siguientes actividades:

- Autenticación de usuarios: por medio de la huella digital, usuario y clave el usuario puede ingresar al sistema
- Registro de usuarios: realizar el registro de usuarios en la base datos, donde se darán los datos laborales del empleado:
    - Cédula del empleado
    - Nombre y apellido
    - Cargo y departamento en el que labora
    - Correo electrónico
    - Teléfono de contacto
- Actualización de datos: en esta sección se pueden actualizar y modificar los datos ingresados en el módulo registro de usuarios, a excepción de la cédula, ya que esta es única.
- Eliminación de usuarios: el administrador puede eliminar a cualquier usuario, las eliminaciones se registrarán automáticamente en la base de datos.
- Crear categorías de documentos
- Crear flujo de remitentes
- Acceso a un módulo para la gestión y control de documentos digitales y digitalizados, donde podrá realizar el proceso de digitalización de documentos así como la distribución del mismo.
- Salir o Cerrar Sesión: esta sección del sistema permite al administrador finalizar la sesión de trabajo.

c) Administrador de documentos

- Crear categorías de documentos
- Crear flujo de remitentes
- Acceso a un módulo para la gestión y control de documentos digitales y digitalizados, donde pude realizar el proceso de digitalización de documentos así como la distribución del mismo.
- Salir o Cerrar Sesión: esta sección del sistema permite al administrador finalizar la sesión de trabajo.

d) Administrador de usuarios

- Registro de usuarios: realizar el registro de usuarios en la base datos, donde se darán los datos laborales del empleado.
    - Cedula del empleado
    - Nombre y apellido
    - Cargo y departamento en el que labora
    - Correo electrónico
    - Teléfono de contacto
- Actualización de datos: en esta sección se podrán actualizar y modificar los datos ingresados en el módulo registro de usuarios, a excepción de la cédula, ya que esta es única.
- Eliminación de usuarios: el administrador de usuario también podrá dar eliminar a cualquier usuario, las eliminaciones se registrarán automáticamente en la base de datos.

1.8. Requerimientos funcionales y no funcionales

A continuación se presentan los requerimientos funcionales y no funcionales del sistema.

a.- Requerimientos funcionales

- Control de acceso físico a espacios físicos y lógicos
- Generar un historial sobre el control de acceso físico para su uso en auditorias
- Multiplataforma
- Gestión y clasificación de documentos
- Uso de la firma electrónica mediante la integración de las tecnologías de certificados digitales y huella dactilar
- Búsqueda, recuperación y presentación de los documentos firmados y a ser firmados.
- Generar historial sobre los documentos firmados.
- Generación de informes de distinta índole
- Visualizar los flujos de los documentos y sus estatus.
- Visualizar flujos creados.

b.- Requerimientos no funcionales

- Usable
- Correctitud de la aplicación
- Escalabilidad
- Disponibilidad del sistema
- Mantenibilidad

1.9. Recursos computacionales

A continuación se describen los recursos de hardware y software a utilizar para el desarrollo del sistema web.

1.9.1. Hardware

Es necesario contar con una plataforma de red con acceso a Internet, donde las máquinas del cliente no requieren de grandes especificaciones a nivel de hardware. Las especificaciones serán necesarias en el servidor web donde se alojará la aplicación.

Básicamente, las principales características con las que debe contar un servidor web son las siguientes:

- Procesador: de un núcleo o doble núcleo.
- Memoria RAM: al menos 2GB o más.
- Disco duro: al menos 250GB.
- Equipo biométrico por ejemplo el modelo U.ARE.U 4000b, dada su excelente relación calidad-costo, la obtención de una excelente calidad en la imagen y una alta confiabilidad y lo más importante, el acceso no sólo a su código fuente sino también la posibilidad de usar SDK genéricos.

1.9.2. Software

Para el desarrollo del sistema se utilizó las siguientes herramientas: Linux, Java, PostgreSQL, Apache Web Server y Tomcat, Eclipse, Verifinger de Neurotechnology, iTEXT, BouncyCastle.

- Linux: es un sistema operativo que está basado en UNIX, entonces, es un sistema operativo muy robusto y estable. Para el desarrollo del sistema se utilizará éste sistema operativo, porque sirve perfectamente como servidor Web e incluye algunas de las herramientas necesarias para el desarrollo de la aplicación.
- JAVA: es un lenguaje de programación orientado a objetos, independiente de la plataforma, fácil de usar. Para el desarrollo web, Java cuenta con 2 tecnologías principales: Servlets y JSP.
- SERVLET: los servlets permiten gestionar elementos HTTP, atendiendo solicitudes y procesando respuestas.
- JSP: la principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento HTML en el archivo con extensión JSP. Por tanto JSP tiene ventajas respecto a sus competidores, por lo que se usará JSP para el desarrollo de la aplicación.
- POSTGRESQL: es una de las herramientas más usadas para base de datos, es confiable, rápido, fácil de usar. Es una base de datos libre y para aplicaciones web es la más conveniente. Por lo tanto, se usará PostgreSQL como base de datos para el desarrollo de la aplicación.
- APACHE: Apache es un robusto servidor de páginas web, que se usa generalmente bajo entorno Linux. La razón por la que se ha escogido esta herramienta es debido a que es la herramienta oficial como servidor web en entorno Linux. Además de ser multiplataforma y software libre, que es precisamente la razón por la que se ha optado por este servidor.
- TOMCAT: TOMCAT es un servidor de aplicaciones popular y soporta aplicaciones con Java, por lo que se ha escogido para el desarrollo del sistema.

- ECLIPSE: es un entorno de desarrollo integrado de código abierto multiplataforma que emplea distintos módulos, plugins, para proporcionar distintas funcionalidades al usuario. Al tratarse de software libre y dado sus poderosas ventajas y características, éste IDE se ha escogido para el desarrollo del sistema.
- VERIFINGER: en relación al SDK a utilizar para la programación del lector de huellas U.ARE.U 4000b, se utilizará un SDK genérico, específicamente Verifinger de Neurotechnology, el cual es un paquete de desarrollo de software (SDK) innovador que permite integrar la biometría en un amplio rango de aplicaciones. Tiene soporte y compatibilidad con sistemas operativos Windows, Mac, Android y Linux. Para el uso del SDK en un entorno de producción se requiere un número de licencias de Fingreprint Client como número de dispositivos quieran acceder al sistema biométrico y una única licencia Fingerprint Matcher que será la que se instalará en el servidor donde estará instalado el sistema. Una vez comprada la licencia no tiene fecha de caducidad. La tabla a continuación (Tabla 1) muestra el detalle de los costos por número de licencias expresados en euros (€).

Tabla 1 Costos de licencia

Cantidad	FingerprintClient	FingerprintMatcher
1-9	€ 35.00	€ 25.00
10-19	€ 26.00	€ 18.00
20-49	€ 23.00	€ 16.00
50-99	€ 20.00	€ 14.00
100-199	€ 18.00	€ 12.50

200-499	€ 16.00	€ 11.00
500-999	€ 14.00	€ 10.00

- iTEXT: iText es una biblioteca Open Source para crear y manipular archivos PDF en Java.
- BOUNCYCASTLE: la librería BouncyCastle es un desarrollo en lenguaje Java, es una la librería de libre distribución con una gran cantidad de algoritmos de criptografía. Hace uso del lenguaje Java sin ningún tipo de restricción, por lo que se convierte en una plataforma ideal para los procesos de encriptar la información.

1.10. Metodología de desarrollo de software

Como metodología para el desarrollo se utilizó una adaptación de XP, que es una nueva tendencia, ya que trabaja de manera interactiva.

La metodología XP, es un estándar de trabajo interactivo, que cuenta con 4 fases, es ideal para desarrollos web o proyectos a corto plazo o corto-mediano plazo, por tender a variaciones y a modificaciones. Además cuenta con fases cortas, se pueden realizar cambios sin perjudicar el fin de la investigación, y no se basa en estándares rígidos ya que trata de ser lo más flexible para buscar la comodidad de ambas partes involucradas en el desarrollo. Es perfecta para trabajar en grupos pequeños de trabajo, con pocos equipos tecnológicos, roles y recursos.

1.11. Alcance

El alcance de este Trabajo Especial de Grado está delimitado de acuerdo a los siguientes aspectos:

- La generación de documentos que maneje la institución
- La generación de certificados digitales
- El uso de certificados digitales de estándar diferente al estándar pkcs#12
- La modificación del estándar X.509 para certificados digitales

1.12. Planificación de Actividades

A continuación se presenta las diferentes actividades que se llevaron a cabo para lograr los objetivos antes mencionados.

Tabla 2 Planificación de actividades

	Actividad 1	**Actividad 2**	**Actividad 3**	**Actividad 4**
Módulo de control de acceso físico y lógico	Estudio de equipos biométricos lectores de huella	Estudio de SDK de biometría basado en huella dactilar	Implementación de control de acceso a través de la huella digital	.
Módulo de gestión de documentos	Estudio de librerías para la manipulación de documentos en formato PDF	Estudio del estándar PKCS12(estándar de criptografía de clave pública)	Obtener datos de la firma digital	Implementación de la firma digital en documentos con formato PDF
Módulo de consultas y reportes	Implementación del módulo con información del sistema			

CAPITULO 2. MARCO TEORICO

En este capítulo se presentan las bases conceptuales sobre biometría (con énfasis en la huella dactilar) y firma digital, consideradas para comprender el contexto en el cual se desarrolló el presente Trabajo Especial de Grado.

2.1. Seguridad de los datos

La seguridad está finamente ligada a la certeza. Para entender esta definición, hay que aclarar que no existe seguridad absoluta, más bien, lo que se intenta es minimizar el impacto y/o riesgo. Las técnicas para llegar a una correcta organización están basadas en cuatro pilares fundamentales que hacen que la información se encuentre protegida.

2.1.1. Pilares de la seguridad

Estos pilares se ocupan principalmente de proteger cuatro aspectos de la información:

- **Confidencialidad:** La información puede ser accedida únicamente por las personas que tienen autorización para hacerlo.
- **Integridad:** Cuando nos referimos a integridad, queremos decir que estamos totalmente seguros de que la información no ha sido borrada, copiada o alterada, no sólo en su trayecto, sino también desde su origen.
- **Disponibilidad:** Estar en línea y disponible cuando se requiere por alguien autorizado. También hace referencia al método de precaución contra posibles daños, tanto en la información como en el acceso a la misma: ataques, accidentes o, simplemente, descuidos pueden ser los factores que obligan a diseñar métodos para posibles bloqueos.

- **Autenticidad:** Nos informa que el archivo en cuestión es el real, es decir, la información de la fuente está actualizada y no sea falsa.

2.2. Biometría

La biometría deriva de las palabras griegas "bios" (vida) y "metros" (medida), la definición de biometría es: "*Estudio mensurativo o estadístico de los fenómenos o procesos biológicos*", [RAE, 2001]; por tanto, el concepto de biometría consiste en técnicas que permiten la medición e identificación de alguna característica propia de un organismo vivo o bien por medio de algún patrón de comportamiento.

La biometría consiste en medir una de las características del cuerpo humano con el fin de identificar un individuo. Eligiendo para ello una característica de fuerte variabilidad de un individuo a otro [Royer, 2007].

El análisis documental ha permitido la identificación de las principales técnicas biométricas existentes. Si bien algunas de ellas aún se encuentran en estados de madurez insuficientes para poder encontrarse en el mercado de una forma eficiente y razonable en cuanto a costes, muchas otras se encuentran suficientemente maduras y ofrecen actualmente soluciones de autenticación en muy diversos ámbitos.

Las técnicas biométricas existentes se clasifican en función del rasgo humano analizado, el cual puede ser fisiológico donde encontramos el reconocimiento de huella dactilar, de iris, de retina y facial, entre otros o de comportamiento donde se encuentran el reconocimiento de voz, de firma o de la manera de andar, entre otros.

Algunos de los principales métodos utilizados en la biometría son: Reconocimiento facial, la huella, la geometría de la mano, el iris, la voz, las venas, las

orejas, el pulso cardiaco, la radiografía dental, el ADN, la forma de escribir a mano y la forma de digitar o teclear en el computador.

A continuación se expone una figura con las principales características biométricas:

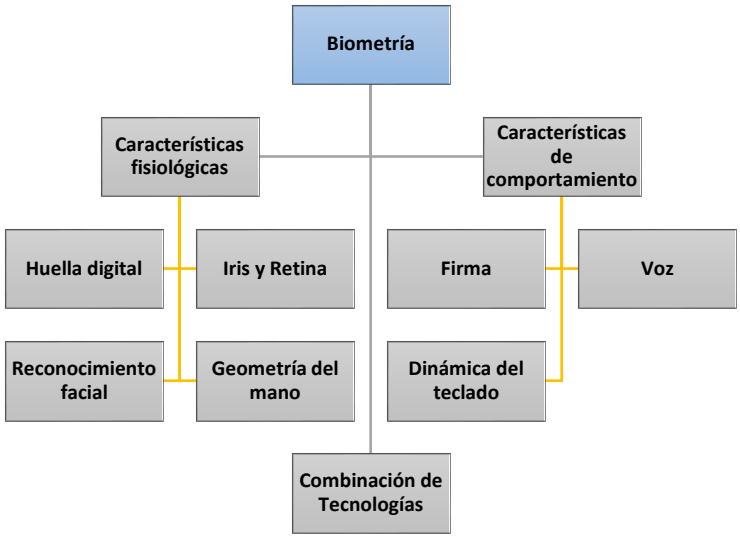


Figura 5: Esquema de características biométrica

Existen otras características biométricas, para ello normalmente se evalúan las siguientes referencias:

- Universabilidad: Toda persona que emplee el sistema debe tener la característica biométrica.
- Permanencia: debe mantenerse la característica biométrica igual en el tiempo sin sufrir cambios mayores.
- Medible: debe ser capaz la característica biométrica de ser medible cuantitativamente.
- Distintiva: debe ser única la característica biométrica para cada persona.

- Rendimiento: hace referencia al rendimiento de la aplicación tanto para extraer la plantilla de la característica como en la velocidad para verificarla e identificarla
- Aceptabilidad: representa el nivel de aceptación del usuario para usar un sistema que implemente la característica biométrica.
- Robustez: indica el grado de seguridad del sistema frente a ataques fraudulentos al mismo.

La tabla 3 muestra un resumen de las principales características en distintos tipos de biometría:

Tabla 3 Resumen de las características por biometría

	Iris	**Retina**	***Huella***	**Mano**	**Escritura**	**Voz**	**Rostro**
Fiabilidad	Muy alta	Muy alta	***Alta***	Alta	Media	Alta	Alta
Facilidad	Media	Baja	***Alta***	Alta	Alta	Alta	Alta
Aceptación	Media	Media	***Media***	Alta	Muy alta	Alta	Muy alta
Estabilidad	Alta	Alta	***Alta***	Media	Baja	Media	Media
Costo equipo	Alto	Alto	***Bajo***	Alto	Medio	Bajo	Alto

2.2.1. Identificación por huellas dactilares

La identificación de huellas digitales es una de las técnicas más antiguas y ciertamente la más difundida de los medios de identificación usados actualmente. La huella digital de un individuo es definida por una compleja combinación de patrones de líneas, arcos, lazos, y círculos. Un lector de huellas lee la huella mediante el uso de una luz intermitente a través de una lámina de vidrio, sobre el cual el usuario ha plasmado uno o varios de sus dedos, cuya reflexión se digitaliza, donde a través de un software de computadora especializado permitirá codificar los distintos patrones digitalizados encontrados en la imagen y las plantillas resultantes pueden ser encriptados en forma opcional y almacenadas en una base de datos central o sobre una tarjeta individual de cada usuario.

Las huellas digitales pueden ser obtenidas de diferentes maneras. El procedimiento común envuelve la captura de la imagen de la impresión hecha en algún papel lo cual no es muy práctico para la implementación de un sistema de identificación de civiles, para lo que se podría utilizar mejor un sistema de registro utilizando un escáner óptico. Un escáner registra y analiza una imagen del dedo que se coloca sobre una lámina de vidrio.

Ventajas de la huella digital

- La huella digital de cada persona es única y permanece inalterable de por vida.
- Grandes bases de datos de huellas digitales existen actualmente.
- Se ha invertido gran cantidad de dinero en el desarrollo y perfeccionamiento de técnicas de procesamiento de huellas digitales.
- Es una técnica no intrusiva, es decir, el colocar el dedo en un lector no hace sentir al usuario que su cuerpo está siendo invadido por algún ente.

- La tasa de error de los algoritmos que identifican o autentican huellas digitales es muy baja.

Desventajas

- Ejecutar una búsqueda uno a uno en una base de datos puede ser lento, a menos que se tengan varios sistemas identificadores trabajando a la vez repartiéndose el trabajo. Realmente esta desventaja no es solamente para la tecnología de identificación de huellas digitales, ya que todas las técnicas biométricas sufren del mismo requerimiento.
- La imagen escaneada puede ser borrosa o inexacta debido a heridas, sucio en el dedo o en la lámina del escáner.

Clasificación

Los patrones de huellas digitales están divididos en 4 tipos principales, todos ellos matemáticamente detectables. En la tabla 4 se muestra una breve descripción de cada uno de los tipos fundamentales de huella digital.

Tabla 4 Tipos fundamentales de huella digital [GBA, 2013

	Arco Se clasifica con la letra A para los pulgares y con el Nº 1 para los demás dedos. Es el primer tipo del sistema, está formado por líneas más o menos paralelas que atraviesan el dactilograma de un extremo a otro y carece de deltas, líneas directrices y núcleo.
	Presilla Interna Se clasifican con la letra I para los pulgares y -con el Nº 2 para los demás dedos. Está formado por un delta que se encuentra ubicado a la derecha del observador y sus líneas directrices que envuelven a las nucleares se dirigen hacia la izquierda.
	Presilla Externa Se clasifica con la letra E para los pulgares y con el Nº 3 para los demás dedos. Está formado por un delta que se encuentra ubicado a la izquierda del observador y sus líneas directrices que envuelven a las nucleares se dirigen hacia la derecha.
	Verticilo Se clasifica con la letra V para los pulgares y con el Nº 4 para los demás dedos. Es el cuarto y último tipo del sistema, está formado por dos deltas, uno a la derecha y el otro a la izquierda y sus líneas directrices circunscriben al núcleo en diferentes formas.

Esta clasificación es útil al momento de la verificación en la identificación electrónica, ya que el sistema sólo busca en la base de datos del grupo correspondiente. Los puntos característicos se encuentran en las crestas papilares. El conocimiento de estos puntos es de suma importancia para comprobar la igualdad entre dos dactilogramas mediante el cotejo correspondiente.

Muchos dactilóscopos hablan de diez (10), los puntos necesarios para probar la identidad entre dos dactilogramas. En la siguiente figura (Ver figura 7) aparecen 8 puntos característicos que hay en un dedo, éstos se repiten indistintamente para formar entre 60 y 120 (por ejemplo 10 horquillas 12 empalmes 15 islotes, etc.). A estos puntos también se llaman minutas, o minucias, término utilizado en la medicina forense que significa "punto característico".

Figura 6: Esquema de la huella dactilar [GBA, 2013]

La tabla a continuación (Ver Tabla 5) muestra en detalle los 8 puntos característicos presentes en una huella digital.

Tabla 5 Puntos característicos de una huella digital

[GBA, 2013]

Tipo	Definición	Ejemplo
Bifurcación	Línea que en su trayecto se abre o bifurca, formando un ángulo más o menos agudo.	
Cortada	Línea que se interrumpe o corta una o varias veces durante su recorrido.	
Empalme	Entre dos líneas paralelas sale una a fin de unirse a otra en diagonal.	
Encierro	Es una línea dada que se le une otra formando un ojal.	
Extremo de línea	Es la línea que queda interrumpida en uno de sus extremos, o en ambos sin solución de continuidad.	
Horquilla	Es aquella que en algún lugar de su recorrido se une a otra sin formar ángulo.	
Islote	Línea que es un poco más grande que el punto formado por 2 o más puntos.	
Punto	Es la mínima expresión de una cresta papilar.	

Procedimiento

Con este conjunto de puntos, el software biométrico de huella digital genera un modelo en dos dimensiones, que se almacena en una base de datos, con la debida referencia a la persona que ha sido objeto del estudio. Para ello, la ubicación de cada punto característico o minucia se representa mediante una combinación de números (x.y) dentro de un plano cartesiano, los cuales sirven como base para crear un conjunto de vectores que se obtienen al unir las minucias entre sí mediante rectas cuyo ángulo y dirección generan el trazo de un prisma de configuración única e irrepetible.

Para llevar a cabo el proceso inverso o verificación dactilar, se utilizan estos mismos vectores, no las imágenes capturadas por el sensor.

Tabla 6 Ejemplo del proceso de captura de una huella [criminalistaenred, 2013]

Fase 1	**Fase 2**	**Fase 3**	**Fase 4**
			
El dedo es leído por un captor de huellas	El dedo es codificado por el sistema	Una plantilla es generada y la imagen es comprimida en formato WSQ (opcional)	El sistema guarda y reconoce un conjunto de números que solo podrán ser reconocidos como una plantilla

En la tabla 6 se muestra un ejemplo de las fases llevadas a cabo en el proceso de captura de huella que se usan en algunos sistemas.

Dispositivo para identificación

El sistema de identificación automatizada de huellas dactilares, tiene un índice de seguridad del 99.9% ya que verifica la identidad de una persona, basada en las características de sus huellas digitales.

Para tratar los datos de la huella se utiliza un algoritmo que permite asociar la huella que se desea identificar, con otras de similares características, almacenadas en la base de datos.

2.2.3. Fases de un sistema biométrico

Los pasos básicos en un sistema biométrico genérico son:

- **Registro**

Captura la imagen de la impresión dactilar mediante un lector en vivo con posibilidad o no de huella rodada. Existe la opción de captura sobre una impresión entintada usando un escáner plano. En la siguiente figura (Ver figura 8) se muestran algunos lectores de huella digital.

Figura 7: Aparatos de recogida de huella dactilar

- **Extracción de plantillas de una huella digital (procesado):**

Consiste en la extracción de puntos característicos o minucias (típico de 30 a 100 puntos). El procesado se compone de las siguientes etapas (Ver Figura 9):

- PASO 1: Adquisición de la huella dactilar mediante un lector biométrico.
- PASO 2: Proceso de mejora de imagen para obtener mejor resultados
- PASO 3: Binarización de la imagen (convertir de escala de grises a blanco y negro)
- PASO 4: Esquelitización o adelgazamiento, para reducir el grosor de las crestas en la imagen binaria a un solo pixel.
- PASO 5: Extracción de minucias y datos de posicionamiento analizando cada pixel de la imagen y analizando sus vecinos
- PASO 6: Guardado de minucias en la base de datos para posterior comparación de los puntos característicos o minucias de la imagen

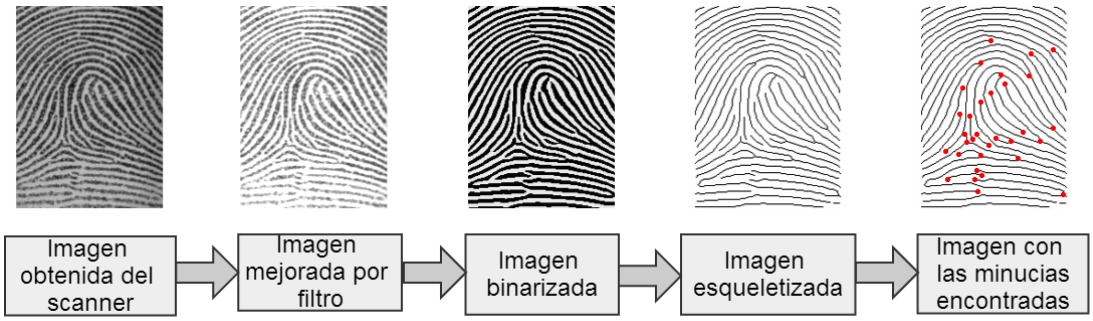


Figura 8: Detalle del procesado de una huella

La figura 10 muestra los detalles de las minucias que se pueden encontrar en una huella digital.

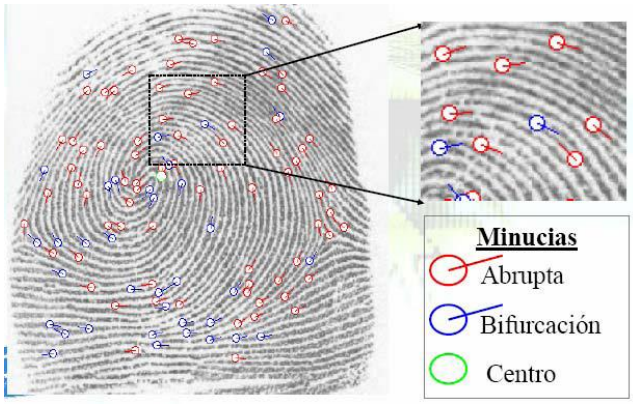


Figura 9: Minucias

- **Almacenamiento**

Baja necesidad de espacio de almacenamiento (típico < 500 Bytes). Típicamente se almacenan de 2 a 4 impresiones, aunque hay proyectos en los que se almacenan las 10 (visa Schengen), en el proceso de almacenamiento solo se toman en cuenta las minucias detectadas (Ver Figura 11)

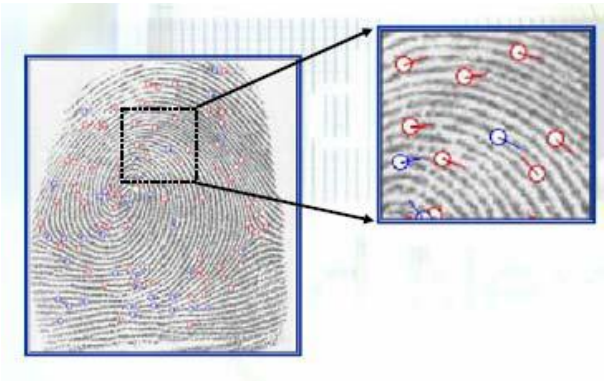


Figura 10: Almacenamiento de una huella

- **Verificación**

Es el procedimiento matemático por el cual se determinan similitudes y diferencias entre los puntos característicos de dos impresiones dactilares (Ver figura 12). Se tiene en cuenta giros, deformaciones elásticas, cicatrices, etc. Dependiendo de un umbral que se fije, el cotejo da un resultado positivo o negativo.

Problemas a resolver en la verificación:

- ✓ Aparición de Minucias Falsas.
- ✓ Desaparición de Minucias Verdaderas.
- ✓ Giro de la Impresión Dactilar.
- ✓ Diferente Área de Superposición.
- ✓ Presión Diferente.
- ✓ Entre otras...

Figura 11: Verificación de una huella

2.2.4. Funcionamiento, rendimiento y suplantación

En un sistema de huella dactilar típico, la persona se registra en el sistema. Idealmente cuando se verifica con éxito, implica que su huella concuerda por encima de un umbral establecido; (en función de la modalidad) cuando entra, casi todas sus características concuerdan.

Entonces cuando alguna otra persona intenta identificarse, esto no sucede y el sistema no le permite el acceso. Las tecnologías actuales tienen tasas de error que varían ampliamente (desde valores bajos como el 60%, hasta altos como el 99,9%), veamos a continuación dos aspectos interesantes [Trigero, M., 2011]:

- **Rendimiento**

El rendimiento de la huella dactilar como medida biométrica se define generalmente en términos de tasa de falso positivo (False Acceptance Rate o FAR), la tasa de falso negativo (False Non Match Rate o FNMR, también False Rejection Rate o FRR), y el fallo de tasa de alistamiento (Failure-to-enroll Rate, FTR o FER).La figura a continuación (Ver figura 13) muestra en detalle el rendimiento de un sistema biométrico.

Figura 12: Rendimiento biométrico [criminalistaenred, 2012]

En los sistemas biométricos basados en huella dactilar el FAR y el FRR pueden transformarse en los demás cambiando cierto parámetro. Una de las medidas más comunes de los sistemas biométricos reales es la tasa en la que el ajuste en el cual acepta y rechaza los errores es igual: la tasa de error igual (Equal Error Rate o EER), también conocida como la tasa de error de cruce (Cross-over Error Rate o CER).

Cuanto más bajo es el EER o el CER, se considera que el sistema es más exacto. Las tasas de error anunciadas implican a veces elementos idiosincrásicos o subjetivos. Por ejemplo, un fabricante de sistemas biométricos basados en huella dactilar fijó el umbral de aceptación alto, para reducir al mínimo las falsas aceptaciones; en la práctica, se permitían tres intentos, por lo que un falso rechazo se contaba sólo si los tres intentos resultaban fallidos (por ejemplo escritura, habla, etc.), las opiniones pueden variar sobre qué constituye un falso rechazo [criminalistaenred, 2012].

A pesar de estas dudas, los sistemas biométricos basados en huella dactilar tienen un potencial para identificar a individuos con un grado de certeza muy alto.

2.3. Firma digital

Es aquella firma electrónica, que consiste en una cadena de caracteres asociada a un mensaje digital que garantizan la autenticidad del origen e integridad del mismo. Para generarla se utilizan métodos criptográficos, esto significa que si un documento firmado digitalmente es verificado correctamente se presume salvo prueba en contrario que proviene del suscriptor del certificado asociado y que no fue modificado.

Es un mensaje firmado con la clave privada del remitente puede ser verificado por cualquier persona que tenga acceso a la clave pública del remitente, lo que demuestra que el remitente tenía acceso a la clave privada (y por lo tanto, es probable que sea la persona asociada con la clave pública utilizada) y la parte del mensaje que no se ha manipulado. Sobre la cuestión de la autenticidad.

2.3.1. Características de la firma digital

- **Únicas:** Las firmas deben poder ser generadas solamente por el firmante y por lo tanto infalsificable. Por tanto la firma debe depender del firmante.
- **Infalsificables:** Para falsificar una firma digital el atacante tiene que resolver problemas matemáticos de una complejidad muy elevada, es decir, las firmas han de ser computacionalmente seguras. Por tanto la firma debe depender del mensaje en sí.
- **Verificables:** Las firmas deben ser fácilmente verificables por los receptores de las mismas y, si ello es necesario, también por los jueces o autoridades competentes.

- **Innegables:** El firmante no debe ser capaz de negar su propia firma.
- **Viables:** Las firmas han de ser fáciles de generar por parte del firmante.

2.3.2. Certificado Digital

Es un documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y una clave pública, es decir, es una clave pública firmada por una autoridad en la que confiamos (esto nos permite confiar en que la clave pública que contiene se puede usar para verificar firmas digitales). La figura 14 muestra algunos campos que contiene un certificado digital.

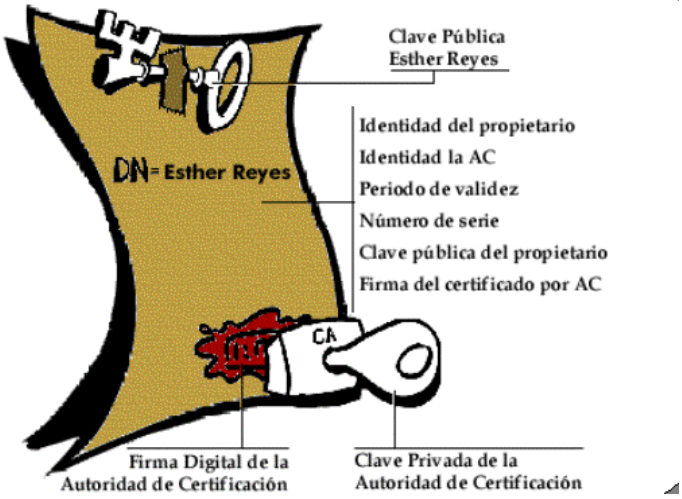


Figura 13: Certificado Digital

En el proceso de entrega de un certificado digital se encuentran involucrados varios entes. Entre los que destacan la Autoridad Certificadora y el solicitante o poseedor del certificado digital. La figura 15 es un ejemplo de los entes que pueden estar involucrados en el proceso de solicitud y verificación de un certificado digital.

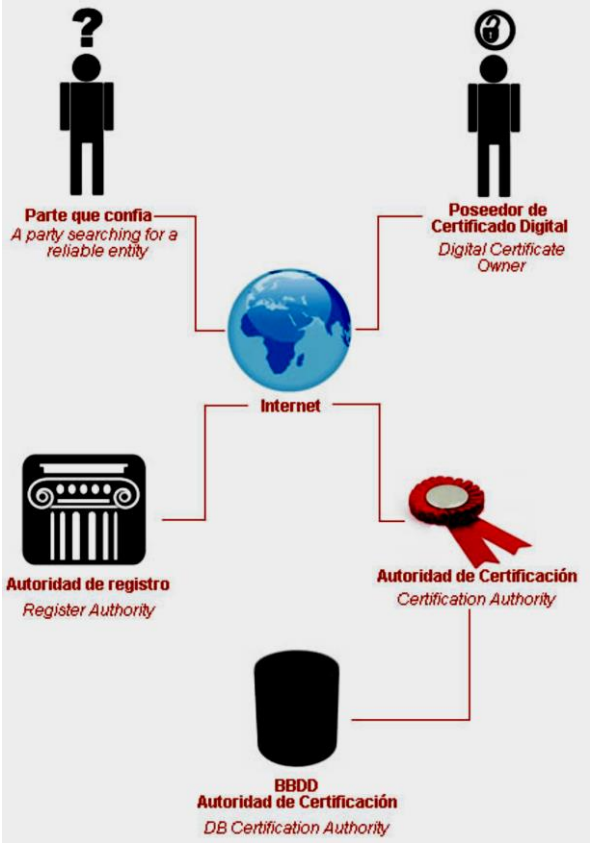


Figura 14: Certificado Digital

2.3.2.1. Formato

Si bien existen variados formatos para certificados digitales, los comúnmente empleados se rigen por el estándar UIT-T X.509 [William Stallings, 2004].

El certificado contiene usualmente el nombre de la entidad certificada, número de serie, fecha de expiración, una copia de la clave pública del titular del certificado (utilizada para la verificación de su firma digital) y la firma digital de la autoridad emisora del certificado de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación.

En la actualidad casi todas las aplicaciones de comercio electrónico y transacciones seguras requieren un certificado digital, se ha propagado tanto su uso

que se tiene ya un formato estándar de certificado digital, este es conocido como X509 v3.

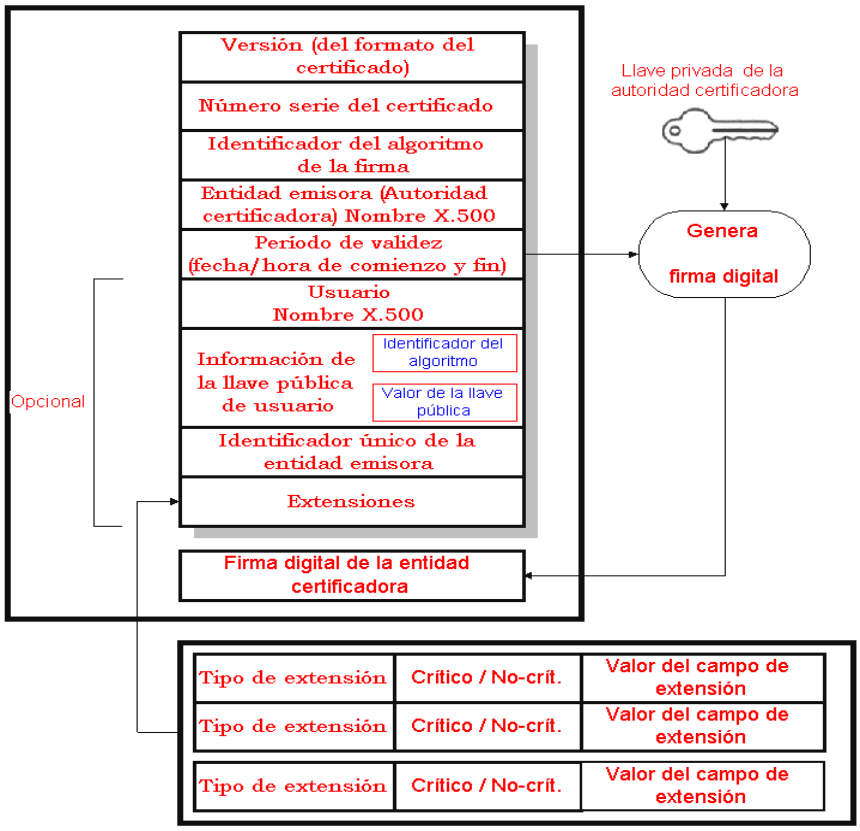


Figura 15: Formato del Certificado Digital

Un certificado digital, que vaya de acuerdo al standard X509v3, contiene la siguiente información:

- Identificación del titular del certificado: Nombre, dirección, etc.
- Copia de la llave pública del titular del certificado.
- Copia del certificado.
- Fecha de validez del certificado (fecha de expiración).
- Número de serie.
- Nombre de la Autoridad Certificadora (identificación).

- Firma digital de la Autoridad Certificadora.

El documento contiene el nombre de un sujeto y su llave pública.

2.3.2.2. Ventajas

El certificado digital permite autentificar y garantizar la confidencialidad de las comunicaciones entre ciudadanos, empresas u otras instituciones públicas a través de las redes abiertas de comunicación. Se garantiza que únicamente el ciudadano puede acceder a la información, evitando suplantaciones.

Los certificados digitales funcionan como puentes para realizar los trámites y transacciones a través de internet. En vez de ir a la ventanilla de una Administración, es la ventanilla la que viene al ciudadano cuando la necesite. Eso es lo que ofrece el certificado digital, un fichero cuya misión es verificar la identidad de una persona cuando ésta realiza un trámite o transacción a través de Internet y que garantiza que el documento firmado electrónicamente no ha sido manipulado.

Aparte de evitar desplazamientos y colas, se ahorra tiempo y dinero, se gana en comodidad se puede instalar en cualquier ordenador, aunque cuando se solicita por primera vez en Internet luego, necesariamente, debe descargarse en el mismo equipo desde el que se realizó la petición, y también supone un ahorro en tecleo de datos y papel para los organismos que prestan servicios. En el caso de presentar la declaración de la renta «on-line» y que salga a devolver, es fácil que en unos pocos días se tenga el dinero. Las declaraciones «on-line» no tienen preferencia para la devolución, simplemente es que los datos llegan antes».

2.3.2.3. Partes de un certificado digital

Las tres partes más importantes de un certificado digital son:

1. Una clave pública
2. La identidad del implicado: nombre y datos generales,
3. La firma privada de una tercera entidad llamada autoridad certificadora que todos reconocen como tal y que válida la asociación de la clave pública en cuestión con el tipo que dice ser.

Algunos de los datos más importantes de este formato son los siguientes:

- Versión: 1,2 o 3
- Número de Serie: 0000000000000000
- Emisor del Certificado: VeriMex
- Identificador del Algoritmo usado en la firma: RSA, DSA o CE
- Periodo de Validez: De Enero 2002 a Dic 2003
- Sujeto: Anita la Gordita
- Información de la clave pública del sujeto: la clave, longitud, y demás parámetros
- Algunos datos opcionales, extensiones que permite la v3
- Firma de la Autoridad Certificadora

Un certificado digital entonces se reduce a un archivo de uno o dos octetos de tamaño, que autentica a un usuario de la red.

La figura 17 muestra el proceso de solicitud de un certificado digital.

Figura 16: Ciclo del Certificado Digital

2.3.7. Decreto Nº 1.204, del año 2.001

El decreto Nº 1.204 del año 2001 que hace referencia a los mensajes de datos y firmas electrónicas. Expresa que mediante la implementación del gobierno electrónico el ciudadano venezolano o extranjero tiene acceso, desde cualquier lugar del mundo, a la información sobre el funcionamiento y gestión de cada uno de los entes estatales y gubernamentales del país, la utilidad de estas tecnologías y de este decreto-ley que las hace más seguras, aumenta exponencialmente día a día.

Este marco legal y técnico que adopta el país para el desarrollo de la firma electrónica es compatible con el que ya existe en otros países. La aplicación de criterios legales diferentes a los aplicados en otros países en cuanto a los efectos legales de la firma electrónica y cualquier diferencia en los aspectos técnicos, en virtud de las cuales las firmas electrónicas son consideradas seguras, resultaría perjudicial para el desarrollo futuro de las relaciones y en especial del comercio electrónico que es una modalidad mercantil que está creciendo y englobando

transacciones de todo tipo a nivel mundial y, por consiguiente, para el crecimiento económico del país y su incorporación a los mercados globales.

Debido a la evolución acelerada de la tecnología, los países con legislaciones más recientes sobre el tema, han optado al igual que el nuestro, por proyectos simples, tecnológicamente neutros y dinámicos, en los cuales se mantienen los grandes aciertos de modelos anteriores (aplicación indistinta a todo tipo de actos y contratos, tanto en el sector público como en el privado y la homologación con los documentos en formato tradicional). El mecanismo adoptado ha sido la elaboración de normas legales de carácter general, que validan y homologan los actos y contratos celebrados por estos medios, y que contienen provisiones reglamentarias para su implementaci6n. Con los elementos básicos principales contenidos en este decreto-ley se brinda seguridad y certeza jurídica a las comunicaciones, transacciones, actos y negocios electrónicos que utilicen los mecanismos previstos en él.

Se observa que en el artículo 16 del capítulo IV, resaltan información relevante sobre firmas electrónicas "La Firma Electrónica que permita vincular al Signatario con el mensaje de datos y atribuir la autoría de éste, tendrá la misma validez y eficacia probatoria que la ley otorga a la firma autógrafa. A tal efecto, salvo que las partes dispongan otra cosa, la firma electrónica deberá llenar los siguientes aspectos:

1. Garantizar que los datos utilizados para su generación puedan producirse sólo una vez, y asegurar, razonablemente, su confidencialidad.
2. Ofrecer seguridad suficiente de que no pueda ser falsificada con la tecnología existente en cada momento.
3. No alterar la integridad del mensaje de datos. A los efectos de este artículo, la firma electrónica podrá formar parte integrante del mensaje de

datos, o estar inequívocamente asociada a éste; enviarse o no en un mismo acto.

2.3.8. Pilares del certificado digital

El Certificado digital es el medio técnico que proporciona los datos electrónicos de los elementos de autenticación del firmante, integridad de la información y no repudio de lo firmado. Es decir, se trata de la generación de un entorno en el que las comunicaciones y transacciones sean seguras y fiables. Para todo ello, el Certificado Digital, se basará en tres pilares tecnológicos:

- **Criptografía simétrica:** Este mecanismo utiliza una misma llave para cifrar y descifrar la comunicación.
- **Criptografía asimétrica:** Este mecanismo utiliza dos claves: una de las llaves ser pública, podrá ser conocida por todos, y otra, que ser privada, deberá estar custodiada por su propietario (Ver figura 18).

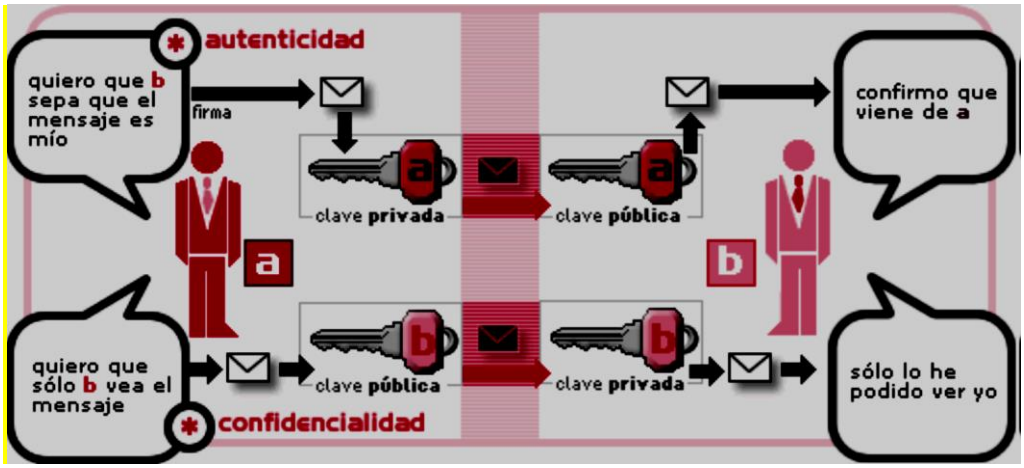


Figura 17: Ciclo del Certificado Digital

- **Función hash:** Nos permite asegurarnos que nuestra comunicación llegue a su destino sin que haya sido modificada, es decir, la integridad. Las funciones

hash transforman un mensaje de longitud arbitraria en un número fijo de bits, de tal forma que dos mensajes diferentes generaran dos secuencias HASH distintas. Así vamos a identificar de forma única al mensaje original.

Descripción de una firma digital mediante huella electrónica (hash)

Se genera en primer lugar una huella digital del documento mediante un algoritmo de hash, ésta será una huella electrónica o hash. Esta huella es una secuencia de unos y ceros de una determinada longitud. Un determinado documento sólo puede generar una huella. Si ese documento se modifica, aunque sólo sea en una coma, la huella será totalmente distinta.

El hash que se ha generado a partir del documento se encripta con la clave privada del emisor del documento, y se obtiene la firma digital de este documento (Ver Figura 19).

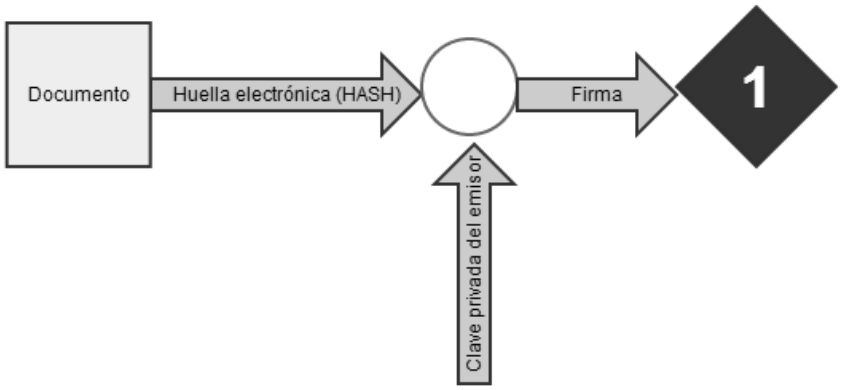


Figura 18: Firma digital de un documento

La firma obtenida garantiza:

- Que el documento ha sido firmado por el emisor. No es posible que haya sido firmado por otra persona a menos que haya robado la clave privada del emisor.
- Que el contenido del documento no ha sido manipulado desde la emisión hasta la recepción del mismo.

Envío del documento firmado

Dependiendo del formato elegido, el documento y la firma pueden ir en el mismo fichero o en ficheros separados (Ver Figura 20).

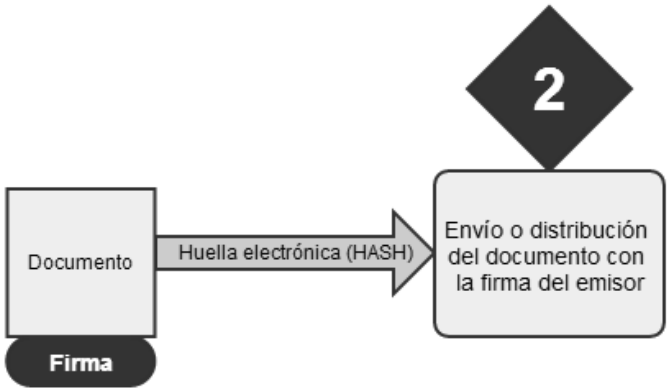


Figura 19: Envio del documento con la firma digital

Recepción del documento y comprobación

Cuando el documento llega al destinatario se lleva a cabo el proceso de comprobación, para asegurar que el documento no ha sido manipulado y para garantizar que efectivamente lo envía la persona real (y no alguien que suplanta su identidad).

En primer lugar se calcula el hash del documento utilizando el mismo algoritmo de HASH que utilizó el emisor.

Por otra parte se desencripta el hash original del documento (contenido en la firma). Para descifrar éste hash se utiliza la clave pública del emisor.

En el caso de los certificados digitales, la clave pública se obtiene a través de la entidad certificadora, que garantiza que pertenece precisamente a esa persona (emisor).

Una vez extraída el hash de la firma se la compara con el hash obtenido directamente del documento.

Si los dos hash son iguales podemos garantizar que el documento no ha sido manipulado y que ha sido enviado por el emisor real. La figura 21 muestra cómo se realiza este proceso.

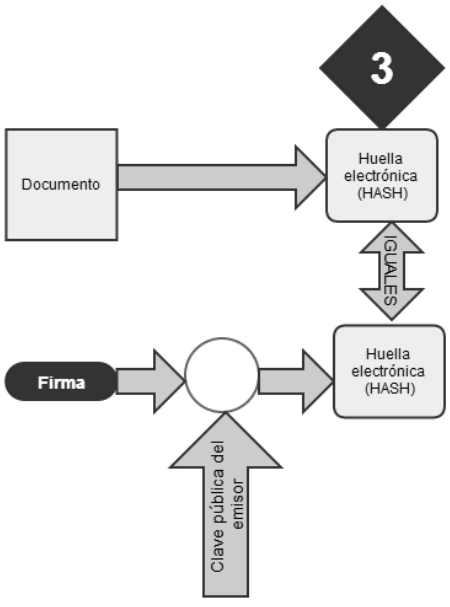


Figura 20: Comprobación del documento mediante la firma digital

El procedimiento de firma en origen y de comprobación en recepción lo realizan automáticamente las aplicaciones de cifrado. Es un proceso prácticamente transparente para el usuario.

2.4. Estudio y análisis comparativo de librerías y/o herramientas

A continuación se muestra el estudio realizado a las librerías y herramientas utilizadas durante el desarrollo del presente Trabajo Especial de Grado.

2.4.1. SDK (Software Development Kit o Paquete de desarrollo de software)

Al desarrollar una aplicación de software que involucre un lector de huellas dactilares, es necesario el envío y captura de información entre el lector de huellas y el computador. Para este ello se hace uso de librerías que facilitan el proceso, siendo las librerías y la documentación necesaria para utilizarlas durante el desarrollo de software denominadas SDK (*Software Development Kit* o Paquete de desarrollo de software).

En el mercado se encuentran dos tipos de SDK para lectores de huella:

- SDK Específicos: Son SDK creados por las empresas que fabrican los lectores de huellas. Además de cumplir con las funciones principales que se necesitan para controlar un lector de huellas, estos SDK incluyen funciones específicas para una marca y modelo de lector de huella y no son compatibles con otras marcas de lectores de huellas. Un ejemplo es el SDK BioNano de Anviz.
- SDK Genérico: Cumple con las funciones principales para controlar los lectores de huella, es decir, detectar y activar el lector de huella, además de capturar y encontrar los rasgos característicos (minucias) que hacen única una huella dactilar, con esto podemos identificar y verificar a las personas con su

huella dactilar. Un SDK genérico funciona para varias marcas y modelos de lectores de huella. Un ejemplo es el SDK de la empresa Griaule Biometrics o el Verifinger de Neurotechnology.

Ventajas

- **Ventajas de un SDK Genérico**
  - ✓ El software desarrollado con un SDK genérico funcionara para varias marcas y modelos lectores sin tener que cambiar el código fuente, economizando tiempo y dinero.
  - ✓ Cumple con las funciones principales que son capturar la huella y encontrar rasgos característicos (minucias) para identificar y verificar el personal.
  - ✓ Si el lector falla puede ser reemplazado por una mayor cantidad de lectores. No se requiere un modelo o marca especial.
  - ✓ El precio de un SDK genérico es menor que el SDK de un lector de huellas especifico.

- **Ventajas de un SDK Específico**
  - ✓ Al ser elaborado por la misma empresa que lo fabrica es 100% funcional dando una mayor velocidad y precisión en el reconocimiento de la huella.

Ejemplos de SDK Genérico

- **Fingerprint SDK (griaule biometrics)**

Fingerprint SDK es un paquete de desarrollo de software (SDK) innovador que permite integrar la biometría en un amplio rango de aplicaciones. Gracias a su

soporte para docenas de lenguajes de programación, riqueza en el código de los ejemplos y su documentación completa [griaulebiometrics, 2012]

Su principal objetivo es permitir el desarrollo de aplicaciones personalizadas tales como: sistemas de autorización, sistema de transacciones, tiempo de atendimiento, identificación en puntos de venta y acceso físico a través de autentificación por huellas dactilares. Fingerprint SDK posee dos diferentes versiones:

1. **Fingerprint SDK para Windows:**

Soporta muchos lenguajes de programación a través de DLL, Java, ActiveX o .NET.

2. **Fingerprint SDK para Java**

Permite el desarrollo multi-plataforma de programas en Java que funcionan en Microsoft Windows o en GNU/Linux.

Características

- ✓ Soporte al desarrollo de aplicaciones empresariales y Web
- ✓ Soporte a Windows y a Linux
- ✓ Independencia en el sensor, soporta múltiples lectores de huellas dactilares (26 diferentes sensores son soportados)
- ✓ Fácil e intuitivo, casi todas las librerías de reconocimiento de huellas dactilares son provistas con una única interfaz en forma de DLL, donde usted necesita crear archivos de importación para el lenguaje que usted usa, entre

otros obstáculos. Fingerprint SDK te ofrece ActiveX, DLL, Java y componentes .NET;

- ✓ Consolidación de la plantilla de huellas dactilares. Mejora la tasa de reconocimiento a través de la mejora en la calidad de la plantilla. Elimina la necesidad de usar varios ejemplos del mismo dedo, por ende reduce el tamaño de la base de datos y el tiempo de identificación.

- **VeriFinger SDK**

VeriFinger SDK se basa en la tecnología de reconocimiento de huella dactilar y está dirigido a desarrolladores de sistemas biométricos e integradores. El SDK permite el desarrollo rápido de aplicaciones biométricas usando la funcionalidad del algoritmo VeriFinger para Microsoft Windows, Linux y Mac OS X [neurotechnology, 2013]

Están disponibles los siguientes SDKs de VeriFinger:

- ✓ El SDK VeriFinger Standard, que está dirigido a la mayoría de desarrolladores de sistemas biométricos y permite el desarrollo de aplicaciones biométricas de Microsoft Windows, Linux, Android o Mac OS X.
- ✓ El SDK VeriFinger (ampliado), está dirigido a desarrolladores que desean comenzar rápidamente el desarrollo del sistema de identificación de huellas digitales cliente/servidor basado en redes.

2.4.2. Tecnologías para el Desarrollo Web

En esta sección se expondrán los componentes principales y las herramientas que existen para hacer posible el desarrollo de la aplicación web objeto de este

trabajo. En el desarrollo web se tienen herramientas para realizar el diseño, programación y depuración de una aplicación. Todas las herramientas son de gran importancia, por lo cual se deben elegir las más adecuadas a las necesidades de la aplicación y capacidades del desarrollador.

A continuación se define qué es una aplicación web, cómo está estructurada y su clasificación. Además, se describen las tecnologías utilizadas en el desarrollo de la aplicación del Trabajo Especial de Grado.

2.4.2.1. Aplicación Web

Aplicaciones Web son las aplicaciones que pueden ser accedidas por los usuarios a través de un servidor de internet o intranet, utilizando un navegador. Estas aplicaciones son ligeras, actualizables en tiempo real y si es por internet el cliente puede acceder desde cualquier parte del mundo. [Aplicación Web, 2009]. Las aplicaciones Web se utilizan para tiendas virtuales, blogs, compartir información, correo Web, wikis, foros, redes sociales, entre otros. [ALEGSA, 2008].

En la actualidad, las aplicaciones web son muy populares debido a la facilidad de actualización y mantenimiento sin tener que distribuir e instalar software de forma presencial a miles de personas, en distintas partes del mundo. Este tipo de aplicaciones, gracias a las ventajas que ofrecen las comunicaciones y el flujo de la información a través de Internet, así como el bajo costo de implementación y desarrollo, han llevado a grandes compañías a nivel mundial a inclinarse hacia el software basado en la web como vía de comunicación con sus clientes o usuarios.

Existen diversos tipos de aplicaciones, con distintos propósitos y funcionalidades pero la gran mayoría sigue la arquitectura cliente-servidor, la cual se describe a continuación.

2.4.2.2. Arquitectura Cliente-Servidor

Es una arquitectura para el desarrollo de aplicaciones web, donde básicamente el cliente le hace peticiones al servidor que le da respuesta, en otras palabras, es un entorno en el que las transacciones se dividen en procesos independientes para el intercambio información, recursos o servicios.

Se denomina cliente al proceso que inicia el diálogo o solicita los recursos tales como manejo de la interfaz de usuario, captura y validación de los datos de entrada, generación de consultas e informes sobre las bases de datos y el servidor al proceso que responde a las solicitudes. [Cliente-Servidor, 2010].

En la Figura 22 se puede observar el nivel más básico de la arquitectura cliente-servidor [Barranco de Areba, 2001].

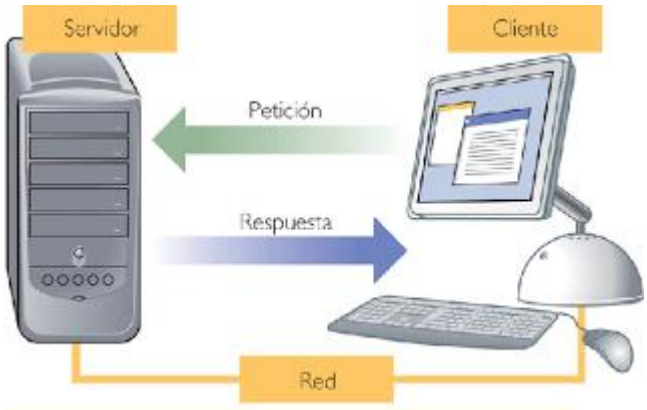


Figura 21: Arquitectura Cliente-Servidor

Cuando los procesos cliente y servidor están en uno o más computadores en forma independiente en la red, entonces, el servidor puede proporcionar servicios a más de un cliente, y un cliente puede solicitar servicios a varios servidores que se encuentren disponibles en la red sin importar la ubicación y las características físicas

que este posea. La red es la que permite unir los servidores y clientes, por lo tanto, es el medio de comunicación entre clientes y servidores. [Valle, 2005]

La arquitectura cliente-servidor proporciona las ventajas de la arquitectura de aplicaciones basadas en componentes, destacándose:

- Función distribuida entre el cliente y el servidor
- Interfaz de presentación gráfica
- Multitarea dado que los servicios pueden ser invocados síncrona o asíncronamente, de modo que un cliente puede realizar distintas operaciones asíncronas al mismo tiempo
- Inversión del control, el cliente es quien determina el diálogo o navegación a realizar por los servicios.

2.4.2.3. Tecnologías del lado del cliente

Por obtener mayor velocidad de respuesta y agilidad en las aplicaciones web, es necesario realizar acciones en el cliente.

A continuación se describen brevemente las tecnologías utilizadas y las acciones que realizan.

- Lenguaje de Marcas de Hipertexto (HTML): Es un lenguaje de programación que permite describir la estructura y el contenido en forma de texto, mediante etiquetas, no es un lenguaje compilado por lo tanto los errores de programación no son detectados. Su principal base son los hipervínculos o los link, que permiten el enlace entre páginas HTML o documentos, permitiendo así la navegación por el Word Wide Web.

El Lenguaje HTML está compuesto por dos partes, cabecera y cuerpo. En la cabecera va la identificación de la página HTML y se reconoce porque comienza con" <head>" y termina con "</head>", de igual manera el cuerpo que lleva contenido del documento, comienza con "<body>" y termina con "</body>".

- Hojas de Estilo en Cascada (Cascading Style Sheets - CSS) El aspecto visual de las interfaces de usuario del sistema se manejó con CSS, que es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML. Las Hojas de estilo se definen entre el "<head>" y el "</head>", y permite cambiar las características de una página Web sin modificar su contenido, sus principales usos son para cambiar imágenes, fondos, tamaño de letras, textos, organización del documento. CSS es la mejor forma de separar los contenidos de su presentación. Esta separación presenta numerosas ventajas, ya que obliga a crear documentos HTML bien definidos. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes [Eguiluz, 2009].
- JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas, definiendo página web dinámica, como aquella que incorpora efectos como texto que aparecen y desaparecen, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario.

  Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios [Eguiluz, 2009].

- Una de las librerías más destacadas y desarrolladas en JavaScript es JQuery, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web, entre otros [JQuery, 2013]. Durante el desarrollo de la aplicación se utilizó ampliamente esta tecnología, para manejar los eventos y comportamientos de los componentes de la interfaz de usuario
- Ajax (Asynchronous JavaScript and XML) Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad y velocidad en las aplicaciones [Techlosofy, 2008].

2.4.2.4. Tecnologías del lado del servidor

Las tecnologías del lado del servidor son las encargadas de realizar el procesamiento de las peticiones del cliente, mediante la interpretación de una secuencia de comandos para generar la respuesta y enviarla de vuelta en un lenguaje soportado por el navegador web.

A continuación se describe algunas de las tecnologías utilizadas del lado del servidor.

- Sistemas Manejadores de Bases de Datos (SMBD) Un Sistema Manejador de Base de Datos (SMBD) consiste en un conjunto de datos relacionados entre sí y un conjunto de herramientas de software (y/o hardware) para tener acceso a esos datos. El SMBD permite definir, procesar y administrar la base de datos y sus aplicaciones. El SMBD puede organizar y presentar los datos

seleccionados de una base de datos. Esta capacidad permite, a quien toma decisiones rastrear, probar y consultar el contenido de la base de datos para extraer las respuestas a las preguntas no recurrentes y no previstas en informes regulares.

Entre las funciones del SMBD se tiene la definición y manipulación de datos, seguridad, integridad, recuperación y concurrencia de los datos, diccionario de datos y, en general, ejecutar todas estas funciones de la forma más eficiente posible. El objetivo primordial de un SMBD es proporcionar un entorno para recuperar información y almacenar nueva información en la base de datos, para lo cual debe proporcionar a los usuarios una visión abstracta de los datos. Es decir, los detalles de cómo se almacena y se mantienen los datos, son transparentes para los usuarios [Campoy, 1999].

Los datos e información almacenados por la aplicación web desarrollada, son almacenados en una base de datos relacional, utilizando PostgreSQL como sistema manejador, el cual se explica a continuación.

- PostgreSQL es un sistema manejador de base de datos. Este software proporciona un servidor de base de datos SQL muy rápido, multi-hilo, multi usuario y robusto. El servidor PostgreSQLestá diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido.

El lenguaje de programación utilizado para el desarrollo de la aplicación fue Java, por ser un lenguaje moderno, orientado a objetos, de código abierto, robusto, seguro y portable que integra gran cantidad de componentes y herramientas de trabajo que facilitan el desarrollo de aplicaciones.

El desarrollo se realizó utilizando el modelo de aplicación que define Java EE (Java Plataform Enterprise Edition), la cual integra un conjunto coordinado de tecnologías que ayudan a reducir significativamente la complejidad del desarrollo de aplicaciones. Tomando como base Java SE (Java Platform Standard Edition), Java EE añade capacidades y tecnologías adicionales para proporcionar una plataforma completa, estable y segura para el desarrollo de aplicaciones Web empresariales.

Con la finalidad de trabajar con la tecnología Java EE, se utilizó la versión Java SE 6. Se utilizó el JDK (Java Development Kit) 7 actualización 17. El JDK es un producto que proporciona un conjunto de herramientas que permite el desarrollo de aplicaciones Java [Oracle, 2013].

Para el desarrollo de la aplicación y la interfaz, se utilizó la plataforma Eclipse para desarrolladores Java EE en su versión Juno. Eclipse es un entorno de desarrollo integrado que permite el desarrollo modular de aplicaciones, facilitando los procesos de compilación, depuración y ejecución de las mismas. Está escrito en lenguaje Java y proporciona una interfaz intuitiva para la construcción de aplicaciones Web Java EE.

- Java EE: Durante el desarrollo de esta aplicación se utilizaron parte de los componentes que define el modelo de aplicación de Java EE. Este modelo define la utilización del lenguaje de programación Java y su máquina virtual, para la construcción de aplicaciones empresariales que ofrezcan algún tipo de servicio a clientes, empleados, etc. Dichas aplicaciones pueden ser inherentemente complejas, pueden acceder gran cantidad de información de múltiples fuentes, pueden ofrecer sus servicios a una variedad de clientes, por lo tanto necesitan la portabilidad, seguridad y escalabilidad que el modelo de programación Java EE ofrece [Oracle, 2013].

La plataforma Java EE utiliza un modelo distribuido multinivel de capas para sus aplicaciones empresariales. La lógica de la aplicación es dividida en componentes de acuerdo a sus funcionalidades, los diferentes componentes que conforman la aplicación final pueden estar instalados en diferentes lugares (Cliente o Servidor) dependiendo de la tarea que cumplan dentro de la aplicación Java EE. La Figura 23, tomada de [Oracle, 2013], muestra la arquitectura multinivel del modelo de aplicación Java EE.

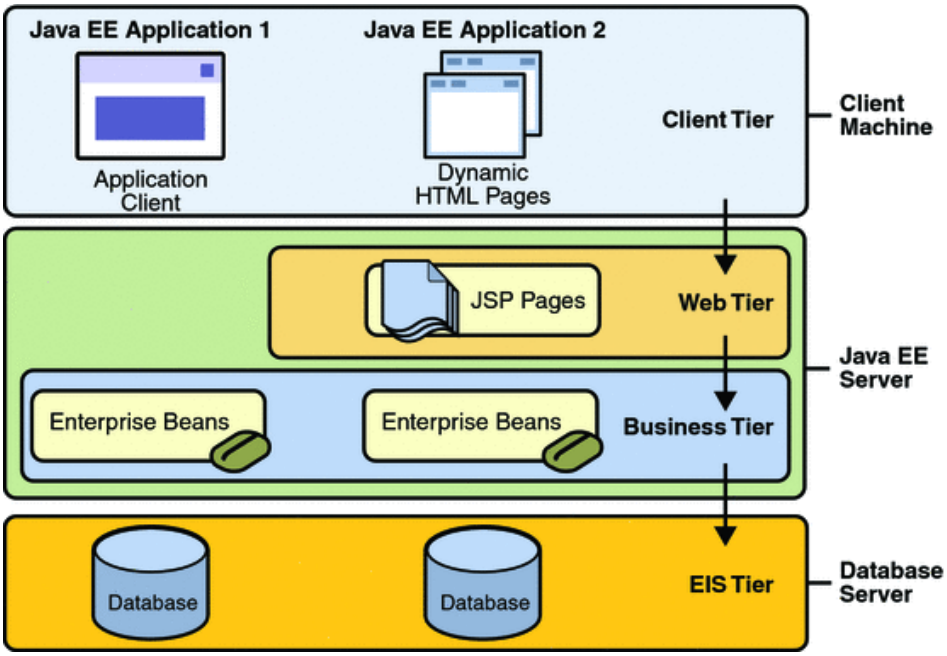


Figura 22: Arquitectura Multinivel Java EE

[Oracle, 2013]

Componentes de Java EE

Un componente Java EE es una unidad funcional de software independiente que forma parte de una aplicación junto con las demás clases y archivos, capaz de comunicarse con otros componentes.

La especificación Java EE define los siguientes componentes:

3. Aplicaciones cliente, las cuales se ejecutan del lado del cliente.
4. Java Servlets, Java Server Faces (tecnología no utilizada en el presente T.E.G.) y Java Server Pages, las cuales son tecnologías web que se ejecutan del lado del servidor.
5. Enterprise Java Beans (tecnología no utilizada en el presente T.E.G.), los cuales son componentes de negocio que se ejecutan en el servidor.

- Página de servidor Java (JSP) es una tecnología de Java que permite a los desarrolladores Web mantener y construir páginas Web dinámicas. Como parte de la familia de tecnologías Java, JSP permite el desarrollo de aplicaciones independientes de la plataforma donde se ejecuten. JSP permite separar la interfaz de usuario de los elementos de generación de contenido, permitiendo a los desarrolladores modificar el diseño general de la página sin alterar el proceso de generación de contenido dinámico.

  Las principales características de la tecnología JSP son:

1. Es un lenguaje para implementar paginas JSP, las cuales son documentos basados en texto que describen como procesar una solicitud y construir una respuesta.
2. Un lenguaje de expresión que permite acceder a objetos del lado del servidor.
3. Ofrece mecanismos para implementar extensiones al lenguaje JSP.

- Java Servlets: son módulos que extienden los servidores petición-respuesta, que corren dentro de un contenedor de servlets o un servidor de aplicaciones (ver Figura 24). Estos módulos reciben los requerimientos del cliente y

dinámicamente generan una respuesta que es devuelta regularmente en forma de documento HTML.

Cada Servlet tiene su propio ciclo de vida:

- Un servidor carga e inicia el servlet.
- El servlet maneja cero (0) o muchas peticiones.
- El servidor elimina el servlet.

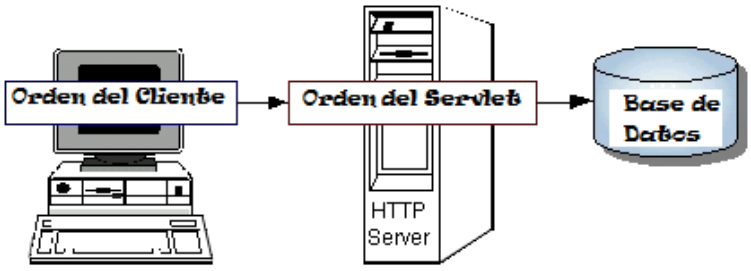


Figura 23: Java Servlets

1. Inicializar un Servlet (primera fase)

Cuando el servidor carga un servlet (lo hace ejecutando el método “init”). La inicialización se completa antes de manejar peticiones de clientes y antes de que el servlet sea destruido (muchos servlets pueden ser llamados al mismo tiempo ya que no presentan problemas de concurrencia durante su inicialización).

2. Interactuar con Clientes (segunda fase)

Después de la inicialización, el servlet puede manejar peticiones de clientes para el procesamiento necesario de respuesta a las peticiones recibidas. (Hay que tener en cuenta los problemas entre requerimientos concurrentes con acceso a variables compartidas).

3. Destruir un Servlet (tercera fase)

Los servlets se ejecutan hasta que el servidor los destruye, por cierre el servidor o bien a petición del administrador del sistema. Cuando un servidor destruye un servlet, ejecuta el método destroy del propio servlet.

Contenedor de servlets

Un contenedor de servlets es un servidor Web especializado que soporta la ejecución de servlets. Integra las funcionalidades básicas de los servidores Web tradicionales con ciertas especificaciones y extensiones de la tecnología Java Servlet. Cada servlet dentro de la aplicación se encuentra registrado en el contenedor. Los servlets registran en el contenedor la información pertinente de la funcionalidad que cumplen dentro del sistema. De igual forma los servlets registran un mecanismo de identificación que permitirá al contenedor localizarlos, para su posterior inicialización y canalización de solicitudes.

El contenedor de servlets utilizado para el desarrollo de este trabajo fue Apache Tomcat en su versión 6.0. Apache Tomcat es un contenedor de servlets desarrollado por la ASF (Apache Software Foundation) que implementa las especificaciones de las tecnologías Java Servlets y Java Server Pages, a la vez que proporciona un ambiente de servidor Web HTTP nativo para Java.

Al momento de la implementación del MVC, tenemos a los servlets en la parte del controlador.

2.4.3. Herramientas para automatizar las pruebas unitarias

En los últimos años se han creado una serie de frameworks que permiten automatizar las pruebas unitarias, permitido definir estas y ejecutarlas en reiteradas ocasiones. Estos frameworks son denominados xUnit.

Estos frameworks se basan en los conceptos de Test Case y Test Suite, el primer es la prueba unitaria del componente a testear, la segunda nos permite agrupar varias pruebas unitarias para poder ejecutarlas en forma conjuntas.

La finalidad de los Test Case es probar funcionalidad de un determinado modulo. En lenguajes orientados a objetos seria probar los métodos de una clase. Los Test Suite permiten agrupar todas las pruebas de un componente, permitiendo ejecutar de forma conjunta todas las pruebas definidas para un Componente.

A medida que el desarrolló va avanzando y se van definiendo mas caso de pruebas, estas herramientas nos permiten hacer pruebas de regresión. De esta forma si hay cambios en clases ya testeadas se les puede volver a evaluar su correctitud. Esto es muy útil en el caso de la metodología XP donde cuando hay cambios se evalúan y hace una refactorización para mantener de forma simple el producto, estas herramientas nos permiten poder reevaluar que tras los cambios se sigan cumpliendo las pruebas anteriormente definidas.

Las principales ventajas de la utilización de pruebas unitarias automáticas en el desarrollo son:

1. Fomentan el cambio: Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se ha dado en llamar

refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.

2. Simplifica la integración: Puesto que permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
3. Documenta el código: Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.
4. Los errores están más acotados y son más fáciles de localizar: dado que tenemos pruebas unitarias que pueden desenmascararlos.

JUNIT

JUnit es un framework xUnit para la plataforma Java, esta nos permite definir los casos de prueba unitarios para las clases importantes del sistema. Permite ejecutar las clases de forma controlada para poder evaluar si esta se comporta de la manera esperada.

JUnit ejecutará las pruebas sobre la clase y comparara contra los resultados esperados, si los resultados son correctos indicara que las pruebas fueron exitosas, en caso que alguno de los resultados no sea el esperado Junit indicara que se produjo un fallo he indicara el o los métodos que no pasaron sus evaluaciones. Es importante destacar que Junit seguirá con las pruebas aunque ya se haya detectado una falla.

Este framework es Open Source y fue desarrollado por Kent Beck y Erich Gamma, el primero uno de los creadores de XP y el segundo conocido por ser uno de los líderes del proyecto Eclipse y por sus trabajos en Patrones de Diseño.

JUnit, los casos de prueba son clases que derivan de la clase TestCase, e implementan métodos sin parámetros de nombre testXXX, donde XXX es una descripción de lo que está probando ese método [Beck, 2000].

Las pruebas implementadas que extenderán de la clase TestCase tiene la posibilidad de sobre-escribir los métodos setUp() y tearDown(), los cuales son invocados antes y después de cada método de test. Esto permite inicializar y liberar recursos entre la ejecución de los distintos métodos en la clase, permitiendo asegurarse de que no hay efectos colaterales entre la ejecución de los distintos test.

El propio framework incluye formas de ver los resultados (runners) que pueden ser en modo texto, gráfico (AWT o Swing) o como tarea en Ant.

En la actualidad las herramientas de desarrollo como NetBeans y Eclipse cuentan con plugins que permiten que la generación de las plantillas necesarias para la creación de las pruebas de una clase Java se realice de manera automática, facilitando al programador enfocarse en la prueba y el resultado esperado, y dejando a la herramienta la creación de las clases que permiten coordinar las pruebas.

Los métodos de prueba consisten en verificar que los resultados obtenidos sean los esperados. Para esto JUnit ofrece un conjunto de métodos (assert, assertEquals, assertTrue y otros) que permiten realizar comparaciones entre objetos o comprobar la veracidad de una condición determinada.

2.5. Patrón Modelo-Vista-Controlador (MVC)

Es un patrón de diseño, comúnmente utilizado en aplicaciones Web, que separa los datos de la aplicación, la interfaz de usuario y la lógica de control en tres (3) componentes distintos (Ver Figura 25).

Las actividades asociadas a los componentes se describen a continuación:

- Modelo: Define las reglas del negocio (funcionalidad de la aplicación) y representa las estructuras de los datos.
- Controlador: Gestiona las acciones del usuario actuando como intermediario entre las vistas y el modelo.
- Vistas: es la información presentada al usuario del sistema.

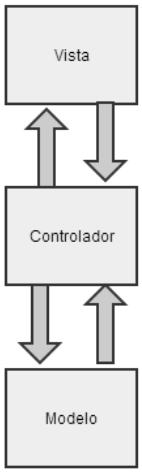


Figura 24: Modelo Vista Controlador

A continuación se presenta una secuencia de tareas del MCV las cuales se combinan para ofrecer una respuesta satisfactoria y coherente con respecto a la acción realizada por el usuario de la aplicación:

1. El usuario interactúa con la interfaz e introduce un evento cualquiera.
2. El controlador recibe y gestiona el evento introducido por el usuario.
3. El controlador se comunica con el modelo para realizar la tarea solicitada por el usuario. En algunas situaciones se deben modificar los datos del modelo.

4. Después de recibir la respuesta del modelo el controlador la entrega a la vista para que sea mostrada al usuario. En algunos casos el modelo se comunica con la vista para entregarle los resultados de la solicitud.
5. La vista espera nuevas interacciones por parte del usuario, si éstas ocurren se repite el ciclo nuevamente. [Modelo Vista Controlador, 2013]

La Figura 26, es un claro ejemplo del uso de Servlets y paginas JSP siguiendo el patrón de diseño MVC.

Las paginas JSP representan la capa de presentación y la tecnología de Servlets representa la capa de procesamiento de tareas. Los servlets actúan como controladores al procesar las solicitudes y crear los JavaBeans que las paginas JSP necesiten. Los controladores son responsables de decidir a qué página JSP debe encaminarse una solicitud. Las paginas JSP obtienen los objetos creados por los servlets y extraen el contenido dinámico para ser mostrado al usuario.

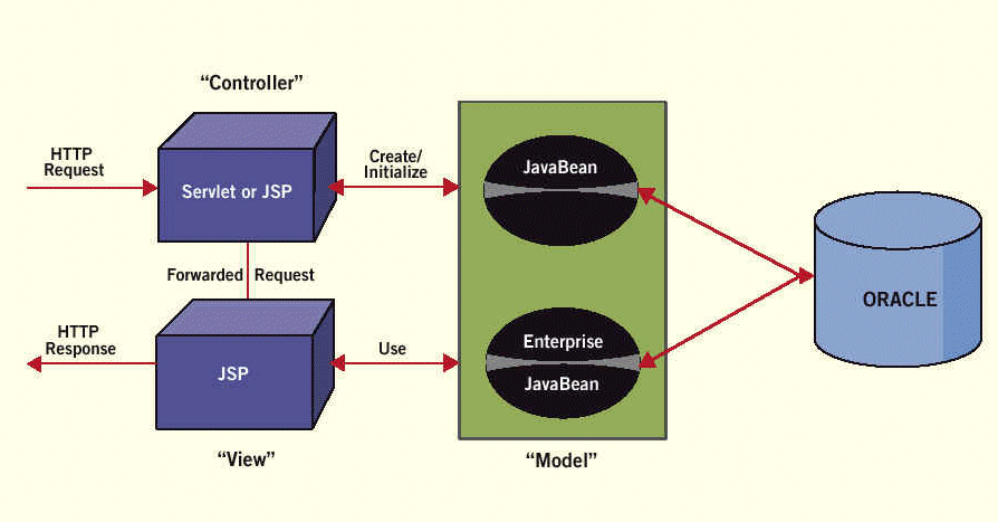


Figura 25: MVC con Servlet y JSP [Oracle, 2013]

2.6. Formato de Documento Portátil (PDF)

El formato de documento portátil (PDF) fue creado por Adobe Systems hace ya quince (15) años, permite entre otras cosas obtener y poder visualizar información desde cualquier aplicación y en cualquier sistema informático. La ventaja que se asoma de lo antes mencionado es que dicha información se puede compartir prácticamente con cualquier persona sin tener que preocuparse del lugar donde se encuentre y tampoco en que sistema esté trabajando. [Adobe, 2013]

2.6.1. Características del formato de documento portátil (PDF)

- Multiplataforma: Los archivos PDF se pueden ver e imprimir en cualquier plataforma. Ejemplo: Mac OS, Microsoft Windows, Unix y una gran variedad de plataformas móviles.
- Extensible: Alrededor de 1800 empresas alrededor del mundo ofrecen asesoría y soluciones basadas en PDF.
- Sofisticado en cuanto a la integridad de la información: Los archivos PDF muestran la misma información que los archivos originales. Ejemplo: texto, dibujos 3D, fotos, hipervínculos, etc.
- Más seguro: Los archivos PDF que se crean pueden ser firmados electrónicamente o también se les puede proteger con contraseña.

La gran aceptación que ha obtenido PDF a través de los años se debe a la posibilidad de imprimir exactamente lo que se ve en la pantalla sin la necesidad de adaptar márgenes o la visualización de las imágenes, como ocurre con otros formatos.

2.6.2. Estándares de los archivos PDF

- PDF/A: Aprobado por la ISO a finales de 2005, se principal objetivo es facilitar la preservación de los documentos electrónicos. Un aspecto

importante de los archivos PDF es la metadata (datos sobre los datos) contenida en las propiedades, cualquier cambio que se realice en el documento PDF se ve reflejado en la metadata de dicho documento. Como la metadata se encuentra en formato XML se puede modificar, así las empresas pueden crear su propio estándar para que los documentos electrónicos perduren a través del tiempo. [Adobe, 2013]

- PDF/E: Aprobado por la ISO en 2007, se caracteriza por proporcionar directrices para el intercambio fiable de documentos y diseños. Los grandes beneficiados de este estándar son los ingenieros, arquitectos, y profesionales dedicados a lo construcción ya que permite el intercambio de eficiente de documentos que soportan el contenido complejo en 3D [Adobe, 2013]
- PDF/X: Fue desarrollado para facilitar el intercambio de páginas finales y listas para impresión. Los profesionales que más se benefician son los que se encuentran en el mundo de las artes gráficas. [Adobe, 2013]

CAPÍTULO 3. Marco Aplicativo

Para el logro exitoso de los objetivos planteados en el Capítulo 1 se sigue un método de desarrollo basado en una adaptación de XP. Tal y como se plantea en el Capítulo 2: Marco Teórico, para el desarrollo de la aplicación se decidió utilizar JSP, HTML5, CSS3, JavaScript como tecnologías de desarrollo y para apoyar las funcionalidades gráficas de la aplicación se utiliza la librería JQuery.

En las siguientes secciones se describe como se realiza la adaptación del proceso de desarrollo de software Programación Extrema (XP) para el desarrollo de la aplicación Web. Además de exponerse detalladamente cada una de las iteraciones realizadas para el desarrollo de la aplicación basados en el método de desarrollo XP, especificando las cuatro etapas correspondientes a cada iteración.

3.1. Proceso de desarrollo de software: Metodología XP

Enfoque desarrollado por Kent Beck que se caracteriza por poner más esfuerzo en la adaptabilidad del proyecto, en la comunicación con el cliente y en el trabajo en equipo. Como es sabido los requerimientos cambian eventualmente con el transcurso del tiempo y poner esfuerzos en la fácil adaptación a los cambios de requerimientos de los clientes pregona una mejor práctica que forzar los resultados y el camino del proyecto a unos requerimientos inícíales rígidos e invariables [Beck, 2004]

Las claves adoptadas por XP para el éxito de un desarrollo son:

- Comunicación
- Simplicidad
- Retroalimentación (Feedback)

- Coraje / valor

Por otro lado surge la aseveración de que se debe usar XP cuando los proyectos poseen requerimientos cambiantes, el riesgo del proyecto es alto y el equipo de desarrollo es en parejas.

3.1.1. Características de XP (Programación Extrema)

- Desarrollo iterativo e incremental: pequeñas mejoras al código en intervalos cortos de tiempo.
- Programación en parejas: se deben llevar a cabo las tareas del proyecto con dos (2) personas sentadas en un (1) sólo puesto.
- Retroalimentación frecuente con el cliente
- Refactorización del código: reescribir ciertas partes del código para aumentar su legibilidad y corregir los errores arrojados por las pruebas.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, XP propone que todo el personal pueda corregir y extender cualquier parte del proyecto.
- Simplicidad en el código: las cosas se deben hacer siempre simples, y agregar un poco más de trabajo si y sólo si es necesario.

3.1.2. Adaptación de XP

A continuación se describen las actividades que se llevaron a cabo en la adaptación de XP para el desarrollo de la aplicación (Ver figura 27).

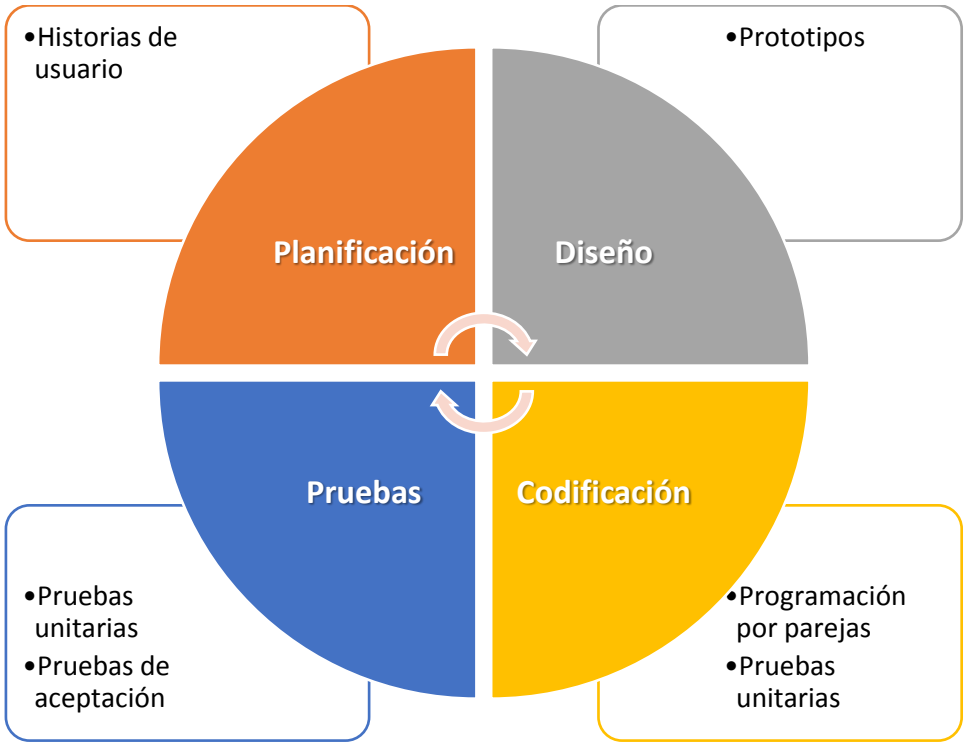


Figura 26: Adaptación de la metodología XP

En el resultado del estudio y análisis realizado en el seminario, se seleccionó la metodología XP como método de desarrollo de software para llevar a cabo el desarrollo de la aplicación web que se presenta en este Trabajo Especial de Grado, debido a las ventajas descritas anteriormente, entre ellas la simplicidad, agilidad y adaptación al cambio; sin embargo se decidió agregar un conjunto de artefactos que permitirán realizar un mejor levantamiento de información. Entre estos artefactos se tienen:

- Lista de requerimientos funcionales y no funcionales.
- Prototipo en papel
- Levantamiento de requerimientos

Se diseño diagramas BPMN (Notación de Modelado de Procesos de Negocio) en los procesos actuales de manejo de documentos y los diversos procesos que resultaron de la automatización, mediante una aplicación web para dicha solicitud los cuales se pueden observar en la figura 28 y 29 respectivamente.

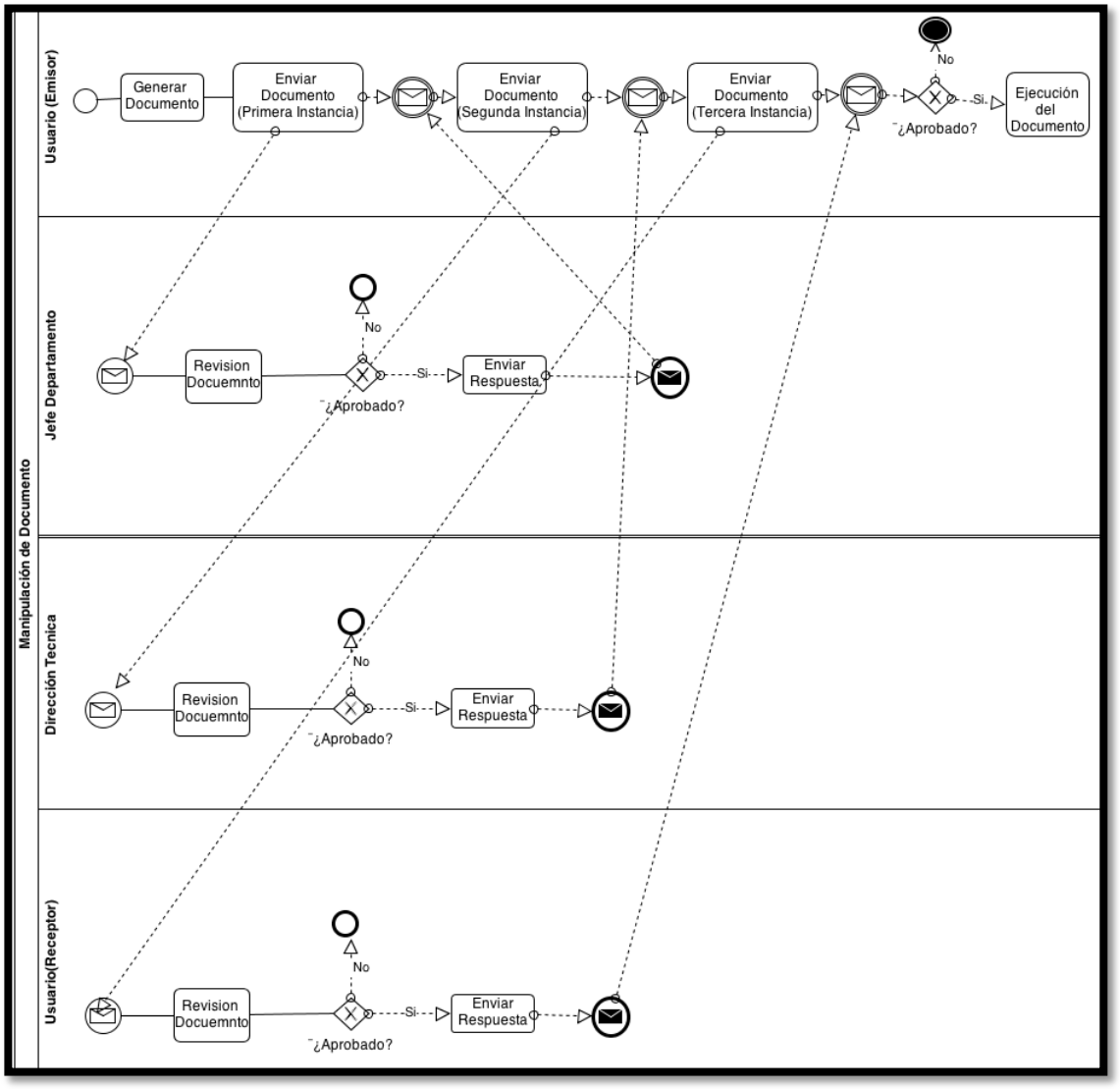


Figura 27: Modelo de proceso para la manipulación de documentos

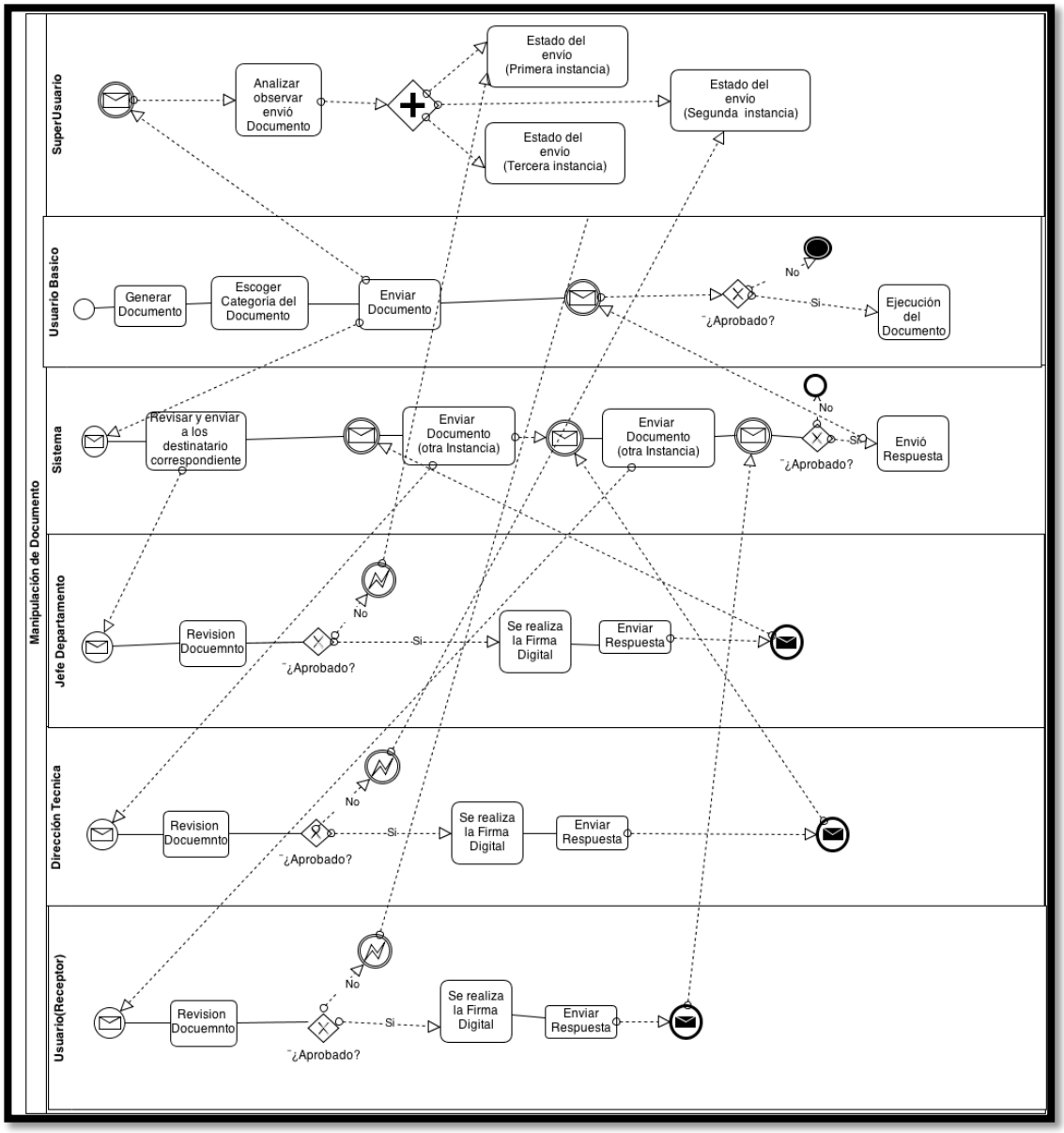


Figura 28: Modelo de proceso para la manipulación de documentos automatizada

3.1.3. Iteraciones

Las iteraciones presentes en el desarrollo estuvieron basadas en objetivos, es decir, se cumplían con los requerimientos presentes en las Historias de Usuarios (son una representación de los requerimientos del software, es la vía mediante la cual se

realiza la comunicación y son escritas por los clientes) antes de comenzar la siguiente iteración.

3.1.4. Planificación

Según la metodología XP en la fase de planificación el cliente debe explicar las necesidades que requiere que realice el sistema. A éste documento se le conoce como historias de usuarios, son necesarias aproximadamente de 20 a 80 Historias de Usuarios y se recomienda que duren aproximadamente de una (1) a tres (3) semanas cada una. Se definen los tiempos de entrega de la aplicación para recibir feedback del cliente. Se deben realizar reuniones constantemente con el equipo de desarrollo para así identificar problemas, proponer soluciones y obtener aquellos puntos a los que se le va a dar mayor importancia. En el presente Trabajo Especial de Grado se realizaron 43 historias de usuarios, las cuales tenían duración de una (1) a dos (2) semanas. Esto debido a que se realizó una adaptación de la metodología.

En este trabajo, las Historias de Usuarios se definen en la Tabla 7, la cual posee un campo ID que hace referencia al número de Historia de Usuario, también tiene una fecha que indica el inicio de la iteración, un campo para los requerimientos y la última columna llamada Tipo.

Cabe destacar que se tienen tres (3) variantes de Historias de Usuarios que describen el estatus de los requerimientos: nueva, corrección y mejora.

Tabla 7 Formato de Historias de Usuario

Id	Fecha	Requerimiento	Tipo

3.1.5. Diseño

La metodología XP propone que el diseño de soluciones debe estar basado siempre en la simplicidad. La refactorización resulta bastante útil para lograr alcanzar los objetivos. Se deben utilizar metáforas del sistema, así se ayuda al equipo de desarrollo a comprenderlo mejor y se facilita el trabajo de los mismos. Las tarjetas CRC (Clases, Responsabilidades, Colaboración) ayudan al equipo de desarrollo a definir las actividades durante el desarrollo del sistema. Se utilizan diagramas del Lenguaje de Modelado Unificado (UML): Casos de Uso y diagramas BPMN.

En este trabajo, para la fase de diseño se explica con palabras y/o realizan los diagramas o prototipos necesarios para la fácil comprensión y solución del problema asociado a cada iteración.

3.1.6. Codificación

La metodología XP propone que el grupo de desarrollo debe ser pequeño, dos (2) programadores y el cliente. Éste último deberá estar siempre disponible para formar parte del equipo de desarrollo. Los programadores deben trabajar en pareja para así fomentar el intercambio constante de ideas a lo largo del desarrollo. Las pruebas unitarias deberán ser creadas antes del código fuente, así se asegura una programación ágil con bajas probabilidades de errores. Se deben realizar las integraciones constantemente para evitar obtener un software fragmentado y difícil de unir. Además también se evitan las confusiones de los programadores. Se recomienda dejar para el final la optimización del código y también utilizar estándares de programación.

En éste trabajo, la codificación se realiza en parejas pero no se trabaja en conjunto con la misma Historia de Usuario sino que cada programador se encarga de

realizar y alcanzar el objetivo deseado con cada requerimiento dependiendo de la complejidad del mismo, es decir, mientras más complejo el requerimiento se torna se trabaja en parejas en el mismo lugar de trabajo, mientras que si la complejidad no es tan alta cada programador realiza la programación por separado.

3.1.7. Pruebas

La metodología XP establece que todos los módulos del sistema deben poseer pruebas asociadas. No se debe entregar ningún módulo del sistema sin antes haber sido sometido a prueba. Finalmente, en el denominado período de pruebas de aceptación o de caja negra se definen las entradas del sistema y sus salidas. Y además se recomienda automatizar las pruebas para que el sistema pueda ser probado varias veces.

Para éste trabajo, se llevaron a cabo las pruebas de aceptación recomendadas por XP pero con la diferencia de que dichas pruebas son realizadas después de que ya se tienen varios requerimientos completos. Por otra parte, se obvia la automatización de las pruebas.

3.1.8. Pruebas Unitarias

Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. El concepto de "módulo" varía de acuerdo al lenguaje de programación que estemos utilizando; por ejemplo, en Java sería una clase. Las pruebas unitarias nos aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema.

Los programadores realizan estas pruebas cuando: la interfaz de un método no es clara, la implementación es complicada, para testear entradas y condiciones

inusuales, luego de modificar algo. Éstas deben contemplar cada módulo del sistema que pueda generar fallas. Para poder integrar el código realizado al ya existente, el mismo debe aprobar satisfactoriamente todos los casos de prueba definidos.

En la metodología XP los programadores deben escribir las pruebas unitarias para cada módulo antes de escribir el código. No es necesario escribir casos de prueba para todos los módulos, sólo para aquellos en que exista la posibilidad de que puedan fallar.

Luego de escribir el código, los programadores ejecutan las pruebas, las cuales deben resultar 100% efectivas para que el código pueda integrarse al sistema. En caso contrario hay que solucionar los errores y ejecutar nuevamente los casos de prueba hasta lograr que ninguno de ellos.

Las pruebas son automatizadas utilizando herramientas como JUnit, de forma tal de poder soportar un testing continúo y mantener organizados los casos de pruebas.

3.1.9. Actores y Responsabilidades

Existen diferentes roles (actores) y responsabilidades en XP para diferentes actividades y propósitos durante el proceso. Para este Trabajo Especial de Grado los roles existentes son:

- Desarrollador: quien es el responsable de tomar las decisiones técnicas y de llevar a cabo la codificación, el diseño y realizar las pruebas de software.
- Cliente: quien es parte del equipo, determina qué construir y cuándo, desarrolla pruebas funcionales del software para determinar cuando está completo un determinado aspecto.

Las personas encargadas para cada rol son:

- Rol Desarrollador: Diana Alfaro
- Rol Desarrollador: Rut Martínez
- Rol Cliente: Prof. Carlos Acosta
- Rol Cliente: Profa. Ana Morales
- Rol Cliente: Profa. Adriana Liendo

3.2. Desarrollo de la Aplicación

En ésta sección se explica cómo se realizó la implementación de la adaptación de XP en el proceso de desarrollo del sistema.

3.2.1. Iteración 0: Modelo de la Aplicación

En esta iteración consiste en realizar el modelo de datos de la aplicación y el diseño de la aplicación en general.

Planificación

En la tabla 8 se muestran las historias de usuario desarrolladas en esta iteración.

Tabla 8 Historias de Usuario (Iteración 0: Modelo de Datos)

Id	Fecha	Requerimiento	Tipo
001		Realizar el modelo de datos de la aplicación y el diseño de la aplicación.	Nueva

Diseño

En esta iteración se realiza una breve descripción, de los principales problemas encontrados, y a través de un análisis con diagramas UML (Lenguaje de Modelado Unificado) como casos de uso y E-R (Diagrama de Entidad Relación), como se modela el diseño de la aplicación con el uso de estos modelos y diagramas.

- **Modelo de datos**

Mediante un análisis a la solución se definió la estructura y las tablas del sistema que contribuya al manejo de la data dentro de la aplicación (Ver Figura 30)

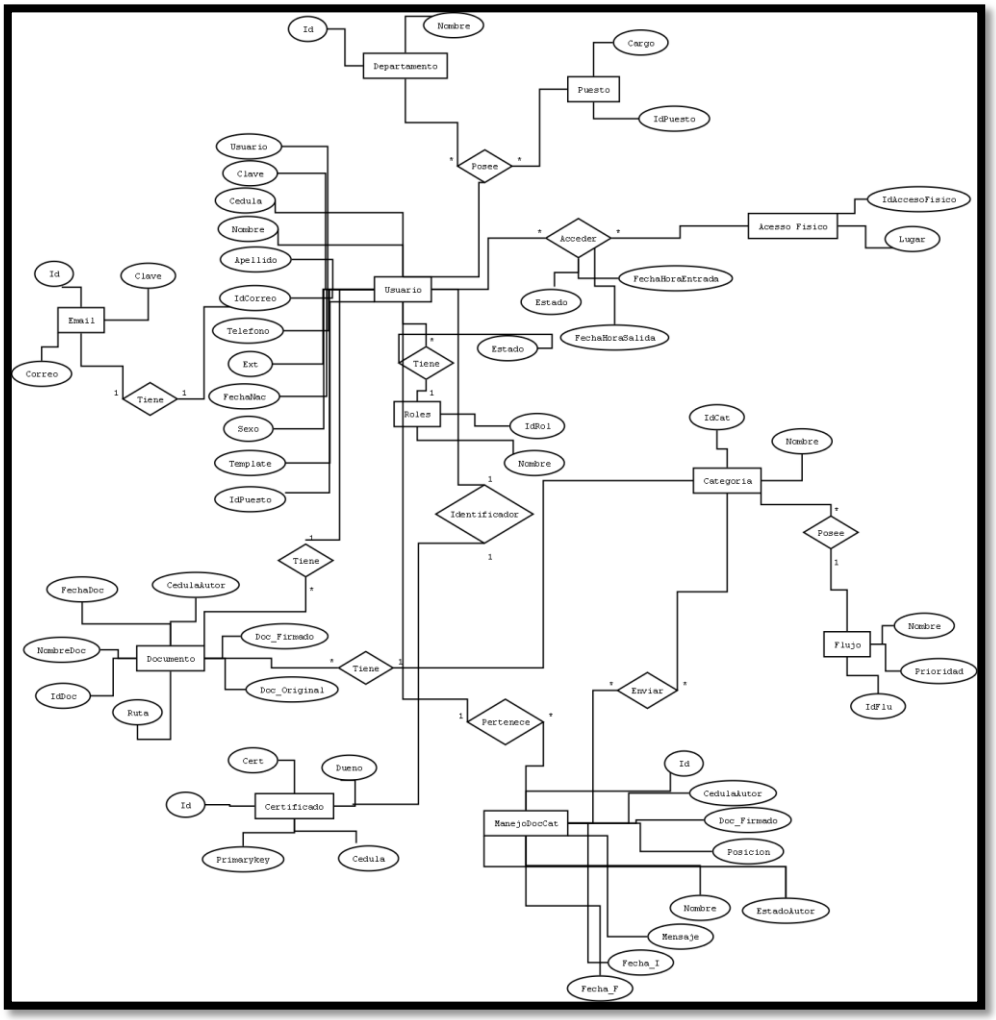


Figura 29: Modelo de Datos

En el modelo de datos de la Figura 30 se muestra la base del sistema, compuesto principalmente de 9 tablas para el almacenamiento de la información del sistema, permitiendo independencia entre las tablas para futuras adiciones al sistema.

- **Modelo Funcionabilidad**

Se analizó las principales funcionalidades que se ejecutan en el sistema para el cumplimiento del objetivo que sería el correcto traslado del documento dentro de la organización. Se muestra en la figura 31 y la figura 32

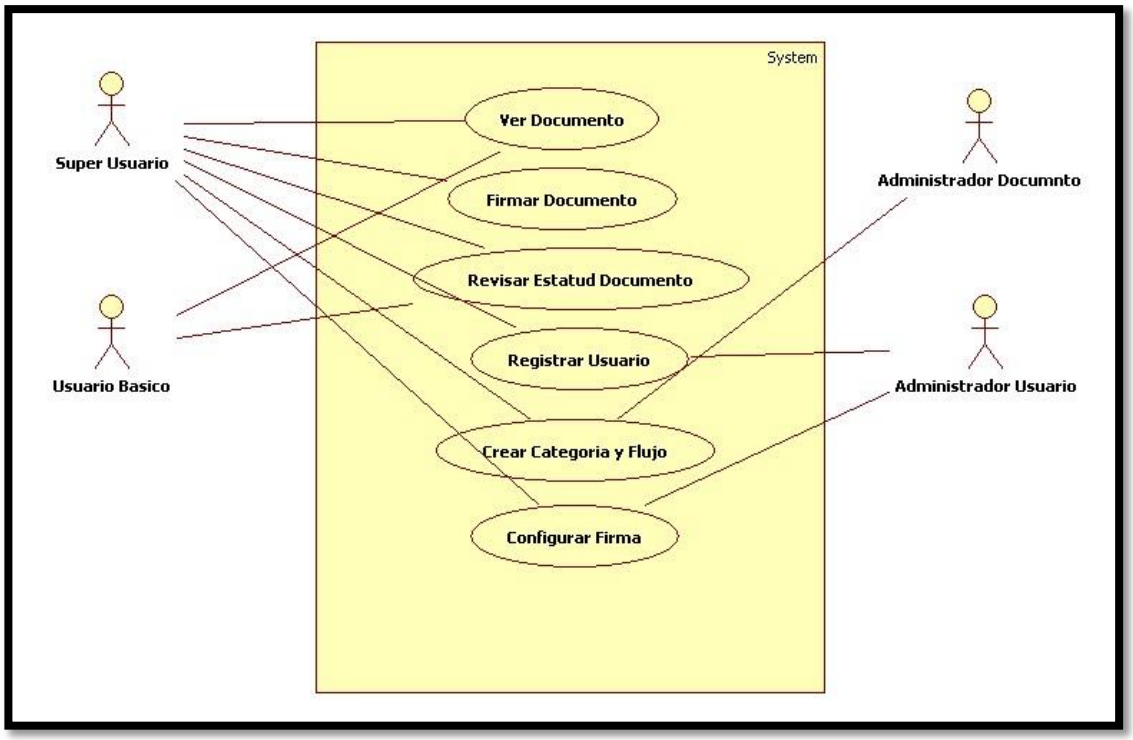


Figura 30: Caso de Uso nivel 1

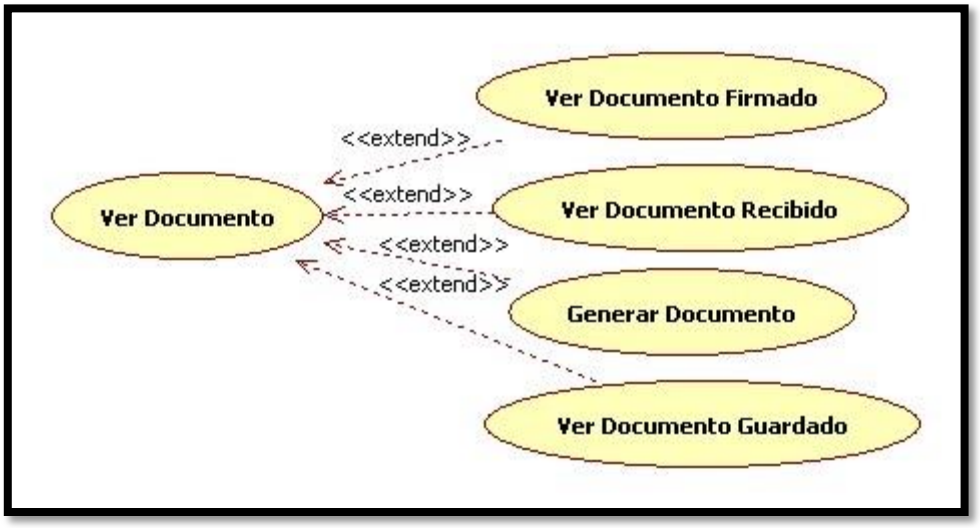


Figura 31: Caso de Uso nivel 2

Tabla 9 Caso de Uso: Ver documentos recibidos

Caso de uso 1	
Nombre	Ver documento recibido.
Actor	Usuario
Descripción	Puedo observar los documentos que ha recibido y que esperan por su aprobación o rechazo.
Pre-condición	Debe autenticarse en el sistema.

Tabla 10 Caso de Uso: Ver documentos firmados

Caso de uso 1.1	
Nombre	Ver documentos firmados
Actor	Usuario
Descripción	Puede observar los documentos que ha firmado
Pre-condición	Debe autenticarse en el sistema.

3.3.2. Iteración 1: Prototipo de Interfaz de Usuario

Esta iteración consiste en realizar un prototipo de interfaz para todas las vistas de la aplicación que permita al usuario un manejo sencillo e intuitivo.

Planificación

En la tabla 11 se muestran las historias de usuario desarrolladas en esta iteración.

Tabla 11 Historias de Usuario (Iteración 1: Prototipo de Interfaz de Usuario)

Id	Fecha	Requerimiento	Tipo
002		Crear las interfaces para el sistema	Nueva
003		Aplicar la plantilla CSS a las interfaces	Nueva
004		Diseñar un logo	Nueva
005		Determinar la disposición de elementos en la interfaz	Nueva

Diseño

En la fase de diseño se desarrolla un prototipo de interfaz, donde se establece la disposición de los elementos básicos que conforman todas las vistas de la aplicación.

Para el prototipo se utilizan diferentes tonalidades de gris ya que transmite neutralidad, lujo y elegancia, y permiten al usuario navegar por la aplicación sin causarle fatiga.

En el caso del texto, el tipo de letra a utilizar es Arial con un tamaño variable entre 11 y 24 pixeles dependiendo del tipo de título. Además se utilizan los siguientes colores:

- Naranja: para resaltar elementos de la interfaz al posar el mouse o ratón sobre ellos
- Gris oscuro: para los títulos
- Gris claro: para el texto

A continuación se describe y presenta el prototipo para la vista principal (ver Figura 33) de la aplicación:

- Menú Superior: Este elemento está ubicado en la parte superior de las vistas. Ocupa el 100% del ancho de la página web con una altura de 38 pixeles. Muestra el nombre de usuario (una vez ingresado al sistema) que cuenta con un pequeño menú desplegable (ver Figura 34) que permite ver el perfil del usuario, otras opciones para el usuario que tiene disponible y la opción de cerrar sesión; acceso rápido a los documentos del usuario, opciones de la aplicación y un campo de búsqueda general.
- El nombre de la aplicación es: "Sistemas de Control de Acceso y Firma Digital" El logo (ver Figura 35) fue diseñado de forma tal que representará el uso de la biometría por huella digital en el sistema.
- Menú Principal: ubicado a la derecha, contiene los campos de "Principal" que mostrará información general del sistema, "Usuarios" para ver información de todos los usuarios del sistema con opción de modificar, eliminar o registrar un usuario nuevo, "Documentos" para realizar el envió o verificar el estatus de documentos enviados previamente, "Flujo" permite visualizar los flujos de documentos creados o crear uno propio, "Categoría" para visualizar las

categorías de documentos creadas o crear una categoría y “Control de Acceso” para información sobre los controles de accesos físicos y al sistema de un usuario particular o en una fecha determinada.

- Barra de accesos directos: está compuesto por el logo del sistema ubicado del lado izquierdo de la página web y del lado derecho se encuentran las funcionalidades más importantes que conforman el sistema (“Usuario”, “Documentos”, “Flujos” y “Categorías”).
- Contenido: ubicado en el centro de la página, mostrará la información de acuerdo a las opciones elegidas del “Menú Superior”, “Menú Principal” o a través de la “Barra de Accesos Directos”.

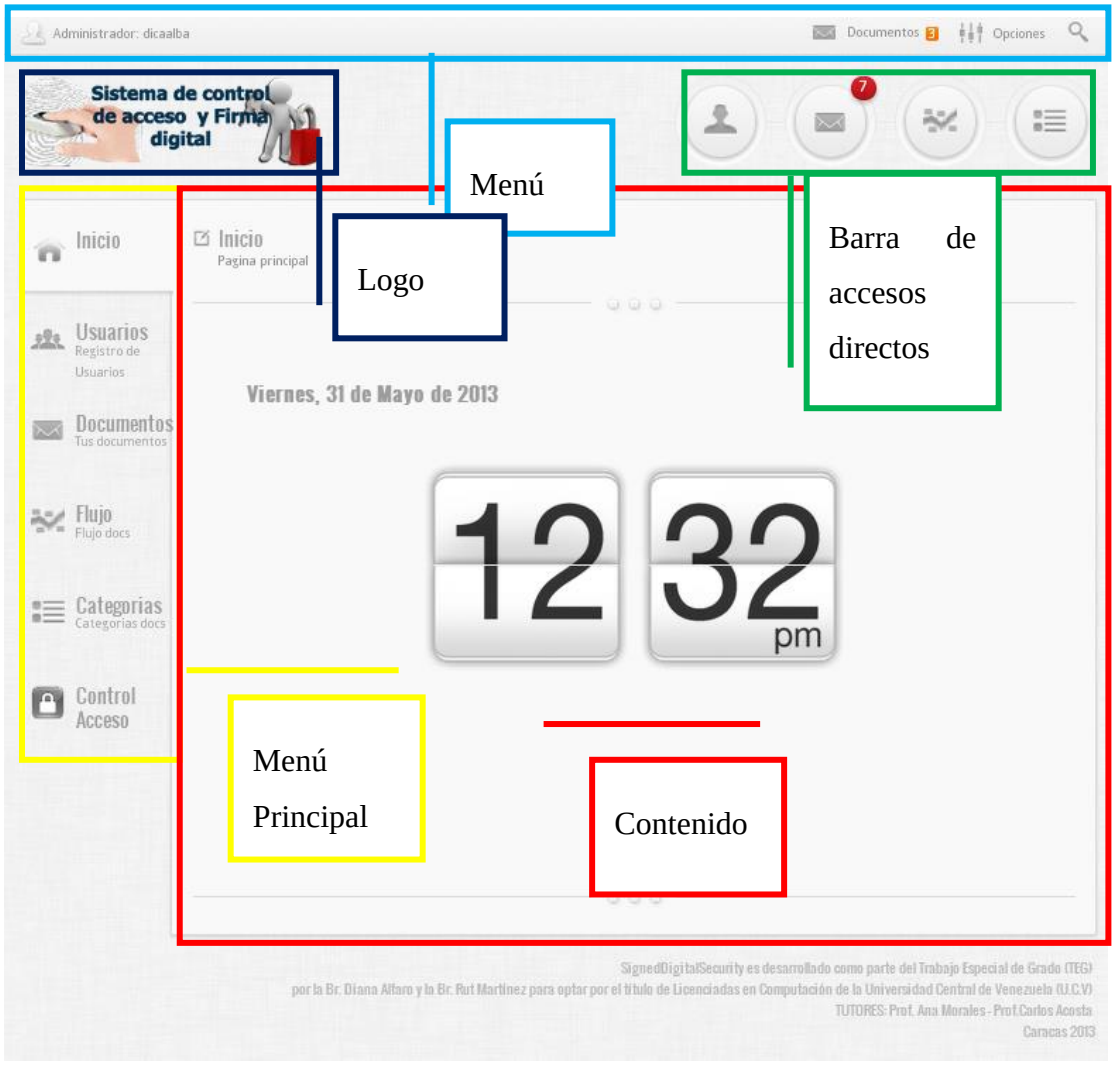


Figura 32: Vista Principal del Prototipo de Interfaz

Figura 33: Vista del menú desplegable sobre el nombre de usuario

Figura 34: Logo del Sistema

Codificación

En la Figura 36 se muestra parte de la codificación correspondiente a la plantilla utilizada en todas las vistas de la interfaz de usuario, cumpliendo con el diseño propuesto, se observa las secciones o partes modeladas con campos DIV y con el uso de las nuevas etiquetas que provee HTML5 (<header>, <aside>, <footer>).

```
<!DOCTYPE HTML>
<html lang="es">
<head>
<body>

        <div class="top_panel">
        <div class="wrapper">
            <div class="user">
                <img src="../Images/user_avatar.png" alt="user_avatar" class="user_avatar" />
                <span class="label"><%=usuario %></span>

                <div class="top_tooltip">
            </div>
            <div class="top_links">
        </div>
    </div>

    <header class="main_header">
        <div class="wrapper">
    </header>



    <div class="wrapper contents_wrapper">

    <aside class="sidebar">


        <div class="contents">
            <div class="grid_wrapper">
        </div>
    </div>
    <footer>
</body>
</html>
```

Figura 35: Código de la Vista Principal del Sistema

Pruebas

Luego de realizar pruebas de funcionalidad y usabilidad en conjunto con los tutores del Trabajo Especial de Grado y algunos usuarios del sistema, se formularon las siguientes sugerencias para el prototipo de interfaz:

- Las metáforas del menú principal deben ser acordes a lo que se pretende reflejar
- El diseño de los formularios debe mantenerse uniforme a lo largo del sistema.
- Cambiar el color gris claro del texto a un color gris más oscuro con el fin de que las capturas del sistema que se utilicen para el presente documento sean más legibles.
- Eliminar la barra de accesos directos por resultar redundante
- Centrar el logo del sistema a fin de que no quede un espacio vacío motivado a la eliminación de la barra de accesos directos.

3.3.3. Iteración 2: Interfaz de Usuario Final

Esta iteración consiste en aplicar las recomendaciones obtenidas sobre los prototipos de interfaces de usuario descritos en la Iteración 0.

Planificación

En la tabla 12 se muestran las historias de usuario a desarrollar en esta iteración.

Tabla 12 Historias de Usuario (Iteración 2: Interfaz de Usuario Final)

Id	Fecha	Requerimiento	Tipo
006		Crear las interfaces para el sistema	Renovada

Diseño

Para el diseño de la interfaz, se toma en consideración las observaciones realizadas en la iteración 1. Se cambian las metáforas del menú principal, se realiza la unificación en el diseño de los formularios, el cambio de color del texto, así como la eliminación de la barra de accesos directos.

El resultado de las modificaciones realizadas puede observarse en la figura a continuación (Ver Figura 37)

Figura 36: Vista Final del Prototipo de Interfaz

Codificación

Respecto a la codificación los cambios fueron únicamente a nivel de hojas de estilo para un color gris oscuro (“color: #0A0A0A;”) para las fuentes del sistema.

Pruebas

En conjunto con los usuarios del sistema se obtuvo una aceptación positiva de la interfaz mostrada.

3.3.4. Iteración 3: Módulo de Acceso

El objetivo de esta iteración es realizar las funciones de acceso a la aplicación, como el registro, la autenticación y recuperación de usuario y/o clave del usuario.

Planificación

Las historias de usuario para esta iteración son presentadas en la tabla 13.

Tabla 13 Historias de Usuario (Iteración 3: Módulo de Acceso)

Id	Fecha	Requerimiento	Tipo
007		Registro de Usuario	Nueva
008		Agregar la funcionalidad de registro de usuario	Nueva
009		Enviar correo para recuperar usuario y/o clave	Nueva
010		Agregar usuarios al sistema	Nueva
011		Editar datos personales	Nueva
012		Cambiar clave	Nueva

Diseño

El diseño se basa en la Figura 38, que muestra la disposición y relaciones de los objetos principales definidos en el módulo de acceso de la aplicación.

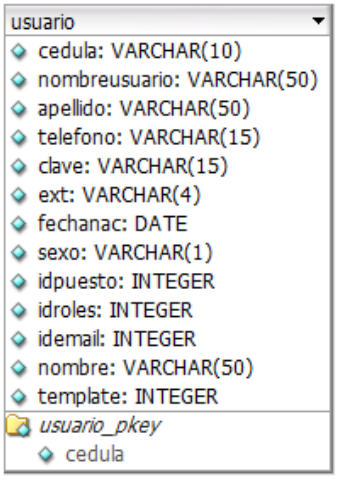


Figura 37: Diseño de la Base de Datos para el Módulo de Acceso

Cuando el usuario desee ingresar al sistema, debe colocar su huella dactilar(ver Figura 39) para que el sistema se encargue de buscar su huella en los registros de la base de datos, de no existir dicha huella el sistema mostrará un mensaje de "Huella NO identificada" (ver Figura 40) en caso contrario, el sistema mostrará un mensaje de "Huella identificada" (ver Figura 41) y abrirá una nueva ventana en el navegador donde ahora el usuario deberá ingresar su nombre de usuario y clave(ver Figura 42). En caso de que exista discrepancia de los datos se emite un mensaje de "Usuario y/o clave no válidos". Luego de validar la información se procede a cargar los datos y configuración (ver Figura 37).

Figura 38: Módulo de Acceso de Usuario

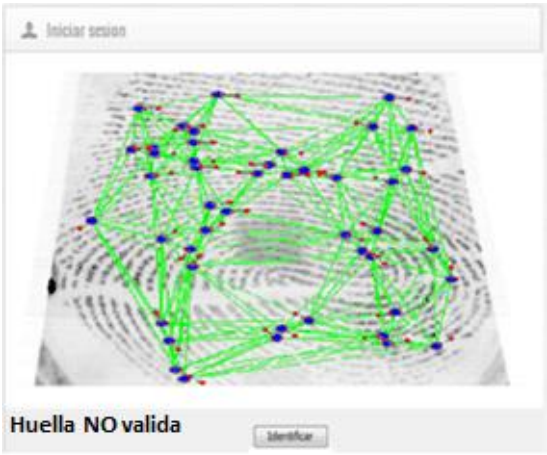


Figura 39: Módulo de Acceso de Usuario (Huella no válida)

Figura 40: Módulo de Acceso de Usuario (Huella válida)

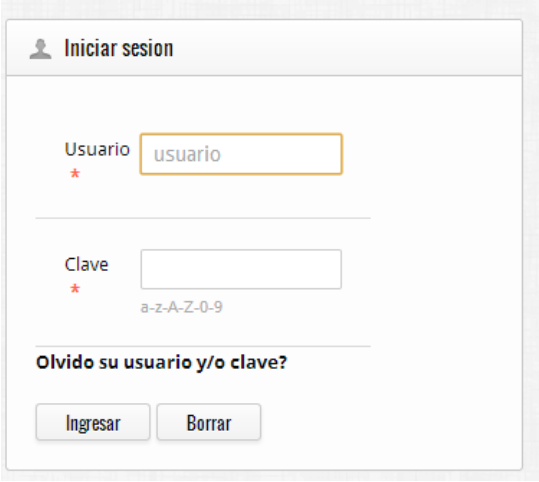


Figura 41: Módulo de Acceso de Usuario (Solicitud de usuario y clave)

Para registrar un nuevo usuario se desarrolla una sección en donde se solicita que sean completados todos los campos requeridos para el registro del nuevo usuario tales como: cédula, nombres, apellidos, correo, cargo, departamento, tipo de Usuario en los que se puede elegir entre: Usuario (Usuario básico), Administrador, Administrador de usuarios y Administrador de documentos (ver Figura 43). Una vez realizado el registro se procederá a enviar un email de confirmación de registro con el enlace al que tiene que acceder para ingresar al sistema y una clave autogenerada por la aplicación (ver Figura 44). El usuario al ingresar al enlace enviado por email deberá proceder a colocar su huella digital del nuevo usuario y proceder a dar clic en el botón “Registrar” (ver Figura 45) en el que debemos indicar la cédula, la clave enviada por correo, la clave nueva y un nombre de usuario para usar en la aplicación. Luego de registrarse en el sistema, el usuario podrá modificar y agregar nueva información de sus datos personales, como por ejemplo: fecha de nacimiento, sexo, entre otros.

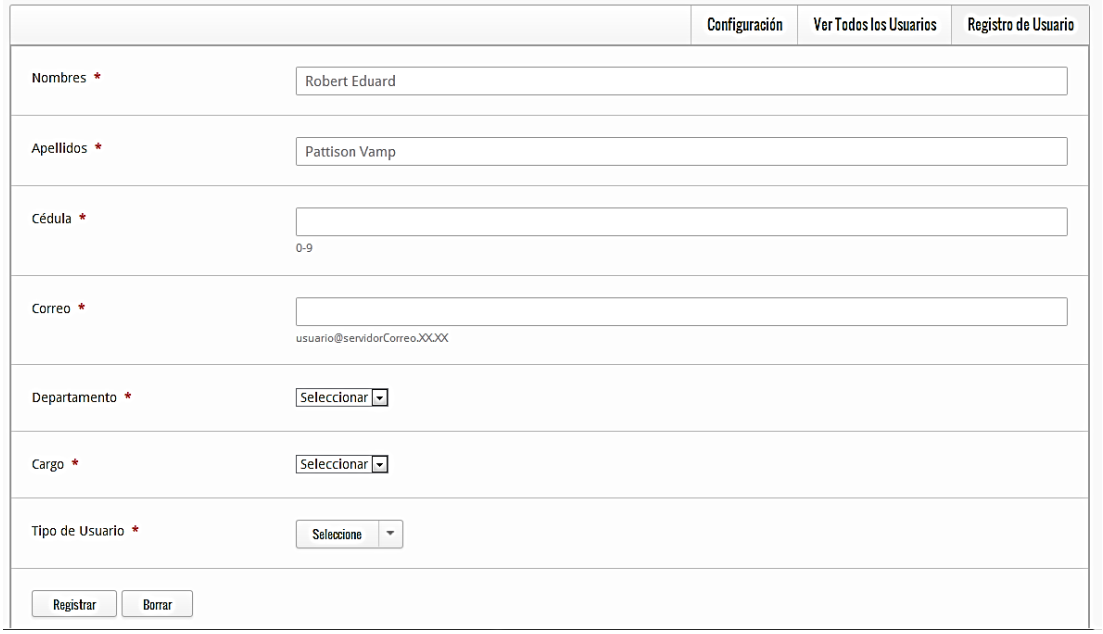


Figura 42: Vista para el Registro de Usuario

signed.digital.teg@gmail.com

para correo@servidorCorreo.com

Este es un correo de verificación

Por favor haga clic en el siguiente enlace para seguir con la registro de sus datos.

Enlace

Usuario con cédula n° 18633030

Clave generada por el sistema QPMJFFYT

Figura 43: Envio de email de confirmación de registro

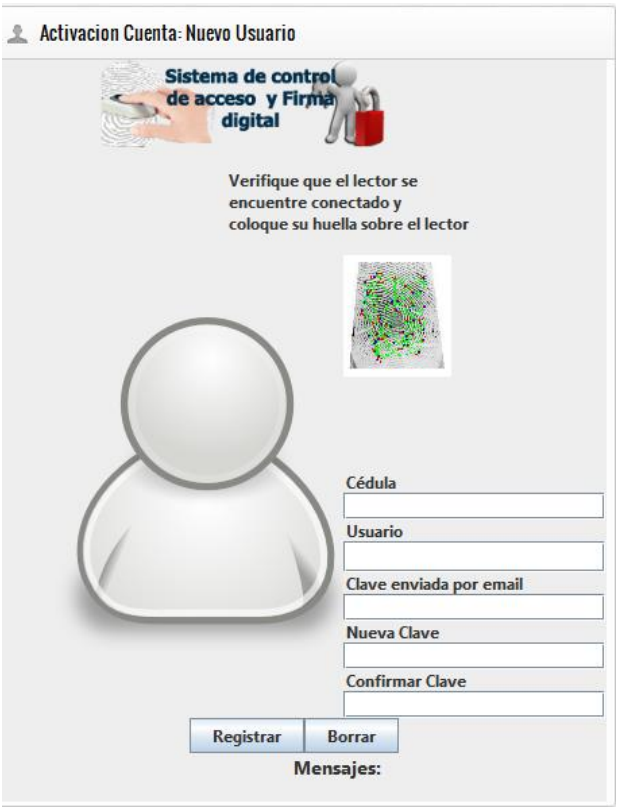


Figura 44: Registro de huella y otros datos del usuario

Codificación

Para crear un nuevo registro, se desarrolla una función que almacena los datos solicitados en la tabla "Usuario" de la base de datos. En la Figura 46 se presenta una porción del código de la función de almacenamiento.

```
String sql = "";
String cedula= request.getParameter("cedula");
String nombre= request.getParameter("nombre");
String apellido= request.getParameter("apellido");
String correo= request.getParameter("correo");
String cargo= request.getParameter("cargo");
String dpto= request.getParameter("departamento");
String tipoUsuario= request.getParameter("tipoUsuario");

int idcorreo=0;
aleatoria=getCadenaAlfanumAleatoria(8);
bd.Conexion();

try {
  PrintWriter out = response.getWriter();
  String sqlCorreo="INSERT INTO email (correo) VALUES ('"+correo+"')";//primero se gurada el correo

  Statement setenciaCorreo= bd.instruccion;
  arrojo = setenciaCorreo.executeUpdate(sqlCorreo);

   Statement inst=bd.canal.createStatement();
   String idmanejo="SELECT last_value FROM email_id_seq"; //obtengo el id  tabla correo
   ResultSet t= inst.executeQuery(idmanejo);

      if(t.next())
          idcorreo= t.getInt("last_value");

  sql = "insert into usuario (cedula, nombre, apellido, idemail, idpuesto, idroles, "
              +" clave) values (" + "'" + cedula + "', '" +nombre + "','" + apellido + "',"
              + "'" + idcorreo + "'," + "'" + cargo + "'," + "'" + tipoUsuario + "'," + "'" + aleatoria + "'" +")";

  Statement setencia= bd.instruccion;
  arrojo = setencia.executeUpdate(sql);

} catch (SQLException e) {
   e.printStackTrace();
}finally{
   bd.cerrarConexion();
}
```

Figura 45: Código para registrar un nuevo usuario

También se tomó en cuenta que el usuario olvidará su usuario y/o clave y deseará recuperarla (s), y para resolver dicho caso se creó un enlace con el nombre de "Olvido su usuario y/o clave?" (Ver Figura 47) en el que podrá recuperarlos ya que el sistema obtuvo previamente su huella digital y con dicha información puede obtener los otros datos del usuario realizando una consulta a la base de datos.

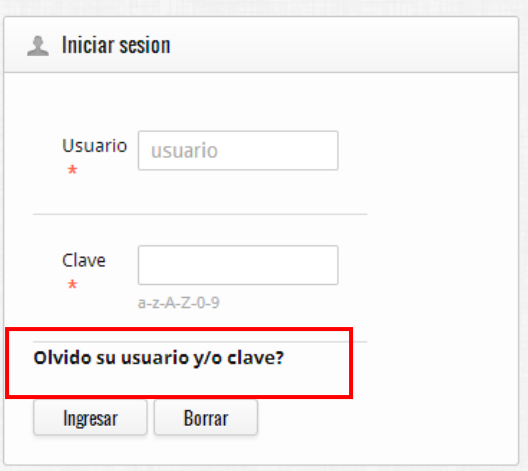


Figura 46: Recuperar clave

Al usuario le llega su nombre de usuario y la clave al correo (Ver Figura 48), con la que posteriormente en conjunto con su huella dactilar podrá ingresar a la aplicación. Una vez el usuario ingrese al sistema, podrá ingresar a la sección "Mi Perfil" donde podrá cambiar la clave por otra si así lo desea (Ver Figura 49).

Para el envío de correo se creó un Mailer con la configuración del correo (SMTP, usuario, clave, puerto y dominio).

signed.digital.teg@gmail.com

para correo@servidorCorreo.com

Su nombre de usuario es: dicaalba

Su clave es: QPMJFFYT

Figura 47: Correo que se envía para notificar cambio de clave

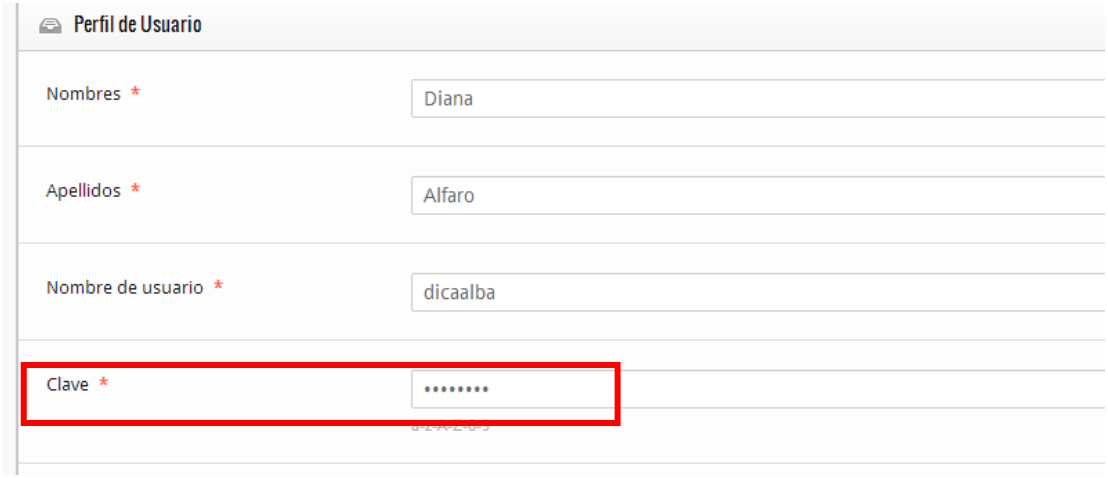


Figura 48: Cambiar clave

En caso de cambio de clave por parte del usuario, el sistema la compara con la que se encuentra en la tabla sesiones, de ser igual, actualiza la tabla y almacena la nueva clave.

La figura 50 muestra mediante un diagrama BPM el proceso de cambio de clave.

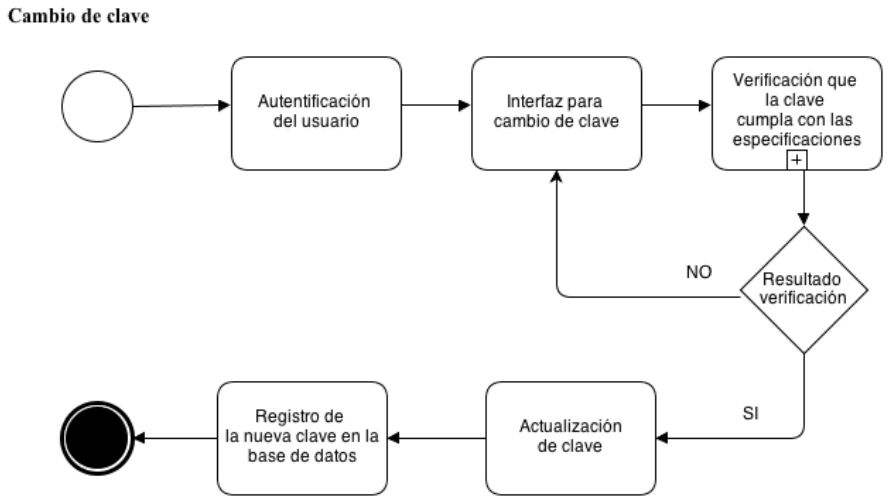


Figura 49: Diagrama BPM de cambio de clave

Pruebas

En la carga de datos y configuración de usuario, se lleva a cabo la comprobación de los siguientes aspectos asegurándose que los datos coincidan con la información almacenada anteriormente:

- Nombre de usuario
- Datos personales

En el caso de la recuperación de usuario y/o clave se confirma el envío correcto al correo con el nombre de usuario y la clave solicitada por el usuario. Se realizaron pruebas a la funcionalidad de cambio de clave verificando que se ejecute satisfactoriamente el cambio deseado.

3.3.5. Iteración 4: Creación y manejo de sesiones de usuarios

En esta iteración se crean y manejan las sesiones de los usuarios.

Planificación

En la Tabla 14 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 14 Historias de Usuario

(Iteración 4: Creación y manejo de sesiones de usuarios)

Id	Fecha	Requerimiento	Tipo
013		Manejo de sesiones	Nueva

Diseño

Para manejar pertinentemente las sesiones de los usuarios se utilizan los campos nombre de usuario, tipo de usuario, huella y clave de la tabla usuarios.

Codificación

Para lograr el objetivo en ésta iteración se creó la vista para la sesión del usuario. Cuando el usuario desea entrar en el sistema e introduce su huella digital, su nombre de usuario y su clave de acceso, éstas se comparan con las que se tienen guardadas en la base de datos, si coinciden, se le otorga acceso a las funcionalidades que posee el usuario. (Ver ejemplo de código en la Figura 51)

```
String usuario = request.getParameter("usuario");

HttpSession sesionOk = request.getSession();
sesionOk.setAttribute("usuario",usuario);
```

Figura 50: Identificación por tipo de usuario

3.3.6. Iteración 5: Generación de reportes de documentos enviados

En esta iteración se realiza la generación de reportes en un documento con formato PDF para los documentos enviados por el usuario.

Planificación

En la Tabla 15 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 15 Historias de Usuario (Iteración 5: Generación de reportes de documentos enviados)

Id	Fecha	Requerimiento	Tipo
014	01/09/2013	Generación de documento pdf con la información de los documentos enviados por el usuario	Nueva

Diseño

Para el diseño de esta iteración se presenta una lista desplegable donde se puede elegir qué tipo de estatus quiere que se genere (aceptado, rechazado, todos, en espera) donde el estatus "todos" indica que para la generación del reporte se tomará en cuenta todos los documentos enviados (Ver figura 52).

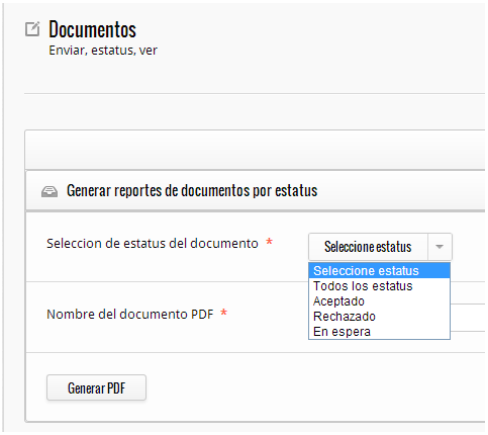


Figura 51: Diseño de la vista de generación de reportes

Además el usuario puede especificar el nombre del documento a generar. Luego de pulsar el botón correspondiente genera reporte se descargara automáticamente el PDF generado (Ver figura 53 y figura 54).

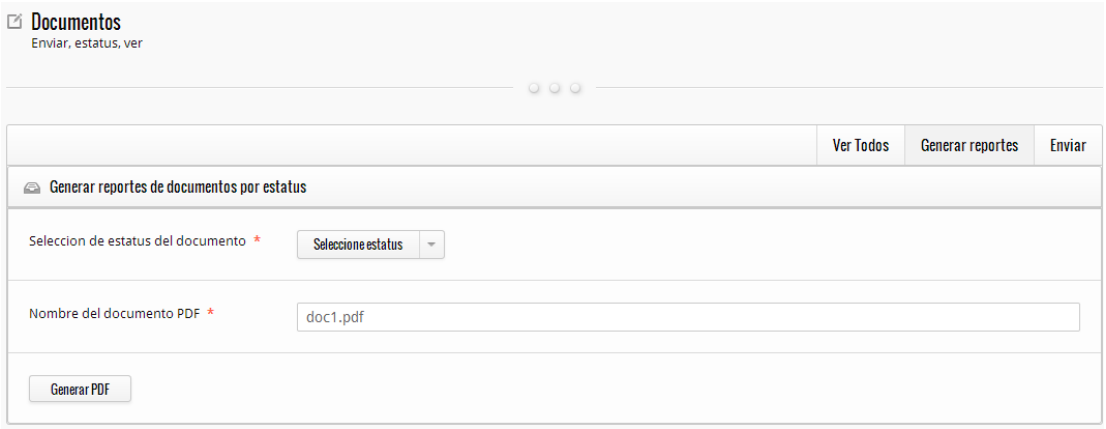


Figura 52: Pantalla para indicar los datos necesarios para la generación de reportes

PRUEBADOC

Documentos

Fecha Inicio	Nombre del Documento	Categoria del Documento	Posición del Flujo Receptor	Departamento del Receptor	Cargo del Receptor	Estatus	Observaciones	Fecha Finalización
2013-09-22 16:46:11.403	PDF prueba.pdf	memo	1	informatica	gerente	aceptado	El documento ha sido aceptado por 0 personas	2013-09-22 17:44:07.717
2013-09-22 23:06:21.655	1028.pdf	diana	2	informatica	gerente	aceptado	El documento ha sido aceptado por 1 personas	2013-09-24 01:16:19.304
2013-09-23 19:33:17.883	prueba.pdf	diana	2	informatica	gerente	aceptado	El documento ha sido aceptado por 1 personas	2013-09-24 01:22:41.763
2013-09-23 19:33:17.883	prueba.pdf	diana	2	informatica	gerente	aceptado	El documento ha sido aceptado por 1 personas	2013-09-24 01:22:41.763

Figura 53: Ejemplo de documento PDF generado

Codificación

Para el desarrollo de esta iteración para la generación de reportes de estatus de documentos enviados se utiliza la librería ITEXT para JAVA. Que nos permite generar documentos con formato PDF dinámicamente.

De los datos obtenidos del formulario de la figura 53 se realiza la respectiva consulta a la base de datos, donde los datos que nos retorna la consulta serán los que se ingresarán en el documento PDF a generar, para ello se dibuja una tabla y se insertan en ella el resultado de la consulta a la base de datos (Ver figura 55).

```
        //Iterar Mientras haya una fila siguiente
    while (rs.next()) {           //Agregar 9 celdas

        String lugar ="";
        if(rs.getInt("posicion")==1){
            lugar="0";
        }else{
            lugar=Integer.toString(rs.getInt("posicion")-1);

        }
            cell = new PdfPCell(new Paragraph(rs.getString("fecha_i"),fuente8 )); //Fecha inicio
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombre_doc"),fuente8));//Nombre del documento
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombredoc"),fuente8));//Nombre categoria
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("posicion"),fuente8)); //posicion
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombre_departamento"),fuente8)); // Nombre departamento
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombre_cargo"),fuente8)); //Nombre cargo
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("estado_autor"),fuente8) );//Estatus
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph("El documento ha sido aceptado por "+lugar+" personas",fuente8)); //Observaciones
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("fecha_f"),fuente8)); // Fecha fin
            tabla.addCell(cell);


    }
```

Figura 54: Creación de una tabla en un documento PDF

Luego de completado el documento PDF se procede a su exportación para la posterior descarga por parte del usuario. (Ver figura 56).

```
//Metodo principal del ejemplo
public void GenerarPDF(String titulo, String nomPDF, String statusDoc) {
    strRotuloPDF = titulo;
    strNombreDelPDF = nomPDF;
    try{

        //Hoja tamanio carta, rotarla (cambiar a horizontal)
        document = new Document(PageSize.LETTER.rotate() );

        writer = PdfWriter.getInstance(document,
                // direccionar el PDF-stream a un archivo
                new FileOutputStream(strNombreDelPDF));

        agregarMetaDatos(document);

        document.open();


        agregarContenido(document);

        document.close();

        System.out.println("Se ha generado el PDF: "+ strNombreDelPDF);

    } catch (Exception e)  {
        e.printStackTrace();
    }
}
```

Figura 55: Creación de un documento PDF

Pruebas

Las pruebas de aceptación que se realizaron para esta iteración incluye la validación del formulario (Ver figura 53), así como la correcta generación de reportes para cada uno de los estatus de documentos seleccionados y el formato en que estos datos son mostrados en el documento.

3.3.7. Iteración 6: Carga de archivo de certificado digital

En esta iteración se realiza la carga al servidor del archivo de certificado digital con extensión PFX (estándar PKCS12).

Planificación

En la Tabla 16 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 16 Historias de Usuario (Iteración 6: Carga de archivo de certificado digital)

Id	Fecha	Requerimiento	Tipo
014		Carga de archivo de certificado digital	Nueva

Diseño

Para realizar la firma digital de documentos con formato PDF se modifica el modelo de datos y se crea una nueva tabla llamada certificados. Dicha tabla cuenta con los campos de cédula de usuario que hace referencia al campo cedula de la tabla usuarios y el campo certificado, el cual representa el certificado digital de cada usuario, éste archivo debe estar en formato con extensión PFX (estándar PKCS12).

Codificación

Para lograr el objetivo en ésta iteración se creó la vista para la carga del certificado digital, el cual es un archivo con extensión .pfx, además de ingresar los datos correspondientes al nombre del propietario del certificado y la clave privada de éste. (Ver ejemplo de código en la Figura 57)

Figura 56: Carga de archivo de certificado digital

Pruebas

Las pruebas de aceptación que se hicieron para verificar que el certificado digital se cargara pertinentemente fue la siguiente: Se trató de cargar un archivo con distinta extensión a .pfx obteniendo una respuesta por parte del sistema de "Formato de Certificado no válido (debe ser .pfx)".

3.3.8. Iteración 7: Firma digital de documentos pdf

En esta iteración se realiza la firma digital de un documento con formato PDF.

Planificación

En la Tabla 17 se observan las historias de usuarios desarrolladas en ésta iteración.

Tabla 17 Historias de Usuario
(Iteración 7: Firma digital de documentos)

Id	Fecha	Requerimiento	Tipo
015		Firma Digital de documentos con formato .pdf	Nueva

Diseño

Para realizar la firma digital de documentos con formato PDF se requiere de los datos del certificado digital de la persona que va a firmar el documento. La información se encuentra almacenada en un archivo con extensión .pfx que corresponde al campo "certificado" de la tabla llamada certificados.

Codificación

Para lograr el objetivo en ésta iteración se procede a tomar el documento con extensión .pdf y los datos del certificado de usuario que va a firmar el documento, los cuales se encuentran almacenados en la tabla certificados, la firma digital de un documento puede ocurrir de dos maneras: al momento de enviar un documento a un flujo de personas determinados, el documento es firmado previamente al envió.

La otra manera es cuando las personas que pertenecen a un flujo firman un documento que les ha sido enviado, estando con ello de acuerdo con el documento (Ver ejemplo de código en la Figura 58)

```
KeyStore ks = KeyStore.getInstance("pkcs12");
System.out.println("Segundo paso para leer doc");

ks.load(new FileInputStream("cert.pfx"), privateKey.toCharArray());
System.out.println("Tercer paso para leer doc");

String alias = (String)ks.aliases().nextElement();
System.out.println("Cuarto paso para leer doc");

PrivateKey key = (PrivateKey)ks.getKey(alias, privateKey.toCharArray());
System.out.println("Quinto paso para leer doc");

Certificate[] chain = ks.getCertificateChain(alias);
System.out.println("Sexto paso para leer doc");

// Recibimos como parámetro de entrada el nombre del archivo PDF a firmar
PdfReader reader = new PdfReader(args[0]);
System.out.println("Leyo el documento");

FileOutputStream fout = new FileOutputStream("firma.pdf");
System.out.println("Creo el documento auxiliar");

// Añadimos firma al documento PDF
PdfStamper stp = PdfStamper.createSignature(reader, fout, '?');
PdfSignatureAppearance appearance = stp.getSignatureAppearance();
PdfSignatureAppearance sap = stp.getSignatureAppearance();
sap.setCrypto(key, chain, null, PdfSignatureAppearance.WINCER_SIGNED);
sap.setReason("Firma PKCS12");
sap.setLocation("Caracas");

    appearance.setSignatureGraphic(Image.getInstance(RESOURCE));
   // appearance.setRenderingMode(PdfSignatureAppearance.RenderingMode.GRAPHIC);

// Añade la firma visible. Podemos comentarla para que no sea visible.
sap.setVisibleSignature(new Rectangle(100,100,200,200),1,null);
stp.close();
```

Figura 57: Firma digital de documentos

En la Figura 58 “cert.pfx” es el nombre de un certificado digital utilizado para realizar pruebas y “firma.pdf” es el nombre del archivo que se genera con una firma digital.

Pruebas

Las pruebas de aceptación que se hicieron para verificar que la firma de documentos estuviese trabajando pertinentemente fue la siguiente:

Se realizó el envió de un documento pdf a un flujo creado previamente el cual no está firmado, luego del envió se pudo comprobar que el documento enviado se encuentra firmado por quien envió el documento, en este caso el autor. (Ver figura 59)

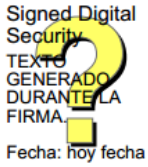


Figura 58: Firma digital de un documento pdf

3.3.9. Iteración 8: Verificación y validación de firmas digitales

En esta iteración se realiza la verificación y validación firma digital en un documento con formato PDF.

Planificación

En la Tabla 18 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 18 Historias de Usuario (Iteración 8: Verificación y validación de firmas digitales)

Id	Fecha	Requerimiento	Tipo
016		Verificación y validación firma digital en un documento con formato PDF.	Nueva

Diseño

Para realizar la verificación y validación de una firma digital, se procede a comprobar que la firma digital que tiene el documento PDF sea proveniente de un certificado digital reconocido, es decir, que la autoridad certificadora de dicho certificado digital reconozca dicho certificado.

Codificación

Para lograr el objetivo en ésta iteración comprueba que el certificado sea válido, de ser así se puede obtener los datos del certificado, como por ejemplo el nombre del dueño de ese certificado de lo contrario el certificado no es válido (Ver ejemplo de código en la Figura 60)

```
Random rnd = new Random();
KeyStore kall = PdfPKCS7.loadCacertsKeyStore();

String pdfAVerificar = request.getParameter("archivo");

PdfReader reader = new PdfReader(pdfAVerificar);
AcroFields af = reader.getAcroFields();
ArrayList names = af.getSignatureNames();
for (int k = 0; k < names.size(); ++k) {
   String name = (String)names.get(k);
   int random = rnd.nextInt();
   FileOutputStream out = new FileOutputStream("revision_" + random + "_" + af.getRevision(name) + ".pdf");

   byte bb[] = new byte[8192];
   InputStream ip = af.extractRevision(name);
   int n = 0;
   while ((n = ip.read(bb)) > 0)
   out.write(bb, 0, n);
   out.close();
   ip.close();

   PdfPKCS7 pk = af.verifySignature(name);
   Calendar cal = pk.getSignDate();
   Certificate pkc[] = pk.getCertificates();
   Object fails[] = PdfPKCS7.verifyCertificates(pkc, kall, null, cal);
   if (fails == null) {
       System.out.print(pk.getSignName());
   }
   else {
       System.out.print("Firma no válida");
   }
   File f = new File("revision_" + random + "_" + af.getRevision(name) + ".pdf");
   f.delete();
}
```

Figura 59: Verificación y validación de firmas digitales

Pruebas

Las pruebas de aceptación se realizaron para verificar que el certificado de un documento sea válido, sin embargo los certificados que se generaron para hacer las respectivas pruebas con el sistema han sido creados utilizando OpenSSL por lo que no tienen validez legal a diferencia de los certificados emitidos por entidades certificadoras conocidas, como por ejemplo: Verisign.

3.3.10. Iteración 9: Adición de múltiples firmas digitales a un documento

En esta iteración se realiza la verificación y validación firma digital en un documento con formato PDF.

Planificación

En la Tabla 19 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 19 Historias de Usuario (Iteración 9: Verificación y validación de firmas digitales)

Id	Fecha	Requerimiento	Tipo
017		Adición de múltiples firmas a un documento PDF	Nueva

Diseño

Para realizar la adición de más de una firma digital en un documento con formato PDF se hace uso del código empleado en la iteración anterior (Ver Iteración 9: Verificación y validación de firmas digitales) con la excepción de que se ubica la imagen de la firma digital en una posición del documento de acuerdo al número de firmas que existen previamente, con el fin de evitar sobreponer u ocultar las otras firmas del documento.

Codificación

Para lograr el objetivo en ésta iteración se asignó un tamaño fijo a las firmas y la ubicación donde son colocadas se basan en el número de firma que representan (Ver ejemplo de código en la Figura 61)

```
// Añade la firma visible. Podemos comentarla para que no sea visible.

System.out.println("posicion"+ posicion);
java.util.Date fecha = new java.util.Date();
sap.setVisibleSignature(new Rectangle(inicio*=posicion,100,fin,200),1,null);
```

Figura 60: Adición de múltiples firmas a un documento

Pruebas

Las pruebas de aceptación que se hicieron para verificar que un documento pueda contener más de una firma, se basó en crear un documento el cual contendría al menos tres (3) firmas, ajustando el tamaño de cada una a fin de ocupar el menor espacio posible.

3.3.11. Iteración 10: Módulo de acceso

En esta iteración se realizan cambios a nivel de interfaz de usuario y código referente a como ingresar al sistema y al registro por primera vez de la huella digital del usuario

Planificación

En la Tabla 20 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 20 Historias de Usuario (Iteración 10: Módulo de Acceso)

Id	Fecha	Requerimiento	Tipo
018	21/08/2013	Inicio de sesión en el Sistema	Mejorada
019	21/08/2013	Enviar correo para recuperar clave	Mejorada
020	21/08/2013	Registro de Usuario	Mejorada

Diseño

En la iteración 1 se tenía una única interfaz donde era necesario colocar la huella y si ésta era correcta, procedía a solicitar nombre de usuario y clave registrados con esa huella. Por recomendaciones del usuario se divide la petición de información en dos ventanas, primero aparece una pantalla con un diseño de interfaz más sofisticado con mensajes de ayuda y una sección más pequeña donde se muestra la huella capturada (Ver figura 62)

Figura 61: Pantalla de captura de la huella digital

En este punto puede ocurrir dos cosas, la primera es que la huella no esté almacenada en la base de datos, dándole al usuario un mensaje de "Huella NO identificada" (Ver figura 63), en caso de que la huella si este almacenada en la BD se abrirá en el navegador una nueva ventana solicitando al usuario que ingrese su nombre de usuario y clave (Ver figura 64)

Figura 62: Pantalla de captura de una huella digital no válida

Figura 63: Pantalla de captura de una huella digital valida

Codificación

Para lograr el objetivo en ésta iteración se creó la vista para la captura de la huella en un applet, este applet mantiene una comunicación con el servidor de base de datos de la aplicación donde se encuentran almacenadas las huellas de los usuarios del sistema. En caso de encontrarse ocurrencia el applet redireccionará a la vista de petición de nombre de usuario y clave desarrollada en JSP (Ver figura 65), donde si además coinciden su nombre de usuario y su clave de acceso, si coinciden, se le

otorga acceso a las funcionalidades que posee el usuario. (Ver ejemplo de código en la Figura 66).

```
String dataString = "" + encontro+"."+cedula;//true/18030633
System.out.println(dataString);
// URL newURL = new URL("http://localhost:8080/SignedDigitalSecurity/HTML/japplet.jsp?" + dataString);
URL newURL = new URL("http://localhost:8080/SignedDigitalSecurity/HTML/registroCA.jsp?s=" + dataString);

// writeLog("Clave y huella valida.");
System.out.println("Clave y huella valida.");
getAppletContext().showDocument(newURL, "_blank");
```

Figura 64: Redirección de petición de applet a jsp

También se agregó la opción de recuperar el nombre de usuario y/o la clave. Mediante el enlace mostrado (Ver figura 67) al hacer clic se le enviara al correo registrado dicha información para su posterior acceso.

```
/***SELECT****/
try {

    BD bd = new BD();
    bd.Conexion();
    Statement instruccion=bd.instruccion;

    String sentencia="select usuario.cedula, roles.nombre "+
    "from usuario inner join roles on usuario.idroles=roles.idroles "+
    "where clave='"+id+"' and nombreUsuario='" + usuario + "'";

    tabla = instruccion.executeQuery(sentencia);

    if(tabla.next()){

        cedula=tabla.getString("cedula");
        tipoUsuario=tabla.getString("nombre");

        System.out.println("ci " +cedula);
        creado=true;
    }
    bd.Cerar();
    System.out.println("Conexion cerrada");
}catch(Exception e) {

    System.out.println(e);

}
```

Figura 65: Verificación de nombre de usuario y clave

```
sql = "Select * from v_emailCI where cedula='"+request.getParameter("cedula")+"'";

ResultSet tabla;
try {
    tabla = inst.executeQuery(sql);
    if(tabla.next()){
        usuarioClave(tabla.getString("correo"), tabla.getString("nombreusuario"), tabla.getString("clave"));
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Figura 66: Recuperar usuario y/o clave

Pruebas

Las pruebas de aceptación que se realizaron consistieron en verificar que efectivamente al dar clic en la opción para recuperar nombre de usuario y clave, se recibiera en el buzón de correo los datos requeridos para continuar con el proceso de inicio de sesión en el sistema.

3.3.12. Iteración 11: Envió de correo y almacenamiento de documentos

En esta iteración se realiza él envió de los documentos mediante correo electrónico y el almacenamiento de los documentos enviados como adjuntos en la bases datos.

Planificación

En la Tabla 21 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 21 Historias de Usuario (Iteración 11: Envió de email y almacenamiento de documentos)

Id	Fecha	Requerimiento	Tipo
020	25/06/2013	Envió de correo electrónico con documentos firmados adjuntados.	Nueva
021	25/06/2013	Almacenar correctamente los documentos originales y firmados en la base de datos	Nueva

Diseño

Para el proceso envió de documentos vía correo electrónico se utilizó la librería "javamail" para java y para tener un respaldo de dichos documentos que se envían como adjunto al correo, se guarda una copia en la base de datos. En el envío de documentos se diseñó la vista de la siguiente figura (Ver figura 68), en la que debe completar los campos para un correcto envío.

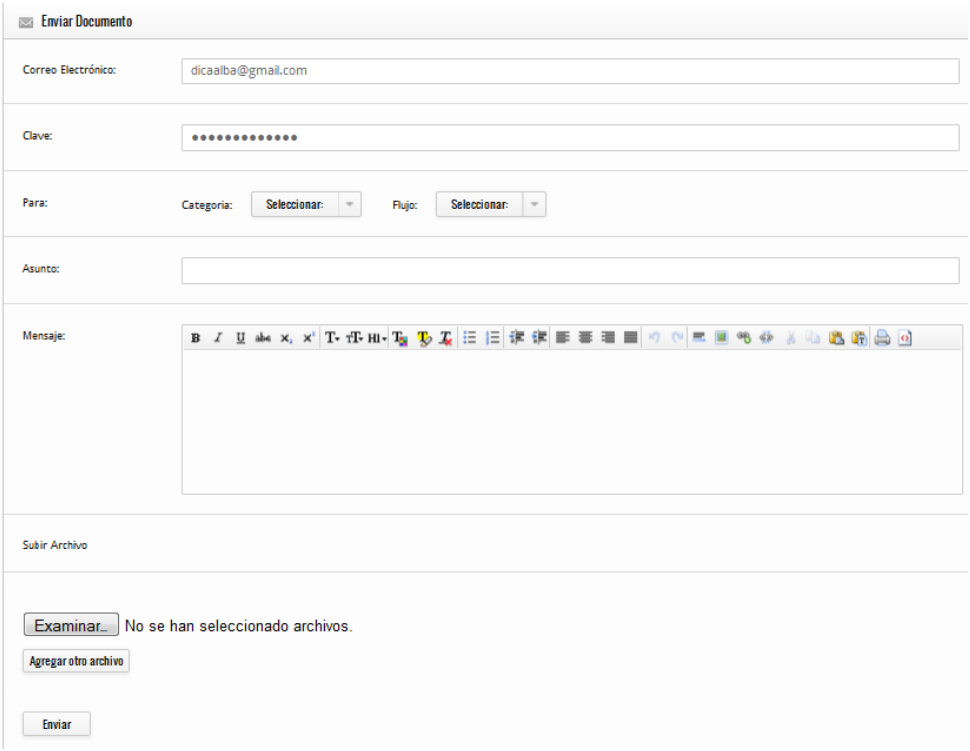


Figura 67: Vista para el envío de correo

Codificación

La librera de "javamail" proporciona métodos fáciles y rápidos a la hora de hacer uso del correo electrónico. Permite especificar los datos del servidor de correo, en este caso se ha creado una cuenta en el servidor de correo de gmail (signed.digital.teg@gmail.com) la cual es utilizada a modo de correo de administración del sistema para el envío de correos a los usuarios por parte del sistema. La figura siguiente (Ver figura 69) muestra el proceso de envió de un mensaje de correo electrónico, donde se especifica el cuerpo del mensaje a enviar (asunto, mensaje y archivos adjuntos si los hubiese) y la(s) persona (s) destinataria(s) del correo.

```
// Se envia el correo.
Transport t = session.getTransport("smtp");
t.connect(de, clave);
t.sendMessage(message, message.getAllRecipients());
t.close();
```

Figura 68: Envio de correo

Para el proceso de almacenamiento de documentos, se hace uso de la tabla documento creado en el modelado de datos del sistema (iteración 0), en donde se almacena dos copias del documento: la del documento original y una segunda copia del documento firmado digitalmente (Ver figura 70).

```
File docs = new File(item.getName());

byte[] fileBytes = getBytesFromFile(docs);
try {

PreparedStatement ps = conn.prepareStatement("INSERT INTO documento(nombre_doc,
doc_original, doc_firmado) VALUES(?,?,?)");

nombre [i]=docs.getName();
ps.setString(1, docs.getName());
ps.setBytes(2, fileBytes);

SignPDF firmar = new SignPDF();

ps.setBytes(3, fileBytes);
```

Figura 69: Almacenar documento

Pruebas

Las pruebas de aceptación que se hicieron para verificar que se enviara correctamente los documentos fue la siguiente: se realizó el envío de varios documentos con formato PDF a distintos destinos, el primer destinatario del flujo asignado para el (los) documento (s) enviado (s) recibe un correo electrónico donde se le notifica que le ha llegado un (los) documento (s) que necesita (n) su aprobación. Comprobando de esta manera que funciona correctamente el proceso de envío en el sistema.

3.3.13. Iteración 12: Diseño de flujo para envió de documentos

En esta iteración se realiza diseño de flujo de un documento, el flujo consiste en un conjunto de receptores (con un orden en particular) para el envío de documentos con el fin de que los receptores del flujo aprueben (firmen digitalmente) el documento)

Planificación

En la tabla 22 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 22 Diseño de flujo para envió de documentos (Iteración 12: Diseño de flujo para envió de documentos)

Id	Fecha	Requerimiento	Tipo
022	30/06/2013	Creación de categorías y flujo	Nueva
023	30/06/2013	Visualización de flujos	Nueva

Diseño

Para realizar diseño de flujo se creó en la base de datos una tabla para almacenar categoría de documento que se relacionó con otra tabla que almacena el flujo. Donde un flujo es un conjunto de receptores (empleados de la organización) y por categoría se entiende a la clasificación de los distintos tipos de documentos que se manejan dentro de una organización y por flujo. Esta información será utilizada al momento de realizar el envío de un documento.

La figura 71 que se muestra a continuación, permite la creación de diferentes categorías de documentos, las cuales luego de ser creadas podrán ser asignadas a un flujo que es creado usando la vista de la figura 72.

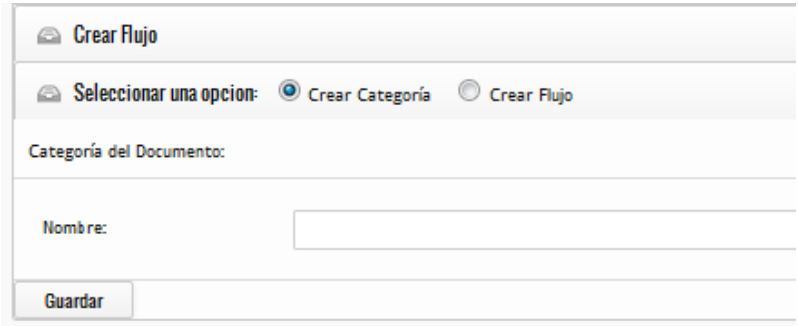


Figura 70: Vista para crear categoría

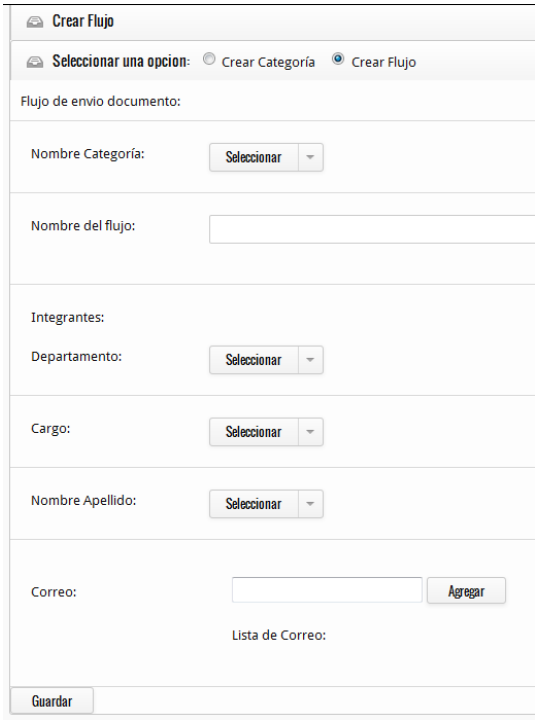


Figura 71: Vista para crear flujo

Codificación

Para el desarrollo de esta iteración se ha creado dos tablas en el modelo de datos del sistema (iteración 0), estas tablas son flujo y categoría. Se toma en

consideración que una categoría de documento puede pertenecer a más de un flujo a la vez y que un flujo puede contener una cantidad de destinatarios limitada a diez (10). La creación y vinculación de una categoría a un flujo determinará quienes son los destinatarios y el orden en el que los mismos han de recibir el (los) documento (s) que espera(n) su aprobación.

En la figura 73 que se muestra a continuación se detalla el proceso de creación de una categoría de documento y la posterior creación del flujo que podría ir vinculado a dicha categoría.

```
    try {

    Statement instruccion=bd.instruccion;
    String query_1="INSERT INTO flujo (nombreflujo, integrantesflujo,categoria) VALUES('"+
    nombreFlujo +"','{"+ integrantesFlujo +"}','"+ categoria +"')";
    sentencia.executeQuery(query_1);
    System.out.println("valores");
}catch(Exception e){
    System.out.println(e);
}
        bd.Cerar();//-----Cerrar Base de Datos----
        %>
        <script language="javascript">
        alert("Flujo registrado correctamente");
        location.href="flujo.jsp";
        </script>
        <%
}

//se guarda la categoria con el flujo

if(request.getParameter("nombreDoc")!=null && request.getParameter("crear_categoria")!=null ){

    BD bd = new BD();
    bd.Conexion();
    Statement sentencia=bd.instruccion;

    //insertar la categoria
    try {

            String query_3="INSERT INTO categoria(nombreDoc) VALUES('" +
            request.getParameter("nombreDoc") + "')";
             sentencia.executeQuery(query_3);

    }catch(Exception e){
        System.out.println(e);
    }
}
```

Figura 72: Guardar los datos para el flujo de receptores

Pruebas

Para el correcto funcionamiento de esta iteración se realizó en un primer momento la creación de una categoría, seguidamente se procedió a crear otra

categoría con el mismo nombre a lo cual el sistema respondió con un mensaje al usuario indicando “Categoría ya existe”. Luego de haber creado una categoría, se procede a crear un flujo que ira vinculado a esa categoría, proceso en el cual el sistema permite añadir una cantidad de destinatarios limitada a diez (10). Luego de haber sido creado el flujo asociado a una categoría, se verifico en el sistema que dicho flujo haya sido creado correctamente y para ello el sistema tiene la opción de mostrar los detalles de todos flujos que han sido.

3.3.14. Iteración 13: Configuración de puesto de trabajo

En esta iteración se realiza la configuración de los departamentos y los cargos asociados a ese departamento dentro de una organización.

Planificación

En la siguiente tabla se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 23 Configuración de puesto de trabajo

(Iteración 13: Configuración de puesto de trabajo)

Id	Fecha	Requerimiento	Tipo
024	03/07/2013	Crear nuevo departamento	Nueva
025	03/07/2013	Crear un puesto de trabajo (departamento-cargo)	Nueva

Diseño

Para realizar la configuración de puesto de trabajo se diseñó dos tablas en el modelado de datos del sistema (iteración 0), una para guardar información de los departamentos (tabla departamento) y otra para la relación entre departamento y cargo que sería el puesto de trabajo (tabla puesto).

La figura 74 que se muestra a continuación, muestra la vista diseñada para la creación de los departamentos de una organización. Luego de haber creado los diferentes departamentos se puede proceder a crear un puesto, indicando el nombre del cargo y el departamento al cual pertenece (Ver figura 75)

Figura 73 Vista para crear departamento

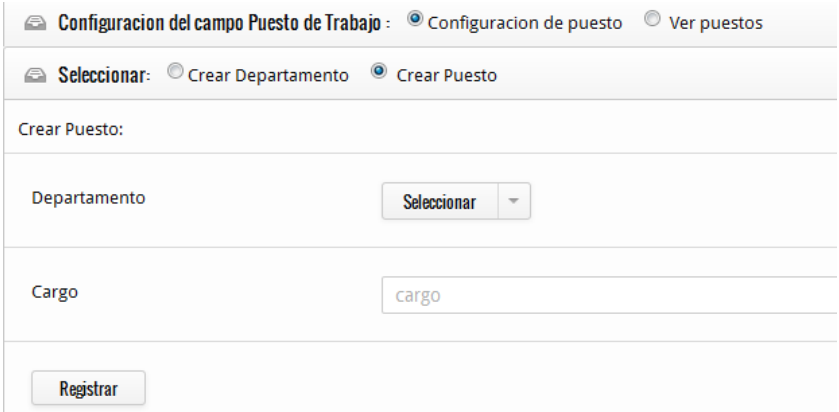


Figura 74: Vista para crear puesto de trabajo

Codificación

Para la creación de un puesto de trabajo se toma en cuenta que los cargos (tabla puesto) asociados a los departamentos se pueden repetir entre los diferentes departamentos, pero no se pueden repetir los nombres de los departamentos (tabla departamento). La información capturada de las vistas de las figuras 74 y 75 es almacenada en las tablas creadas en el modelo de datos del sistema (iteración 0), tal como se muestra en la figura 76.

```
    String sentencia2="select * from departamento where nombre_departamento like '"+departamento+"' ";
    ResultSet t = sentencia.executeQuery(sentencia2);

    if(t.next()){
        %>
        <script language="javascript">
        alert("Error. Este Departamento ya existe");
        location.href="usuario.jsp";
        </script>
        <%
    }

    Statement instruccion=bd.instruccion;
    String query="INSERT INTO departamento ( nombre_departamento) VALUES('"+departamento+"')";
    sentencia.executeQuery(query);

}catch(Exception e){
    System.out.println(e);
}


try {
    String query_4="INSERT INTO puesto ( cargo, iddepartamento) VALUES('"+idc+"','"+idd +"') ";
    sentencia.executeQuery(query_4);

}catch(Exception e){
    System.out.println(e);
}
```

Figura 75: Guardar los datos para el puesto de trabajo

Pruebas

Para verificar que los puestos de trabajo se crearan correctamente se realizaron las siguientes pruebas, se procedió primero a crear un departamento que una vez creado no se puede repetir. Ya creado el departamento se crea el cargo relacionado a ese departamento. El puesto de trabajo creado se puede comprobar que aparece en el vista que se ha diseñado para mostrar los puestos de trabajo existentes (Ver figura 77).

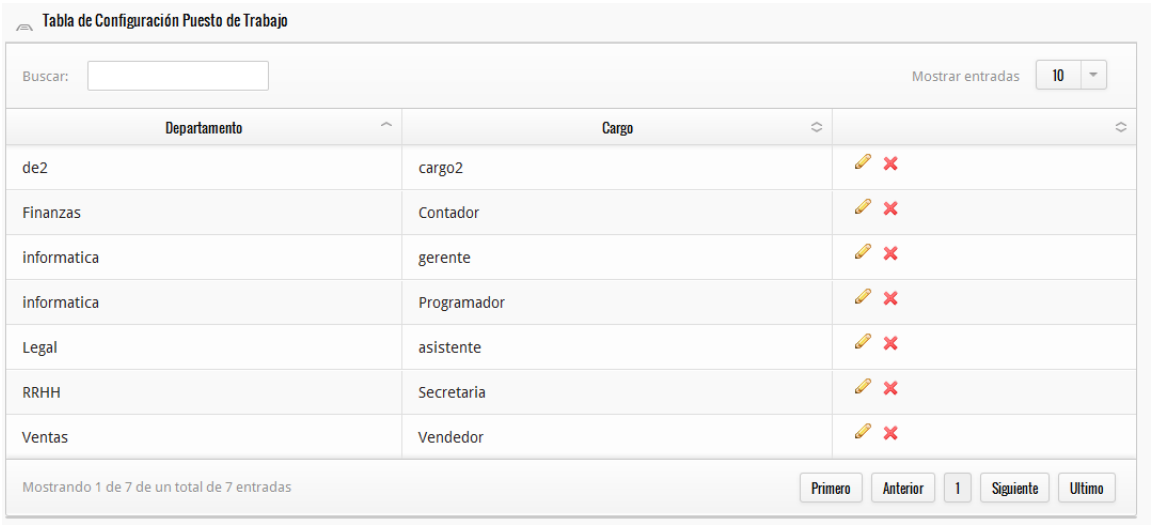


Figura 76: Vista de puestos de trabajos creados

3.3.15. Iteración 14: Configuración control acceso físico

En esta iteración se realiza la configuración para el control del acceso físico a un determinado usuario.

Planificación

En la tabla se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla Configuración Control Acceso Físico

(Iteración 14: Configuración Control Acceso Físico)

Id	Fecha	Requerimiento	Tipo
026	01/08/2013	Gestión de ubicación	Nueva
027	01/08/2013	Gestión de permisos	Nueva

Diseño

Para la gestión de ubicación, se realiza dos acciones principales, primero guarda la ubicación donde estará el equipo para controlar el acceso físico de un determinado lugar, para ello se diseñó una tabla en la base de datos para guardar la dirección MAC del equipo y la descripción del lugar (Ver figura 78) y después permite visualizar las ubicaciones controladas (Ver figura 79), donde se tiene la opción de eliminar una ubicación.

Figura 77: Vista para asignar una ubicación

Figura 78: Listado de ubicaciones permitidas

Una vez asignados los permisos a las diversas ubicaciones, se puede proceder a determinar que usuarios del sistema pueden tener acceso a esa ubicación y especificar un rango de fecha para dicho acceso (Ver figura 80).

Figura 79: Vista para el permiso de un usuario a una ubicación

Codificación

De la información capturada en el formulario de la figura 81 se procede a almacenar los datos de la ubicación de la computadora a la que se le asignaran los permisos y el tiempo (horas) en que pueden tener acceso al espacio físico donde se encuentra la computadora, para ello se hace uso de la tabla acceso (iteración 0) en la que quedará registrado todos los permisos asignados a una ubicación especifica (Ver figura 82).

```
if(request.getParameter("tiempo")!=null && request.getParameter("mac")!=null
    && request.getParameter("ubicacion")!=null && !request.getParameter("permiso").equals("-1") ){

    t= request.getParameter("tiempo");
    m = request.getParameter("mac");
    u = request.getParameter("ubicacion");
    p=request.getParameter("permiso");

    BD bd = new BD();
    bd.Conexion();

try {
    Statement instruccion=bd.instruccion;
    String sentencia="INSERT INTO acceso (mac,lugar,permisotodo,tiempomax) VALUES('"+
            m +"','"+ u +"','"+ p +"',"+ t +")";
    instruccion.executeUpdate(sentencia);
}catch(Exception e){
    System.out.println(e);
    %>
    <script language="javascript">
    alert("Error!. Ya existe esta dirección mac ");
    location.href="GestionControlAcceso.jsp";
    </script>
    <%
}finally{

    bd.Cerar();
}

<script language="javascript">
    alert("Ubicacion Registrado.");
    location.href="GestionControlAcceso.jsp";
</script>
<%

}
```

Figura 80: Ubicación

De la misma manera, para la información obtenida del formulario de la figura 82 se hace uso de la tabla “registroestado” creada en el modelado de datos del sistema (iteración 0), en la que almacenará los usuario que tienen acceso a la ubicación especificada en la figura 83 siempre y cuando la misma ya no haya sido asignada previamente en el mismo rango de fechas.

```
try{
            String sentencia2="select * from registroestado inner join acceso on acceso.mac =registroestado.mac where  '"+
                    i+"'>='"+f+"' and acceso.mac like '"+m+"' and ((final >= '"+i+"'  and inicio <= '"+i+
                    "' ) or (final >=  '"+f+"' and inicio <= '"+f+"'))";
            ResultSet t = sentencia.executeQuery(sentencia2);

            if(t.next()){
                %>
                <script language="javascript">
                    alert("Error. Fecha en conficto");
                    location.href="GestionControlAcceso.jsp";
                </script>
                <%

            }else{
                try {

                    Statement instruccion=bd.instruccion;
                    String sentencia5="INSERT INTO registroestado (cedula,estado,inicio,final,mac) VALUES('"+
                        c+"','"+e+"','"+i+"','"+f+"','"+m+"')";
                    instruccion.executeQuery(sentencia5);

                }catch(Exception e){
                    System.out.println(e);
                }
                %>
                <script language="javascript">
                    alert("Se guardo correctamente");
                    location.href="GestionControlAcceso.jsp";
                </script>
                <%
            }
    }catch(Exception f){
            System.out.println(f);
    }
```

Figura 81: Permisos

En la figura 79 se muestra todos las ubicaciones que han sido creadas, las mismas pueden ser modificadas o eliminadas, en este último caso siempre y cuando las ubicaciones no estén previamente asignadas a los usuarios (Ver figura 80).

```
if(request.getParameter("deletemac")!=null){

    try{
        String sentencia="delete from  acceso where mac='"+request.getParameter("deletemac")+"'";

        System.out.println("Eliminar  ... :"+sentencia);

        Statement instruccion=bd.instruccion;
        arrojo = instruccion.executeUpdate(sentencia);
        %>
        <script language="javascript">
        alert("Se elimino correctamente");
        location.href="GestionControlAcceso.jsp";
        </script>
        <%


    } //fin try no usar ; al final de dos o mas catchs

    catch(SQLException e) {


        %>
        <script language="javascript">
        alert("Error . Tiene dependencia");
        location.href="GestionControlAcceso.jsp";
        </script>
        <%

    };
```

Figura 82: Eliminar Mac

Pruebas

Las pruebas realizadas a esta iteración incluyeron el doble registro de una misma dirección MAC, a lo cual el sistema notifica al usuario que dicha dirección MAC ya existe.

Para la asignación de los permisos a un usuario en particular, el tiempo que se le otorga está en un rango de 1-24 horas, el usuario podrá indicar el número de horas haciendo uso de un control que contiene ese rango, por lo cual el número de horas no sobrepasara el rango asignado. De la misma forma, el sistema no permite asignar al mismo usuario dos o más permisos en el mismo rango de fecha, en caso de que esto ocurra el sistema notificará al usuario con un mensaje “Error. Fechas en conflicto”.

3.3.16. Iteración 15: Control Acceso

En esta iteración se otorga o niega el acceso físico a una ubicación a un determinado usuario.

Planificación

En la tabla 24 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 24 Control Acceso Físico (Iteración 15: Control Acceso Físico)

Diseño

Id	Fecha	Requerimiento	Tipo
028	15/08/2013	Autenticar usuario	Nueva
029	15/08/2013	Verificar permisos	Nueva

Para realizar el control de acceso físico se obtiene la huella de la persona mediante un equipo biométrico, el cual se encuentra ubicado a la entrada del espacio físico al que desea acceder. Las huellas de los usuarios del sistema se encuentran almacenadas en la tabla "huella" diseñada en el modelado de datos del sistema (iteración 0), el sistema verifica que la huella se encuentre almacenada en dicha tabla y además verifica que el usuario asociado a ese número de cédula tenga permiso para acceder a dicha ubicación (iteración 14). En caso de ser una huella valida el sistema mostrará al usuario un mensaje de "Acceso Otorgado" (Ver figura 84) y luego de ello registrará la entrada o salida del usuario del espacio físico según sea el caso cada vez que coloque su huella sobre el lector, siempre y cuando el usuario tenga permisos para acceder a la ubicación.

En caso de que la huella no este registrada el sistema mostrará un mensaje de "Acceso NO otorgado" (Ver figura 85)

Figura 83: Acceso otorgado

Figura 84: Acceso no otorgado

Codificación

Para registrar la entrada o salida del usuario de una ubicación, se procede primero a la captura de la huella registrada del usuario, si la huella es válida (Ver figura 84) se obtiene la cédula de dicho usuario y con la dirección MAC de la computadora se hace una primera verificación que consiste en revisar si el usuario está activo o no, por ejemplo si un usuario está de vacaciones o reposo, su estado pasa de estar "Activo" a "Inactivo", si es un usuario "Activo" se procede a verificar como segundo paso (Ver figura 85) si el usuario está registrando una entrada o una salida, si el usuario marca su salida luego del tiempo permitido el sistema notificará al usuario que su entrada/salida no fue registrada en el tiempo estipulado (Ver figura 86) y además apenas el usuario exceda las horas permitidas para estar en una ubicación, el sistema enviará un correo al administrador del sistema notificando la irregularidad.

```
if(c.EstadoUsuario(cedula,bd.instruccion,mac)){
        if(c.VerificacionAcceso(cedula, bd.instruccion,bd.canal,mac)){
            String tipo=c.tipo();
            if(tipo.equals("entrada")){
                %>
                <script language="javascript">
                    alert("Se registro la entrada");
                    window.close();
                </script>
                <%
            }else if(tipo.equals("salida")){
                %>
                <script language="javascript">
                    alert("Se registro la salida");
                    window.close();
                </script>
                <%
            }
        }else{
            %>
            <script language="javascript">
            var mostrar =  document.getElementById("tipo");
            alert("Error . No se registro su  "+'".$tipo."'+" en el tiempo estipulado");
            window.close();
            </script>
            <%
        }
}else{
    %>
    <script language="javascript">
    alert("Acceso Denegado.");
    window.close();
    </script>
    <%

}
```

Figura 85: Verificar permiso y acceso a un espacio físico

```
//verifico si tiene permiso a todo
String s="select permisotodo from acceso where mac like '"+accesomac+"'";
System.out.println("senmtencia permiso  "+s);
//System.out.println("Entro consulta permiso "+s);
ResultSet t= instruccion.executeQuery(s);
if(t.next()){
    permiso= t.getString(1);
}else{
    permiso="negado";
}

String sentencia="select estado, inicio , final  from registroestado where mac like '"+
accesomac+"' and cedula like '"+cedula+"' and DATE(final) >= '"+
        diactual+"' and DATE(inicio) <= '"+diactual+"'";
tabla= instruccion.executeQuery(sentencia);

//System.out.println("Entro consulta fecha "+sentencia);

if(tabla.next()){
    String estado=tabla.getString(1);
  if(estado.equals("inactivo")){
          b="no";
  }
  if(estado.equals("activo")){
      b="si";
  }


}
```

Figura 86: Detalles del método de verificación de permiso

Pruebas

Para verificar el correcto funcionamiento de esta iteración, se ingresó a la vista de la figura 84, una huella registrada y con permisos, una segunda huella registrada sin permisos y finalmente una tercera huella no registrada. Para los dos primeros casos, el sistema reconoce las huellas, la primera al tener permisos asignados a esa ubicación física permite el registro de su entrada y si la huella es colocada nuevamente el sistema le indica que está ingresando su salida.

En el caso de la segunda huella, no tiene permisos asignados por lo que el sistema le notifica al usuario que no tiene permisos para acceder a la ubicación. Por último la tercera huella al no estar registrada, el sistema le indica al usuario el mensaje que aparece en la figura 85.

3.3.17. Iteración 16: Estadísticas sobre el control de acceso físico

En esta iteración se muestra un conjunto de estadísticas con respecto a los accesos a una ubicación físicas en particular.

Planificación

En la tabla 25 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 25 Estadística de control de Acceso físico (Iteración 16: Estadística de control de Acceso físico)

Id	Fecha	Requerimiento	Tipo
030	9/08/2013	Obtener datos	Nueva
031	9/08/2013	Crear Graficas	Nueva
032	9/08/2013	Visualizar entradas/salidas	Nueva

Diseño

Para realizar las estadísticas sobre los accesos realizados a una ubicación en particular, se utiliza las tablas creadas en el modelo de datos del sistema (iteración 0) las cuales son: "registrousuacc" y la tabla "registroestado", donde en la tabla "registrousuacc" se guarda un historial de las entradas y salidas de la persona en un determinado lugar (ubicación), esta información se muestra en el sistema (Ver figura 88). En la figura 89 se muestra un formulario donde se puede conocer en detalle el historial de entradas y salidas para una ubicación en una fecha específica generando

unas gráficas para las inasistencias, asistencias y faltas de las personas en un rango de fecha indicado (Ver figura 90)

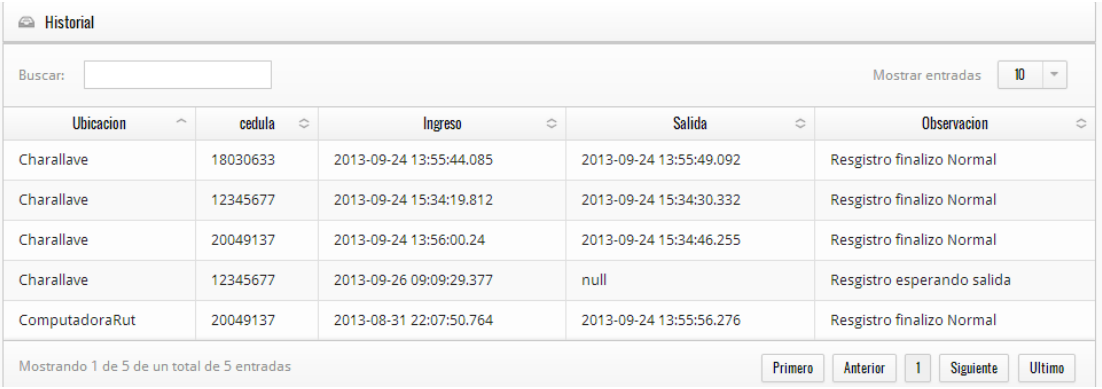


Figura 87: Vista del historial de entradas y salidas para todas las ubicaciones

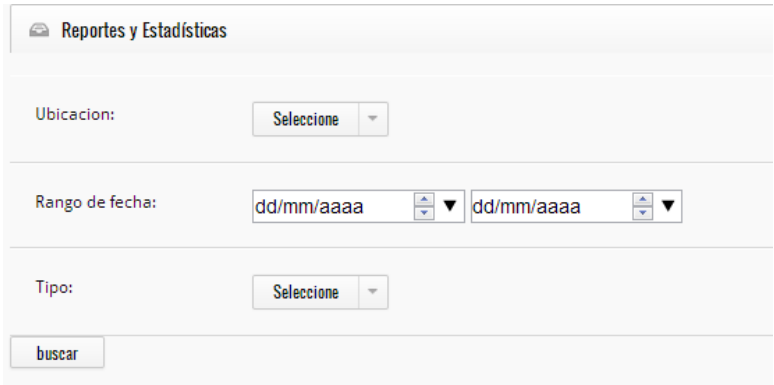


Figura 88: Vista para ingresar los datos para la estadística

Figura 89: Gráficas generadas para los datos ingresados en la figura 88

Codificación

El código de la figura 91 muestra las consultas realizadas en la base de datos a partir de las cuales se toman los datos utilizados para las gráficas generadas en la figura 90.

```
String sentencia1="select count(distinct(cedula)) from registroestado where mac like '"+
    m+"' and estado like 'inactivo'  and DATE(final) >= '"+f+"' and DATE(inicio) <= '"+i+"'";
ResultSet tabla1= instruccion.executeQuery(sentencia1);
if(tabla1.next()){
    inasitencia = tabla1.getInt(1);
}
System.out.println("Inasistencia "+inasitencia  );

String sentencia2="select count(distinct(cedula)) from usuario where cedula not like all "+
    "(select distinct(cedula) from registrousuacc where  estado like 'completo'   and  mac like '"+
    m+"'  and DATE(fechahorasalida) >= '"+f+"' and DATE(fechahoraentrada) <='"+i+"' )";;
ResultSet tabla2= instruccion.executeQuery(sentencia2);
if(tabla2.next()){
    falta   = tabla2.getInt(1);
}

String sentencia6="select count(distinct(cedula)) from usuario ";
ResultSet tabla6= instruccion.executeQuery(sentencia6);
if(tabla6.next()){
    total   = tabla6.getInt(1);
}

System.out.println("Todo "+total);

String sentencia3="select count(distinct(cedula)) from registroestado where mac like '"+
    m+"' and estado like 'activo'  and DATE(final) >= '"+f+"' and DATE(inicio) <='"+i+"' ";
ResultSet tabla3= instruccion.executeQuery(sentencia3);
if(tabla3.next()){
    asistenciarelativo  = tabla3.getInt(1);
}
System.out.println("asistenciarelativo "+asistenciarelativo );

String sentencia4="select count(distinct(cedula)) from registrousuacc where  estado like 'completo'   and  mac like '"+
    m+"'  and DATE(fechahorasalida) >= '"+f+"' and DATE(fechahoraentrada) <='"+i+"' ";
ResultSet tabla4= instruccion.executeQuery(sentencia4);
if(tabla4.next()){
    asistenciareal  = tabla4.getInt(1);
}
```

Figura 90: Datos para generar gráficas

Pruebas

Se realizaron las siguientes pruebas para comprobar el correcto funcionamiento de la iteración, entre las pruebas realizadas fue la validación en el rango de fechas a fin de que la fecha inicio no fuera mayor que la fecha final, a lo que el sistema detecto los datos como inválidos. De la misma forma se generó varias gráficas a distintas ubicaciones mostrando resultados diferentes para cada caso.

3.3.18. Iteración 17: Generación de reportes

En esta iteración se realiza la generación de reportes en un documento con formato PDF.

Planificación

En la tabla 26 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 26 Historias de Usuario (Iteración 17: Generación de reportes de documentos)

Id	Fecha	Requerimiento	Tipo
033	01/09/2013	Generación de documento pdf	Renovada

Diseño

Para el diseño de esta iteración por ejemplo, se desea obtener el listado de todos los documentos enviados, para ello se tiene una lista desplegable donde el usuario puede elegir qué tipo de estatus quiere que se genere (aceptado, rechazado, todos, en espera) donde el estatus "todos" indica que para la generación del reporte se tomará en cuenta todos los documentos enviados (Ver figura 92).

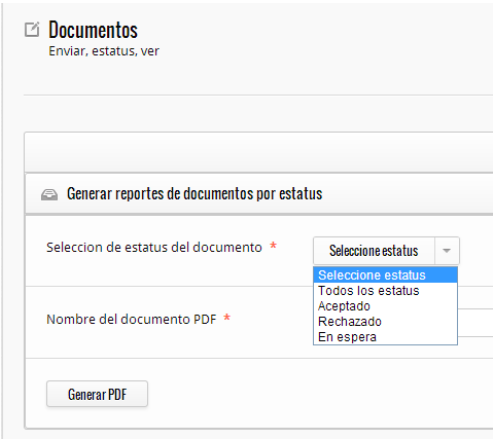


Figura 91: Diseño de la vista de generación de reportes

Además el usuario puede especificar el nombre del documento a generar. Luego de pulsar el botón correspondiente genera reporte se descargara automáticamente el PDF generado (Ver figura 93 y figura 94).

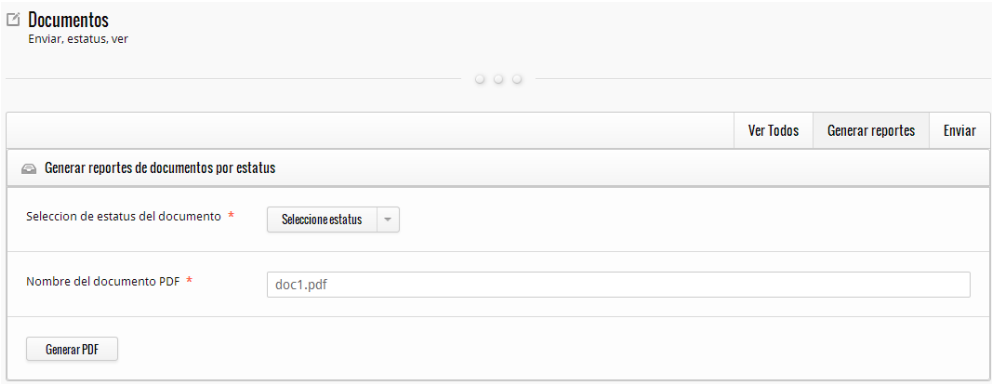


Figura 92: Pantalla para indicar los datos necesarios para la generación de reportes

PRUEBADOC

Documentos								
Fecha Inicio	Nombre del Documento	Categoria del Documento	Posición del Flujo Receptor	Departamento del Receptor	Cargo del Receptor	Estatus	Observaciones	Fecha Finalización
2013-09-22 16:46:11.403	PDF prueba.pdf	memo	1	informatica	gerente	aceptado	El documento ha sido aceptado por 0 personas	2013-09-22 17:44:07.717
2013-09-22 23:06:21.655	1028.pdf	diana	2	informatica	gerente	aceptado	El documento ha sido aceptado por 1 personas	2013-09-24 01:16:19.304
2013-09-23 19:33:17.883	prueba.pdf	diana	2	informatica	gerente	aceptado	El documento ha sido aceptado por 1 personas	2013-09-24 01:22:41.763
2013-09-23 19:33:17.883	prueba.pdf	diana	2	informatica	gerente	aceptado	El documento ha sido aceptado por 1 personas	2013-09-24 01:22:41.763

Figura 93: Ejemplo de documento PDF generado

Codificación

Para el desarrollo de esta iteración para la generación de reportes de estatus de documentos enviados se utiliza la librería ITEXT para JAVA. Que nos permite generar documentos con formato PDF dinámicamente.

De los datos obtenidos del formulario de la figura 93 se realiza la respectiva consulta a la base de datos, donde los datos que nos retorna la consulta serán los que se ingresaran en el documento PDF a generar, para ello se dibuja una tabla y se insertan en ella el resultado de la consulta a la base de datos (Ver figura 95).

```
        //Iterar Mientras haya una fila siguiente
    while (rs.next()) {          //Agregar 9 celdas

        String lugar ="";
        if(rs.getInt("posicion")==1){
            lugar="0";
        }else{
            lugar=Integer.toString(rs.getInt("posicion")-1);

        }
            cell = new PdfPCell(new Paragraph(rs.getString("fecha_i"),fuente8 )); //Fecha inicio
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombre_doc"),fuente8));//Nombre del documento
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombredoc"),fuente8));//Nombre categoria
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("posicion"),fuente8)); //posicion
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombre_departamento"),fuente8)); // Nombre departamento
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("nombre_cargo"),fuente8)); //Nombre cargo
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("estado_autor"),fuente8) );//Estatus
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph("El documento ha sido aceptado por "+lugar+" personas",fuente8)); //Observaciones
            tabla.addCell(cell);
            cell = new PdfPCell(new Paragraph(rs.getString("fecha_f"),fuente8)); // Fecha fin
            tabla.addCell(cell);


    }
```

Figura 94: Creación de una tabla en un documento PDF

Luego de completado el documento PDF se procede a su exportación para la posterior descarga por parte del usuario. (Ver figura 96).

```
//Metodo principal del ejemplo
public void GenerarPDF(String titulo, String nomPDF, String statusDoc) {
    strRotuloPDF = titulo;
    strNombreDelPDF = nomPDF;
    try{

        //Hoja tamanio carta, rotarla (cambiar a horizontal)
        document = new Document(PageSize.LETTER.rotate() );

        writer = PdfWriter.getInstance(document,
                // direccionar el PDF-stream a un archivo
                new FileOutputStream(strNombreDelPDF));

        agregarMetaDatos(document);

        document.open();


        agregarContenido(document);

        document.close();

        System.out.println("Se ha generado el PDF: "+ strNombreDelPDF);

    } catch (Exception e)  {
        e.printStackTrace();
    }
}
```

Figura 95: Creación de un documento PDF

Pruebas

Las pruebas de aceptación que se realizaron para esta iteración incluye la validación del formulario (Ver figura 93), así como la correcta generación de reportes para cada uno de los estatus de documentos seleccionados y el formato en que estos datos son mostrados en el documento.

3.3.19. Iteración 18: Generación de gráfico sobre flujo de un documento

En esta iteración se realiza la generación de un gráfico sobre el flujo en el que se encuentra un documento determinado.

Planificación

En la tabla 27 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 27 Historias de Usuario (Iteración 18: Generación de gráfico sobre flujo de un documento)

Id	Fecha	Requerimiento	Tipo
034	08/09/2013	Generación de gráfico sobre flujo de un documento	Nueva

Diseño

Para el diseño de esta iteración se presenta un gráfico en el que se presenta el flujo determinado para un documento, indicando el autor del documento, las personas que has firmado/rechazado o que se encuentran en espera de firmar/rechazar un documento. En la figura 97 se puede apreciar un ejemplo del flujo de un documento.

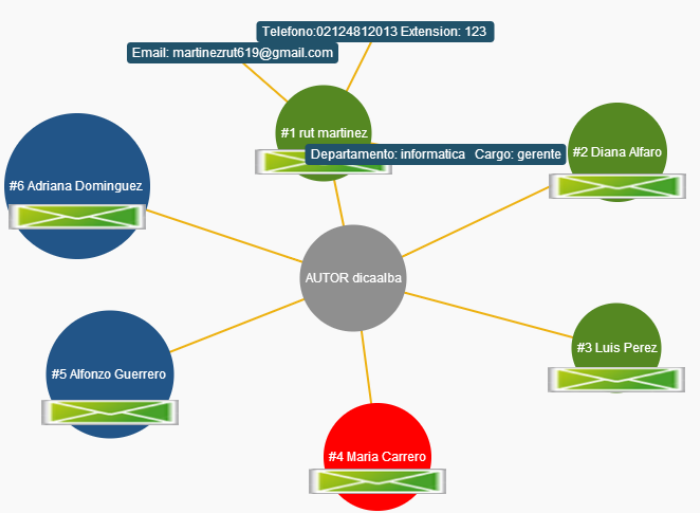


Figura 96: Diseño de la vista de generación de gráficos

También se brinda una leyenda para cada uno de los colores del gráfico (Ver figura 98).

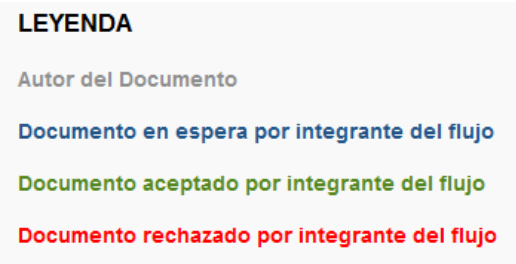


Figura 97: Leyenda que indica el significado de cada color del gráfico

Además se agrega información adicional sobre cada uno de los usuarios involucrados en el flujo, tales como: nombre del departamento, cargo/puesto que ocupa, teléfono y correo de contacto, dicha información puede ser visualizada al posar el cursor del mouse sobre alguno de los nombres del gráfico, tal como se aprecia en la figura 99.

Figura 98: Detalles de un usuario involucrado en el flujo

Codificación

Para el desarrollo de esta iteración para la generación de gráficos dinámicos se hace uso de un plugin de JQUERY llamado “Arbor”

Este plugin se basa en nodos, donde cada nodo contiene cierta información de cada integrante del flujo y los nodos se conectan entre sí. A mayor número de nodos el gráfico resultante adapta el espacio que ocupa en la pantalla con el fin de distribuir uniformemente los nodos.

En la siguiente figura se muestra el proceso de unión entre nodos (Ver figura 100).

```
edges:{
  ion:{ innovation:{},
      participation:{},
        international:{},
        collaboration:{},
        facilitation:{},
        solution:{}
  },
  innovation:{ innovation1:{}, innovation2:{}, innovation3:{} },
 participation:{ participation1:{}, participation2:{}, participation3:{} },
  international:{ international1:{}, international2:{}, international3:{} },
  collaboration:{ collaboration1:{}, collaboration2:{},  collaboration4:{}  },
  facilitation:{ facilitation1:{}, facilitation2:{}, facilitation3:{}} ,
  solution:{ solution1:{}, solution2:{}, solution3:{} },
 }
}


 // inicializamos arbor
var sys = arbor.ParticleSystem()
sys.parameters({stiffness:900, repulsion:2000, gravity:false, dt:0.015})
sys.renderer = Renderer("#viewport");
sys.graft(data);
```

Figura 99: Creación del gráfico que muestra el flujo de un documento

Pruebas

Las pruebas de aceptación que se realizaron para esta iteración, el usuario desea que el gráfico generado represente una estructura lineal. Sin perder los detalles que se muestran actualmente en el flujo que se genera.

3.3.20. Iteración 19: Generación de gráfico sobre flujo de un documento

En esta iteración se realiza la versión final del gráfico que se genera con los datos del flujo en el que se encuentra un documento determinado.

Planificación

En la tabla 28 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 28 Historias de Usuario (Iteración 19: Generación de gráfico sobre flujo de un documento-Renovada)

Id	Fecha	Requerimiento	Tipo

035	28/09/2013	Cambiar el gráfico generado sobre flujo de un documento	Renovada

Diseño

Para el diseño de esta iteración se realiza un cambio en el diseño del gráfico presentado en la figura 97, en gráfico sigue manteniendo su función de representar un flujo determinado para un documento, indicando el autor del documento, las personas que has firmado/rechazado o que se encuentran en espera de firmar/rechazar un documento. En la figura 101 se puede apreciar un ejemplo del nuevo diseño del flujo de un documento.

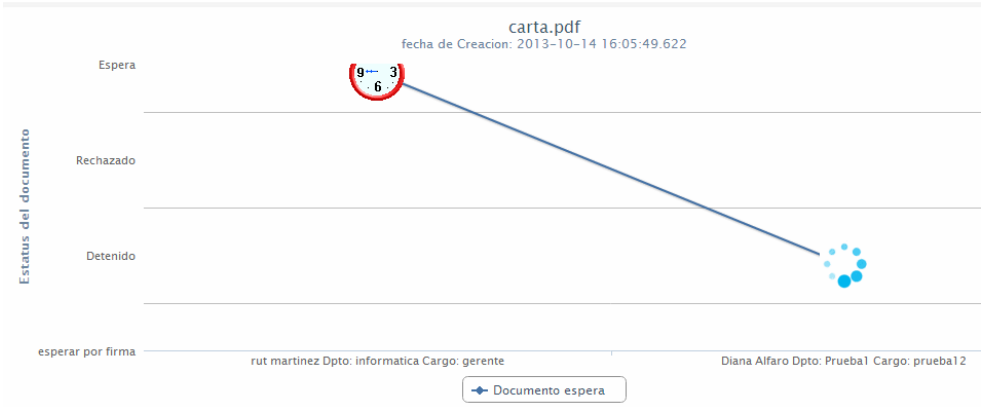


Figura 100: Diseño de la nueva vista de generación de gráficos

También se brinda una nueva leyenda para cada uno de las imágenes que se aprecian en el gráfico (Ver figura 102).

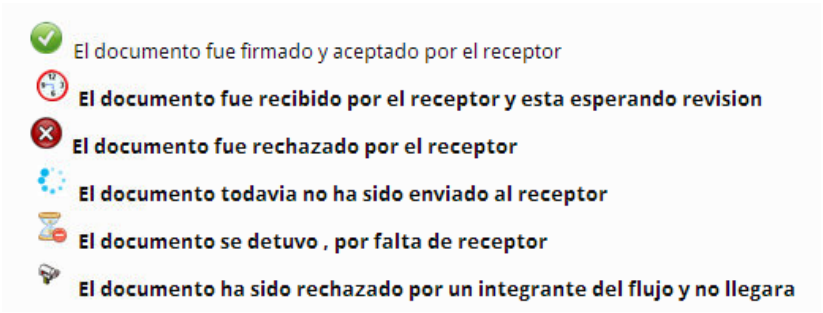


Figura 101: Leyenda que indica el significado de cada imagen del gráfico

Además se agrega información adicional sobre cada uno de los usuarios involucrados en el flujo, tales como: nombre del departamento, cargo/puesto que ocupa, teléfono y correo de contacto, dicha información puede ser visualizada al posar el cursor del mouse sobre alguno de los nombres del gráfico, tal como se aprecia en la figura 103.

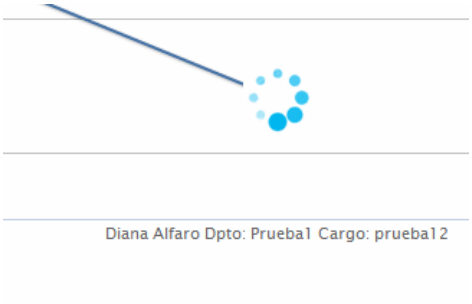


Figura 102: Detalles de un usuario involucrado en el flujo

Codificación

Para el desarrollo de esta iteración para la generación de gráficos dinámicos se hace uso de un plugin de JQUERY llamado “highcharts”

Este plugin es una librería javascript con la que es posible implementar una multitud de opciones para estadísticas y detallar valores.

En la siguiente figura se muestra cómo dependiendo del estado del documento para ese integrante (aceptado, rechazado, espera) se le asigna una posición en el eje Y, así como una imagen de acuerdo al estatus (Ver figura 104).

```
if(i==posicion){
    System.out.println("entro");
    //aceptado
    if(estado_documento.equals("aceptado")){
        integrantes_autor=integrantes_autor+"{y:5.73,marker: {symbol:'url(http:../Images/PDF/aceptar3.png)'}} ";
    }
    //rechazado
    if(estado_documento.equals("rechazado")){
        integrantes_autor=integrantes_autor+"{y:3.9,marker: {symbol:'url(http:../Images/PDF/rechazar3.png)'}} ";

    }
    //espera
    if(estado_documento.equals("espera")){
        integrantes_autor=integrantes_autor+"{y:4.9,marker: {symbol:'url(http:../Images/PDF/espera.gif)'}} ";

    }
```

Figura 103: Creación del gráfico que muestra el flujo de un documento

Pruebas

Las pruebas de aceptación que se realizaron para esta iteración, el usuario estuvo satisfecho con el diseño del gráfico actual.

3.3.21. Iteración 20: Configuración control acceso físico

En esta iteración se realiza algunos cambios en la gestión de ubicación, la gestión de estadística y la gestión de permisos. Además de ello se crea la gestión de horarios que está relacionada con las dos gestiones antes mencionadas.

Planificación

En la tabla 29 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 29 Historias de Usuario (Iteración 20: Configuración control acceso físico-Mejorada)

Id	Fecha	Requerimiento	Tipo
036	30/09/2013	Gestión de estadística	Mejorada
037	30/09/2013	Gestión de permisos	Mejorada
038	30/09/2013	Gestión de ubicación	Mejorada
039	30/09/2013	Gestión de horarios	Nueva

Diseño

Para el diseño de la gestión de estadística se realizan algunos cambios respecto al historial de entradas y salidas para todas las ubicaciones mostrado en la figura 88. En el cual se hacía uso de un mismo lector de huellas que permitía marcar la entrada o la salida. Para esta iteración se asumen dos lectores de huella: una para ingresar la entrada y otro para ingresar la salida. Cada vez que se coloca el dedo sobre el lector (si la persona está autorizada para acceder a esa ubicación), el sistema registrará la fecha y hora del acontecimiento. Ver figura 105.

Ubicacion	Cédula	Observación	Fecha/Hora
ComputadoraRut	20049137	entrada	2013-08-31 22:07:50.764
ComputadoraRutACTUAL	20049137	salida	2013-10-04 01:14:40.304
ComputadoraRutACTUAL	20049137	salida	2013-10-04 01:14:43.381

Figura 104: Nueva presentación para el historial de entradas y salidas

Así mismo, en la gestión de estadística para la generación de reportes se elimina la opción de generar reportes para un usuario en particular (Ver figura 89) y se establece un reporte general dada una ubicación y un rango de fechas (Ver figura 106).

Figura 105: Nueva presentación para la generación de reportes

En la gestión de horarios se permite crear un horario indicando el nombre del horario, la hora de inicio, la hora de fin y los días que formarán parte de ese horario. (Ver figura 109)

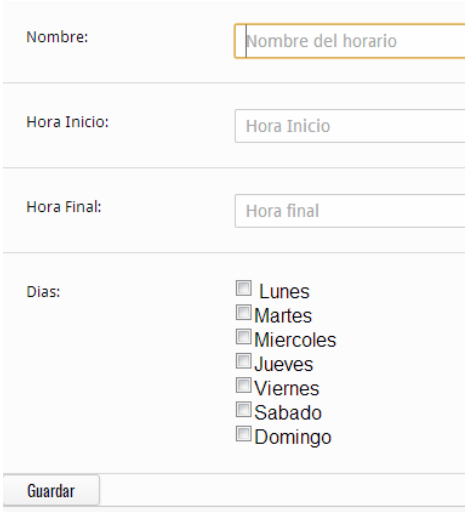


Figura 106: Diseño de la vista para la creación de un horario de trabajo

Una vez creado un horario es posible registrar una ubicación con acceso restringido (gestión de ubicación), a la cual sólo será posible acceder en ese horario (Ver figura 108)

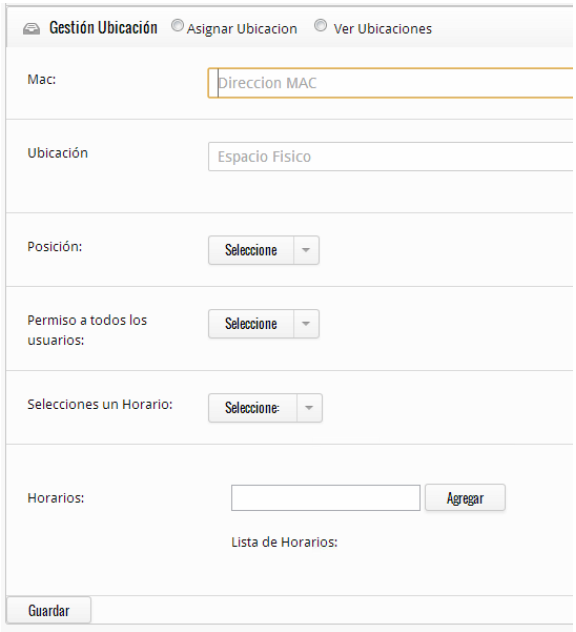


Figura 107: Diseño de la vista para la creación de una ubicación con acceso restringido

Para permitir el acceso a estas ubicaciones con acceso restringido a un usuario en particular, es necesario especificar quién será el que tendrá acceso, a dónde, cuándo (gestión de permisos) entre otros datos que se muestran en la figura 109

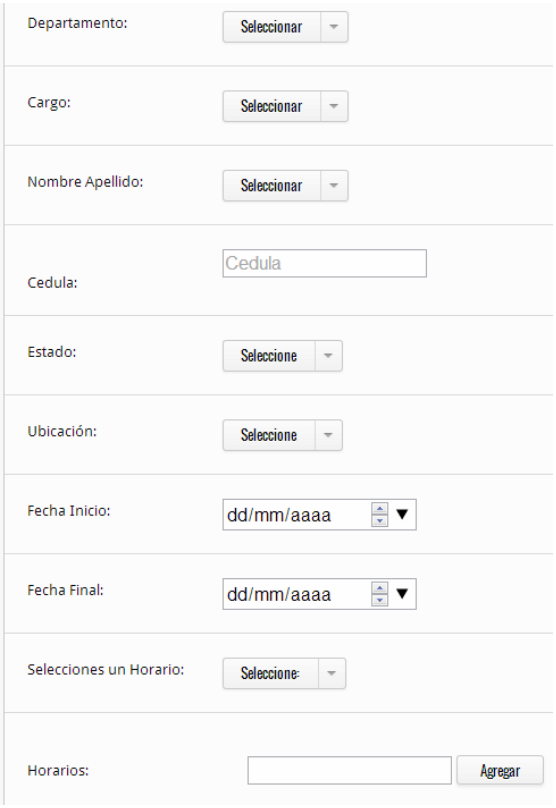


Figura 108: Diseño de la vista de asignación de permisos

Codificación

De los datos obtenidos del formulario de la figura 109 se realiza la respectiva operación en la base de datos para el registro de un nuevo horario de trabajo (Ver figura 110).

```
if(request.getParameter("nombre")!=null && request.getParameter("hora_i")!=null
    && request.getParameter("hora_f")!=null ){

    nombre= request.getParameter("nombre");
    hora_i = request.getParameter("hora_i");
    hora_f = request.getParameter("hora_f");
    String semana="";
    if(request.getParameter("dia1")!=null) lu=request.getParameter("dia1");
    if(request.getParameter("dia2")!=null) ma=request.getParameter("dia2");
    if(request.getParameter("dia3")!=null) mi=request.getParameter("dia3");
    if(request.getParameter("dia4")!=null) ju=request.getParameter("dia4");
    if(request.getParameter("dia5")!=null) vi=request.getParameter("dia5");
    if(request.getParameter("dia6")!=null) sa=request.getParameter("dia6");
    if(request.getParameter("dia7")!=null) dom=request.getParameter("dia7");

    if (lu.equals("1")) semana+= "1,";
    if (ma.equals("2")) semana+= "2,";
    if (mi.equals("3")) semana+= "3,";
    if (ju.equals("4")) semana+= "4,";
    if (vi.equals("5")) semana+= "5,";
    if (sa.equals("6")) semana+= "6,";
    if (dom.equals("0")) semana+= "0,";

    semana=semana.substring(0,semana.length()-1);
    BD bd = new BD();
    bd.Conexion();

try {
    Statement instruccion=bd.instruccion;
    String sentencia="INSERT INTO horario (nombre,hora_i,hora_f,dias) VALUES('"+
        nombre +"','"+ hora_i +"','"+ hora_f +"','{"+ semana +"}')";
    System.out.println(sentencia);
    instruccion.executeUpdate(sentencia);
    bd.Cerar();
}catch(Exception e){
    System.out.println(e);}
```

Figura 109: Registro de un nuevo horario de trabajo

De igual forma, con los datos obtenidos del formulario de la figura 110 se realiza la respectiva operación en la base de datos para el registro de una ubicación con acceso restringido (Ver figura 111).

```
Statement instruccion=bd.instruccion;
String sentencia="INSERT INTO acceso (mac,lugar,permisotodo,todo_horario,nombre_horario,posicion) VALUES('"+
       m +"','"+ u +"','"+ p +"','{"+ idhora +"}','{"+ integranteshora +"}','"+ po+"')";
System.out.println(sentencia);
instruccion.executeUpdate(sentencia);
```

Figura 110: Registro de una ubicación con acceso restringido

Finalmente, para el registro de una asignación de permisos se toman los datos del formulario de la figura 109 (Ver figura 112).

```
Statement instruccion=bd.instruccion;
String sentencia5="INSERT INTO registroestado (cedula,estado,inicio,final,mac) VALUES('"+c+"','"+e+"','"+i+"','"+f+"','"+m+"')";
instruccion.executeQuery(sentencia5);
```

Figura 111: Asignación de permisos

Pruebas

Las pruebas de aceptación que se realizaron para esta iteración, se basaron en validar cada uno de los formularios presentados en las figuras de esta iteración. Pruebas que resultaron exitosas.

3.3.22. Iteración 21: Cambio de SDK de biometría

En esta iteración se realiza un cambio a nivel de software en el sistema. En las iteraciones relacionadas con la captura de la huella digital del usuario, el mecanismo de captura se realizaba por medio del SDK de Griaule Biometrics. Pero debido a un pequeño bug o error de software encontrado en el SDK y que hasta el momento de la redacción del presente documento Griaule Biometrics no tiene solución, se procedió a sustituir por el SDK Verifinger de Neurotechnology.

El error del SDK de Griaule encontrado durante el desarrollo del sistema consiste en la tardía liberación de los recursos (licencia, acceso al lector biométrico) por parte de las librerías de este SDK, error que no está presente con el SDK de Neurotechnology.

Planificación

En la Tabla 30 se observan las historias de usuarios desarrolladas en ésta iteración:

Tabla 30 Historias de Usuario (Iteración 21: Cambio de SDK de biometría)

Id	Fecha	Requerimiento	Tipo
040	05/10/2013	Cambio de SDK de biometría	Nueva
041	05/10/2013	Registro de huella	Renovada
042	05/10/2013	Verificación de huella	Renovada
043	05/10/2013	Firma digital usando la huella	Nueva

Diseño

Para el diseño de esta iteración se realizan mejoras en las vistas de las iteraciones relacionadas a la captura de la huella digital. Como es el caso del proceso de verificación de una huella digital que presenta ahora la vista mostrada en la figura 113.

Figura 112: Nueva vista para la verificación de una huella digital

Al momento de pulsar sobre el botón "Verificar" y colocar el dedo sobre el lector de huella, se puede visualizar la captura de esa huella en la interfaz del sistema, como se muestra en la figura 114.

Figura 113: Captura de una huella para su verificación

De ser la verificación correcta, el sistema mostrará el mensaje de la figura 115, en caso contrario mostrará el mensaje de la figura 116

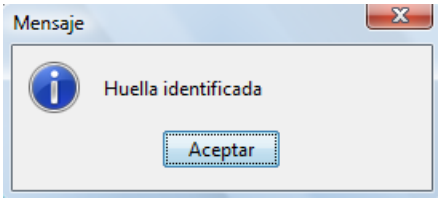


Figura 114: Mensaje de huella identificada

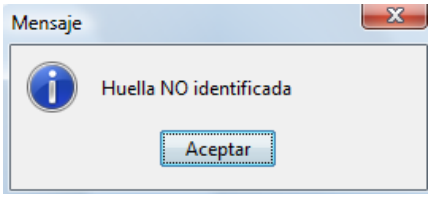


Figura 115: Mensaje de huella no identificada

La vista para el registro de una huella digital en el sistema también fue renovada como se muestra en la figura 117.

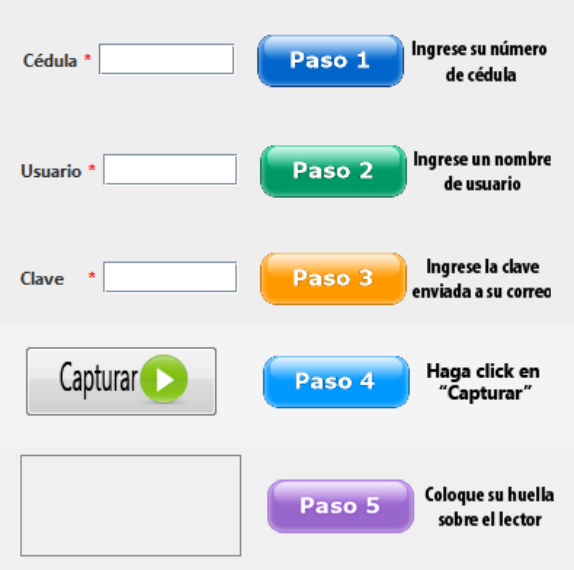


Figura 116: Nueva vista para el registro de una huella digital

Adicionalmente, se añade un paso previo a la firma digital de un documento (iteración 7), para ello antes de proceder a agregar la firma digital a un documento, es necesario colocar la clave del certificado y además la huella digital del firmante, si ambos datos son correctos se agregará la firma al documento. Para lo cual se diseñó la vista que aparece en la figura 118.

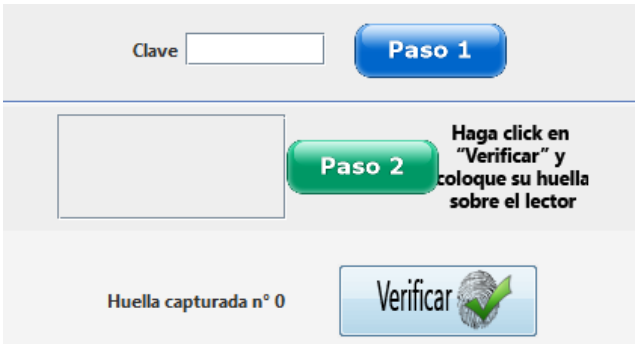


Figura 117: Vista para enlazar la huella digital con la firma digital

Codificación

Los cambios a nivel de código fueron en llamadas a métodos y tipos de datos que varían entre un SDK (Griaule) y otro (Neurotechnology), pero la lógica se mantiene.

En la figura a continuación (figura 120) se puede observar la llamada al método que se encarga de registrar una huella digital junto con otros datos tomados de la vista .

```
db.getRecords().add(txtCedula.getText(), txtUsuario.getText(), txtClave.getText(), template.save(), valorDado);
```

Figura 118: Registro de una huella digital

Para la verificación de la huella digital se compara la huella capturada de la figura 118 con las huellas almacenadas en la base de datos del sistema (ver figura 121)

```
LocalMatcherResult[] results = matcher.identify(bytes, db);

if (results.length==0) {
    JOptionPane.showMessageDialog(getContentPane(), "Huella NO identificada");
} else {
    JOptionPane.showMessageDialog(getContentPane(), "Huella identificada");

    int n=0;
    String ci ="";
    for (LocalMatcherResult result : results) {
        n=result.getNum();
        ci=result.getId();

    }
}
```

Figura 119: Proceso de verificación de una huella digital

Pruebas

Las pruebas de aceptación que se realizaron para esta iteración, se basaron en verificar que el error presente en el SDK anterior (Griaule) no lo presentará también el SDK actual (Verifinger de Neurotechnology).

3.3. Pruebas de Aceptación

En la metodología XP las pruebas de aceptación cumplen con el objetivo de indicarnos cuando las funcionalidades de una iteración han sido completadas exitosamente. A diferencia de las pruebas unitarias realizadas en cada una de las historias de usuario, el criterio de aprobación de las pruebas de aceptación no tiene que ser necesariamente de 100% de efectividad ya que es imposible esperar un código totalmente libre de errores, por lo tanto es necesario definir un criterio de aprobación para saber cuándo el software está listo para ser liberado.

3.3.1. Definición de los casos de prueba

El objetivo de estas pruebas no es tener un conjunto de casos escritos que cubran el 100% del código, sino poder realizar el testing del sistema desde el punto de vista del usuario. En la metodología XP se considera que las pruebas de aceptación deben consistir en un conjunto mínimo (no pobre ni insuficiente) de casos que cubran los requerimientos de negocios fundamentales planteados por el cliente.

Para escribir los casos de prueba se deben tener en cuenta ciertas consideraciones importantes. En primer lugar cada caso debe servir para obtener un feedback rápido y concreto de cómo se está desarrollando la iteración y el proyecto. Se tienen que tratar de evitar los casos de pruebas extensos que incluyan un gran número de pasos. Los casos escritos deben ser concisos y hay que documentar por separado los pasos del caso y los datos de prueba en sí mismos. Es importante señalar también que todos los casos de prueba cumplidos con éxito en iteraciones anteriores se deben seguir cumpliendo en todas las iteraciones, y si se produce un error aunque sea en un único paso del caso de prueba se considera que todo el caso falló.

A continuación se presenta un ejemplo de un caso de prueba de aceptación. En la figura 121 se muestra el resumen del caso, en la figura 122 se detallan los pasos del caso y las acciones que se deben llevar a cabo, y en la figura 123 se especifican los datos de las diferentes pruebas realizadas.

Test Overview	
Acceptance test name	Login
Ref #	1
Story	6
Iteration	1
Critical functionality	Yes
What does this test do?	Tests the login screen, validation of user login, and password
Category	User management
Prerequisite	Records in rows 2, 3, and 7 in TestCase Sheet 1 are in database

Figura 120: Resumen del caso de prueba Login - Crispin (2001)

STEP	COMMAND/URL	ACTION	INPUT DATA	EXPECTED OUTPUT
1	localhost:8080/login.jsp	Enter login name, password, submit	Login and password from TestCase Sheet 2	See expected result in TestCase Sheet 2
2	localhost:8080/login.jsp	Repeat step 1 with all rows in TestCase Sheet 2	TestCase Sheet 2	See expected result in TestCase Sheet 2

Figura 121: Pasos y acciones del caso de prueba 'Login' (abreviado) - Crispin (2001)

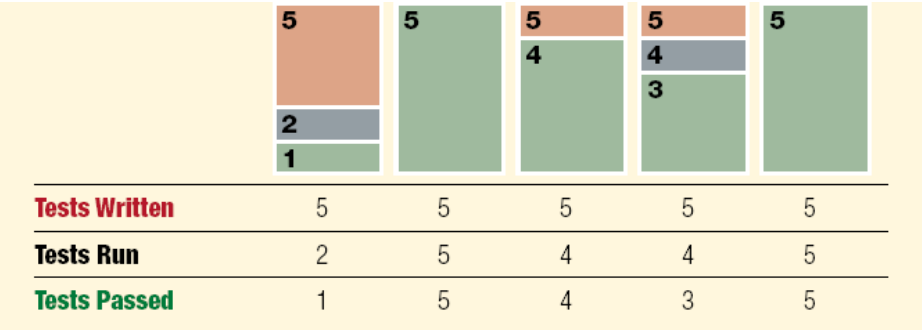
1	LOGIN	PASSWORD	EXPECTED RESULT
2	Testy	tester	Login successful
3	jim-bob	11111	Login successful
4	NULL	NULL	Invalid login and/or password
5	empty	(spaces)	Invalid login and/or password
6	bad (leading space)	password	Invalid login and/or password
7	:";'<>,.	{}|[]\^+@!	Login successful
8	longloginname12345678901234567890 12345678901234567890	longpassword12345678901234567890 12345678901234567890	Invalid login and/or password

Figura 122: Datos de entrada y resultados del caso de prueba 'Login' - Crispin (2001)

3.3.2. Presentación de los resultados de las pruebas de aceptación

Otra importante diferencia entre las pruebas de aceptación y las pruebas unitarias es que para las de aceptación la presentación de los resultados es importante, en cambio para las unitarias no tiene mucho sentido ya que siempre se requiere un 100% de efectividad. Se recomienda la exhibición de los resultados que se obtienen al ejecutar las pruebas de aceptación, generando reportes y gráficas que desplieguen los porcentajes de efectividad obtenidos. Estos índices permiten evaluar si el equipo de desarrollo está realizando un buen trabajo o no. Es importante mantener esta información estadística actualizada y visible para todos los integrantes del proyecto. [Beck ,2000]

El siguiente cuadro de resumen presenta tanto gráfica como numéricamente un ejemplo de los casos de pruebas escritos, ejecutados y exitosos (Ver Figura 124)



Tests Written	5	5	5	5	5
Tests Run	2	5	4	4	5
Tests Passed	1	5	4	3	5

Figura 123: Cuadro de resumen de los casos y resultados de las pruebas de aceptación - Crispin (2001)

La siguiente imagen muestra otro ejemplo presentado e intenta mostrar la cantidad de casos de pruebas de aceptación ejecutados en cada iteración, diferenciándolos entre los casos exitosos, los casos que fallaron y los que no fueron validados por el cliente (Ver figura 125).

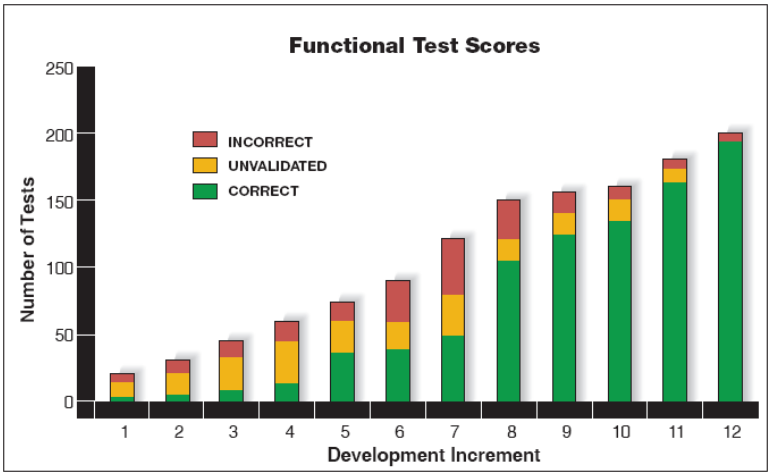


Figura 124: Resultados de las pruebas de aceptación al final de cada iteración - (Jeffries, 1999)

En la figura anterior (Ver figura 125) podemos observar que hay una clara tendencia que indica que en cada iteración se incrementen los casos escritos y ejecutados con respecto a la anterior. Además para que el proyecto culmine de forma exitosa es necesario que en las iteraciones finales los casos con error y los no validados por el cliente disminuyan, llegando a la última iteración con el 100% de los casos validados.

El formato a ser utilizado para la descripción de los casos de prueba a ser realizados se presenta en la tabla 31.

Tabla 31 Formato para la descripción de los casos de prueba

Nombre	Nombre de fácil comprensión del propósito del caso de prueba y su campo de aplicación.
ID	Identificador del caso de prueba
Iteración	La actual iteración sobre el caso de prueba
Nro. de Historia de usuario	Al que se le está realizando el caso de prueba
Descripción	Contiene una breve descripción del propósito de

	la prueba, y la funcionalidad que chequea
Categoría	Módulo al que pertenece el caso de prueba
Pre-requisitos	Suposiciones que deben cumplirse antes de que correr el caso de prueba

La siguiente tabla muestra el formato de los pasos del caso de prueba y las acciones a llevar a cabo (Ver Figura 32).

Tabla 32 Formato de los pasos del caso de prueba y las acciones a realizar

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada

La tabla que se presenta a continuación (Ver Figura 33) corresponderá con el formato a utilizar para especificar los datos de entrada de las diferentes pruebas realizadas así como el resultado de su ejecución.

Tabla 33 Formato de los datos de entrada y el resultado otorgado

Prueba #	Datos de entrada	Resultado

3.3.2.1. Caso de prueba: Ingreso al sistema (Huella Digital)

Para este caso de prueba se verifica el correcto funcionamiento de la primera fase de login o ingreso al sistema, la cual se realiza a través de la captura de la huella digital del usuario.

Descripción

La siguiente tabla (Ver Tabla 34) muestra la descripción para el presente caso de prueba:

Tabla 34 Caso de prueba: Ingreso al sistema (Huella Digital)

Nombre	Ingreso al sistema (Huella digital)
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en determinar que únicamente las huellas de los usuarios que se encuentren registradas son las que podrán acceder a al sistema. Para ello se probará con huellas registradas en la base de datos y aquellas que no lo estén.
Categoría	Módulo de acceso
Pre-requisitos	Tener instalado los controladores (drivers) del lector de huella, así como tener conectado el lector a la unidad USB. El modelo de lector a usar es el UAREU 4000B

Acciones a realizar

La tabla 35 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de ingreso al sistema por huella dactilar.

Tabla 35 Pasos y acciones a realizar: Ingreso al sistema (Huella Digital)

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/ HTML/index.jsp	Colocar el dedo sobre el lector de huellas y luego hacer click en el botón "Verificar"	Huella Digital de la tabla 36 para la prueba #1	Ver resultado de la salida en la tabla 36 para la prueba #1
2	localhost:8080/ CAFD/ HTML/index.jsp	Repetir la acción del paso 1	Huella Digital de la tabla 36 para la prueba #2	Ver resultado de la salida en la tabla 36 para la prueba #2

Tabla 36 Datos de entrada y resultado: Ingreso al sistema (Huella Digital)

Prueba #	Datos de entrada	Resultado
1	Huella digital registrada en la base de datos	Huella valida
2	Huella digital NO registrada en la base de datos	Huella invalida

3.3.2.2. Caso de prueba: Ingreso al sistema (Usuario y clave)

Para este caso de prueba se verifica el correcto funcionamiento de la segunda fase de login o ingreso al sistema, la cual se realiza a través de la solicitud del usuario y la clave.

Descripción

La siguiente tabla (Ver Tabla 37) muestra la descripción para el presente caso de prueba:

Tabla 37 Caso de prueba: Ingreso al sistema (Usuario y Clave)

Nombre	Ingreso al sistema (Usuario y clave)
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en verificar que solo si el usuario y la clave son correctos, es decir, están almacenados en la base de datos; el usuario podrá entonces ingresar a la aplicación web.
Categoría	Módulo de acceso
Pre-requisitos	Huella identificada del caso de prueba de ingreso al sistema (Huella Digital)

Acciones a realizar

La tabla 38 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de ingreso al sistema por nombre de usuario y clave.

Tabla 38 Pasos y acciones a realizar: Ingreso al sistema (Usuario y Clave)

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/ HTML/ iniciarSesion.jsp	Ingresar nombre de usuario y clave y luego hacer click en el botón "Ingresar"	Nombre de usuario y clave de la tabla 39 para la prueba #1	Ver resultado de la salida en la tabla 39 para la prueba #1

Tabla 39 Datos de entrada y resultado: Ingreso al sistema (Usuario y Clave)

Prueba #	Nombre de usuario	Clave	Resultado
1	dicaalba	dicaalba	Ingreso exitoso al sistema
2	dicaalba	123	Usuario y/o clave incorrectos
3	null	null	Usuario y/o clave incorrectos
4	vacio	Espacios en blanco	Usuario y/o clave incorrectos
5	/	clave	Usuario y/o clave incorrectos
6	:";.'?'@!	¨[]{}+?¡	Usuario y/o clave incorrectos

3.3.2.3. Caso de prueba: Recuperar usuario y/o clave

Para este caso de prueba se verifica el correcto funcionamiento para la recuperación del nombre de usuario y/o la clave.

Descripción

La siguiente tabla (Ver Tabla 40) muestra la descripción para el presente caso de prueba:

Tabla 40 Caso de prueba: Recuperar usuario y/o clave

Nombre	Recuperar usuario y/o clave
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en enviar un correo electrónico al usuario recordando sus datos de acceso en caso de que éste haya olvidado su nombre de usuario y/o clave
Categoría	Módulo de acceso
Pre-requisitos	Huella identificada del caso de prueba de ingreso al sistema (Huella Digital)

Acciones a realizar

La tabla 41 que se muestra a continuación indica los pasos y acciones a seguir para realizar el envío del correo electrónico al usuario con los datos de las credenciales de acceso (usuario y clave).

Tabla 41 Pasos y acciones a realizar: Recuperar usuario y/o clave

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/ HTML/iniciarSesion.jsp	Hacerclicken la opción “Olvido su usuario y/o clave”	Ninguno	Ver resultado de la salida en la tabla 42

Tabla 42 Datos de entrada y resultado: Recuperar usuario y/o clave

Prueba #	Resultado
1	Envió por parte del sistema del correo con los datos de ingreso al sistema (usuario y clave) y recepción del correo por parte del usuario

3.3.2.4. Caso de prueba: Editar perfil de usuario

Para este caso de prueba se verifica el correcto funcionamiento la edición del perfil de usuario

Descripción

La siguiente tabla (Ver Tabla 43) muestra la descripción para el presente caso de prueba:

Tabla 43 Caso de prueba: Editar perfil de usuario

Nombre	Editar perfil de usuario
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en verificar que puedan ser modificados los datos del usuario correctamente.
Categoría	Usuario
Pre-requisitos	El usuario debe haber pasado por el proceso de login

Acciones a realizar

La tabla 44 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de modificación de datos del usuario.

Tabla 44 Pasos y acciones a realizar: Editar perfil de usuario

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/ HTML/perfil.jsp	Ingresar los campos a ser modificados	Nombre, apellido, nombre de usuario, clave, fecha de nacimiento, teléfono, extensión, sexo	Ver resultado de la salida en la tabla 45

Tabla 45 Datos de entrada y resultado: Editar perfil de usuario

Prueba #	Datos ingresados	Resultado
1	Nombre: Diana Apellido: Alfaro Nombre de usuario: dicaalba Clave:dicaalba Fecha de Nac. 1987-10-05 Teléfono: 04261352973 Extensión: 0191 Sexo: f	Datos modificados correctamente
2	Nombre: Diana Apellido: Alfaro Nombre de usuario: dicaalba Clave:123 Fecha de Nac. 1987-10-05 Teléfono: 04261352973 Extensión: 0191 Sexo: f	Longitud mínima de la clave es de 8 caracteres
3	Nombre: null Apellido: null Nombre de usuario: null Clave:null Fecha de Nac. null Teléfono: null Extensión: null Sexo: f	Nombre, apellido, nombre de usuario y clave son obligatorios
4	Nombre: (Espacios en blanco) Apellido: (Espacios en blanco) Nombre de usuario: (Espacios en blanco) Clave:(Espacios en blanco) Fecha de Nac. (Espacios en blanco) Teléfono: (Espacios en blanco) Extensión: (Espacios en blanco) Sexo: f	Nombre, apellido, nombre de usuario y clave son obligatorios. Fecha de nacimiento, teléfono y extensión no pueden tener campos en blanco

3.3.2.5. Caso de prueba: Registrar usuario (datos personales/laborales)

Para este caso de prueba se verifica el correcto funcionamiento del registro de nuevos usuarios al sistema.

Descripción

La siguiente tabla (Ver Tabla 46) muestra la descripción para el presente caso de prueba:

Tabla 46 Caso de prueba: Registrar Usuario

Nombre	Registrar usuario
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en verificar el registro de nuevos usuarios al sistema, ingresando para ello sus datos personales
Categoría	Usuario
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios para el registro de usuarios nuevos en el sistema

Acciones a realizar

La tabla 47 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 47 Pasos y acciones a realizar: Registrar Usuario

Paso #	**Comando/URL**	**Acción**	**Datos de Entrada**	**Salida Esperada**
1	localhost:8080/ CAFD/ HTML/usuario.jsp	Ingresar los campos que pide el formulario	Nombre, apellido, cédula, departamento, cargo y tipo de usuario	Ver resultado de la salida en la tabla 48

Tabla 48 Datos de entrada y resultado: Registrar Usuario

Prueba #	**Datos ingresados**	**Resultado**
1	Nombre: Diana Apellido: Alfaro Departamento: Informática Cargo: Gerente Cédula:18030633 Tipo Usuario: Administrador	Usuario registrado. Envió de correo para registro de la huella enviado.
2	Nombre: Diana Apellido: Alfaro Departamento: null Cargo: null Cédula: null Tipo Usuario: Administrador	Debe seleccionar un departamento y un cargo
3	Nombre: null Apellido: null Departamento: null Cédula: null	Debe completar los campos

	Cargo: null Tipo Usuario: null	obligatorios

3.3.2.6. Caso de prueba: Registro de usuarios (huella dactilar)

Para este caso de prueba se verifica el correcto funcionamiento del registro de la huella dactilar de un usuario del sistema

Descripción

La siguiente tabla (Ver Tabla 49) muestra la descripción para el presente caso de prueba:

Tabla 49 Caso de prueba: Registro de huella dactilar

Nombre	Registro de huella dactilar
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en verificar que solo los usuarios que hayan sido registrados sus datos personales y laborales puedan registrar su huella dactilar en el sistema. Así como no permitir el registro de más de una huella para el mismo usuario.
Categoría	Usuario
Pre-requisitos	Registro de usuario (datos personales/laborales)

Acciones a realizar

La tabla 50 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro la huella dactilar de un usuario y otros datos.

Tabla 50 Pasos y acciones a realizar: Registro de huella dactilar

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	URL personalizada, se envía al correo electrónico después de haber realizado el registro de los datos del usuario	Ingresar los datos solicitados y colocar el dedo a ser registrado sobre el lector biométrico	Cédula, nombre de usuario, clave enviada al correo, clave nueva de la tabla 51	Ver resultado de la salida en la tabla 51

Tabla 51 Datos de entrada y resultado: Registro de huella dactilar

Prueba #	Datos ingresados	Resultado
1	Cédula: 18030633 Usuario: dicaalba Clave enviado por correo:jak3sdsa Clave: dicaalba Huella dactilar	Registro completado. Redirección a la página de inicio
2	Cédula: null Usuario: null Clave enviado por correo:null Clave: null Huella dactilar	Registro no completado, campos obligatorios
3	Cédula: 18030630 Usuario: dicaalba Clave enviado por correo:jak3sdsa	Cédula no encontrada. Huella no registrada

	Clave: dicaalba Huella dactilar	(Contacte al administrador)

3.3.2.7. Caso de prueba: Manejo de sesiones de usuario

Para este caso de prueba se verifica el correcto funcionamiento las sesiones en el sistema

Descripción

La siguiente tabla (Ver Tabla 52) muestra la descripción para el presente caso de prueba:

Tabla 52 Caso de prueba: Manejo de sesiones de usuario

Nombre	Manejo de sesiones de usuario
ID	1
Iteración	1
Nro. de Historia de usuario	4
Descripción	El presente caso de prueba consistirá en verificar que una vez identificado un usuario en el sistema, las variables de sesión que tiene asociadas se manejen correctamente
Categoría	Módulo de acceso
Pre-requisitos	Identificación en el sistema

Acciones a realizar

La tabla 53 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de manejo de sesiones en el sistema

Tabla 53 Pasos y acciones a realizar: Manejo de sesiones de usuario

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	Cualquier URL presente después del proceso de login por huella, usuario y clave	Cerrar la sesión actual	Sin datos de entrada	Ver resultado de la salida en la tabla 54
2	Cualquier URL presente después del proceso de login por huella, usuario y clave	Dejar de interactuar con el sistema por 10 minutos o más	Sin datos de entrada	Ver resultado de la salida en la tabla 54

Tabla 54 Datos de entrada y resultado: Manejo de sesiones de usuario

Prueba #	Resultado
1	Sesión cerrada completamente
2	Sesión cerrada al cabo de 10 minutos o más

3.3.2.8. Caso de prueba: Generación de reportes

Para este caso de prueba se verifica el correcto funcionamiento para la generación de reportes del sistema.

Descripción

La siguiente tabla (Ver Tabla 55) muestra la descripción para el presente caso de prueba:

Tabla 55 Caso de prueba: Generación de reportes

Nombre	Generación de reportes
ID	1
Iteración	1
Nro. de Historia de usuario	17
Descripción	El presente caso de prueba consistirá en verificar que se genere correctamente el (los) reportes solicitados en un archivo de extensión PDF
Categoría	Documentos
Pre-requisitos	Identificación en el sistema

Acciones a realizar

La tabla 56 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de generación de reportes.

Tabla 56 Pasos y acciones a realizar: Generación de reportes

Paso #	**Comando/URL**	**Acción**	**Datos de Entrada**	**Salida Esperada**
1	localhost:8080/ CAFD/HTML/ Generar_Reporte_Documento.jsp	Click en el botón "Generar PDF"	Estatus del documento en la tabla 57	Ver resultado de la salida en la tabla 57

Tabla 57 Datos de entrada y resultado: Generación de reportes

Prueba #	Estatus del documento	Resultado
1	Todos	PDF generado
2	Aceptado	PDF generado
3	Rechazo	PDF generado
4	En espera	PDF generado
5	null	Seleccione estatus del documento

3.3.2.9. Caso de prueba: Carga de archivo de certificado digital

Para este caso de prueba se verifica el correcto funcionamiento de la carga del archivo de certificado digital, el cual es necesario para la firma digital de los documentos.

Descripción

La siguiente tabla 58 muestra la descripción para el presente caso de prueba:

Tabla 58 Caso de prueba: Carga de archivo de certificado digital

Nombre	Carga de archivo de certificado digital
ID	1
Iteración	1
Nro. de Historia de usuario	6
Descripción	El presente caso de prueba consistirá en verificar la correcta carga de un certificado digital que tenga la extensión .PFX
Categoría	Firma Digital

Pre-requisitos	Contar con un certificado digital de extensión .PFX

Acciones a realizar

La tabla 59 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de la carga del certificado digital.

Tabla 59 Pasos y acciones a realizar: Carga de archivo de certificado digital

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/CAFD/HTML/opciones.jsp	Completar los datos requeridos y hacer click en "Subir"	Dueño, clave privada del certificado y el certificado de la tabla 60	Ver resultado de la salida en la tabla 61

Tabla 60 Datos de entrada y resultado: Carga de archivo de certificado digital

Prueba #	Datos ingresados	Resultado
1	Nombre del dueño del certificado: dicaalba Clave privada: dicaalba Certificado digital: archivo.pfx	Certificado cargado
2	Nombre del dueño del certificado: dicaalba Clave privada: dicaalba Certificado digital: archivo.doc	El certificado debe tener extensión .pfx

3	Nombre del dueño del certificado: dicaalba Clave privada: dicaalba Certificado digital: null	Debe añadir un certificado

3.3.2.10. Caso de prueba: Firma digital

Para este caso de prueba se verifica el correcto funcionamiento de la firma digital en un documento PDF.

Descripción

La siguiente tabla (Ver Tabla 61) muestra la descripción para el presente caso de prueba:

Tabla 61 Caso de prueba: Firma digital

Nombre	Firma digital
ID	1
Iteración	1
Nro. de Historia de usuario	7
Descripción	El presente caso de prueba consistirá en verificar el funcionamiento de la firma digital en un documento PDF
Categoría	Documentos
Pre-requisitos	Certificado digital cargado en el sistema

Acciones a realizar

La tabla 62 que se muestra a continuación indica los pasos y acciones a seguir para firmar digitalmente un documento, el cual es de extensión PDF. Para ello es necesario tener un certificado digital válido (con extensión .pfx) cargado en el sistema.

Tabla 62 Pasos y acciones a realizar: Firma Digital

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ Documentos_Recibidos.jsp	Click en la metáfora de aceptar un documento (agregar firma digital al mismo)	Datos del certificado almacenados en el sistema	Ver resultado de la salida en la tabla 63

Tabla 63 Datos de entrada y resultado: Firma Digital

Prueba #	Datos ingresados	Resultado
1	Datos del certificado de la tabla "certificado" para el usuario actual	Documento firmado
2	No tiene certificado cargado	Documento no firmado. Debe agregar su certificado digital al sistema
3	Clave privada del certificado digital incorrecta	La clave ingresada no coincide con los datos de su certificado digital

3.3.2.11. Caso de prueba: Firma digital (Con huella digital)

Para este caso de prueba se verifica el correcto funcionamiento de la firma digital en un documento PDF, previa a la verificación de la huella digital del usuario.

Descripción

La siguiente tabla (Ver Tabla 64) muestra la descripción para el presente caso de prueba:

Tabla 64 Caso de prueba: Firma digital (Con huella digital)

Nombre	Firma digital (Con huella digital)
ID	1
Iteración	1
Nro. de Historia de usuario	21
Descripción	El presente caso de prueba consistirá en verificar el funcionamiento de la firma digital en un documento PDF para el cual se procede a verificar la huella digital del usuario y la clave del certificado digital para firmar el documento PDF
Categoría	Documentos
Pre-requisitos	Certificado digital cargado en el sistema Huella digital registrada en el sistema

Acciones a realizar

La tabla 65 que se muestra a continuación indica los pasos y acciones a seguir para firmar digitalmente un documento usando la huella digital. Para ello es necesario

tener un certificado digital válido (con extensión .pfx) y la huella digital registrada en el sistema.

Tabla 65 Pasos y acciones a realizar: Firma Digital (Con huella digital)

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ Documentos_Recibidos.jsp	Click en la metáfora de aceptar un documento (agregar firma digital al mismo)	Clave privada del certificado digital Huella digital del usuario	Ver resultado de la salida en la tabla 66

Tabla 66 Datos de entrada y resultado: Firma Digital (Con huella digital)

Prueba #	Datos ingresados	Resultado
1	Clave privada del certificado digital correcta Huella digital verificada	Documento firmado
2	Clave privada del certificado digital incorrecta Huella digital verificada	Documento no firmado. Huella y/o clave incorrectas
3	Clave privada del certificado digital correcta Huella digital no verificada	Documento no firmado. Huella y/o clave incorrectas

3.3.2.12. Caso de prueba: Modificar usuario

Para este caso de prueba se verifica el correcto funcionamiento en la modificación de los datos de los usuarios del sistema.

Descripción

La siguiente tabla (Ver Tabla 67) muestra la descripción para el presente caso de prueba:

Tabla 67 Caso de prueba: Modificar Usuario

Nombre	Modificar usuario
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	Este caso de prueba consiste en verificarla modificación de los datos de un usuario
Categoría	Usuario
Pre-requisitos	Huella digital registrada en el sistema

Acciones a realizar

La tabla 68 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de modificación de datos personales de un usuario.

Tabla 68 Pasos y acciones a realizar: Modificar Usuario

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ miperfil.jsp	Ingresar los campos en el formulario que van a ser cambiados	Datos personales del usuario que pueden ser modificados	Ver resultado de la salida en la tabla 69

Tabla 69 Datos de entrada y resultado: Modificar Usuario

Prueba #	Datos ingresados	Resultado
1	Nombre: Diana Apellido: Alfaro Nombre de usuario: dicaalba Clave:dicaalba Fecha de Nac. 1987-10-05 Teléfono: 04261352973 Extensión: 0191 Sexo: f	Datos modificados correctamente
2	Nombre: Diana Apellido: Alfaro Nombre de usuario: dicaalba Clave:123 Fecha de Nac. 1987-10-05 Teléfono: 04261352973 Extensión: 0191 Sexo: f	Longitud mínima de la clave es de 8 caracteres
3	Nombre: null Apellido: null Nombre de usuario: null Clave:null Fecha de Nac. null Teléfono: null Extensión: null Sexo: f	Nombre, apellido, nombre de usuario y clave son obligatorios
4	Nombre: (Espacios en blanco) Apellido: (Espacios en blanco) Nombre de usuario: (Espacios en blanco) Clave:(Espacios en blanco) Fecha de Nac. (Espacios en blanco) Teléfono: (Espacios en blanco) Extensión: (Espacios en blanco) Sexo: f	Nombre, apellido, nombre de usuario y clave son obligatorios. Fecha de nacimiento, teléfono y extensión no pueden tener campos en blanco

3.3.2.13. Caso de prueba: Eliminar usuario

Para este caso de prueba se verifica el correcto funcionamiento de la eliminación de un usuario registrado previamente en el sistema.

Descripción

La siguiente tabla (Ver Tabla 70) muestra la descripción para el presente caso de prueba:

Tabla 70 Caso de prueba: Eliminar Usuario

Nombre	Eliminar un usuario
ID	1
Iteración	1
Nro. de Historia de usuario	10
Descripción	El presente caso de prueba consistirá en eliminar algún usuario que se encuentre registrado en el sistema
Categoría	Usuario
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios para eliminar un usuario del sistema

Acciones a realizar

La tabla 71 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba que consiste en eliminar un usuario del sistema.

Tabla 71 Pasos y acciones a realizar: Eliminar Usuario

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML /ver_usuario.jsp	Click en la metáfora de eliminar un usuario	Acción sobre la metáfora de eliminar un usuario	Ver resultado de la salida en la tabla 72

Tabla 72 Datos de entrada y resultado: Eliminar Usuario

Prueba #	Datos ingresados	Resultado
1	Eliminar un usuario diferente al usuario actual	Usuario eliminado
2	Eliminar el usuario actual	No puede eliminar su usuario

3.3.2.14. Caso de prueba: Crear departamento

Para este caso de prueba se verifica el correcto funcionamiento del registro de un departamento al sistema.

Descripción

La tabla 73 muestra la descripción para el presente caso de prueba:

Tabla 73 Caso de prueba: Crear departamento

Nombre	Crear departamento
ID	1
Iteración	1
Nro. de Historia de usuario	13
Descripción	El presente caso de prueba consistirá en crear un nuevo departamento al sistema
Categoría	Manejo de usuario
Pre-requisitos	Tener el rol de Administrador o Administrador de usuario para la creación de departamentos en el sistema

Acciones a realizar

La tabla 74 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba que consiste en crear un departamento y registrarlo en el sistema.

Tabla 74 Pasos y acciones a realizar: Crear Departamento

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML /configuracion_usuario.jsp	Ingresar los campos solictados	Nombre del departamento a crear	Ver resultado de la salida en la tabla 75

Tabla 75 Datos de entrada y resultado: Crear Departamento

Prueba #	Datos ingresados	Resultado
1	Departamento: Informática	Departamento creado
2	Departamento: null	Debe completar los campos obligatorios
3	Departamento: (espacios en blanco)	Debe completar los campos obligatorios
4	Departamento: Informática	Este departamento ya existe

3.3.2.15. Caso de prueba: Crear cargo

Para este caso de prueba se verifica el correcto funcionamiento para la creación de un cargo asociado a un departamento creado previamente en el sistema.

Descripción

La tabla 76 muestra la descripción para el presente caso de prueba:

Tabla 76 Caso de prueba: Crear Cargo

Nombre	Crear cargo
ID	1
Iteración	1
Nro. de Historia de usuario	13
Descripción	El presente caso de prueba consistirá en crear un cargo asociado a un departamento creado previamente

Categoría	Usuario
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios para la creación de un nuevo cargo

Acciones a realizar

La tabla 77 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de creación de un cargo asociado a un departamento creado previamente en el sistema.

Tabla 77 Pasos y acciones a realizar: Crear cargo

Paso #	**Comando/URL**	**Acción**	**Datos de Entrada**	**Salida Esperada**
1	localhost:8080/ CAFD/HTML/ configuracion_usuario.jsp	Ingresar los campos que pide el formulario	Nombre, apellido, cédula, departamento, cargo y tipo de usuario	Ver resultado de la salida en la tabla 78

Tabla 78 Datos de entrada y resultado: Crear Cargo

Prueba #	**Datos ingresados**	**Resultado**
1	Departamento: Informática Cargo: Gerente	Cargo registrado
2	Departamento: null	Debe completar los campos

	Cargo: null	obligatorios
3	Departamento: (espacios en blanco) Cargo: (espacios en blanco)	Debe completar los campos obligatorios
4	Departamento: Informática Cargo: Programador I	Cargo registrado
5	Departamento: Informática Cargo: Gerente	Este cargo ya existe

3.3.2.16. Caso de prueba: Eliminar puesto de trabajo

Para este caso de prueba se verifica el correcto funcionamiento al momento de eliminar un puesto de trabajo.

Descripción

La siguiente tabla (Ver Tabla 79) muestra la descripción para el presente caso de prueba:

Tabla 79 Caso de prueba: Eliminar puesto de trabajo

Nombre	Eliminar puesto de trabajo
ID	1
Iteración	1
Nro. de Historia de usuario	13
Descripción	El presente caso de prueba consistirá en eliminar un puesto de trabajo del sistema, donde un puesto de trabajo es la relación departamento-cargo.
Categoría	Usuario

Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios para eliminar un puesto de trabajo.

Acciones a realizar

La tabla 80 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 80 Pasos y acciones a realizar: Eliminar puesto de trabajo

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	www.ciens.ucv.ve/ SignedDigitalSecurity/ historial_dp_usuario.jsp	Click en la metáfora de eliminar un puesto de trabajo ✖	Acción sobre la metáfora de eliminar un puesto de trabajo	Ver resultado de la salida en la tabla 81

Tabla 81 Datos de entrada y resultado: Eliminar puesto de trabajo

Prueba #	Datos ingresados	Resultado
1	Eliminar un puesto de trabajo asignado a uno o varios usuarios	Puesto de trabajo con dependencia, no se puede eliminar
2	Eliminar un puesto de trabajo no asignado	Puesto de trabajo eliminado

3.3.2.17. Caso de prueba: Modificar puesto de trabajo

Para este caso de prueba se verifica el correcto funcionamiento en la modificación de puestos de trabajos creados previamente en el sistema.

Descripción

La siguiente tabla (Ver Tabla 82) muestra la descripción para el presente caso de prueba:

Tabla 82 Caso de prueba: Modificar puesto de trabajo

Nombre	Modificar puesto de trabajo
ID	1
Iteración	1
Nro. de Historia de usuario	13
Descripción	El presente caso de prueba consistirá en modificar un puesto de trabajo creado previamente en el sistema
Categoría	Usuario
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios para el registro de usuarios nuevos en el sistema

Acciones a realizar

La tabla 4.53 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba que consiste en modificar un puesto de trabajo creado previamente en el sistema.

Tabla 83 Pasos y acciones a realizar: Modificar puesto de trabajo

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ historial_dp_usuario.jsp	Ingresar los campos que pide el formulario	Nombre del cargo	Ver resultado de la salida en la tabla 84

Tabla 84 Datos de entrada y resultado: Modificar puesto de trabajo

Prueba #	Datos ingresados	Resultado
1	Departamento: Informática Cargo: Gerente Nuevo Cargo: Programador I	Cargo modificado
2	Departamento: Informática Cargo: Gerente Nuevo Cargo: null	Debe completar los campos obligatorios
3	Departamento: Informática Cargo: Gerente Nuevo Cargo: (espacios en blanco)	Debe completar los campos obligatorios

3.3.2.18. Caso de prueba: Crear categoría

Para este caso de prueba se verifica el correcto funcionamiento al momento de crear una categoría de documento en el sistema.

Descripción

La siguiente tabla (Ver Tabla 85) muestra la descripción para el presente caso de prueba:

Tabla 85 Caso de prueba: Crear categoría

Nombre	Crear categoría
ID	1
Iteración	1
Nro. de Historia de usuario	12
Descripción	El presente caso de prueba consistirá en registrar una categoría de documento en el sistema
Categoría	Gestión de documentos
Pre-requisitos	Tener el rol de Administrador o Administrador de documentos para el registro de una categoría de documento nuevos en el sistema

Acciones a realizar

La tabla 86 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba que consiste en registrar una categoría de documento en el sistema.

Tabla 86 Pasos y acciones a realizar: Crear categoría

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/	Ingresar los campos que	Nombre de la categoría a	Ver resultado de la salida en

Prueba #	Datos ingresados	Resultado
1	Categoría: memos	Categoría creada
2	Categoría: null	Debe completar los campos obligatorios
3	Categoría: (espacios en blanco)	Debe completar los campos obligatorios
4	Categoría: memos	Nombre de categoría ya existe

	flujocc.jsp	pide el formulario	registrar	la tabla 87

Tabla 87 Datos de entrada y resultado: Crear categoría

3.3.2.19. Caso de prueba: Crear flujo

Para este caso de prueba se verifica el correcto funcionamiento para la creación de un flujo sobre una categoría de documento ya creada.

Descripción

La siguiente tabla (Ver Tabla 88) muestra la descripción para el presente caso de prueba:

Tabla 88 Caso de prueba: Crear flujo

Nombre	Crear flujo
ID	1
Iteración	1
Nro. de Historia de	12

usuario	
Descripción	El presente caso de prueba consiste en verificar la creación de un flujo sobre una categoría
Categoría	Gestión de documentos
Pre-requisitos	Tener el rol de Administrador o Administrador de documentos para crear un nuevo flujo de un documento

Acciones a realizar

La tabla 89 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba que consiste en la creación de un flujo sobre una categoría de documento ya creada.

Tabla 89 Pasos y acciones a realizar: Crear flujo

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ flujocf.jsp	Ingresar los campos que pide el formulario	Nombre de la categoría, nombre del flujo, departamento, cargo y nombre de los integrantes del flujo	Ver resultado de la salida en la tabla 90

Tabla 90 Datos de entrada y resultado: Crear flujo

Prueba #	Datos ingresados	Resultado
	Categoría: memos Flujo: RRHH	

1	Departamento: Informática Cargo: Gerente Nombre: Diana Alfaro	Flujo creado
2	Categoría: null Flujo: null Departamento: null	Debe completar los campos obligatorios
3	Flujo: (espacios en blanco)	Debe seleccionar un departamento, un cargo y un tipo de usuario

3.3.2.20. Caso de prueba: Eliminar flujo

Para este caso de prueba se verifica el correcto funcionamiento en la eliminación de un flujo.

Descripción

La siguiente tabla (Ver Tabla 91) muestra la descripción para el presente caso de prueba.

Tabla 91 Caso de prueba: Eliminar flujo

Nombre	Eliminar flujo
ID	1
Iteración	1
Nro. de Historia de usuario	12
Descripción	El presente caso de prueba consistirá en eliminar un flujo creado
Categoría	Gestión de documentos
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios para el registro de

	usuarios nuevos en el sistema

Acciones a realizar

La tabla 92que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba para eliminar un flujo que ha sido creado previamente en el sistema.

Tabla 92 Pasos y acciones a realizar: Eliminar flujo

Paso #	**Comando/URL**	**Acción**	**Datos de Entrada**	**Salida Esperada**
1	localhost:8080/ CAFD/HTML/flujo.jsp	Click en la metáfora de eliminar un flujo 	Acción sobre la metáfora de eliminar un flujo	Ver resultado de la salida en la tabla 93

Tabla 93 Datos de entrada y resultado: Eliminar flujo

Prueba #	Datos ingresados	Resultado
1	Eliminar flujo con estatus: No asignado	Se eliminó correctamente
2	Eliminar flujo con estatus: Contingencia	Se eliminó correctamente
3	Eliminar flujo con estatus: Aceptado	Se eliminó correctamente
4	Eliminar flujo con estatus: Detenido	Error. Tiene dependencia

3.3.2.21. Caso de prueba: Enviar documento

Para este caso de prueba se verifica el correcto funcionamiento del registro de nuevos usuarios al sistema.

Descripción

La tabla 94 muestra la descripción para el presente caso de prueba:

Tabla 94 Caso de prueba: Enviar documento

Nombre	Enviar documento
ID	1
Iteración	1
Nro. de Historia de usuario	11
Descripción	El presente caso de prueba consistirá en verificar el correcto envió de un documento a un flujo creado
Categoría	Gestión de documentos
Pre-requisitos	Ser un usuario registrado en el sistema

Acciones a realizar

La tabla 95 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba que consiste en verificar el correcto envió de un documento a un flujo creado.

Tabla 95 Pasos y acciones a realizar: Enviar documento

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/docs.jsp	Ingresar los campos que pide el formulario	Clave del correo, categoría, flujo, asunto y cuerpo del mensaje, archivo(s) adjunto (s)	Ver resultado de la salida en la tabla 96

Tabla 96 Datos de entrada y resultado: Enviar documento

Prueba #	Datos ingresados	Resultado
1	Clave: ******** Categoría: memos Flujo: RRHH Asunto: Memo colectivo Cuerpo: mensaje Memo.pdf	Correo enviado
2	Clave: null Categoría: null Flujo: null Asunto: null Cuerpo: null Memo.pdf	Debe completar los campos obligatorios
3	Clave: (espacios en blanco) Categoría: (espacios en blanco) Flujo: (espacios en blanco) Asunto: (espacios en blanco) Cuerpo: (espacios en blanco) Memo.pdf	Debe completar los campos obligatorios
	Clave: ********	

4	Categoría: memos Flujo: RRHH Asunto: Memo colectivo Cuerpo: mensaje Sin adjunto	Ingrese un archivo adjunto al correo

3.3.2.22. Caso de prueba: Acceso físico

Para este caso de prueba se verifica el correcto funcionamiento del control de acceso físico a una ubicación registrada.

Descripción

La siguiente tabla (Ver Tabla 97) muestra la descripción para el presente caso de prueba:

Tabla 97 Caso de prueba: Acceso físico

Nombre	Acceso físico
ID	1
Iteración	1
Nro. de Historia de usuario	15
Descripción	El presente caso de prueba consistirá en verificar el correcto funcionamiento del control de acceso físico a una ubicación registrada.
Categoría	Gestión de Control de Acceso
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 98 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba.

Tabla 98 Pasos y acciones a realizar: Acceso físico

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ AccesoFisico.jsp	Click en botón "Verificar" y colocar la huella en el lector	Huella dactilar	Ver resultado de la salida en la tabla 99

Tabla 99 Datos de entrada y resultado: Acceso físico

Prueba #	Datos ingresados	Resultado
1	Huella registrada en el sistema y con permisos para acceder a esa ubicación	Registro su entrada/salida
2	Huella registrada en el sistema y sin permisos para acceder a esa ubicación	Acceso denegado
3	Huella NO registrada en el sistema	Huella no válida

3.3.2.23. Caso de prueba: Crear ubicación

Para este caso de prueba se verifica el correcto funcionamiento del registro de una ubicación física.

Descripción

La tabla 100 muestra la descripción para el presente caso de prueba:

Tabla 100 Caso de prueba: Crear ubicación

Nombre	Crear ubicación
ID	1
Iteración	1
Nro. de Historia de usuario	15
Descripción	El presente caso de prueba consistirá en verificar la correcta creación funcionamiento del registro de una ubicación física.
Categoría	Gestión de Control de Acceso
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 101 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba.

Tabla 101 Pasos y acciones a realizar: Crear ubicación

Paso #	**Comando/URL**	**Acción**	**Datos de Entrada**	**Salida Esperada**
1	localhost:8080/ CAFD/HTML/ manejo_ubicacion1.jsp	Ingresar los campos que pide el	Dirección MAC, ubicación, posición, permiso,	Ver resultado de la salida en la tabla 102

		formulario	horario	

Prueba #	Datos ingresados	Resultado
1	MAC: 74-F0-6D-83-B5-76 Ubicación: Sala de servidores Posición: entrada Permiso: si Horario: A	Ubicación creada
2	MAC: null Ubicación: null Posición: null Permiso: null Horario: null	Debe completar los campos obligatorios

Tabla 102 Datos de entrada y resultado: Crear ubicación

3.3.2.24. Caso de prueba: Eliminar ubicación

Para este caso de prueba se verifica el correcto funcionamiento en la eliminación de una ubicación.

Descripción

La tabla 103 muestra la descripción para el presente caso de prueba:

Tabla 103 Caso de prueba: Eliminar ubicación

Nombre	Eliminar ubicación
ID	1
Iteración	1
Nro. de Historia de usuario	15
Descripción	El presente caso de prueba consistirá en eliminar una ubicación
Categoría	Gestión de control de acceso

Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 104 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 104 Pasos y acciones a realizar: Eliminar ubicación

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ manejo_ubicacion2.jsp	Click en la metáfora de eliminar una ubicación	Acción sobre la metáfora de eliminar una ubicación	Ver resultado de la salida en la tabla 105

Tabla 105 Datos de entrada y resultado: Eliminar ubicación

Prueba #	Datos ingresados	Resultado
1	Ubicación con usuarios con permisos asignados	Ubicación eliminada
2	Ubicación con usuarios sin permisos asignados	Error. Tiene dependencia

3.3.2.25. Caso de prueba: Crear horario de trabajo

Para este caso de prueba se verifica el correcto funcionamiento del registro de un nuevo horario de trabajo.

Descripción

La tabla 106 muestra la descripción para el presente caso de prueba:

Tabla 106 Caso de prueba: Crear horario de trabajo

Nombre	Crear horario de trabajo
ID	1
Iteración	1
Nro. de Historia de usuario	12
Descripción	El presente caso de prueba consistirá en verificar el correcto funcionamiento del registro de un nuevo horario de trabajo.
Categoría	Gestión de Control de Acceso
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 107 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 107 Pasos y acciones a realizar: Crear horario de trabajo

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ manejo_horario1.jsp	Ingresar los campos que pide el formulario	Nombre del horario, hora de inicio y fin, días del horario	Ver resultado de la salida en la tabla 108

Tabla 108 Datos de entrada y resultado: Crear horario de trabajo

Prueba #	Datos ingresados	Resultado
1	Nombre del horario: A Inicio: 8:00 fin: 16:00 días: lunes	Horario registrado
2	Nombre del horario: null Inicio: null fin: null días: null	Debe completar los campos obligatorios
3	Nombre del horario: A Inicio: 8:00 fin: 16:00 días: null	Debe completar los campos obligatorios

3.3.2.26. Caso de prueba: Eliminar horario de trabajo

Para este caso de prueba se verifica el correcto funcionamiento al eliminar un horario de trabajo.

Descripción

La siguiente tabla (Ver Tabla 109) muestra la descripción para el presente caso de prueba:

Tabla 109 Caso de prueba: Eliminar horario de trabajo

Nombre	Eliminar horario de trabajo
ID	1
Iteración	1
Nro. de Historia de usuario	12
Descripción	El presente caso de prueba consistirá en verificar el correcto funcionamiento al eliminar un horario de trabajo
Categoría	Gestión de Control de Acceso
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 4.80 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 110 Pasos y acciones a realizar: Eliminar horario de trabajo

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ manejo_horario2.jsp	Click en la metáfora de eliminar un horario de	Acción sobre la metáfora de eliminar un horario de trabajo	Ver resultado de la salida en la tabla 111

		trabajo		

Tabla 111 Datos de entrada y resultado: Eliminar horario de trabajo

Prueba #	Datos ingresados	Resultado
1	Horario asignado	Error. Tiene dependencia
2	Horario sin asignar	Horario eliminado

3.3.2.27. Caso de prueba: Crear permisos

Para este caso de prueba se verifica el correcto funcionamiento del registro de permisos de acceso.

Descripción

La siguiente tabla (Ver Tabla 112) muestra la descripción para el presente caso de prueba:

Tabla 112 Caso de prueba: Crear permisos

Nombre	Crear permisos
ID	1
Iteración	1
Nro. de Historia de usuario	12
Descripción	El presente caso de prueba consistirá en verificar el registro de permisos de acceso
Categoría	Gestión de Control de Acceso
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 113 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 113 Pasos y acciones a realizar: Crear permisos

Paso #	Comando/URL	Acción	Datos de Entrada	Salida Esperada
1	localhost:8080/ CAFD/HTML/ manejo_permiso1.jsp	Ingresar los campos que pide el formulario	Departamento, Cargo, Nombre, Apellido, Estado, Ubicación, Fecha Inicio, Fecha Final, Horario	Ver resultado de la salida en la tabla 114

Tabla 114 Datos de entrada y resultado: Crear permisos

Prueba #	Datos ingresados	Resultado
1	Departamento: Informática Cargo: Gerente Nombre: Diana Apellido: Alfaro Estado: Activo Ubicación: Sala de servidores Fecha Inicio:05/10/2013 Fecha Final: 25/10/2013 Horario: A	Permiso asignado
2	Departamento: Informática Cargo: Gerente Nombre: Diana	Debe completar los campos

	Apellido: Alfaro Estado: Activo Ubicación: Sala de servidores Fecha Inicio: null Fecha Final: null Horario: A	obligatorios

3.3.2.28. Caso de prueba: Eliminar permisos

Para este caso de prueba se verifica el correcto funcionamiento para eliminar un permiso.

Descripción

La siguiente tabla (Ver Tabla 115) muestra la descripción para el presente caso de prueba:

Tabla 115 Caso de prueba: Eliminar permisos

Nombre	Eliminar permisos
ID	1
Iteración	1
Nro. de Historia de usuario	12
Descripción	El presente caso de prueba consistirá en verificar la eliminación de un permiso
Categoría	Gestión de Control de Acceso
Pre-requisitos	Tener el rol de Administrador o Administrador de usuarios

Acciones a realizar

La tabla 116 que se muestra a continuación indica los pasos y acciones a seguir para realizar el caso de prueba de registro de datos personales y laborales de un usuario.

Tabla 116 Pasos y acciones a realizar: Eliminar permisos

Paso #	**Comando/URL**	**Acción**	**Datos de Entrada**	**Salida Esperada**
1	localhost:8080/ CAFD/HTML/ manejo_permiso2.jsp	Click en la metáfora de eliminar un permiso ✖	Acción sobre la metáfora de eliminar un permiso	Ver resultado de la salida en la tabla 117

Tabla 117 Datos de entrada y resultado: Eliminar permisos

Prueba #	**Datos ingresados**	**Resultado**
1	Selección de un permiso	Permiso eliminado

3.3.3. Resultados de las pruebas de aceptación

La tabla 118 muestra un resumen donde se presenta numéricamente los casos de pruebas escritos, ejecutados y exitosos, mientras que la figura 4.6 muestra gráficamente los casos de pruebas escritos, ejecutados y exitosos.

Tabla 118 Cuadro de resumen de los casos y resultados de las pruebas de aceptación

Pruebas escritas	8
Pruebas con JUnit	17
Pruebas exitosas	25

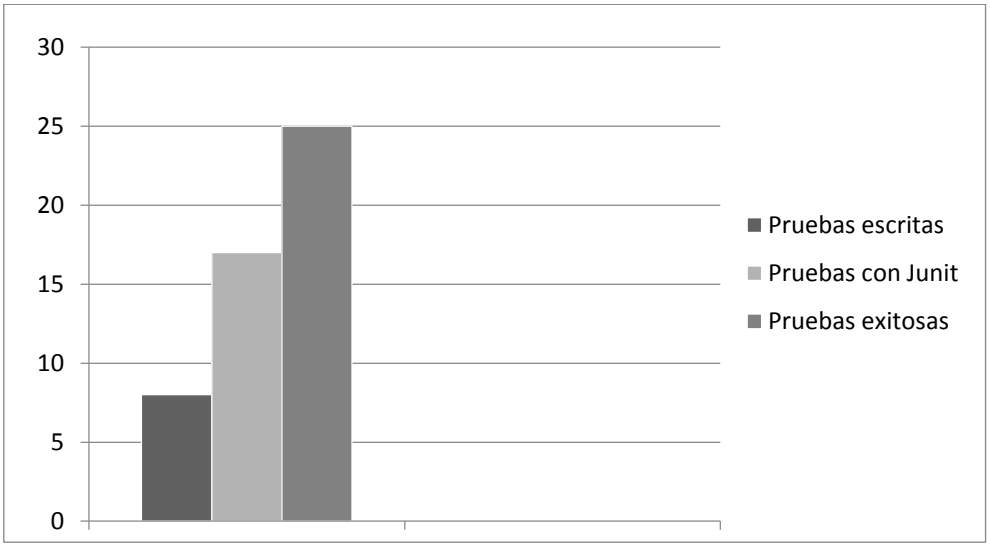


Figura 125: Gráfico resumen de los casos y resultados de las pruebas de aceptación

En las pruebas de aceptación realizadas en FUNVISIS se tuvieron los siguientes inconvenientes al momento durante la ejecución de las mismas:

- ✓ El formato del certificado que poseen se encuentra en formato .p12, para lo cual es necesario que realicen la conversión al formato .pfx para utilizarlo en el sistema.

- ✓ La versión desactualizada del navegador (Firefox Mozilla 3.6.20 año 2009) ocasionó que ciertos elementos de interfaz no se visualizarán correctamente, esto debido a que para el desarrollo de la aplicación

web se utilizó HTML5 y CSS3. Igualmente algunos mensajes en JavaScript no se mostraban correctamente. A lo que FUNVISIS solucionará con la actualización de la versión del navegador.

Tanto en la tabla 118 como en la figura 126 se puede concluir que los casos de pruebas realizados al sistema resultaron 100% exitosos.

CONCLUSIONES

En este Trabajo Especial de Grado se controló el acceso a espacios físicos así como la automatización en el proceso de gestión y autenticación de documentos a través del diseño y desarrollo de una aplicación web basado en la integración de la tecnología de huella dactilar y los certificados digitales.

Para realizar la aplicación web, se recurrió a una adaptación del 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 realizó 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. Pero teniendo como principal desventaja poca iteración con usuario final en el proceso del desarrollo.

Aunque existen otras características biométricas mejores pero más costosas, la biometría basada en huella dactilar es una buena forma de autenticar usuarios a bajo costo, debido a que valida características inherentes al usuario y que teóricamente el único que puede tener tales características es el verdadero usuario, de una forma eficiente.

De igual manera, el uso de certificados digitales, provee un excelente medio para garantizar la autenticidad de un documento. El uso de la verificación de la huella dactilar antes de proceder a firmar digitalmente un documento con un certificado digital brinda un mayor nivel de seguridad a la aplicación web, permitiendo que sólo el dueño del certificado digital sea quien firme el documento y se gestionó la manipulación de documentos que se traslada de un departamento a otro dentro de

FUNVISIS mediante un módulo que permite enviar varios documentos , adjuntar el certificado en el documento y observar el estatus de dichos documentos .

Tomando en cuenta los puntos anteriores, en la aplicación web también se controló y gestionó el acceso físico mediante autenticación de huella dactilares de esta forma se mantuvo bitácoras de entrada y salidas a un determinado espacio físico. Se logró implementar como medida de seguridad con el uso de la huella dactilar, usuario y clave para el acceso lógico del sistema.

Para la implementación de este sistema se hizo con java, es un poderoso lenguaje que permite realizar aplicaciones a nivel de escritorio como aplicaciones web, su amplio uso en diversos ámbitos y la posibilidad de integrar distintos SDK durante el desarrollo de la aplicación permitió incrementar los niveles de producción del equipo de desarrollo así como minimizar los tiempos de entrega de código funcional.

El conjunto de pruebas realizadas durante la aplicación de la metodología XP, permitieron realizar las correcciones recomendadas por el usuario y así reducir las ocurrencias de errores al finalizar el desarrollo de la aplicación.

LIMITACIONES

Entre las limitaciones que se hallaron en la realización del Trabajo Especial de Grado se encuentran:

No se adquirió certificados digitales válidos y reconocidos por entidades certificadoras como VeriSign, lo que limito a la aplicación web al uso de certificados digitales generados con herramientas como OpenSSL, certificados que al no estar reconocidos no son válidos para los exploradores de Internet o visualizadores de PDF. Los mecanismos utilizados para la generación de los certificados los exporta con formato pkcs#12. El formato pkcs#11 que utiliza tarjetas o módulos que contienen el certificado digital no fue implementado en el sistema debido a que se encuentra fuera del alcance de este Trabajo Especial de Grado por la falta de un certificado digital en ese formato.

De la misma forma, al no adquirir un certificado digital reconocido no fue posible eliminar los mensajes de advertencia en los applets para la captura de huella, debido a que por medidas de seguridad Java bloquea la ejecución automática de los applets que no estén firmados por entes reconocidos.

Otro punto importante es la adquisición del SDK para el manejo de las huellas dactilares, debido a que los SDK bajo la filosofía de SL para el reconocimiento biométrico basados en huella dactilar no son lo suficientemente estables y confiables, se tuvo que optar por un SDK de software propietario, que sumista un SDK de prueba valido por un determinado tiempo.

Por último encriptar la información de se maneja dentro del sistema, como medida de seguridad.

TRABAJOS FUTUROS

Algunas de las mejoras y trabajos futuros que pudieran implementarse en el sistema desarrollado en este Trabajo Especial de Grado, incluyen:

- Permitir firmar digitalmente con otros formatos de certificado digital. Por ejemplo, el formato pkcs#11.
- Utilizar algún mecanismo de encriptación sobre el documento PDF durante el envió del mismo en el sistema. Un mecanismo de encriptación podría darse utilizando el hash de la huella digital del autor del documento.
- La clave con la que se genera el certificado digital pudiera ser un segmento de datos de la huella digital del usuario cifrado con algún algoritmo. Aunque esto repercuta en la adquisición de certificados digitales ya que como paso previo se deberá tener los datos de la huella dactilar.
- Modificar la estructura de un certificado digital X.509 v3 para que acepte como campo las minucias de una huella digital.

RECOMENDACIONES FINALES

El campo de la seguridad informática es muy dinámico debido a que cada día surgen nuevos tipos de ataque que amenazan los esquemas de protección más recientes; los que hoy se creen virtualmente invulnerables mañana pueden dejar de serlo. Todo esto conduce a mecanismos de seguridad más robustos para los que eventualmente surgirán ataques que los dejarán obsoletos, y así sucesivamente.

Es por ello que se debe estar constantemente actualizado en cuanto a nuevos para la seguridad y mantener actualizados los sistemas.

A continuación se lista una serie de recomendaciones a tener en cuenta:

- Colocar los lectores de huella dactilar en los lugares donde se necesite controlar el acceso a espacios físicos.
- Para el acceso lógico al sistema web será necesario que el personal también cuente con un lector de huellas conectado a su computador.
- Para mayor eficiencia del dispositivo digital deben de tener en cuenta que el dedo tendrá que estar limpio al ingresar su huella así como limpiar el lente óptico del dispositivo regularmente.
- Mantenimiento constante al sistema, actualización de la base de datos.
- Adquirir las licencias necesarias para el uso del sistema biométrico.
- Para mayor seguridad colocar una cámara digital si por algún motivo se llega a dañar el lector de huella o robado poder determinar quién fue el culpable.
- Adquirir certificados válidos por entes reconocidos mundialmente.

REFERENCIAS BIBLIOGRÁFICAS

- RAE, (2001): Diccionario de la Real Academia, 22ª Edición (2001),
- Trigero, M. (2011): Sistema De Control Y Gestión De Personal Para Pymes, Basado En Sistemas Biométricos. [Versión electrónica].
  Consultado el 20 de septiembre de 2012. Disponible en:
  http://www.dspace.espol.edu.ec/handle/123456789/19200
- Jean-Marc Royer (2007). Seguridad en la informática de empresa: riesgos, amenazas, prevención y soluciones. (2007) .Ediciones ENI
- Meyers Nathan (2000), Pág.276 Programación Java en Linux, Edición Especial, Prentice Hall, Madrid.
- ALG SISTEMAS (2007). Introducción a la biometría por huella dactilar, Tecnologías Biométricas de Identificación. [Artículo en línea].
  Consultado el 15 de mayo de 2012. Disponible en:
  http://www.algdrainvac.com/PRESENTA-TECNOBIO-TB-distribuidores-2.pdf
- Rebecca Murphey (2012).Fundamentos de Jquery. [Versión electrónica].
  Consultado el 10 de octubre de 2012.Disponible en:
  http://librojquery.com/
- Categorías de Software Libre y No Libre. [Versión electrónica].
  Consultado el 10 de octubre de 2012.
  Disponible en:
  http://www.gnu.org/philosophy/categories.es.html#OpenSource
- Gaceta oficial Nº 38.095 (2004), Decreto N° 3.390
- SIR (2001). [Versión electrónica].
  Consultado el 25 de enero de 2013. Disponible en:
  http://sir.com.ve/site/index.html
- BIOCOM (2012). La firma digital de documentos médicos informatizados [Versión electrónica].

Consultado el 29 de enero de 2013. Disponible en:
http://www.biocom.com/informatica_medica/legalrec_firma_digital.html

- Blackberryvzla (2012). Firma para digital v.1.4.1 para BlackBerry [Versión electrónica].
  Consultado el 29 de enero de 2013. Disponible en:
  http://www.blackberryvzla.com/2012/02/firma-para-digital-v141-para-blackberry.html
- Isigma Asesoría Tecnológica (2010). Firma electrónica para certificados digitales. [Versión electrónica].
  Consultado el 1 de febrero de 2013. Disponible en:
  http://www.clicksignworld.com/es
- Josef Cacek (2013).JSIGNPDF. [Versión electrónica].
  Consultado el 1 de febrero de 2013. Disponible en:
  http://jsignpdf.sourceforge.net/
- Novenca Security Systems (2012). [Versión electrónica].
  Consultado el 1 de febrero de 2013. Disponible en:
  http://www.novenca.com
- Adobe (2012). [Versión electrónica].
  Consultado el 14 de febrero de 2013.
  Disponible en:
  http://www.adobe.com/es/products/acrobat/electronic-signatures-e-signatures.html
- BioTrackCorp (2012) [Versión electrónica].
  Consultado el 14 de febrero de 2013. Disponible en:
  http://www.biotracksoftware.com/espanol/espanol.htm
- GBA (2013) [Versión electrónica].
  Consultado el 2 de agosto de 2013. Disponible en:
  http://www.mseg.gba.gov.ar/Dap2/sistema%20dactil%20introduccion.html
- criminalistaenred (2013) [Versión electrónica].

Consultado el 2 de agosto de 2013. Disponible en:
http://www.criminalistaenred.com.ar/Lectores%20de%20huellas.html

- INTECO (2011) [Versión electrónica].
  Consultado el 2 de agosto de 2013. Disponible en:
  www.inteco.es
- BECK (2000) Extreme Programming Explained: Embrance Change. Massachusetts: Addison-Wesley.
- CRISPIN Lisa. 2001. Extreme Rules of the Road [Versión electrónica]
  Consultado el 5 de septiembre de 2013.Disponible en internet: http://www.testing.com/agile/crispin-xp-article.pdf
- CRISPIN, L.; TIP HOUSE. 2002. Testing Extreme Programming. Massachusetts: Addison-Wesley.
- JEFFRIES, Ron: Extreme Testing [Versión electrónica]
  Consultado el 12 de abril de 2012. Disponible en: http://www.xprogramming.com/publications/SP99 Extreme for Web.pdf
- Neurotechnology. [Versión electrónica].
  Consultado el 4 de octubre de 2013. Disponible en:
  www.neurotechnology.com
- Beck, K. A. (2004). Extreme Programming Explained.
  Addison Wesley
- BIOAPI [Versión electrónica].
  Consultado el 4 de octubre de 2013. Disponible en:
  http://www.bioapi.org/
- fprint [Versión electrónica].
  Consultado el 4 de octubre de 2013. Disponible en:
  http://www.freedesktop.org/wiki/Software/fprint/

- William Stallings (2004). Fundamentos de seguridad en redes: aplicaciones y estándares. 2° Edición Pearson

ANEXOS

Prueba de usuario

Prueba de usuario que se realizó en FUNVISIS, con el usuario final.

ID_CP: 1	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Registro de departamentos y cargos de la organización	
Pre-requisitos:	Tener el rol de Administrador o Administrador de usuario para la creación de departamentos en el sistema	
URL:	http://localhost:8888/CAFD/HTML/configuracion_usuario.jsp	
Pasos o secuencia lógica:	Ingresar nombre de departamento (no creado antes) Ingresar nombre del cargo asociado a ese departamento	
Datos de Entrada:	Nombre de departamento: CEDI Nombre del cargo asociado: Jefe (E) del CEDI	
Salida Esperada:	Se guardó correctamente	
Resultados:	Se guardó correctamente	
Observaciones: El campo sólo acepta letras, los caracteres especiales no. Por tal razón, el cargo asociado al departamento fue "Jefe E del CEDI"		
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**		

Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 2	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Registro de departamentos y cargos de la organización	
Pre-requisitos:	Tener el rol de Administrador o Administrador de usuario para la creación de departamentos en el sistema	
URL:	http://localhost:8888/CAFD/HTML/configuracion_usuario.jsp	
Pasos o secuencia lógica:	Ingresar nombre de departamento (duplicado) Ingresar nombre del cargo asociado a ese departamento (duplicado)	
Datos de Entrada:	Nombre de departamento: Informatica Nombre del cargo asociado: Programador	
Salida Esperada:	Error. Datos duplicados	
Resultados:	Mensaje de error: “Error. Datos duplicados”	
Observaciones: Sin observaciones		
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**		
Estado:		

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 3	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Registro de departamentos y cargos de la organización	
Pre-requisitos:	Tener el rol de Administrador o Administrador de usuario para la creación de departamentos en el sistema	
URL:	http://localhost:8888/CAFD/HTML/configuracion_usuario.jsp	
Pasos o secuencia lógica:	Ingresar nombre de departamento Ingresar nombre del cargo asociado a ese departamento (duplicado)	
Datos de Entrada:	Nombre de departamento: _____ Nombre del cargo asociado: _____	
Salida Esperada:	Se guardó correctamente	
Resultados:		
Observaciones: ESTA PRUEBA NO LA HICE, NO ESTÁ CLARA: Para esta prueba debo crear un departamento nuevo? Si el departamento puede ser duplicado y el cargo es duplicado, es la misma prueba 2. Si el departamento es uno nuevo y el cargo está duplicado, a pesar de que es un cargo asociado a otro departamento me dice que hay un error, pues el cargo está duplicado.		

(Ejecutor de la prueba)	**(Responsable del sistema)**	**(Responsable del sistema)**
Estado:		
Concluido	x **En proceso**	x **Rechazado**

ID_CP: 4	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Registro de usuarios	
Pre-requisitos:	Tener el rol de Administrador o Administrador de usuarios para el registro de usuarios nuevos en el sistema	
URL:	http://localhost:8888/CAFD/HTML/usuario.jsp	
Pasos o secuencia lógica:	Ingresar nombre Ingresar apellido Ingresar cédula (no duplicada) Ingresar nombre de usuario Ingresar departamento Ingresar cargo Seleccionar tipo de usuario	

Datos de Entrada:	Ingresar nombre: Daniel Ingresar apellido: Moreno Ingresar cédula (no duplicada):10823354 Ingresar nombre de usuario: dmoreno Ingresar departamento: CEDI Ingresar cargo: Jefe E del CEDI Seleccionar tipo de usuario: Usuario
Salida Esperada:	Usuario registrado. Envío de correo con la clave para el ingreso al sistema
Resultados:	Usuario registrado. Envío de correo con la clave para el ingreso al sistema
Observaciones: Sin observaciones	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 5	**Módulo a probar:**	**Fecha:**

	Módulo de gestión de usuarios	07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Registro de usuarios	
Pre-requisitos:	Tener el rol de Administrador o Administrador de usuarios para el registro de usuarios nuevos en el sistema	
URL:	http://localhost:8888/CAFD/HTML/usuario.jsp	
Pasos o secuencia lógica:	Ingresar nombre Ingresar apellido Ingresar cédula (duplicada) Ingresar nombre de usuario Ingresar departamento Ingresar cargo Seleccionar tipo de usuario	
Datos de Entrada:	Ingresar nombre: Miguel Ingresar apellido: Palma Ingresar cédula (duplicada): 10823354 Ingresar nombre de usuario: mpalma Ingresar departamento: Informatica Ingresar cargo: Analista de Sistemas Seleccionar tipo de usuario: Usuario	
Salida Esperada:	Usuario no registrado. Cédula ya registrada en el sistema	

Resultados:	Usuario no registrado. Correo electrónico no puede estar duplicado
Observaciones: Efectivamente no deja registrar el usuario, pero el mensaje no es correcto.	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 6	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Ingreso al sistema	
Pre-requisitos:	Tener nombre de usuario y clave registrados en la base de datos	
URL:	http:// localhost:8888/CAFD/HTML/iniciarSesion.jsp	
Pasos o secuencia lógica:	Ingresar un nombre de usuario correcto Ingresar una clave correcta Hacer clic en el botón “Ingresar”	
Datos de Entrada:	Nombre de usuario: dmoreno	

	Clave: M2xsm1R9	
Salida Esperada:	Ingreso exitoso al sistema	
Resultados:	Ingreso exitoso al sistema	
Observaciones: Sin observaciones		
Daniel Moreno **(Ejecutor de la prueba)**	**(Responsable del sistema)**	**(Responsable del sistema)**
Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 7	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Ingreso al sistema	
Pre-requisitos:	Tener nombre de usuario y clave registrados en la base de datos	
URL:	http:// localhost:8888/CAFD/HTML/iniciarSesion.jsp	
Pasos o secuencia lógica:	Ingresar un nombre de usuario incorrecto Ingresar una clave incorrecta Hacer clic en el botón “Ingresar”	
Datos de Entrada:	Nombre de usuario: dmoreno	

	Clave: dmoreno	
Salida Esperada:	Negar acceso al sistema	
Resultados:	Se negó el acceso al sistema	
Observaciones: Sin observaciones		
Daniel Moreno **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**		
Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 8	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Ingreso al sistema	
Pre-requisitos:	Tener nombre de usuario y clave registrados en la base de datos	
URL:	http:// localhost:8888/CAFD/HTML/iniciarSesion.jsp	
Pasos o secuencia lógica:	Ingresar un nombre de usuario incorrecto Ingresar una clave correcta Hacer clic en el botón "Ingresar"	
Datos de Entrada:	Nombre de usuario: syepez Clave: M2xsm1R9	

Salida Esperada:	Negar acceso al sistema	
Resultados:	Se negó el acceso al sistema	
Observaciones: Sin observaciones		
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**		
Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 9	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Carga del certificado digital	
Pre-requisitos:	Ser usuario registrado en el sistema. Contar con un certificado digital de extensión .PFX	
URL:	http://localhost:8888/CAFD/HTML/opciones.jsp	
Pasos o secuencia lógica:	Ingresar nombre del dueño del certificado digital Cargar en el formulario el certificado digital con extensión .pfx	
Datos de Entrada:	Nombre del dueño del certificado digital: Adriana Liendo Nombre y extensión del certificado digital: firma.pfx	
Salida Esperada:	Certificado cargado	

Resultados:	Certificado cargado
Observaciones: Si el usuario carga en varias oportunidades el certificado, el mismo no se sustituye sino que mantiene todos los certificados, lo cual genera problemas a la hora de hacer uso del certificado. Esta observación ya ha sido atendida.	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 10	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Carga del certificado digital	
Pre-requisitos:	Ser usuario registrado en el sistema. Contar con un certificado digital de extensión .PFX	
URL:	http://localhost:8888/CAFD/HTML/opciones.jsp	
Pasos o secuencia lógica:	Ingresar nombre del dueño del certificado digital Cargar en el formulario el certificado digital sin extensión .pfx	

Datos de Entrada:	Nombre del dueño del certificado digital: Adriana Liendo Nombre y extensión del certificado digital: firma.p12
Salida Esperada:	El certificado debe tener extensión .pfx
Resultados:	El certificado debe tener extensión .pfx
Observaciones: A pesar de que el formato pkcs#12 usa ambas extensiones, el sistema sólo acepta la extensión pfx. En tal sentido, es necesario hacer previamente las conversiones necesarias.	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 11	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 04/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Creación de categorías de documentos	
Pre-requisitos:	Tener el rol de Administrador o Administrador de documentos para crear un nuevo flujo de un documento	
URL:	http://localhost:8888/CAFD/HTML/flujocc.jsp	
Pasos o secuencia lógica:	Ingresar nombre de la categoría del documento (no duplicada)	

Datos de Entrada:	Categoría: Notificación
Salida Esperada:	Categoría registrada correctamente, puede crear el flujo correspondiente
Resultados:	Categoría registrada correctamente, puede crear el flujo correspondiente
Observaciones: Una vez creada la categoría, la misma no se puede modificar	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 12	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Creación de categorías de documentos	
Pre-requisitos:	Tener el rol de Administrador o Administrador de documentos para crear un nuevo flujo de un documento	
URL:	http://localhost:8888/CAFD/HTML/flujocc.jsp	
Pasos o secuencia lógica:	Ingresar nombre de la categoría del documento (duplicada)	
Datos de Entrada:	Categoría: Notificacion	
Salida Esperada:	Error. Ya existe la categoria	
Resultados:	Error. Ya existe la categoria	

Observaciones:		
(Ejecutor de la prueba)	**(Responsable del sistema)**	**(Responsable del sistema)**
Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 13	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Creación de un flujo de documento asociado a una categoría (Caso 1: Flujo aceptado)	
Pre-requisitos:	Tener el rol de Administrador o Administrador de documentos para crear un nuevo flujo de un documento	
URL:	http://localhost:8888/CAFD/HTML/flujocf.jsp	
Pasos o secuencia lógica:	Seleccionar nombre de la categoría del documento Ingresar nombre del flujo Seleccionar departamento Seleccionar cargo Seleccionar nombre de los integrantes del flujo	
Datos de Entrada:	Catagoría: Notificación Nombre del Flujo: Notificacióndos	

	Primer nivel: Departamento: Informática Cargo: Administrador de Sistemas Nombre y Apellido: Adriana Liendo Segundo nivel: Departamento: Informática Cargo: Programador Nombre y Apellido: Mirna Freitez
Salida Esperada:	Flujo creado correctamente
Resultados:	Flujo creado correctamente
Observaciones: Si se indica departamento, cargo, nombre y apellido, se carga automáticamente el correo del usuario. Sin embargo, si después de haber hecho eso se cambia el cargo, los datos del usuario se mantiene en el formulario y el correo se borra, si se hace click en agregar, se añade un paso o nivel en el flujo con el correo vacio (sin el comentario vacio).	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

Concluido	**x** **En proceso**	**Rechazado**

ID_CP: 14	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	

Plataforma:	Aplicación JAVA
Descripción:	Creación de un flujo de documento asociado a una categoría (Caso 2: Flujo detenido)
Pre-requisitos:	Tener el rol de Administrador o Administrador de documentos para crear un nuevo flujo de un documento
URL:	http://localhost:8888/CAFD/HTML/flujocf.jsp
Pasos o secuencia lógica:	Seleccionar nombre de la categoría del documento Ingresar nombre del flujo Seleccionar departamento Seleccionar cargo Seleccionar nombre de los integrantes del flujo
Datos de Entrada:	Catagoría: Notificación Nombre del flujo: NotificacionTres Primer nivel: Departamento: Informática Cargo: Administrador de Sistemas Nombre y Apellido: Adriana Liendo Segundo nivel: Departamento: Presupuesto Cargo: Analista Nombre y Apellido: Vacio
Salida Esperada:	Flujo creado correctamente
Resultados:	Flujo creado correctamente

Observaciones: Sin observación		
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**		
Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 15	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Creación de un flujo de documento asociado a una categoría (Caso 3: Flujo contingencia)	
Pre-requisitos:	Tener el rol de Administrador o Administrador de documentos para crear un nuevo flujo de un documento	
URL:	http://localhost:8888/CAFD/HTML/flujocf.jsp	
Pasos o secuencia lógica:	Seleccionar nombre de la categoría del documento Ingresar nombre del flujo Seleccionar departamento Seleccionar cargo Seleccionar nombre de los integrantes del flujo	
Datos de Entrada:		
Salida Esperada:		

Resultados:		
Observaciones: No recuerdo la diferencia entre entre detenido y contingencia, no sabría cómo hacer para probarlo		
Adriana Liendo **(Ejecutor de la prueba)**	**(Responsable del sistema)**	**(Responsable del sistema)**
Estado:		
Concluido	x **En proceso**	x **Rechazado**

ID_CP: 16	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Envío de un documento a un flujo (Caso 1: Flujo aceptado)	
Pre-requisitos:	Ser un usuario registrado en el sistema Tener cargado un certificado digital en el sistema	
URL:	http:// localhost:8888/CAFD/HTML/enviar.jsp	
Pasos o secuencia lógica:	Seleccionar categoría Seleccionar flujo Ingresar asunto del correo y Ingresar cuerpo del mensaje	

	Cargar archivo(s) adjunto (s) en formato PDF en el formulario
Datos de Entrada:	Categoría: Notificación Flujo: NotificaciónInformática Asunto: Prueba 1 Cuerpo del mensaje: Mensaje de prueba Archivo Adjunto: Prueba_viático
Salida Esperada:	Correo enviado
Resultados:	Correo enviado, junto con el archivo adjunto
Observaciones:	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 17	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Envío de un documento a un flujo (Caso 2: Flujo detenido)	

Pre-requisitos:	Ser un usuario registrado en el sistema Tener cargado un certificado digital en el sistema
URL:	http:// localhost:8888/CAFD/HTML/enviar.jsp
Pasos o secuencia lógica:	Seleccionar categoría Seleccionar flujo Ingresar asunto del correo y Ingresar cuerpo del mensaje Cargar archivo(s) adjunto (s) en formato PDF en el formulario
Datos de Entrada:	Categoría: Notificación Flujo: Notificacióntres Asunto: Prueba 2 Cuerpo del mensaje: Mensaje de prueba2 Archivo Adjunto: Prueba_viático
Salida Esperada:	Correo enviado
Resultados:	No es posible enviar el documento con el flujo indicado pues no está entre las opciones disponibles para selección
Observaciones:	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 18	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Envío de un documento a un flujo (Caso 3: Flujo en contingencia)	
Pre-requisitos:	Ser un usuario registrado en el sistema Tener cargado un certificado digital en el sistema	
URL:	http:// localhost:8888/CAFD/HTML/enviar.jsp	
Pasos o secuencia lógica:	Seleccionar categoría Seleccionar flujo Ingresar asunto del correo y Ingresar cuerpo del mensaje Cargar archivo(s) adjunto (s) en formato PDF en el formulario	
Datos de Entrada:		
Salida Esperada:	Correo enviado	
Resultados:		
Observaciones: Me ocurre lo mismo que en el caso de la prueba 15		
Adriana Liendo		

(Ejecutor de la prueba)	(Responsable del sistema)	(Responsable del sistema)
Estado:		
Concluido	x **En proceso**	x **Rechazado**

ID_CP: 19	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Envío de un documento a un flujo (Caso 3: Flujo en contingencia)	
Pre-requisitos:	Ser un usuario registrado en el sistema Tener cargado un certificado digital en el sistema	
URL:	http:// localhost:8888/CAFD/HTML/enviar.jsp	
Pasos o secuencia lógica:	Seleccionar categoría Seleccionar flujo Ingresar asunto del correo y Ingresar cuerpo del mensaje Cargar archivo(s) adjunto (s) en formato PDF en el formulario	
Datos de Entrada:	Categoría: _____ Flujo: _____ Asunto del correo: _____	

	Cuerpo del mensaje: _____ Sin archivos adjuntos
Salida Esperada:	Ingrese un archivo adjunto al correo
Resultados:	
Observaciones:	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

Concluido	**En proceso**	**Rechazado**

ID_CP: 20	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Firmar (aceptar) un documento de un flujo	
Pre-requisitos:	Ser usuario registrado en el sistema Certificado digital cargado en el sistema	
URL:	http://localhost:8888/CAFD/HTML/Documentos_Recibidos.jsp	

Pasos o secuencia lógica:	Clic en la metáfora de aceptar un documento (agregar firma digital al mismo) Ingresar clave privada del certificado digital	
Datos de Entrada:	Clave privada del certificado digital: ****** (la clave es privada)	
Salida Esperada:	Documento firmado	
Resultados:	Documento firmado	
Observaciones: La clave del certificado no se debe mostrar mientras se está ingresando		
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**		
Estado:		
x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 21	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Firmar (aceptar) un documento de un flujo	
Pre-requisitos:	Ser usuario registrado en el sistema Certificado digital cargado en el sistema	
URL:	http://localhost:8888/CAFD/HTML/Documentos_Recibidos.jsp	

Pasos o secuencia lógica:	Clic en la metáfora de aceptar un documento (agregar firma digital al mismo) Ingresar clave privada del certificado digital (clave incorrecta)
Datos de Entrada:	Clave privada del certificado digital: 123456
Salida Esperada:	La clave ingresada no coincide con los datos de su certificado digital
Resultados:	La clave ingresada no coincide con los datos de su certificado digital
Observaciones: La clave del certificado no se debe mostrar mientras se está ingresando	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 22	**Módulo a probar:** Módulo de gestión de documentos	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Rechazar un documento de un flujo	
Pre-requisitos:	Ser usuario registrado en el sistema Certificado digital cargado en el sistema	

URL:	http://localhost:8888/CAFD/HTML/Documentos_Recibidos.jsp
Pasos o secuencia lógica:	Clic en la metáfora de rechazar un documento
Datos de Entrada:	--- El sistema solicita la clave del certificado
Salida Esperada:	Documento rechazado
Resultados:	Documento rechazado
Observaciones: El sistema solicita la clave del certificado, a pesar de no usarla, pues el documento rechazado no agrega la firma.	
Adriana Liendo **(Ejecutor de la prueba)** **(Responsable del sistema)** **(Responsable del sistema)**	
Estado:	

x **Concluido**	**En proceso**	**Rechazado**

ID_CP: 23	**Módulo a probar:** Módulo de gestión de usuarios	**Fecha:** 07/11/2013
Nombre del Sistema:	CAFD	
Plataforma:	Aplicación JAVA	
Descripción:	Eliminar un usuario involucrado en un flujo (el cual aún no ha firmado o rechazado el documento en el flujo al que pertenece)	
Pre-requisitos:	Tener el rol de Administrador o Administrador de usuarios para eliminar un usuario del sistema	

URL:	http://localhost:8888/CAFD/HTML/ ver_usuario.jsp
Pasos o secuencia lógica:	Clic en la metáfora para eliminar un usuario
Datos de Entrada:	--- Se eliminó el usuario mfreitez, el cual tenía un documento en espera por firmar.
Salida Esperada:	Usuario eliminado
Resultados:	Usuario eliminado

Observaciones:

El documento donde estaba involucrado el usuario que se borró, ya no se encuentra en la bandeja de enviados. Al revisar en la bandeja de recibidos, sí se puede ver (pues mi usuario era la primera persona del flujo afectado, y al ver detalles ya no muestra la gráfica del estado, sino la leyenda nada más. El flujo ahora está en estado detenido.

Adriana Liendo

(Ejecutor de la prueba) **(Responsable del sistema)** **(Responsable del sistema)**

Estado:

Concluido	**x** **En proceso**	**Rechazado**