



## Universidad de las Ciencias Informáticas

Módulo de administración para el Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones de la República Bolivariana de Venezuela. Segunda iteración.

### Trabajo de Diploma para optar por el título de Ingeniero Informático.

Autor: Luis Orlando Jiménez Cepero

Tutor(es): Ing. Mairelys Boeras Velázquez

Ing. Rodolfo Venero Noriega

Habana, Julio del 2011

*“Si quieres ser sabio, aprende a interrogar  
razonablemente, a escuchar con atención, a responder  
serenamente y a callar cuando no tengas nada que decir”.*

*Johann Kaspar Lavater*

# Declaración de autoría

Declaro ser autor del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Para que así conste firmo el presente a los \_\_\_\_ días de julio del 2011.

Autor:

---

Luis Orlando Jiménez Cepero

Tutores:

---

Ing. Mairelys Boeras Velázquez

---

Ing. Rodolfo Venero Noriega

# Datos de contactos

Ing. Mairelys Boeras Velázquez: Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Instructora con tres años de experiencia.

Correo electrónico: [mbohera@uci.cu](mailto:mbohera@uci.cu)

Ing. Rodolfo Venero Noriega: Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Recién graduado.

Correo electrónico: [rvenero@uci.cu](mailto:rvenero@uci.cu)

# Dedicatoria

*Este trabajo está dedicado a una persona muy especial en mi vida que es mi madre.*

# gradecimientos

*Agradezco ante todo a las personas que siempre me apoyaron en los momentos más difíciles, mi mamá, mi papá, mi padrastro.*

*A todos los que confiaron en mí y me dieron ánimo para seguir, mi abuela Yolanda, mi tía Leonor, mi primo Tonito, mi tata Angelina, todas mis primas Martica, Normita, Tania, Mabel, mi primo Orli. En fin personas especiales que sienten gran aprecio por mí y estuvieron preocupadas en todo el transcurso de este trabajo.*

*A todos los que aportaron un granito para el desarrollo de este trabajo, en especial a Ángel por su dedicación y paciencia, las tantas noches sin dormir, sin comer para lograr que todo saliera adelante. A Leski un nuevo amigo que me dio fuerza y confianza en los momentos tensos para que no me diera por vencido. A la profesora Alina, que a pesar que estuvo en los últimos momentos su ayuda fue incondicional, sin ella no creo que este sueño se hubiese logrado.*

*A todo el grupo de trabajo del proyecto Pasaporte Diplomático por aclararme tantas dudas y dedicarme parte de su tiempo, en especial a Annie, Idia, Erick, Sandy y Sotes que cuando me aproxime a ellos con grandes dudas siempre fueron aclaradas.*

*A todos mis amigos, los que estuvieron cerca y los que a pesar de la distancia estuvieron preocupados y me ayudaron en todo lo que estuvo a su alcance.*

Los sistemas de emisión de documentos de identificación son de gran importancia para todos los países del mundo. Actualmente el proceso de emisión de documentos de identificación en el Ministerio del Poder Popular de Relaciones Exteriores de la hermana República Bolivariana de Venezuela se realiza en sistemas informáticos obsoletos y muy lentos. Para dar solución a esta problemática, a través del convenio Cuba-Venezuela comienza en la Universidad de las Ciencias Informáticas el desarrollo del proyecto Transformación y Modernización del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones. Este sistema se está desarrollando con el propósito de automatizar los procesos referentes a la solicitud, enrolamiento, personalización, ensobrado y entrega de pasaportes diplomáticos, de servicio y acreditaciones.

La presente investigación, es resultado de la necesidad del desarrollo de las funcionalidades pertenecientes a la Segunda iteración de desarrollo del módulo de administración, las cuales son: gestionar usuarios, gestionar roles, gestionar plan de citas, definir convenio de supresión de visa y definir otorgamiento de acreditación. Durante el desarrollo de este trabajo se especificarán los principales elementos para lograr el desarrollo de esta solución.

Palabras claves: administración, documentos de identificación.

## Índice general

<i>Introducción</i> .....	1
<i>Capítulo 1 Fundamentación teórica</i> .....	6
1.1 Introducción .....	6
1.2 Documentos de Identificación .....	6
1.3 Modelos de control de acceso .....	7
1.3.1 Modelo de control de acceso discrecional .....	7
1.3.2 Modelo de control de acceso obligatorio .....	8
1.3.3 Modelo de control de acceso basado en tareas .....	8
1.3.4 Modelo de control de acceso basado en roles .....	9
1.4 Sistemas semejantes .....	10
1.4.1 Sistema de emisión de pasaporte Datacard PB6500 .....	10
1.4.2 Sistema de Emisión de Pasaportes de Yemen .....	11
1.4.3 EMIPAS .....	11
1.5 Metodologías de desarrollo .....	11
1.5.1 Metodologías ágiles .....	12
1.5.1.1 FDD (Feature Driven Development) .....	12
1.6 Lenguajes utilizados .....	14
1.6.1 Lenguaje de programación .....	14
1.6.1.1 JAVA como lenguaje de programación .....	15
1.6.2 Lenguaje de modelado .....	15
1.6.2.1 UML como lenguaje de modelado .....	16
1.7 Herramientas de desarrollo .....	16
1.7.1 NetBeans 6.9 .....	17
1.7.2 Eclipse Galileo .....	17
1.7.3 Apache Tomcat .....	18
1.8 Herramienta de modelado .....	18



1.8.1	Visual Paradigm 6.4 .....	18
1.9	Sistema Gestor de Base de Datos.....	19
1.9.1	PostgreSQL .....	19
1.10	Marcos de trabajo .....	20
1.10.1	Spring 2.5.5.....	20
1.10.2	Hibernate 3 .....	21
1.11	Conclusiones parciales .....	22
<i>Capítulo 2 Propuesta de Solución. ....</i>		<i>23</i>
2.1	Introducción .....	23
2.2	Modelo conceptual del dominio .....	23
2.2.1	Descripción del modelo conceptual de dominio.....	24
2.3	Construcción de la lista de funcionalidades .....	25
2.3.1	Lista de funcionalidades .....	25
2.4	Planeación de iteraciones por funcionalidades.....	26
2.5	Requisitos no funcionales.....	29
2.6	Definición de la arquitectura.....	30
2.6.1	Arquitectura cliente/servidor.....	31
2.7	Patrones de diseño.....	34
2.8	Modelo de clases del diseño.....	36
2.8.1	Diagrama de clases del diseño.....	36
2.8.2	Descripción de las clases del diseño.....	37
2.9	Diseño de la base de datos.....	40
2.9.1	Descripción de las tablas de la base de datos.....	42
3.1	Estándar de codificación.....	43
2.10	Diagrama de paquete del diseño .....	45
2.11	Conclusiones parciales .....	47
<i>Capítulo 3 Implementación y Prueba .....</i>		<i>48</i>

3.2	Introducción .....	48
3.3	Diagrama de componentes .....	48
3.4	Modelo de despliegue .....	49
3.5	Pruebas .....	50
3.5.1	Pruebas unitarias .....	51
3.5.2	Pruebas de Caja Negra .....	52
3.6	Resultado de las pruebas:.....	58
3.7	Conclusiones parciales .....	60
	<i>Conclusiones</i> .....	62
	<i>Recomendaciones</i> .....	63
	<i>Bibliografía</i> .....	64
	<i>Glosario de términos</i> .....	67
	<i>Anexos</i> .....	70
	Anexo 1 Descripción de las Funcionalidades.....	70
	Anexo 2 Diagramas de clases .....	110
	Anexo 3 Descripción de clases .....	113
	Anexo 4 Descripción de las tablas de la base de datos .....	116
	Anexo 5 Diagramas de componentes .....	129
	Anexo 6 Plantillas de pruebas de caja negra .....	133

## Índice de tablas

Tabla 1 Niveles del modelo RBAC. (33)	10
Tabla 2 Lista de funcionalidades	26
Tabla 3 Planeación de la funcionalidad: Gestionar Usuario	27
Tabla 4 Planeación de la funcionalidad: Gestionar roles	28
Tabla 5 Planeación de la funcionalidad: Gestionar planificación de citas	28
Tabla 6 Planeación de la funcionalidad: Gestionar otorgamiento de acreditación	28
Tabla 7 Planeación de la funcionalidad: Gestionar convenio de supervisión	29
Tabla 8 Descripción de la clase ControlUsuarioFTaskImpl	38
Tabla 9 Descripción de la clase ControlUsuarioControlador	39
Tabla 10 Descripción de la clase AccesoFacadeImpl	40
Tabla 11 Descripción de la tabla tbl_acceso_usuario	42
Tabla 12 Descripción de la tabla tbl_acceso_perfil	43
Tabla 13 Criterios de calidad	44
Tabla 14 Reglas específicas para la nombres	45
Tabla 15 Descripción de variables. Caso de prueba: Gestionar Usuario	53
Tabla 16 Caso de prueba: Gestionar Usuario	58

## Índice de Figuras

Figura 1 Representación del nivel RBAC básico	10
Figura 2: Fases de desarrollo de la metodología FDD	13
Figura 3 Modelo conceptual de dominio: "módulo administración"	24
Figura 4: Arquitectura cliente - servidor	32
Figura 5 Vista Lógica de la arquitectura	33
Figura 6 Diagrama de clases del diseño de la funcionalidad Gestionar usuario	37
Figura 7 Diseño de la base de datos	41
Figura 8 Diagrama de Paquete de la capa de Proceso	46
Figura 9 Diagrama de Paquete de la capa de Componentes	46
Figura 10 Diagrama de componentes de la funcionalidad Gestionar usuario	49
Figura 11 Diagrama de Despliegue	50
Figura 12 Ejecución de la clase AdministrationTest	51
Figura 13 Resultado de las Pruebas Unitarias	59
Figura 14 Resultado de las pruebas de Caja Negra	60

## Introducción

Un documento de identificación es un documento legal diseñado por el Estado para identificar a su titular, este puede contener datos requeridos para el uso en operaciones posteriores que demanden la certificación de identidad de su portador. Entre los documentos de identificación actuales se encuentran los pasaportes, cédula de identidad, licencia de conducción, entre otros.

Los pasaportes diplomáticos y de servicio son los documentos concedidos a los funcionarios nombrados por el país (Venezuela) para cumplir misiones en representación de su estado en el mundo, un ejemplo claro cuando se nombra a un ciudadano venezolano embajador en otro país en representación de Venezuela. En el caso de la acreditación es el documento otorgado al personal diplomático extranjero que cumple misión en Venezuela, un ejemplo sería el embajador de Cuba en Venezuela.

Debido a la vital importancia de estos documentos, las instituciones gubernamentales son las encargadas de emitirlos de forma oficial teniendo en cuenta la evaluación y aprobación de los datos solicitados al interesado.

El Ministerio del Poder Popular de Relaciones Exteriores (MPPRE) de la República Bolivariana de Venezuela lleva a cabo los procesos de solicitud, enrolamiento, personalización, ensobrado y entrega de documentos de identificación. Parte de estos procesos actualmente automatizados por sistemas informáticos lentos y obsoletos de forma general presentan las siguientes dificultades:

- ✓ La información de control de los trámites que se maneja no es actualizada ni confiable.
- ✓ Existen grandes dificultades para la realización de consultas a la información de los trámites, ya sea en los archivos físicos como en los sistemas informáticos.
- ✓ Un gran número de las actividades de los procesos de solicitud, elaboración, personalización y entrega de los pasaportes y las acreditaciones, se realizan de forma manual, lo que origina poca seguridad en los documentos y demoras en los servicios.

La Universidad de las Ciencias Informáticas (UCI) desempeña un rol importante en la industria del software cubano por constituir un centro de altos estudios donde sus estudiantes y profesores son capaces de producir software para la solución eficiente de problemas de informatización aplicables tanto dentro como fuera del país. La Universidad se encuentra dividida en distintas facultades y estas a su vez

# Introducción

en distintos centros de investigación y desarrollo, como por ejemplo el Centro de Identificación y Seguridad Digital (CISED) que se encarga del desarrollo de soluciones informáticas orientadas a las técnicas de identificación y seguridad digital.

Considerando las estrechas relaciones de intercambio entre la República de Cuba y la República Bolivariana de Venezuela, las cuales se fundamentan en la histórica hermandad entre sus pueblos y Gobiernos, el CISED y el Ministerio del Poder Popular para las Relaciones Exteriores de la República Bolivariana de Venezuela, en conjunto, están desarrollando el Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones. Este sistema posibilita una serie de funciones, las cuales son realizadas por el personal involucrado en el proceso de emisión que pueden ser: secretaria, solicitante, administrador etc. Actualmente el sistema no cuenta con una gestión de los usuarios que interactúan con el mismo como tampoco cuenta con un mecanismo que les conceda permisos a los usuarios para el acceso a las funciones específicas que cada uno de ellos realiza.

El proceso de emisión de estos documentos cuenta con cuatro pasos fundamentales, solicitud, enrolamiento, personalización, ensobrado y entrega. El solicitante del documento, por ser en este caso, un diplomático de la República Bolivariana de Venezuela o que se encuentre cumpliendo estas responsabilidades representando a otra país en ella, necesita por los altos grados de complicación conocer el horario en que debe presentarse en la oficina para realizar su planificación. No existe en el sistema un mecanismo que posibilite definir los horarios para realizar las citas previas al inicio del trámite. En el caso de que sea un funcionario solicitante de una acreditación, debe existir una vía que permita verificar que el país al que represente el funcionario posea convenio de supresión de visa con Venezuela para que se excluya la captura de estos datos en el sistema y posteriormente sea validada por la secretaria, la cual no está disponible en estos momentos. El proceso que se inicia a partir de estas solicitudes, luego continúa con la captación de datos, personalización y culminando con la entrega del documento. Pero se ha determinado que no existe forma de especificar para tener en cuenta durante la personalización si se le otorga carnet y/o sello en el pasaporte según el tipo de funcionario que sea.

Después de argumentada la situación problemática el problema científico queda formulado de la siguiente manera: ¿Cómo facilitar la gestión de usuarios, roles y elementos de la configuración del proceso de negocio del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones de la República Bolivariana de Venezuela?

# Introducción

Por lo que el objeto de estudio de la investigación lo constituyen: los procesos de gestión de usuarios, roles y elementos de configuración del proceso de negocio pertenecientes a los módulos de administración de sistemas para la emisión de documentos de identificación. Se plantea campo de acción: los procesos de gestión de usuarios, roles y elementos de configuración del proceso de negocio pertenecientes al módulo de administración del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones de la República Bolivariana de Venezuela.

Por todo lo antes planteado la idea a defender del presente trabajo es: el desarrollo de funcionalidades gestión de usuarios, gestión de roles y gestión de elementos de configuración del proceso de negocio del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones, facilitará la gestión de usuarios, roles y elementos de configuración del proceso de negocio del sistema.

Objetivo General: Desarrollar las funcionalidades que faciliten la gestión de usuarios, roles y elementos de configuración del proceso de negocio del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones de la República Bolivariana de Venezuela.

A partir de un análisis para dar cumplimiento al objetivo general se derivan los siguientes objetivos específicos:

- ✓ Realizar un estudio del arte sobre módulos de administración de sistemas de emisión de documentos de identificación que gestionen usuarios, roles y elementos de configuración.
- ✓ Efectuar un estudio de las herramientas y tecnologías necesarias para desarrollar la solución.
- ✓ Diseñar y Desarrollar la propuesta de solución.
- ✓ Validar las funcionalidades implementadas en el sistema.

Para el cumplimiento organizado y bien distribuido de los objetivos se establecen las siguientes tareas de investigación:

- ✓ Estudio y análisis de los módulos de administración de sistemas de emisión de documentos de identificación centrándose en la gestión de usuarios, roles, y elementos de configuración.
- ✓ Estudio de la arquitectura definida para el Sistema de emisión de pasaportes diplomáticos, de servicios y acreditaciones.
- ✓ Definición del modelo de datos.

- ✓ Diseño de las interfaces de usuario para el módulo de administración basadas en los requerimientos funcionales.
- ✓ Desarrollo del proxy para la comunicación de las interfaces con el servicio web expuesto por la capa de proceso.
- ✓ Diseño y aplicación de las pruebas a la solución.

Para llevar a cabo las tareas científicas se pondrán en práctica varios métodos que facilitarán las tareas de la investigación.

Los métodos científicos a utilizar son:

## Métodos teóricos

- ✓ Método Analítico – Sintético: Se realizará un análisis de la bibliografía disponible para hacer un estudio completo del estado del problema a resolver. También para definir los conceptos principales. Sintetizar las características de las herramientas para el desarrollo del sistema y las ventajas del uso de las mismas.
- ✓ Método Histórico – Lógico: Se utilizará para analizar otras soluciones existentes que tengan similitud con los sistemas de emisión de documentos de identificación centrándose en cómo gestionan usuarios, roles y elementos de configuración.

## Métodos empíricos

- ✓ Observación: Este método permitirá investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo que ayuda al planteamiento del problema científico, además de permitir conocer bien el proceso delimitado como objeto de estudio, que contribuye a tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacerlo.
- ✓ Entrevista: Estas se realizarán a los especialistas que están vinculados al desarrollo del proyecto ya que son los que están en contacto directo con el software al que se va a integrar esta solución.

El contenido de la investigación se encuentra distribuido en el documento de la siguiente manera:

- ✓ Capítulo 1: Fundamentación teórica: En este capítulo se realiza un análisis sobre los sistemas existentes en la actualidad para la Emisión de Documentos de Identificación tanto a nivel

# Introducción

nacional como internacional, también se realiza un estudio de las herramientas, tecnologías y metodología definidas para el desarrollo de la solución.

- ✓ Capítulo 2: Propuesta de Solución: Se describe el negocio a través de un modelo de dominio y, a partir de este, se comienza el análisis del sistema a desarrollar. Se enumeran las funcionalidades necesarias para la propuesta de solución, así como se organiza su desarrollo mediante un plan de iteraciones. Mediante los diagrama de clases y la estructura de la base de datos se tendrá una visión de la propuesta de solución que se implementará.
- ✓ Capítulo 3: Implementación y pruebas: Se muestra cómo está estructurada físicamente la solución implementada mediante los diagramas de despliegue y de componentes. Para validar la solución antes de su puesta en práctica se realizan las pruebas necesarias.



# Capítulo 1 Fundamentación teórica

## Capítulo 1 Fundamentación teórica

### 1.1 Introducción

En el presente capítulo se abordarán temas relacionados con los conceptos básicos asociados al dominio del problema y el objeto de estudio. Se hace necesario estudiar la gestión de usuarios, roles y elementos de configuración del proceso de negocio en sistemas de emisión de documentos de identificación existentes para comprender como se realizan estas acciones en un sistemas de información. Se presenta una fundamentación de la metodología, tecnologías y herramientas que se utilizaran para el desarrollo del software.

### 1.2 Documentos de Identificación

La emisión de un documento de identificación es un proceso complejo, que requiere un alto nivel de seguridad y control, basada en normas legislativas establecidas para el lugar donde el documento tiene efecto legal. A continuación se definirán los conceptos para un mejor entendimiento del presente trabajo.

#### Pasaporte

Documento oficial emitido por un Estado para otorgar permiso a la(s) persona(s) identificada(s) en el pasaporte para viajar, ser readmitido(s) y gozar de protección mientras se encuentra(n) en el extranjero.

(1)

#### Pasaporte Diplomático

Para la República Bolivariana de Venezuela, es el documento de identificación en el extranjero que el Ministerio del Poder Popular para Relaciones Exteriores otorga a los funcionarios y personas que cumplirán misiones, funciones o cargos de alta dignidad, de responsabilidad nacional en representación del Gobierno Bolivariano. (2)

#### Pasaporte de Servicio

Para la República Bolivariana de Venezuela, es el documento que se expide por el Ministerio del Poder Popular para Relaciones Exteriores, a funcionarios o particulares que deban salir del país en comisión de servicio al extranjero, al personal de las representaciones diplomáticas y oficinas consulares de la República Bolivariana de Venezuela en el extranjero.(2)

#### Acreditación

La acreditación es la certificación mediante un documento, de que una persona posee las facultades necesarias para desempeñar un cometido. (31)

# Capítulo 1 Fundamentación teórica

## Acreditaciones diplomáticas

Son los documentos concedidos al personal diplomático que respalda su condición ante un Estado y reconoce las funciones y cargos para los que fue designado por su Estado de origen.

Ministerio del Poder Popular para Relaciones Exteriores (MPPRE) de la República Bolivariana de Venezuela

Ministerio que tiene como misión planificar, formular y ejecutar, bajo la conducción directa del presidente, la política internacional del estado y la nación venezolana, a través de la coordinación y armonización de las acciones de los diversos órganos del estado, en lo que concierne a las relaciones internacionales, para propiciar una posición estratégica consecuente con la defensa del interés nacional. Dentro de sus principales objetivos se encuentra la protección de los derechos e intereses de los venezolanos en el exterior, conforme al derecho internacional. Consecuentemente con esto, el MPPRE se encarga de emitir los documentos de identificación al personal diplomático y de servicio que, en dependencia de la función que realizarán en el exterior, se le emite el pasaporte correspondiente. (22)

## Planificación de citas

Para el Sistema de Emisión de pasaportes Diplomáticos, de Servicios y Acreditaciones e el calendario de trabajo referente a la planificación de atención a las solicitudes de documentos de identificación.

## Convenio de supresión de visa

Acuerdo entre los gobiernos de dos países determinados concluyendo una serie de términos relacionados con el visado para ingresar, permanecer o salir del país receptor. Estos términos definen al personal autorizado el tiempo de estancia en el país receptor y las leyes por las que debe regirse.

## 1.3 Modelos de control de acceso

Desde los inicios del desarrollo de software ha sido un requisito la construcción de modelos de seguridad y acceso de diversas maneras sobre los cuales los usuarios que intervienen en los procesos son inicialmente identificados y luego autorizados a su interacción con los mismos. Entre los modelos más comunes se encuentran:

### 1.3.1 Modelo de control de acceso discrecional

En el modelo de control de acceso discrecional (Discretionary Access Control, DAC) es el usuario quien decide cómo proteger el sistema, mediante controles de acceso impuestos por el sistema, es decir, el propietario del objeto es autorizado a permitir u otorgar permisos para este objeto a otros usuarios. Los sujetos pueden ser usuarios, grupos o procesos.

# Capítulo 1 Fundamentación teórica

El modelo DAC brinda al usuario el beneficio de la flexibilidad del modelo, debido a esto es demasiado débil para controlar el acceso a los recursos de forma efectiva. El DAC para realizar operaciones sobre los recursos utiliza las Listas de Control de Acceso (ACLs), que son listas de usuarios y grupos con sus permisos específicos. DAC es apropiado en ambientes donde la compartición de información es más importante que su protección. (32)

## 1.3.2 Modelo de control de acceso obligatorio

En el modelo de control de acceso obligatorio (Mandatory Access Control, MAC) es el sistema quién protege los recursos la autorización para que un sujeto acceda a un objeto depende de los niveles de seguridad que tengan, ya que el administrador es el que indica qué permiso de seguridad tiene el sujeto y el nivel de sensibilidad del objeto.

El modelo de seguridad MAC describe niveles de seguridad a través de un conjunto de clasificaciones discretas que siguen el modelo de clasificación de la información militar donde la confidencialidad de la información es lo más relevante (Desclasificado, Confidencial, Secreto y Sumamente Secreto); esto se conoce como política de seguridad multinivel. Un importante objetivo del modelo MAC es controlar el flujo de información en orden de asegurar su confidencialidad y su integridad, objetivo no alcanzado por los modelos DAC. Sin embargo las políticas que establece este modelo no proporcionan soluciones factibles dado que es demasiado rígido. (32)

## 1.3.3 Modelo de control de acceso basado en tareas

El TBAC (Task Based Access Control) permite controlar el acceso en entornos representados por flujos de trabajo. Da soporte para el modelado de las tareas, la monitorización y gestión del proceso de autorización, garantizando el control de acceso mediante etapas de autorización. La activación y desactivación de los permisos de los estados de protección (son los que definen todos los permisos que pueden ser activados por la etapa de autorización) de forma dinámica en tiempo de ejecución, además permite monitorizar el uso de los permisos en tiempo de ejecución. En el modelo TBAC la especificación de políticas de seguridad es compleja, también la gestión, delegación y revocación de los privilegios son muy primitivas. (32)

# Capítulo 1 Fundamentación teórica

## 1.3.4 Modelo de control de acceso basado en roles

El principal objetivo del modelo de control de acceso basado en roles (Role Based Access Control, RBAC) es prevenir que los usuarios tengan libre acceso a la información de la organización. La NIST<sup>1</sup> define de manera clara y organizada las características de la tecnología RBAC, no sólo reconociendo el beneficio potencial de esta sino que ha propuesto un estándar. El modelo de referencia de RBAC propuesto consta de cuatro componentes: usuarios, roles, operaciones y recursos. En este modelo los usuarios son asignados a uno o varios roles mientras que los permisos y privilegios se asignan a estos roles, es decir, los permisos de acceso están asociados a los roles. El modelo RBAC de NIST está organizado en cuatro niveles sucesivos (Tabla 1 **Niveles del modelo RBAC.**) con el objetivo de aumentar las capacidades funcionales en cada uno de estos. (33)

Nivel	Nombre	Capacidades funcionales
1	RBAC Básico	<ul style="list-style-type: none"><li>• El usuario adquiere los permisos a través de los roles</li><li>• Existe una relación entre usuario-rol</li><li>• Existe una relación entre permiso-rol</li><li>• Apoya la asignación de usuario-rol</li><li>• El usuario puede usar permisos de múltiples roles simultáneamente</li></ul>
2	RBAC Jerárquico	RBAC básico + <ul style="list-style-type: none"><li>• Soporta una jerarquía de roles</li><li>• Nivele 2a soporta las jerarquías arbitrarias</li><li>• Nivele 2b soporta las jerarquías limitadas</li></ul>
3	RBAC Restricciones	RBAC Jerárquico+ <ul style="list-style-type: none"><li>• debe fortalecer la separación de deberes</li><li>• Nivele 3a soporta las jerarquías arbitrarias</li><li>• Nivele 3b soporta las jerarquías limitadas</li></ul>
4	RBAC Simétrico	RBAC Restricciones+

<sup>1</sup>NIST Acrónimo de *National Institute of Standard and Technology* (Instituto nacional de estándares y tecnologías)

# Capítulo 1 Fundamentación teórica

		<ul style="list-style-type: none"><li>• debe apoyar la revisión del permiso-rol con la actuación eficazmente comparable a la revisión del usuario-rol</li><li>• Nivele 4a soporta las jerarquías arbitrarias</li><li>• Nivele 4b soporta las jerarquías limitadas</li></ul>
--	--	---

Tabla 1 Niveles del modelo RBAC. (33)

Después de un análisis de los modelos de control de acceso MAC, DAC, TBAC y RBAC, se llega a la conclusión que los modelos DAC y MAC son inadecuados para cubrir las necesidades del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones. El modelo DAC es débil para controlar el acceso a los recursos de información de forma efectiva, mientras el modelo MAC es demasiado rígido. El caso del modelo TBAC presenta varias carencias cuando es aplicado, una de ella es que reconoce la necesidad de incluir la información de contexto para realizar el control de acceso, otros de los inconvenientes de este modelo es la especificación de políticas de seguridad ya que son complejas y la gestión, delegación y revocación de los privilegios son muy primitivas. Por lo tanto se decide utilizar el modelo de control de acceso RBAC solo hasta el nivel RBAC básico (Figura 1) ya que se ajusta a las características del sistema y a la manera en que se definió otorgar los permisos.

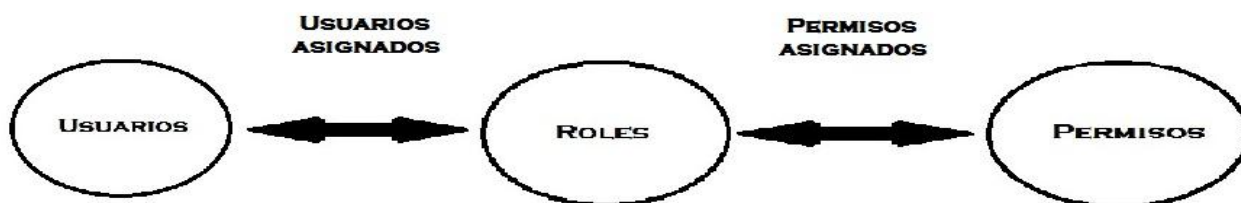


Figura 1 Representación del nivel RBAC básico.

## 1.4 Sistemas semejantes

### 1.4.1 Sistema de emisión de pasaporte Datacard PB6500

Sistema de expedición de pasaporte ofrece una amplia variedad de tecnologías de personalización en línea, incluyendo la capacidad de emitir pasaportes con características de seguridad sofisticadas. Es un sistema fiable para mejorar la seguridad de los documentos e integrar tecnologías avanzadas en los

# Capítulo 1 Fundamentación teórica

pasaportes de lectura mecánica y pasaportes electrónicos. Consta con el módulo de administración del sistema el cual supervisa el flujo de datos, control de acceso y el funcionamiento de los restantes módulos, todo esto consigue proteger el proceso con amplias medidas de seguridad a nivel de sistema, reducir el riesgo de fraude y robo de identidad.

## 1.4.2 Sistema de Emisión de Pasaportes de Yemen

Este es un sistema completamente automatizado para la emisión y gestión de pasaportes de lectura mecánica y pasaportes electrónicos que confirman a los últimos estándares establecidos por la Organización de Aviación Civil Internacional OACI. Dentro de sus módulos está incluido el módulo de administración el cual es el encargado del control de acceso a los servicios y a la clave de las medidas de control automatizado y electrónico, para cumplir su objetivo se basa en el control de acceso basado en roles. El acceso al sistema requiere de dos identificaciones por separado, el código de usuario y una contraseña que sólo pueden ser modificados por los administradores. También el subsistema mantiene un registro de todas las transacciones realizadas por cada usuario.

## 1.4.3 EMIPAS

El sistema está concebido para la personalización y emisión de pasaportes. En su funcionamiento cumple con las normas internacionales de la Organización de la Aeronáutica Civil (OACI), para pasaportes de lectura mecánica. EMIPAS está compuesto básicamente, por cuatro módulos: Personalización, Arraigos, Reportes y Administración. Este último consiste en una aplicación, desde la cual se puede administrar y configurar el sistema. Se realizan las auditorías de seguridad. Se controlan los inventarios, la gestión de bases de datos y se registran las actualizaciones.

Después de haber estudiado los módulos de administración de estos sistemas, se tomó la decisión que ninguno de estos puede ser utilizado como propuesta de solución ya que no se ajustan a los requisitos específicos del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones sino a las del sistema al que pertenece cada uno de ellos.

## 1.5 Metodologías de desarrollo

El desarrollo de software no es sin dudas una tarea fácil. Como resultado a este problema ha surgido una alternativa desde hace mucho: la Metodología. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado. (5)

# Capítulo 1 Fundamentación teórica

Una metodología puede definirse, en un sentido amplio, como un conjunto de métodos o técnicas que ayudan en el desarrollo de un producto software.

Las metodologías persiguen tres necesidades principales:

- ✓ Mejores aplicaciones, tendientes a una mejor calidad.
- ✓ Un proceso de desarrollo controlado.
- ✓ Un proceso estándar en la organización.

## 1.5.1 Metodologías ágiles

Las metodologías ágiles están contenidas en el concepto de desarrollo ágil, el cual se basa en la entrega temprana del software con el uso métodos no formales. No son ajenas al resto de las metodologías, mantienen la comunicación con el cliente, el diseño, la elaboración y entrega, pero enfocan su mayor esfuerzo en la elaboración y entrega del producto. Se clasifican de ágiles por su capacidad de responder rápida y efectivamente ante los cambios. Se apoyan en las habilidades y experiencias personales y del equipo, evitando los extenuantes caminos de las metodologías tradicionales. Las metodologías ágiles presentan diversas ventajas, entre las que se destacan:

- ✓ Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- ✓ Entrega continua y en plazos breves de software funcional.
- ✓ Trabajo conjunto entre el cliente y el equipo de desarrollo.
- ✓ Importancia de la simplicidad, eliminado el trabajo innecesario.
- ✓ Atención continua a la excelencia técnica y al buen diseño.
- ✓ Mejora continua de los procesos y el equipo de desarrollo.

### 1.5.1.1 FDD (Feature Driven Development)

FDD<sup>2</sup> se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero es más similar a este último. FDD está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Las iteraciones se deciden en base a funcionalidades (de ahí el nombre del proceso), que son pequeñas partes del software con significado para el cliente.

---

<sup>2</sup> Desarrollo basado en funcionalidades.

# Capítulo 1 Fundamentación teórica

Un proyecto que sigue FDD se divide en 5 fases como se muestra en la Figura 22:

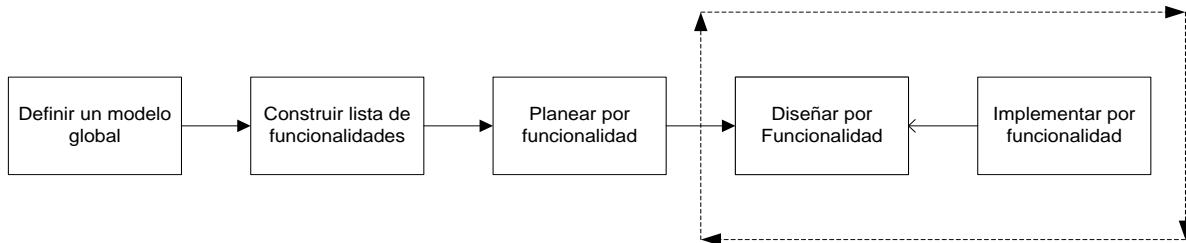


Figura 2: Fases de desarrollo de la metodología FDD.

## 1. Desarrollo de un modelo general:

- ✓ Cuando comienza el desarrollo, los expertos del dominio están al tanto de la visión, el contexto y los requerimientos del sistema a construir.
- ✓ Se divide el dominio global en áreas que son analizadas detalladamente.
- ✓ Los desarrolladores construyen un diagrama de clases o de objetos por cada área.
- ✓ Se construye un modelo global del sistema.

## 2. Construcción de la lista de funcionalidades<sup>3</sup>:

- ✓ Se elabora una lista de funcionalidades que resuma la funcionalidad general del sistema.
- ✓ La lista es elaborada por los desarrolladores y es evaluada por el cliente.
- ✓ Se divide la lista en subconjuntos según la afinidad y la dependencia de las funcionalidades.
- ✓ La lista es finalmente revisada por los usuarios y los responsables para su validación y aprobación.

## 3. Plan de versiones en base a las funcionalidades a implementar:

- ✓ En este punto se procede a ordenar los conjuntos de funcionalidades conforme a su prioridad y dependencia, y se asigna a los programadores jefes.

## 4. Diseño por funcionalidades y Construcción por funcionalidades (7)

- ✓ Se selecciona un conjunto de funcionalidades de la lista.

---

<sup>3</sup> Una funcionalidad es un ítem útil a los ojos del cliente.



# Capítulo 1 Fundamentación teórica

- ✓ Se procede a diseñar y construir la funcionalidad mediante un proceso iterativo.
- ✓ Una iteración puede tomar de unos pocos días a un máximo de dos semanas. El proceso iterativo incluye inspección de diseño, codificación, pruebas unitarias y de caja negra.

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento.

En el proceso de implementar la funcionalidad también se contemplan como partes del mismo (en otros métodos se describen como actividades independientes) la preparación y ejecución de pruebas, así como revisiones del código (para distribuir el conocimiento y aumentar la calidad) e integración de las partes que componen el software. (8).

Los principios de FDD son pocos y simples:

- ✓ Se requiere un sistema para construir sistemas si se pretende escalar a proyectos grandes.
- ✓ Un proceso simple y bien definido trabaja mejor.
- ✓ Los pasos de un proceso deben ser lógicos y su mérito inmediatamente obvio para cada miembro del equipo.
- ✓ Vanagloriarse del proceso puede impedir el trabajo real.
- ✓ Los buenos procesos van hasta el fondo del asunto, de modo que los miembros del equipo se puedan concentrar en los resultados.
- ✓ Los ciclos cortos, iterativos, orientados por rasgos son mejores.

FDD también define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar, además para conocer el estado actual del desarrollo, realizar mejores estimaciones en proyectos futuros. (9)

## 1.6 Lenguajes utilizados

### 1.6.1 Lenguaje de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (10)

# Capítulo 1 Fundamentación teórica

## 1.6.1.1 JAVA como lenguaje de programación

JAVA es un lenguaje de programación desarrollado por Sun Microsystems<sup>4</sup> a principio de los años 90 con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales diversas y que fuera independiente de la plataforma en la que se vaya a ejecutar.

Es un lenguaje simple porque viene de la misma estructura de C y C++. Es orientado a objeto porque tiene presente todos los conceptos en los que se apoya esta técnica, encapsulación, herencia, polimorfismo entre otros. Sirve para crear todo tipo de aplicaciones (locales, intranet o internet). Altamente fiable y seguro porque se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo, gracias a su función de multihilos, ya que por cada hilo que el programa tenga se ejecutarán en tiempo real muchas funciones. Es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código. (11)

Es un lenguaje distribuido, fue construido con grandes capacidades de conexión TCP/IP, existen librerías de rutina para acceder o interactuar con protocolos HTTP o FTP, esto les permite a los programadores acceder a ficheros en la red con tanta facilidad como a los ficheros locales. Proporciona librerías y herramientas para que los programas puedan ser distribuidos y puedan correr en varias máquinas interactuando.

Es robusto, ya que realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. (12)

## 1.6.2 Lenguaje de modelado

Un lenguaje para el modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo. Toda metodología de desarrollo de software utiliza un lenguaje para el modelado de objetos, para la representación de sus diagramas y artefactos.

---

<sup>4</sup> Empresa informática, dedicada al desarrollo de software.

# Capítulo 1 Fundamentación teórica

## 1.6.2.1 UML como lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML ofrece un estándar para describir un plano del sistema, es decir un modelo previo de lo que se pretende desarrollar, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software pero no especifica en sí mismo qué metodología o proceso usar.

(13)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

## 1.7 Herramientas de desarrollo

Las herramientas de desarrollo son aquellos programas o aplicaciones que tengan cierta importancia en el desarrollo de un programa. Pueden ser de importancia vital (como un ensamblador, un compilador o un editor) o de importancia secundaria, como un IDE (Entorno de Desarrollo Integrado).

### Entorno de Desarrollo Integrado

Existen diferentes formas de introducir un programa Java en una computadora. Se puede utilizar un Ambiente de Desarrollo Integrado (IDE, por sus siglas en inglés) o un editor de textos plano. Un IDE es más que una larga pieza de software, que permite introducir, compilar y ejecutar programas. La introducción, compilación y ejecución son parte del desarrollo de un programa y están integradas juntas en

# Capítulo 1 Fundamentación teórica

un ambiente, de ahí el nombre ambiente de desarrollo integrado. Algunos IDE son gratuitos y otros muy caros. (14)

## 1.7.1 NetBeans 6.9

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Al día de hoy hay disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. (15)

## 1.7.2 Eclipse Galileo

La web oficial de Eclipse define al mismo como una plataforma, abierta para todo y para nada en particular (3). Eclipse es una plataforma porque no se encuentra acabada en su totalidad, pero está diseñado para que sea extensible indefinidamente con la adecuada implementación de plug-in. Es un ambiente de desarrollo integrado (IDE) porque provee de herramientas que facilitan el trabajo en el desarrollo de software, porque permite administrar el espacio de trabajo, porque permite compilar, correr y depurar aplicaciones, porque además posee herramientas que permiten compartir elementos y control de versión sobre el código fuente. La característica clave de Eclipse es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plug-ins que van conformando la funcionalidad final. La forma en que los plug-ins interactúan es mediante interfaces o puntos de extensión; así, las nuevas aportaciones se integran sin dificultad ni conflictos. El Eclipse es neutral y apropiado para todo, puesto que ha sido utilizado para desarrollar todo tipo de aplicaciones, y ha sido altamente probado en cualquier ambiente, ya sea en la construcción de servicios web, aplicaciones de escritorio, juegos, etcétera. Como se refleja en lo anterior el Eclipse tiene una gran capacidad y versatilidad, lo cual es permitido por su arquitectura. (3)

# Capítulo 1 Fundamentación teórica

## 1.7.3 Apache Tomcat

Axis2 es un motor de servicios web de Apache. Su arquitectura ha sido diseñada desde cero, teniendo en cuenta las lecciones aprendidas con AXIS1. Este facilita el trabajo con servicios web, ya que podemos añadir nuevas funcionalidades y soportar futuras especificaciones sobre estos, lo que permite desplegar los servicios web bajo requisitos más exigentes. Este proporciona la implementación del protocolo SOAP mediante el cual se puede intercambiar información en un sistema distribuido y descentralizado, este está basado en el lenguaje XML. AXIS2 también proporciona asistentes para trabajar con archivos WDSL (es el lenguaje XML para describir la interfaz publicada por el servicio web, describe la interfaz del servicio, los tipos de datos que usa y dónde se encuentra el servicio), asistentes para la creación de esqueletos de servicio listos para desplegar de forma automática, asistentes para la creación de objetos Java que modelan los mensajes SOAP. Este soporta el modelo de programación asíncrona y síncrona para consumir servicios web, así como protocolos de transporte como HTTP, SMTP, TCP, JMS pudiendo indicar en cada sentido de la comunicación un transporte diferente. Uno de los aspectos fundamentales de AXIS2 es AXIOM (Axis Object Model), AXIOM se trata del modelo de objetos que AXIS2 emplea para serializar y deserializar mensajes SOAP.

## 1.8 Herramienta de modelado

Las Herramientas de modelado son sistemas que proporcionan ayuda automatizada a los ingenieros, analistas y desarrolladores durante el ciclo de vida del desarrollo de un software, garantizando una mejora en la calidad del mismo. Permiten la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta agilizan el trabajo. Facilitan la realización de prototipos y el desarrollo conjunto de aplicaciones, simplifican el mantenimiento de los programas y mejoran y estandarizan la documentación. Además aumentan la portabilidad de las aplicaciones, facilitan la reutilización de componentes y posibilitan un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

### 1.8.1 Visual Paradigm 6.4

El Visual Paradigm es una herramienta que emplea UML como lenguaje de modelado principal. Es fácil de instalar, actualizar y puede ser utilizado durante todo el ciclo de vida del desarrollo de software.

Es una herramienta que soporta varios usuarios trabajando en un mismo proyecto, generando la documentación del mismo automáticamente en varios formatos y permite control de versiones. Brinda la

# Capítulo 1 Fundamentación teórica

posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código.

Visual Paradigm es multiplataforma, lo cual le permite al usuario emplearla en varios sistemas operativos como Windows, Linux, Unix y otros; además se encuentra disponible en distintas versiones, facilitando también las licencias especiales para fines académicos.

Visual Paradigm ofrece:

- ✓ Entorno de creación de diagramas para UML.
- ✓ Diseño centrado en casos de uso y enfocado al negocio.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad en múltiples plataformas. (16)

## 1.9 Sistema Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de lenguajes para definir, manipular y consultar datos garantizando la integridad, confidencialidad y seguridad de los mismos. (6)

### 1.9.1 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS), es una derivación libre del proyecto Postgres, esta versión incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, pero a pesar de esto PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. Una de sus principales características es que soporta distintos tipos de datos, además de los del tipo base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits<sup>5</sup> y permite la creación de tipos propios. (17)

Ventajas de PostgreSQL:

- ✓ Posee una gran escalabilidad. Es capaz de ajustarse al número de unidades centrales de procesamiento (CPU) y a la cantidad de memoria que posee el sistema de forma óptima,

---

<sup>5</sup> Acrónimo de Binary digit (dígito binario). Un bit es un dígito del sistema de numeración binario.

# Capítulo 1 Fundamentación teórica

haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

- ✓ Implementa el uso de rollback, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- ✓ Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- ✓ El único costo asociado a él, es el de conocerlo pues su código fuente está disponible bajo la más liberal de las licencias del código abierto: la licencia BSD (Distribución de Software Berkeley), que permite usarlo modificarlo y distribuirlo en productos comerciales o no comerciales, sin costo alguno.
- ✓ Requiere pocos recursos de hardware y la simplificación del proceso de administración de licencias de software, que no es necesario cuando se usa software libre.
- ✓ PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- ✓ Puede lidiar con gran volumen de datos. (17)

## 1.10 Marcos de trabajo

Un marco de trabajo es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (18)

### 1.10.1 Spring 2.5.5

Spring es un marco de trabajo que se integra con Java, liviano de aplicación, se puede emplear en todo tipo de aplicaciones java, ya sean pequeñas aplicaciones web o voluminosos sistemas que distribuyen su carga entre varios servidores. Diseñado en módulos, con funcionalidades específicas y consistentes con otros módulos, te facilita el desarrollo de funcionalidades específicas y hace que la curva de aprendizaje sea favorable para el desarrollador. (19)

Spring proporciona las siguientes características:

Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control. Esto hace que la configuración de aplicaciones sea rápida y sencilla.

# Capítulo 1 Fundamentación teórica

Funcionalidad de Programación Orientada a Aspectos totalmente integrada en la gestión de configuración de Spring. Se puede aplicar este paradigma de programación a cualquier objeto gestionado por Spring, añadiendo aspectos como la gestión de transacciones declarativa.

Ofrece un soporte especial para Hibernate añadiendo convenientes características de Inversión de control lo que soluciona mucho de los problemas básicos de la integración con Hibernate. Todo ello cumpliendo con las transacciones genéricas de Spring y la jerarquía de excepciones DAO.

## 1.10.2 Hibernate 3

Hibernate es una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Hibernate es una herramienta ORM completa que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente como el producto de código abierto líder en este campo gracias a sus prestaciones, buena documentación y estabilidad. Es valorado por muchos incluso como solución superior a productos comerciales dentro de su enfoque, siendo una muestra clara de su reputación y soporte la reciente integración dentro del grupo JBoss que seguramente generará iniciativas muy interesantes para el uso de Hibernate dentro de este servidor de aplicaciones. (20)

Las características de esta tecnología son las siguientes:

- ✓ Soluciona el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional).
- ✓ Es flexible en cuanto al esquema de tablas utilizado, lo que facilita la adaptación de su uso sobre una base de datos ya existente.
- ✓ Tiene la funcionalidad de crear la base de datos a partir de la información disponible.
- ✓ Ofrece un lenguaje de consulta de datos llamado HQL<sup>6</sup>.
- ✓ Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE.

---

<sup>6</sup> Acrónimo de Hibernate Query Languages.



# Capítulo 1 Fundamentación teórica

## 1.11 Conclusiones parciales

Se realizó un estudio de algunos de los sistemas de emisión de documentos de identificación existentes, centrándose en los módulos de administración concluyendo que ninguno de estos se ajusta a las necesidades de la solución, debido a que estos no concuerdan con las características específicas del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones sino con las características del sistema al cual pertenece. Se analizaron los modelos de control de acceso MAC, DAC, TBAC y RBAC, de estos se seleccionó el modelo RBAC específicamente hasta el nivel RBAC básico para el control de permisos a las funcionalidades del sistema.

Se realizó un estudio de la metodología, tecnologías y herramientas a utilizar dentro de las cuales se encuentran: metodología ágil FDD, como lenguaje de programación Java, lenguaje de modelado UML, como IDE de desarrollo Netbeans 6.9 y Eclipse Galileo. Visual Paradigm 6.4 como herramienta de modelado, PostgreSQL 8.3 como gestor de base de datos, Apache Tomcat como servidor de aplicaciones.

# Capítulo 2 Características y diseño del Sistema

## Capítulo 2 Propuesta de Solución.

### 2.1 Introducción

El módulo de administración a desarrollar cuenta con una serie de funcionalidades para la configuración y control del Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones las cuales son:

- ✓ Gestión de organización.
- ✓ Gestión de tipo de organización.
- ✓ Gestión de usuarios.
- ✓ Gestión de responsable de firma.
- ✓ Gestión de mensajero de organización.
- ✓ Realización de convenio de supresión de visa.
- ✓ Definición de otorgamiento de acreditación.
- ✓ Planificación de citas.
- ✓ Gestión de roles.

Para el desarrollo de este módulo de administración se definieron dos iteraciones. En esta segunda iteración, rigiéndose por las fases de la metodología definida se desarrollarán las siguientes funcionalidades:

- ✓ Gestión de usuarios.
- ✓ Gestión de roles.
- ✓ Planificación de citas.
- ✓ Definición de otorgamiento de acreditación.
- ✓ Definición de convenio de supresión de visa.

Durante este capítulo se identifican los procesos de negocio relacionados con el objeto de estudio, identificándose también las principales funcionalidades.

### 2.2 Modelo conceptual del dominio

El Modelo de Dominio (Figura 33) es una representación visual estática del entorno real del proyecto. Es decir, un diagrama con los objetos que existen (reales) relacionados a la problemática a resolver y las

# Capítulo 2 Características y diseño del Sistema

relaciones que hay entre ellos. Tiene como principal objetivo ayudar a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

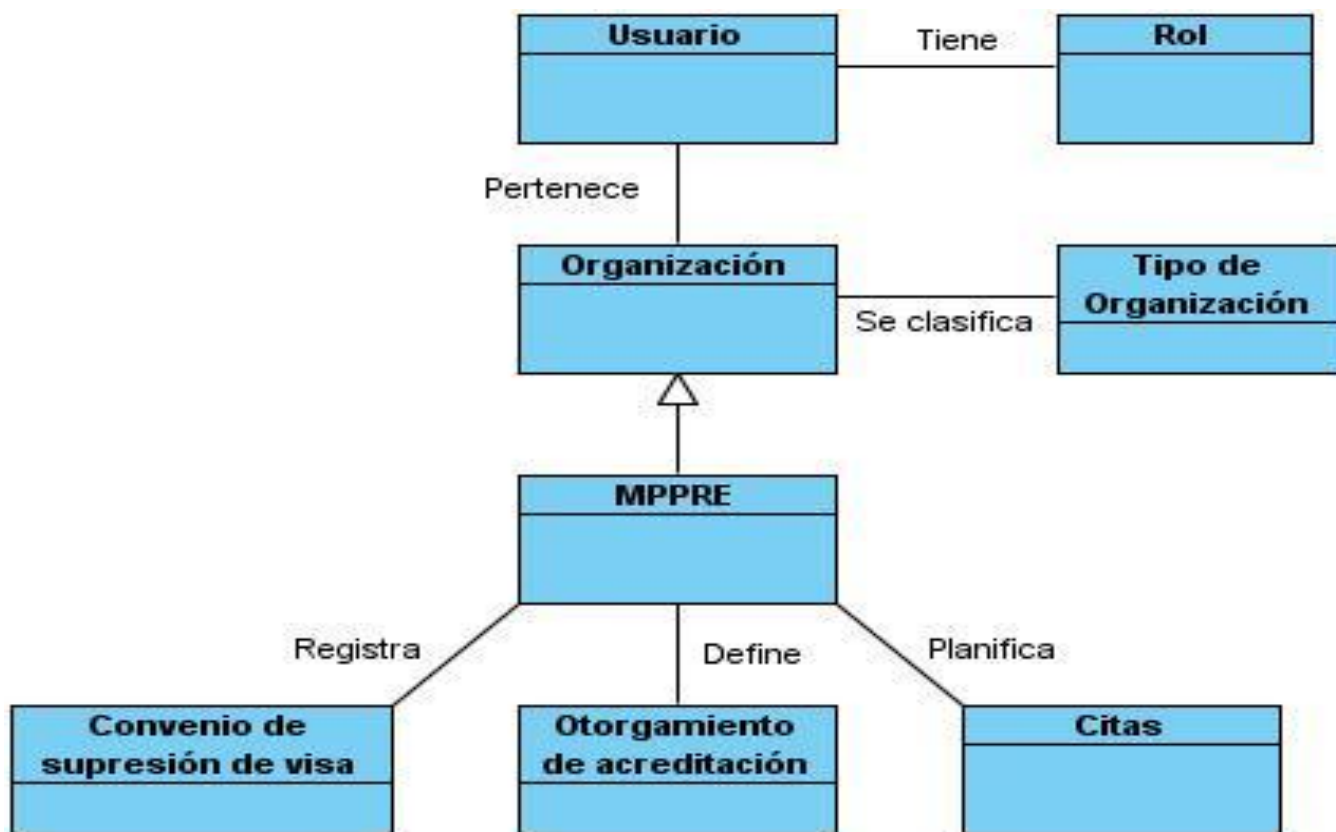


Figura 3 Modelo conceptual de dominio: “módulo administración”.

## 2.2.1 Descripción del modelo conceptual de dominio

El siguiente modelo conceptual de dominio abarca los conceptos principales pertenecientes al módulo de administración del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones.

Primeramente se tiene una Organización que es la entidad donde radican los funcionarios diplomáticos y de servicios quienes solicitan documentos de identificación, la cual se clasifica en diferentes tipos de organizaciones, por ejemplo: Institución Gubernamental, Ministerio, Embajada, Organismo Internacional.

Todos los usuarios pertenecen a una organización a los cuales se les asignan roles que indican los permisos que tendrán en el sistema, estos usuarios pueden ser trabajadores del MPPRE que intervienen en el proceso de emisión de los documentos de identificación o funcionarios de otras organizaciones que realizan las solicitudes correspondientes.

# Capítulo 2 Características y diseño del Sistema

El MPPRE es una de estas organizaciones, específicamente la cual se encarga directamente de los trámites pertenecientes al proceso de Emisión de Pasaportes Diplomático, de Servicio y Acreditaciones.

Para el desarrollo de este proceso se necesita disponer de la información generada por:

- ✓ Convenio de supervisión de visa: En el sistema quedan registrados los países que tiene convenio de supresión de visa con Venezuela. esta información es utilizada por la secretaria cuando revisa la solicitud, donde comprueba la validez de la información sobre visa del titular.
- ✓ Otorgamiento de acreditación: El administrador define para cada titular que solicita acreditación si se le otorga carnet y/o sello en el pasaporte. Esta información se utiliza para llevar a cabo la personalización correspondiente.
- ✓ Citas: Una planificación de trabajo donde se define el horario laborable de la mañana y de la tarde. La cantidad de trámites a realizar esta dado por el tiempo que se definió trabajar entre el tiempo necesario para llevar a cabo un trámite. Este plan es definido por el administrador y revisado por el titular una vez que haya sido aprobada su solicitud a través del cual reserva el día que se presentará en la oficina para captura de datos e información biométrica.

## 2.3 Construcción de la lista de funcionalidades

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Estos describen los servicios que se espera que el sistema cumpla para satisfacer las necesidades del usuario y define el comportamiento del sistema. (29)

Para la realización del Módulo de Administración se llevó a cabo un estudio de todas las funcionalidades que este debe cumplir, con esta información se hizo la especificación de requisitos para tener un mejor entendimiento de lo que el módulo debe hacer y poder implementar el mismo con la calidad requerida.

### 2.3.1 Lista de funcionalidades

Área Temática	Actividades	Funcionalidades
Módulo Administración.	Gestionar Usuarios.	<ol style="list-style-type: none"><li>1. Crear un usuario.</li><li>2. Obtener todos los usuarios.</li><li>3. Modificar datos de un usuario.</li><li>4. Eliminar un usuario.</li><li>5. Desactivar un usuario.</li></ol>
	Definir convenio de supresión	<ol style="list-style-type: none"><li>6. Modificar el convenio de supresión de visa a</li></ol>

# Capítulo 2 Características y diseño del Sistema

	de visa.	<p>un país determinado.</p> <p>7. Obtener países con convenio de supresión de visa.</p> <p>8. Obtener países sin convenio de supresión de visa.</p>
	Definir otorgamiento de acreditación.	<p>9. Obtener listado de la clasificación del personal.</p> <p>10. Modificar los datos para el otorgamiento de acreditación.</p> <p style="padding-left: 40px;">a. Funcionarios</p> <p style="padding-left: 40px;">b. Familiar</p> <p style="padding-left: 40px;">c. Personal de servicio domestico</p>
	Gestionar Planificación semanal de citas.	<p>11. Registrar planificación diaria de citas.</p> <p>12. Modificar planificación diaria de citas.</p> <p>13. Registrar planificación semanal de citas.</p> <p>14. Modificar planificación semanal de citas</p> <p>15. Obtener planificación semanal de citas.</p>
	Gestionar Roles.	<p>16. Crear un rol.</p> <p>17. Modificar un rol.</p> <p>18. Obtener todos los roles.</p> <p>19. Activar un rol.</p> <p>20. Eliminar un rol.</p>

Tabla 2 Lista de funcionalidades.

Una vez que los requisitos funcionales han sido identificados, entonces se describen estos requisitos. La descripción facilita la implementación del sistema ya que en ella se definen las entidades que se van a utilizar, las condiciones que debe cumplir la aplicación inicialmente, se explica la secuencia de actividades entre el usuario y el sistema para llevar a cabo las funcionalidades descritas, así como el resultado que se espera después de realizada cada una de las operaciones. En el Anexo 1 se encuentra la descripción detallada de cada uno de los requisitos funcionales.

## 2.4 Planeación de iteraciones por funcionalidades.

Se hace necesario estimar de forma adecuada el tiempo que les tomará a los programadores diseñar,

# Capítulo 2 Características y diseño del Sistema

implementar y validar cada una de las funcionalidades antes definidas, esto no es más que otra de las actividades de la fase de planificación. Dependiendo de la prioridad y dependencia de las funcionalidades se decide cuáles de ellos se implementarán en las primeras iteraciones del ciclo.

Iteración 1: Se desarrollarán las funcionalidades pertenecientes a la segunda iteración de desarrollo del módulo administración que presenten una alta prioridad.

Iteración 2: Se desarrollarán las restantes funcionalidades pertenecientes a la segunda iteración de desarrollo del módulo administración.

Iteración 1.

Gestionar usuario.			Funcionalidades 5	
Funcionalidad	Modelo general	Diseño	Construcción	Prueba
	Plan	Plan	Plan	Plan
Crear un usuario.	04/02/11	06/02/11	09/02/11	17/02/11
Obtener todos los usuarios.	04/02/11	06/02/11	09/02/11	17/02/11
Modificar los datos de un usuario.	04/02/11	06/02/11	09/02/11	17/02/11
Desactivación de un usuario.	04/02/11	06/02/11	09/02/11	17/02/11
Eliminar un usuario.	04/2/11	06/02/11	09/02/11	17/02/11

Tabla 3 Planeación de la funcionalidad: Gestionar Usuario.

Gestionar roles.			Funcionalidades 5	
Funcionalidad	Modelo general	Diseño	Construcción	Prueba
	Plan	Plan	Plan	Plan
Crear un rol.	04/12/11	12/02/11	18/02/11	23/02/11
Obtener todos los roles	04/12/10	12/02/11	18/02/11	23/02/11
Modificar un rol.	04/12/10	12/02/11	18/02/11	23/02/11
Desactivar un rol	04/12/10	12/02/11	18/02/11	23/02/11
Eliminar un rol	04/12/10	12/02/11	18/02/11	23/02/11

# Capítulo 2 Características y diseño del Sistema

Tabla 4 Planeación de la funcionalidad: Gestionar roles.

Iteración 2

Planificación semanal de citas.			Funcionalidades 6	
Funcionalidad	Modelo general	Diseño	Construcción	Prueba
	Plan	Plan	Plan	Plan
Registrar planificación diaria de citas.	05/12/10	25/02/11	05/03/11	09/03/11
Registrar planificación semanal de citas.	05/12/10	25/02/11	05/03/11	09/03/11
Obtener planificación diaria de citas.	05/12/10	25/02/11	05/03/11	09/03/11
Obtener planificación semanal de citas.	05/12/10	25/02/11	05/03/11	09/03/11
Modificar planificación semanal de citas.	05/12/10	25/02/11	05/03/11	09/03/11
Modificar planificación diaria de citas.	05/12/10	25/02/11	05/03/11	09/03/11

Tabla 5 Planeación de la funcionalidad: Gestionar planificación de citas.

Definir otorgamiento de acreditación.			Funcionalidades 3	
Funcionalidad	Modelo general	Diseño	Construcción	Prueba
	Plan	Plan	Plan	Plan
Obtener listado de la clasificación del personal.	05/12/10	27/02/11	02/03/11	07/03/11
Modificar los datos para el otorgamiento de acreditación.	05/12/10	27/02/11	02/03/11	07/03/11

Tabla 6 Planeación de la funcionalidad: Gestionar otorgamiento de acreditación.

Definir convenio de supresión de visa.			Funcionalidades 5	
Funcionalidad	Modelo general	Diseño	Construcción	Prueba
	Plan	Plan	Plan	Plan

# Capítulo 2 Características y diseño del Sistema

Obtener países sin convenio de supresión de visa.	05/12/10	27/02/11	03/03/11	09/03/11
Obtener países con convenio de supresión de visa.	05/12/10	27/02/11	03/03/11	09/03/11
Modificar convenio de supresión de visa a un país determinado.	05/12/10	27/02/11	03/03/11	09/03/11

Tabla 7 Planeación de la funcionalidad: Gestionar convenio de supervisión.

## 2.5 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Las propiedades son las características que hacen al producto atractivo, usable, rápido y confiable. Normalmente, están vinculados a requisitos funcionales. Una vez que se conoce lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requisitos no funcionales son fundamentales el éxito del producto.

### Requisitos de apariencia o interfaz externa

**RnF1** El sistema debe contar con una interfaz fácil, amigable y sencilla que permita a los usuarios finales interactuar con el sistema.

**RnF2** El tamaño y el tipo de letra deberá ser uniforme para todas las interfaces.

**RnF3** Los iconos y botones que se utilicen en las interfaces, deben de estar relacionados con la función a realizar.

**RnF4** El tamaño de los campos debe corresponderse con el tamaño de la información a tratar.

### Requisitos de usabilidad.

**RnF5** Los usuarios que van a utilizar el sistema deberán tener algún conocimiento previo del manejo de una computadora personal.

### Requisitos de diseño

**RnF6** El lenguaje seleccionado para la aplicación es java.

**RnF7** IDE del lado del cliente Netbeans 6.9.

**RnF8** IDE del lado del servidor Eclipse Galileo.

**RnF9** Visual Paradigm 6.4 para el modelado del sistema.



# Capítulo 2 Características y diseño del Sistema

**RnF10** El gestor de base de datos para la persistencia de información relativa a los procesos y al negocio será PostgreSQL 8.3.

## Requisitos de seguridad

**RnF11** Al sistema sólo podrá acceder el personal autorizado mediante su autenticación.

**RnF12** La autenticación permitirá tener controlado el acceso a la información.

**RnF13** Una vez autenticados los usuarios podrán acceder a la información de forma rápida y segura obteniendo los datos deseados en un momento dado.

**RnF14** Las funcionalidades que brinda el sistema deben corresponderse según el permiso definido para cada usuario según su rol.

**RnF15** Se realizarán copias de seguridad periódicamente.

## Requisitos de eficiencia

**RnF16** Buen rendimiento del sistema.

**RnF17** Recursos de Hardware: como mínimo 1GB de RAM para su funcionamiento y 5 GB de espacio libre en el disco.

## Requisitos legales

**RnF18** La herramienta de modelado seleccionada: Visual Paradigm, aunque es propietaria, en estos momentos la UCI tiene la licencia para su uso.

**RnF19** El lenguaje seleccionado para la implementación: Java es software libre.

## Requisitos políticos-culturales

**RnF20** En las interfaces del sistema no debe contener palabras en otros idiomas distintos al español.

**RnF21** El sistema debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

**RnF22** El sistema debe tener una interfaz que esté acorde con el lugar donde se implantará, es decir, que refleje los ideales de la organización y que sea amena, cumpliendo con las pautas de diseño.

## 2.6 Definición de la arquitectura

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de:

- ✓ Definir los módulos principales.

# Capítulo 2 Características y diseño del Sistema

- ✓ Definir las responsabilidades que tendrá cada uno de estos módulos.
- ✓ Definir la interacción que existirá entre dichos módulos
- ✓ Control y flujo de datos.
- ✓ Secuenciación de la información.
- ✓ Protocolos de interacción y comunicación.
- ✓ Ubicación en el hardware.

La Arquitectura del Software aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. (24)

Para el desarrollo del sistema de Emisión de Pasaporte Diplomáticos, de Servicio y Acreditaciones de la República Bolivariana de Venezuela se definieron un conjunto de patrones que estructuran su arquitectura, dando como resultado que cada módulo de los que integran el sistema se ajustará entonces a estos patrones. El a desarrollar en este trabajo debe ser implementado siguiendo entonces este modelo.

## 2.6.1 Arquitectura cliente/servidor

La arquitectura cliente servidor también llamada modelo cliente – servidor, es una forma de dividir o separar al cliente (en este caso la interfaz gráfica) del servidor como se muestra en la Figura 44. El cliente es aquel que inicia la el diálogo o comunicación solicitando información y el servidor es el que procesa las solicitudes del cliente y retorna una respuesta.

Para que los clientes y los servidores puedan establecer una comunicación se necesita una infraestructura de comunicaciones o protocolo de comunicación, estos brindan los mecanismos básicos para el direccionamiento y transporte de los datos.

Esta arquitectura presenta disímiles ventajas ya que la aplicación cliente pueden actuar tanto como una sola entidad que como entidades separadas y las funciones de cada uno pueden correr sobre plataformas diferentes, además de que la aplicación cliente no necesita conocer la lógica del servidor, ya que solo interactúa con esta a través de una interfaz y los cambios que puedan surgir en el servidor implican pocos o ningún cambio en el cliente.

# Capítulo 2 Características y diseño del Sistema

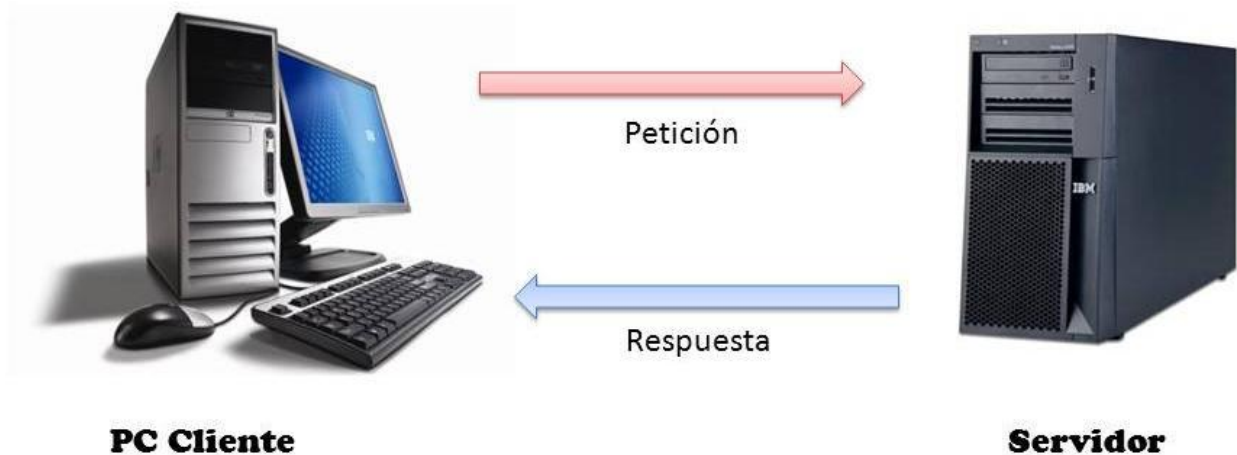


Figura 4: Arquitectura cliente - servidor.

Está compuesta por tres capas lógicas. Estas dan un alto nivel de encapsulamiento de las responsabilidades, permitiendo reducir al máximo el acoplamiento y aumentar la reutilización entre las mismas. Esta distribución de las capas permite que se realicen grandes cambios sin repercutir en las demás capas una vez que estas tengan bien definidas la comunicación entre ellas.

Estas capas están distribuidas de manera tal que los componentes de una capa solo pueden acceder a los componentes de la capa inmediata inferior, de esta manera se reduce la dependencia entre capas, ya que las capas inferiores no tienen conocimiento sobre los detalles de las capas superiores. En la Figura 55 se muestra la vista lógica de las capas.

# Capítulo 2 Características y diseño del Sistema

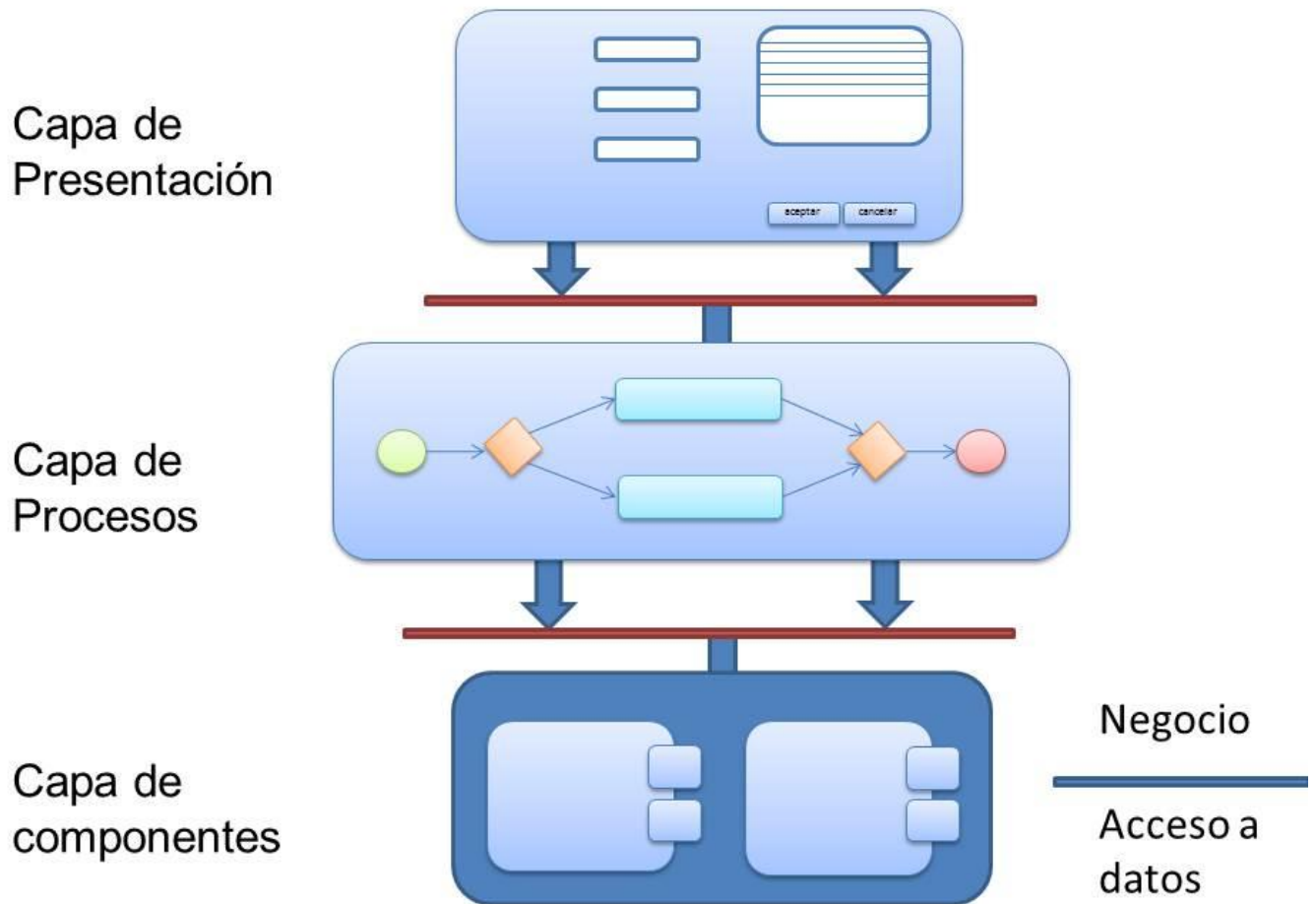


Figura 5 Vista Lógica de la arquitectura.

Descripción de las capas.

A continuación se describen las tres capas que posee la arquitectura utilizadas en el desarrollo del módulo de Administración.

**Capa de presentación:** En esta capa se presencia la interacción entre el cliente y la aplicación de escritorio, de la cual forma parte la propuesta de solución. El cliente se sirve de las funcionalidades expuestas a través de servicios web por la capa inferior (capa de procesos).

**Capa de procesos:** En esta capa se encuentran automatizados los procesos del sistema, tiene la responsabilidad de brindar estos servicios a la capa superior reutilizando las funcionalidades que le brinda

# Capítulo 2 Características y diseño del Sistema

la capa de componentes. Para el caso de la propuesta de solución en específico las funcionalidades desarrolladas en esta capa son tareas libres, es decir, son tareas que no están dentro del proceso por lo tanto se pueden ejecutar en cualquier momento de este.

Capa de componentes: Esta capa es la encargada de implementar todas las funcionalidades utilizadas desde la capa de procesos. Está dividida en otras dos capas, la capa lógica de negocio en la cual se implementan las funcionalidades en de acuerdo a las necesidades identificadas en el negocio y la capa de acceso a datos la cual se encarga de la persistencia de todos los datos utilizados.

## 2.7 Patrones de diseño

Los patrones de diseño son aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software. Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que pueda utilizarse un millón de veces sin tener que hacer dos veces lo mismo.

Los patrones de diseño tienen como objetivo:

- ✓ Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- ✓ Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- ✓ Formalizar un vocabulario común entre diseñadores.
- ✓ Estandarizar el modo en que se realiza el diseño.
- ✓ Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente. (25)

Patrón CRUD<sup>7</sup>: Este es un patrón que se usa en cualquier sistema transaccional ya que puede ser reducido a un juego de acciones como alta o entrada de datos, búsqueda-filtrado, consulta, baja o modificación, todas ellas representadas por la combinación de elementos gráficos dispuestos en una interfaz. (30). Este patrón se ve ejemplificado claramente en la capa de acceso a datos ya que es allí donde se consultan, agregan, modifican y elimina los datos.

---

<sup>7</sup> Acrónimo de *Create, Read, Update, Delete* (Crear, Consultar, Modificar, Eliminar).

# Capítulo 2 Características y diseño del Sistema

Patrones GRASP<sup>8</sup>: Estos patrones son utilizados en la mayoría de las clases diseñadas para el desarrollo de las aplicaciones. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Bajo acoplamiento: Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Debe haber pocas dependencias entre las clases. Uno de los principios para proteger el software frente al cambio es mantener bajo el acoplamiento entre clases, cuanto menor sea el acoplamiento entre clases, menor influencia tendrán los cambios. Este patrón se ve ilustrado en la realización de todas las tareas libres en la capa de proceso, la implementación de cada una de estas recurre a la interfaz indicada ya que esta es la única que contiene los elementos que necesita implementar. Ejemplo de lo anterior sería `ControlUsuarioFTaskImpl` recurre a la interfaz `ControlUsuarioFTask`.

Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es decir se tiene en consideración a qué clase pertenece un método, este principio sugiere que se asigne a la clase que más sepa del método. Esto es una consecuencia del principio alta cohesión, ya que al asignar los métodos a las clases que tienen la información necesaria para ejecutarlo se están creando clases altamente cohesionadas. Este patrón se ve ejemplificado en la clase `GestionPlanCitaFTaskImpl` la cual contiene entre sus métodos `registrarDiaSemana`, `registrarSemana` puesto que es la clase indicada en estas funciones.

Alta cohesión: Cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, se buscan soluciones que asignen los métodos a las clases de forma coherente, de estas forma se obtuvieron clases cohesionadas. Al realizar un cambio en una clase muy cohesionada, todos los métodos que pueden verse afectados, toda la información que necesitamos controlar, estará a la vista, en el mismo fichero. Este patrón tiene estrecha relación con el bajo acoplamiento, ya que un diseño cohesionado tiene un bajo acoplamiento entre clases. Este patrón está presente en todas las clases de tareas libres de la capa de procesos, ya que en cada clase se implementan los métodos que le competen según su

---

<sup>8</sup> Acrónimo de *General Responsibility Assignment Software Patterns* (Patrones de Software para la asignación General de Responsabilidad).

# Capítulo 2 Características y diseño del Sistema

concepción y finalidad. Un ejemplo claro es el método registrarUsuario implementado en la clase ControlUsuarioFTaskImpl.

Patrón controlador: La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario. El patrón controlador sirve de intermediario entre la interfaz y el algoritmo, de tal forma que al recibir los datos del usuario lo envíe a las distintas clases según la petición del usuario. Este patrón se ve ejemplificado en cada una de las clases controladoras de la capa de presentación.

Patrones GOF<sup>9</sup>

Facade (fachada): Este patrón se utiliza para proporcionar una interfaz unificada para un conjunto de funcionalidades lo que facilita su uso. Simplifica el acceso a dicho conjunto de funcionalidades, ya que el necesitado sólo se comunica con ellas a través de una única interfaz. Este patrón es usado en cada una de las tareas libres ya que exponen sus funcionalidades a la capa de presentación a través de interfaces.

## 2.8 Modelo de clases del diseño

Los diagramas de clases del diseño muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Son utilizados para modelar principalmente la vista de diseño estática de un sistema, estos diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

El lenguaje utilizado para especificar una clase de diseño es el mismo que el lenguaje de programación. Seguidamente las operaciones, parámetros, atributos, tipos y demás son especificados utilizando la sintaxis del lenguaje de programación elegido. Se muestran las relaciones de aquellas clases de diseño implicadas con otras clases.

### 2.8.1 Diagrama de clases del diseño

A continuación en la Figura 66 se muestra el diagrama de clases del diseño para las funcionalidades de la Gestión de usuarios. En el Anexo 2 se muestran los diagramas de clases de las restantes funcionalidades.

---

<sup>9</sup> Acrónimo de *Gand of four* (Banda de los cuatro: Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides).

# Capítulo 2 Características y diseño del Sistema

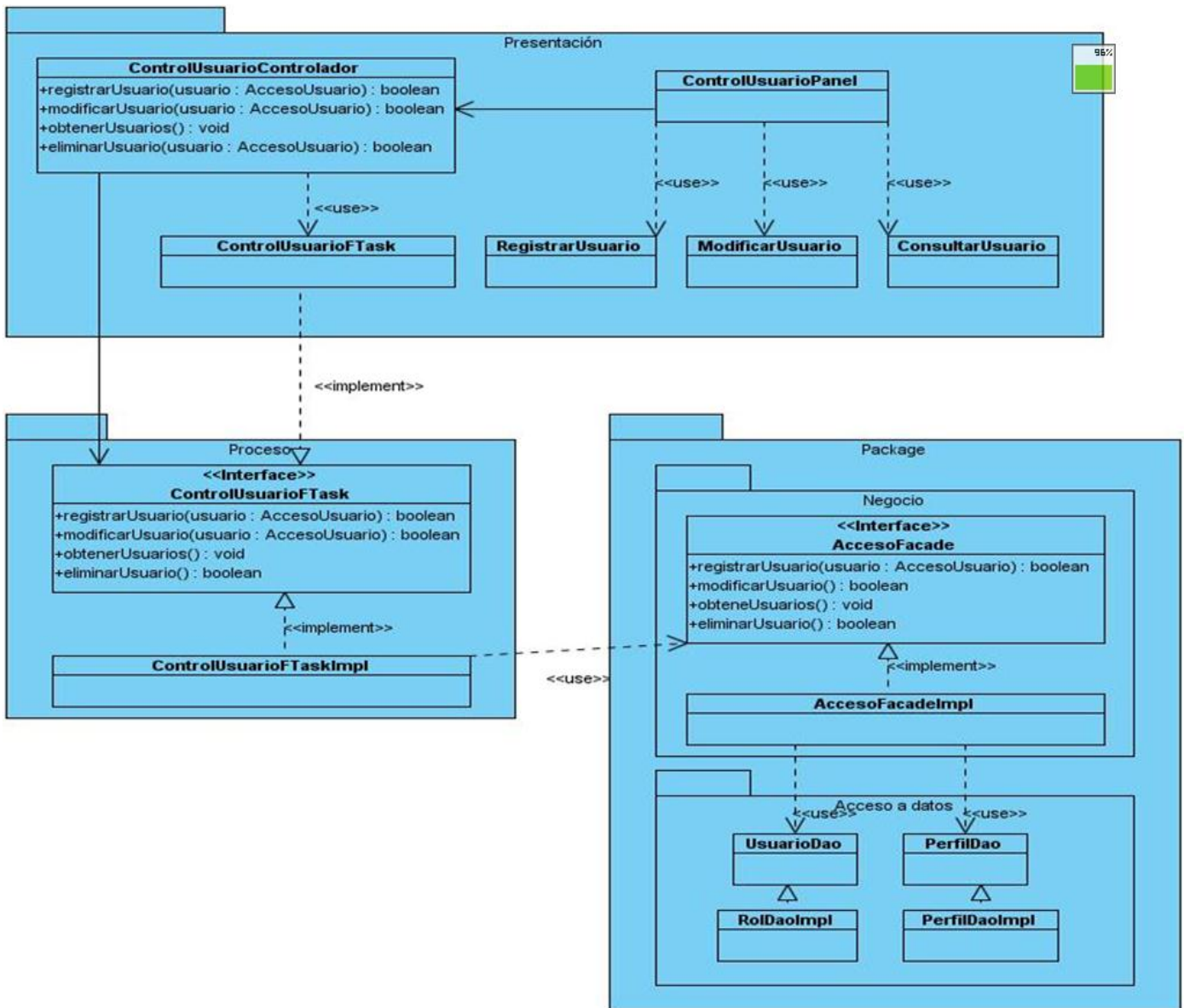


Figura 6 Diagrama de clases del diseño de la funcionalidad Gestionar usuario.

## 2.8.2 Descripción de las clases del diseño

A continuación se describen algunas de las clases más relevantes presentes en el diagrama de clases del diseño anterior presentado. Para cada una de ellas se muestra su nombre, una breve descripción de su función. Se muestra además el objetivo y forma de funcionamiento de todas las operaciones contenidas por la misma, exceptuando aquellos métodos de acceso y transformación (get y set). Consultar Anexo 3



# Capítulo 2 Características y diseño del Sistema

para ver el resto de las descripciones de clases.

Nombre: ControlUsuarioFTask	
Tipo de clase: interfaz	
Está diseñada para la gestión y manejo de todas las acciones básicas que se realizan a los usuarios en la capa de procesos, dándole soporte a la capa de presentación.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	CrearUsuario(AccesoUsuario usuario)
Descripción:	Realiza el proceso de crear un usuario enviando los datos de este a la capa de componentes donde se realizan las acciones necesarias para insertarlo en la base de datos.
Nombre:	ModificarUsuario(AccesoUsuario usuario)
Descripción:	Realiza el proceso de modificar un usuario enviando los datos de la modificación a la capa de componentes donde se realizan las acciones necesarias para ejercer la modificación en la base de datos.
Nombre:	ObtenerUsuarios()
Descripción:	Retorna todos los usuarios del sistema.
Nombre:	eliminarUsuario(String identificador)
Descripción:	Realiza el proceso necesario de eliminar el usuario al cual pertenece el identificador pasado como parámetros.
Nombre:	DesactivarUsuario(String identificador)
Descripción:	Realiza el proceso necesario de eliminar el usuario al cual pertenece el identificador pasado como parámetros.

Tabla 8 Descripción de la clase ControlUsuarioFTaskImpl.

Nombre: ControlUsuarioControlador

# Capítulo 2 Características y diseño del Sistema

Tipo de clase: Controladora	
Está diseñada para el control de las acciones realizadas por el usuario al interactuar con el sistema.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	agregarUsuario(Rol rol)
Descripción:	Registra (o inserta) los datos del usuario que se le pasa como parámetro en la base de datos.
Nombre:	modificarUsuario(Rol rol)
Descripción:	Realiza la función de modificar los datos de un usuario que se le haya pasado por parámetro.
Nombre:	obtenerUsuarios()
Descripción:	Realiza la función de devolver un listado de todos los usuarios.
Nombre:	eliminarUsuario(String identificador)
Descripción:	Elimina el usuario al cual pertenece el identificador pasado como parámetro de la base de datos.
Nombre:	desactivarUsuario(String idRol)
Descripción:	Realiza la operación de desactivar el usuario al cual pertenece el identificador pasado como parámetro de la base de datos.

Tabla 9 Descripción de la clase ControlUsuarioControlador.

Nombre: AccesoFacadeImpl	
Tipo de clase: Interfaz	
Está diseñada para la gestión y manejo de todas las acciones básicas que se realizan a los usuarios en la capa de negocio, dándole soporte a la capa de procesos.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	registrarUsuario(AccesoUsuario accesoUsuario)
Descripción:	Registra (o inserta) el usuario que se le pasa como parámetro en la

# Capítulo 2 Características y diseño del Sistema

	base de datos.
Nombre:	modificarUsuario(AccesoUsuario accesoUsuario)
Descripción:	Realiza la función de modificar los datos de un usuario que se le haya pasado por parámetro.
Nombre:	obtenerUsuarios()
Descripción:	Realiza la función de devolver un listado de todos los usuarios.
Nombre:	EliminarUsuario(String identificador)
Descripción:	Realiza la función de eliminar el usuario al cual pertenece el identificador pasado como parámetro.
Nombre:	DesactivarUsuario(String identificador)
Descripción:	Desactiva el usuario al cual pertenece el identificador pasado como parámetro.

Tabla 10 Descripción de la clase AccesoFacadeImpl.

## 2.9 Diseño de la base de datos

Uno de los pasos fundamentales en la construcción de los componentes, es sin duda el diseño de la base de datos. A continuación en la Figura 77 se muestra la estructura de las tablas de la base de datos.

# Capítulo 2 Características y diseño del Sistema

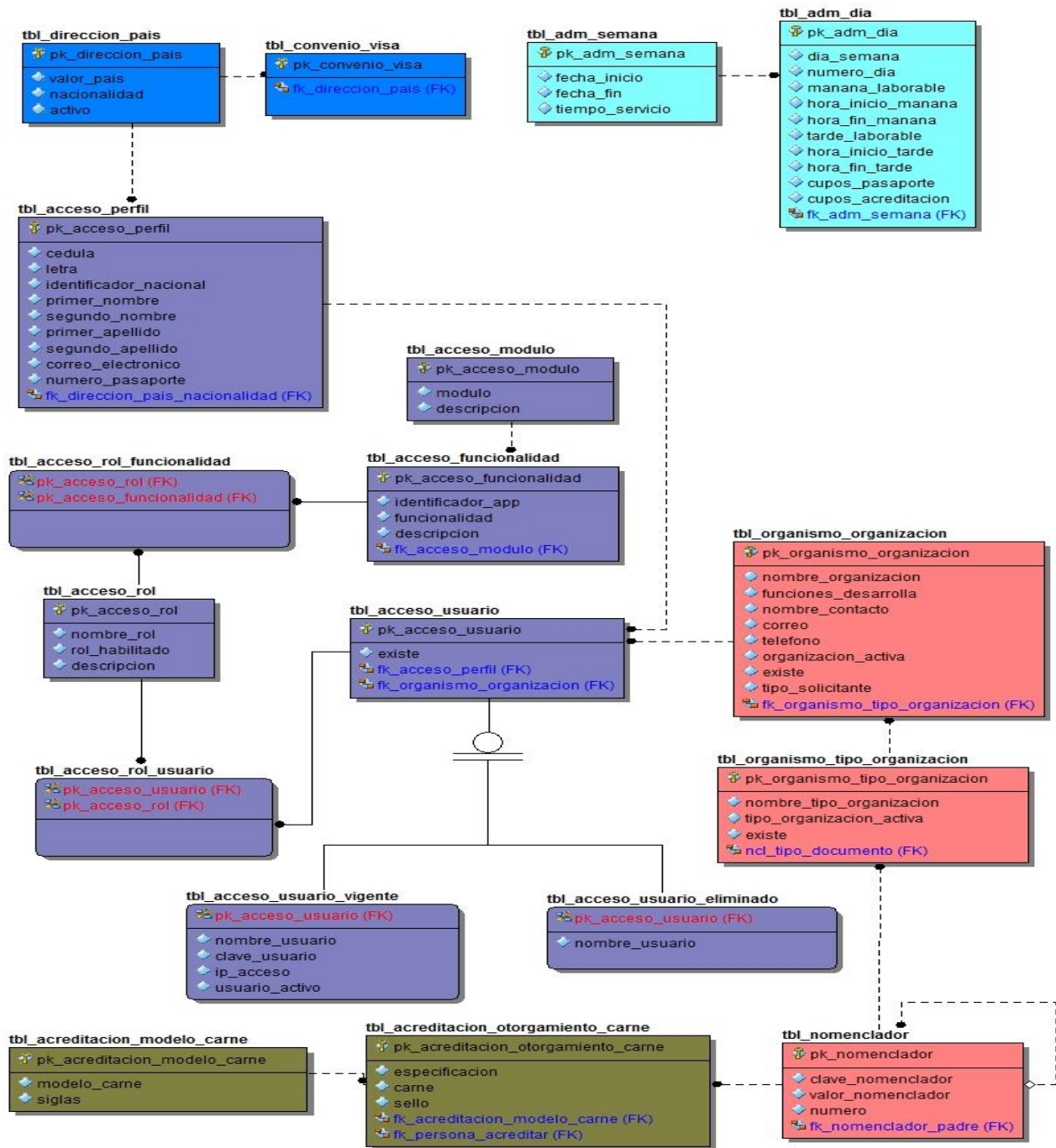


Figura 7 Diseño de la base de datos.

# Capítulo 2 Características y diseño del Sistema

## 2.9.1 Descripción de las tablas de la base de datos

La descripción de una tabla de la base de datos posee su nombre, atributos, tipo de dato y descripción, como se muestra a continuación, pero sólo se documentan las más importantes, el resto se puede encontrar en el Anexo 4.

Nombre de la entidad	tbl_acceso_usuario
Definición	Entidad que representa un usuario.

### Atributos

Nombre del atributo	Tipo de dato	Nulo	Definición
pk_acceso_usuario	CHAR(32)	N	Atributo que representa el identificador de usuario.
existe	BIT	N	Atributo que representa la existencia de usuario.
fk_acceso_perfil	CHAR(32)	N	Atributo que representa el identificador del perfil de usuario.
fk_organismo_organizacion	CHAR(32)	N	Atributo que representa el identificador de organización.

Tabla 11 Descripción de la tabla tbl\_acceso\_usuario.

Nombre de la entidad	tbl_acceso_perfil
Definición	Entidad que representa el perfil de un usuario.

### Atributos

Nombre del atributo	Tipo de dato	Nulo	Definición
pk_acceso_perfil	CHAR(32)	N	Atributo que representa el identificador del perfil de un usuario.
cedula	BIGINT	S	Atributo que representa cedula.
letra	CHAR(1)	S	Atributo que representa letra.

# Capítulo 2 Características y diseño del Sistema

identificador_nacional	VARCHAR(30)	S	Atributo que representa identificador nacional.
primer_nombre	VARCHAR(30)	N	Atributo que representa primer nombre.
segundo_nombre	VARCHAR(30)	S	Atributo que representa Segundo nombre.
primer_apellido	VARCHAR(30)	N	Atributo que representa primer apellido.
segundo_apellido	VARCHAR(30)	S	Atributo que representa Segundo apellido.
correo_electronico	VARCHAR(50)	N	Atributo que representa dirección de correo electrónico.
numero_pasaporte	VARCHAR(20)	S	Atributo que representa número de pasaporte.
fk_direccion_pais_nacionalidad	VARCHAR(4)	N	Atributo que representa el identificador de pais.

Tabla 12 Descripción de la tabla tbl\_acceso\_perfil.

## 3.1 Estándar de codificación

Debido a que el lenguaje de programación seleccionado para el desarrollo del sistema fue Java, a continuación se detallan algunos aspectos en los que se quiere enfatizar a la hora de escribir el código en dicho lenguaje. Muchos de los aspectos que se describirán a continuación se extrajeron del estándar de codificación que define SUN para el uso de este lenguaje. (27)

### Criterios de calidad

Es de vital importancia que durante el desarrollo se tengan en cuenta permanentemente los criterios de calidad mostrados en la (Tabla 13)

Criterio	Objetivo
Facilidad de comunicación	Proporcionar a los usuarios y desarrolladores entradas y salidas fácilmente

# Capítulo 2 Características y diseño del Sistema

	asimilables.
Descripción	Proporcionar y/o plasmar en el código detalles y explicaciones sobre la implementación realizada.
Simplicidad	La implementación realizada debe hacerse de la forma más comprensible posible.

Tabla 13 Criterios de calidad.

## Principios generales de nomenclatura

Los nombres de cada uno de los elementos del programa, clase o interfaz deben ser significativos, su nombre debe explicar, siempre que sea posible, el uso o fin del elemento. La mayoría de los elementos se deben nombrar usando sustantivos (posiblemente compuestos) o formas verbales en imperativo. La forma de construir los nombres será colocando primero el verbo o sustantivo, seguido de cada uno de sus complementos con la primera letra en mayúscula. Ver la descripción específica en la siguiente tabla.

Tipos de identificadores	Reglas para nombres	Ejemplos
Clases	Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Se debe intentar mantener los nombres de las clases simples y descriptivas. Evitar acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML).	class Cliente class ImagenAnimada
Interfaces	Los nombres de las interfaces siguen la misma regla que las clases.	-
Métodos	Los nombres de los métodos deben ser verbos intuitivos que representen lo que se quiere hacer, además empezarán siempre con minúscula. Cuando son compuestos, tendrán la primera letra del nombre en minúsculas, y la primera letra de las palabras siguientes en mayúsculas.	ejecutar(); realizarJugada();
Variables	Excepto las constantes, todas las variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan	int i; char c; float peso;

# Capítulo 2 Características y diseño del Sistema

	<p>con su primera letra en minúscula igual y mayúsculas las que le siguen. Los nombres de variables no deben empezar con los caracteres guion bajo "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje.</p> <p>Los nombres de las variables deben ser cortos pero con significado. Los nombres de variables de un solo carácter se deben evitar, excepto para variables de índices temporales. Nombres comunes para variables temporales son i, j, k, m, y n para enteros; c, d, y e para caracteres.</p>	
Constantes	<p>Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un guión bajo ("_"). (Las constantes ANSI se deben evitar, para facilitar su depuración.)</p>	static final float SIZE = 0.0;

Tabla 14 Reglas específicas para la nombres.

## Declaraciones de las sentencias "package" e "import"

En caso de que la clase pertenezca a un paquete, la sentencia "package" deberá estar incluida dentro del fichero como primera línea. Asimismo después de esta sentencia estará los "import" en caso de su inclusión.

Con el uso de este estándar de codificación será más factible el entendimiento del código ya plasmado para todos los programadores vinculados con el desarrollo de este sistema.

### 2.10 Diagrama de paquete del diseño

El las Figura 88 y Figura 99 se muestran los diagramas de paquete para la capa de proceso y la capa de componentes respectivamente.



# Capítulo 2 Características y diseño del Sistema

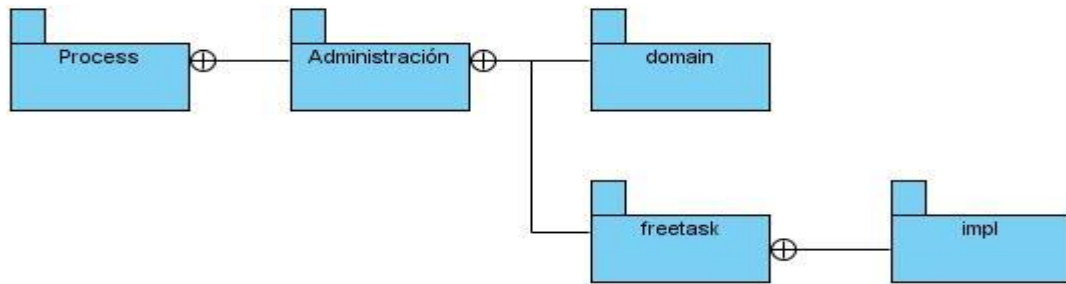


Figura 8 Diagrama de Paquete de la capa de Proceso.

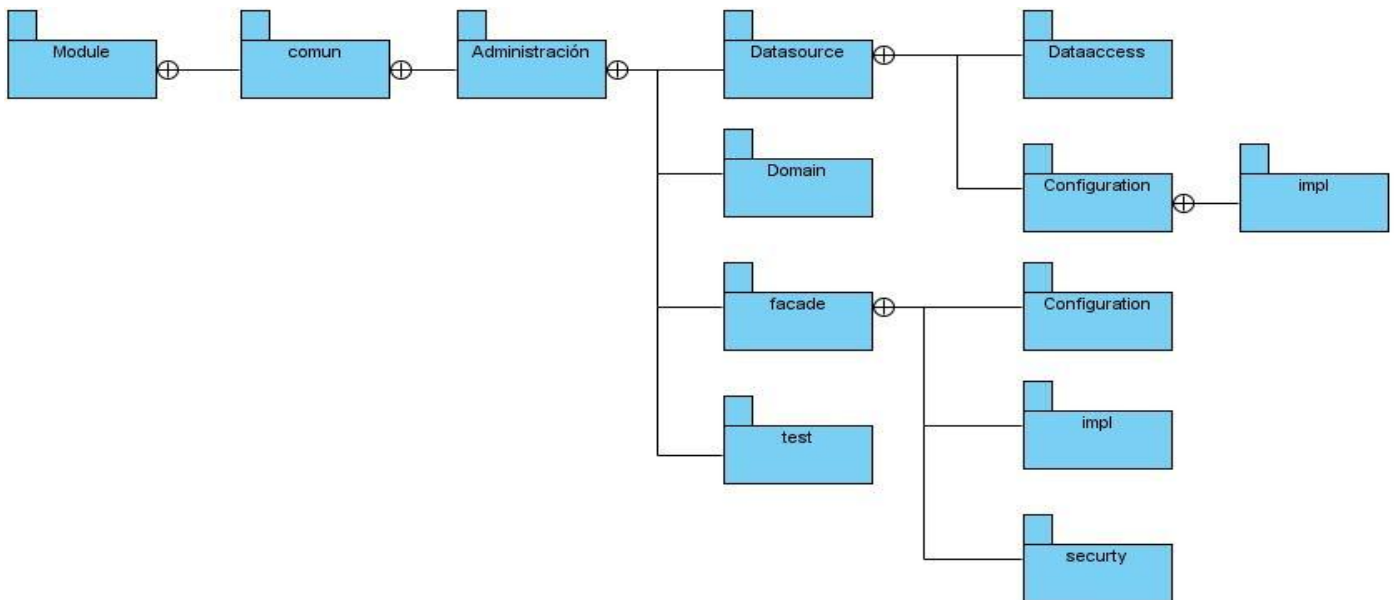


Figura 9 Diagrama de Paquete de la capa de Componentes.

Paquete configuration: Contiene los archivos de configuración de los servicios web, conexión a la base de datos y demás propiedades que necesitan los componentes de negocio para su funcionamiento.

Paquete datasource: Contiene todos los archivos de configuración, clases de consumo de servicios y acceso a datos.

Paquete dataaccess: Se encuentran todas las clases de acceso a datos.

Paquete Impl: Se encuentran las clases que implementan las interfaces correspondientes.

Paquete domain: Contiene todas las clases de dominio o persistentes con que se trabajan en el módulo.

Paquete facade: Contiene todas las clases de negocio con que se trabajan en el módulo.

# Capítulo 2 Características y diseño del Sistema

Paquete test: Contiene todas las clases de pruebas para las diferentes capas de abstracción que se tienen en el módulo.

Paquete freetask: Contiene todas las clases para la solución de funcionalidades que son tareas libres.

## 2.11 Conclusiones parciales

En este capítulo se presentó la realización de los tres primeros procesos secuenciales para el desarrollo de la aplicación usando la metodología de desarrollo definida. Con la ejecución de estos procesos se logró determinar específicamente las funcionalidades del sistema a construir, agrupándolas por afinidad y describiendo cada una de ellas, a través de un plan de iteraciones se le dieron prioridades a las mismas para su construcción. Se realizó un estudio detallado de la arquitectura y patrones a utilizar dando paso a un diseño de clases regido por estos.

# Capítulo 3 Implementación y Prueba

## Capítulo 3 Implementación y Prueba

### 3.2 Introducción

La implementación de un software es la fase que más esfuerzo requiere, en esta fase es donde se le da cumplimiento a cada uno de las funcionalidades que la aplicación debe cumplir. Se presentará el diagrama de componentes y de despliegue correspondientes a la aplicación. Después de implementada la propuesta de solución pasa a la fase de pruebas para la validación de la solución antes de su puesta en explotación.

### 3.3 Diagrama de componentes

Los componentes representan los elementos físicos de un sistema es decir todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otros, un entorno funcional requerido por cualquier proceso predefinido. El diagrama de componentes representa como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponible en el entorno de implementación y en el lenguaje de programación utilizado. A continuación en la Figura 1010 el diagrama de componentes para la funcionalidad Gestionar usuario.

# Capítulo 3 Implementación y Prueba

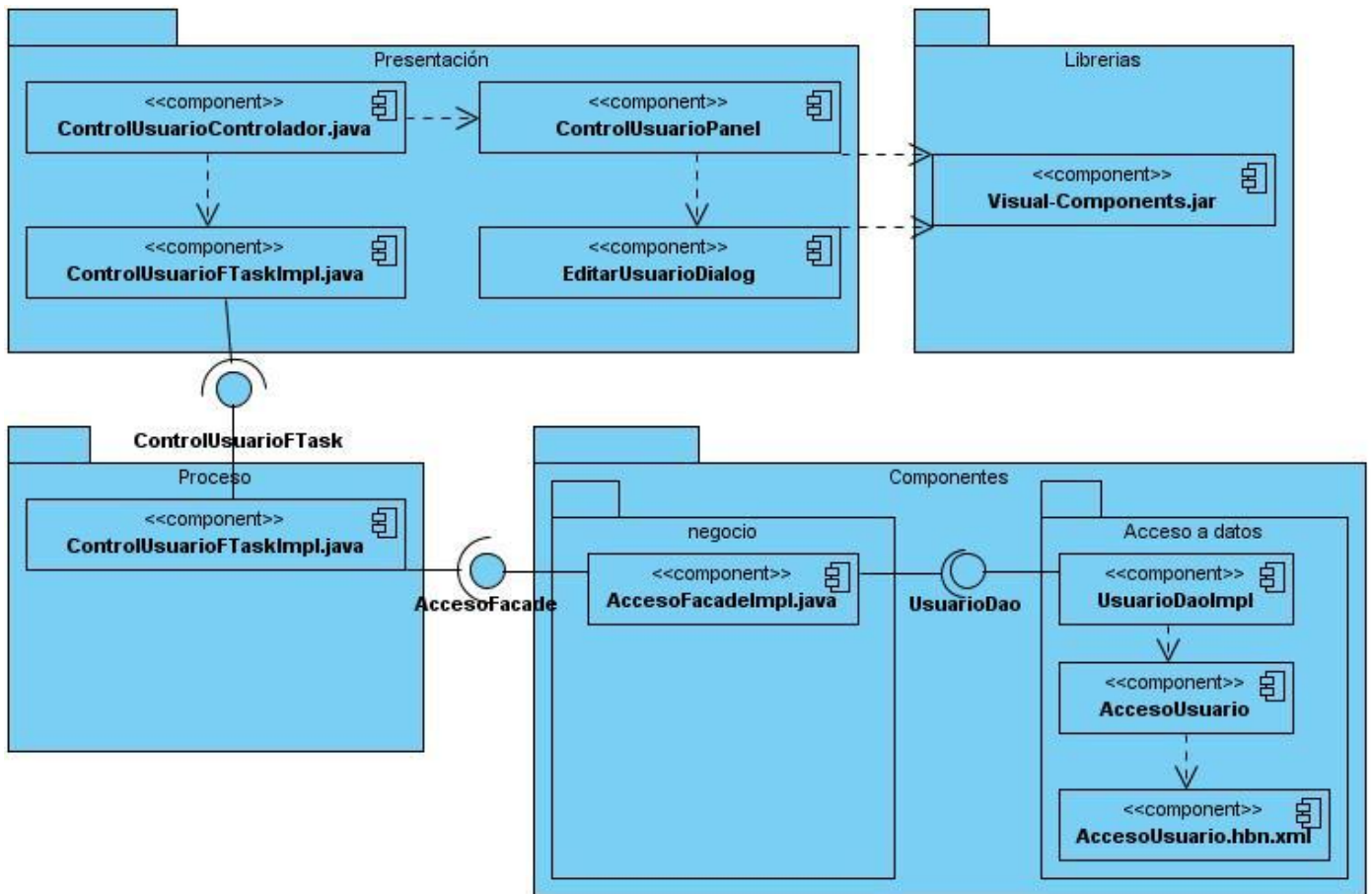


Figura 10 Diagrama de componentes de la funcionalidad Gestionar usuario.

Para ver los diagramas de componentes de las restantes funcionalidades consultar el Anexo 5.

## 3.4 Modelo de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. (26).

# Capítulo 3 Implementación y Prueba

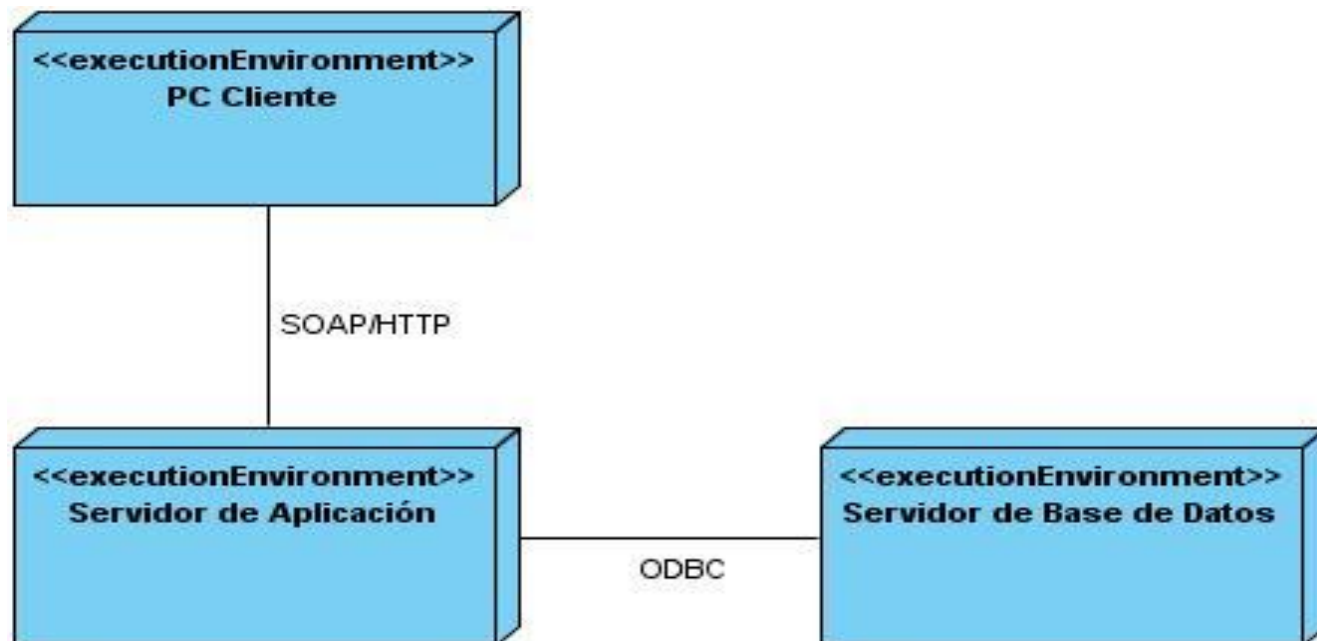


Figura 11 Diagrama de Despliegue.

Como se muestra en la Figura 111 un nodo representa la PC Cliente en la cual estaría desplegado el sistema de emisión para pasaportes y acreditaciones (SEPYA), comunicándose mediante el protocolo SOAP/HTTP<sup>10</sup> al servidor de aplicación donde están publicados los servicios web que satisfacen las peticiones del usuario. En el servidor de aplicación se desarrollan las funcionalidades necesarias para darle respuesta a las solicitudes del usuario, este servidor de aplicación se conecta a través del protocolo ODBC<sup>11</sup> con el servidor de base el cual se encarga del almacenamiento de los datos del Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones.

## 3.5 Pruebas

Para probar la calidad de un software se realizan pruebas que demuestran si las funcionalidades cumplen con las expectativas del cliente. Existen varios tipos de pruebas de software, todas destinadas a encontrar la mayor cantidad de defectos para ofrecer un producto de óptima calidad, las pruebas detectan la

<sup>10</sup> Protocolo simple de acceso a objetos (este protocolo es usado para la construcción de servicios web).

<sup>11</sup> Acrónimo de *Open Data Base Connectivity* (protocolo para la conexión con la base de datos).

# Capítulo 3 Implementación y Prueba

existencia de defectos. Los tipos de prueba que se utilizarán en la propuesta de solución son: las pruebas Unitarias y pruebas de Caja Negra.

## 3.5.1 Pruebas unitarias

Las pruebas unitarias se proponen asegurar que cada una de las funcionalidades implementadas funcionen de modo correcto por separado, para luego poder realizar pruebas a toda la aplicación integrada. Para la realización de las pruebas unitarias se utilizó la herramienta JUnit. Estas pruebas se desarrollaron teniendo en cuenta la planificación prioridad de cada uno de los métodos de acuerdo a la significación que tenga la funcionalidad que implementa. Las pruebas fueron realizadas luego de haber terminado la implementación de las funcionalidades en cada una de las iteraciones definidas en epígrafes anteriores, de este modo verificando el correcto funcionamiento de las mismas de forma independiente.

Para verificar cada clase se creó un caso de prueba con los métodos correspondientes:

Clase AdministrationTest:

A través de la clase AdministrationTest se verifica que proceso de gestión de usuarios sea realizado de una forma correcta. Se cuenta dentro de esta clase con cuatro métodos fundamentales: test\_adicionarUsuario(), test\_modificarUsuario(), test\_eliminarUsuario(), test\_obtenerUsuariosPorCriterio(), el primero es el encargado de validar los datos e insertar un usuario en la base de datos, el segundo se encarga de validar y modificar los datos del usuario insertado en la base de datos, el tercero se encarga de eliminar el usuario al que pertenece el identificador que se le pasa como parámetro y por último el cuarto que es encargado de devolver el usuario según el criterio recibido como parámetro.

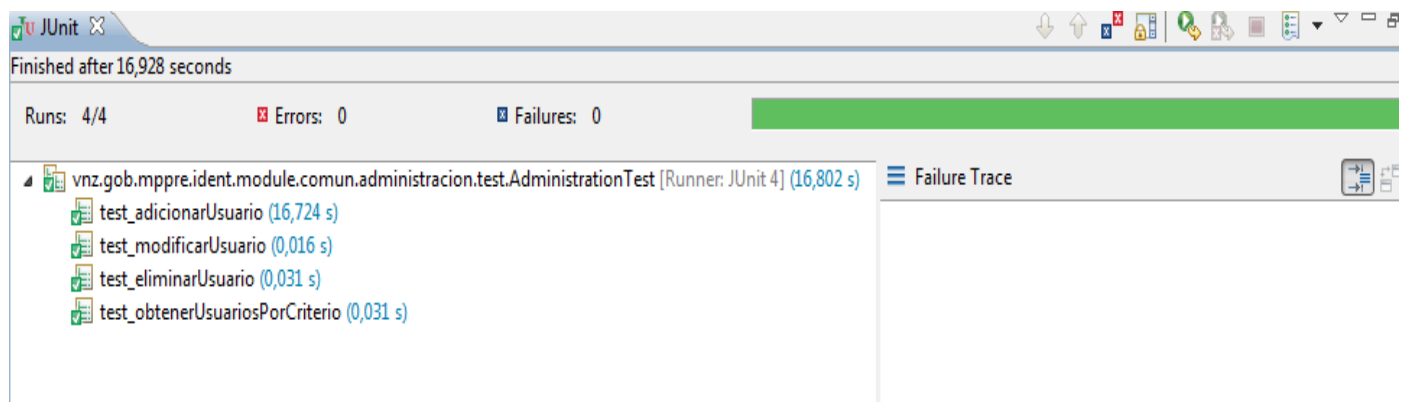


Figura 12 Ejecución de la clase AdministrationTest.

# Capítulo 3 Implementación y Prueba

En la Figura 12 se puede apreciar cada uno de los test ejecutados para la realización de la prueba. La pestaña verde situada delante de cada uno de los métodos indica que fueron realizados de forma satisfactoria, en caso de que estuviera en rojo indicara que ha existido algún error. La barra de progreso de color verde indica que el proceso ha concluido completamente de una forma exitosa.

## 3.5.2 Pruebas de Caja Negra

Las pruebas de caja negra se refieren a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (28)

Los casos de prueba de la caja negra pretenden demostrar que:

- ✓ Las funciones del software son operativas.
- ✓ La entrada se acepta de forma adecuada.
- ✓ Se produce una salida correcta y la integridad de la información externa se mantiene.

Estas intentan encontrar errores de los siguientes tipos:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a bases de datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y de terminación.

Casos de pruebas para las funcionalidades pertenecientes a la Segunda Iteración de desarrollo del módulo de administración del Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones de la República Bolivariana de Venezuela.

- ✓ Gestión de usuario.
- ✓ Gestión de roles.
- ✓ Gestión de planificación semanal de citas.
- ✓ Definición de otorgamiento de acreditación.
- ✓ Definición de convenio de supresión de visa.

# Capítulo 3 Implementación y Prueba

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Primer nombre	Campo de Texto	No	Debe introducir un texto.
[2]	Segundo Nombre	Campo de Texto	Si	Debe introducir un texto.
[3]	Primer apellido	Campo de selección	No	Debe seleccionar un valor
[4]	Segundo apellido	Campo de Texto	No	Debe introducir un texto.
[5]	Correo electrónico	Campo de Texto	No	Debe introducir un texto.
[6]	Nombre de usuario	Campo de texto	No	Debe introducir un texto.
[7]	Número de cédula	Campo de Texto	No	Debe introducir un texto.
[8]	Rol	Campo de selección	No	Debe seleccionar una opción.
[9]	Tipo de solicitante	Campo de selección	No	Debe seleccionar una opción.
[10]	Usuario	Campo de selección	No	Debe seleccionar una opción

Tabla 15 Descripción de variables. Caso de prueba: Gestionar Usuario.



# Capítulo 3 Implementación y Prueba

Flujos	Var 1. Primer nombre	Var 2. Segundo nombre	Var 3. Primer apellido	Var 4. Segundo apellido	Var 5. Correo electrónico	Var 6. Nombre de usuario	Var 7. Numero de cédula	Var 8. Rol	Var 9. Tipo de solicitante	Var 10. usuario	Respuesta del Sistema	Resultado de la Prueba	Observaciones
Crear usuario.	V	V	V	V	V	V	V	V	V	NA	Crea un nuevo usuario con los datos correspondientes a los introducidos.	Satisfactorio	
	I	NA	NA	NA	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Primer nombre no válido".	Satisfactorio	
	NA	I	NA	NA	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Segundo nombre no válido".	Satisfactorio	
	NA	NA	I	NA	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Primer apellido no válido".	Satisfactorio	

# Capítulo 3 Implementación y Prueba

Flujos	Var 1. Primer nombre	Var 2. Segundo nombre	Var 3 .Primer apellido	Var 4. Segundo apellido	Var 5. Correo electrónico	Var 6.Nombre de usuario	Var 7. Numero de cédula	Var 8. Rol	Var 9. Tipo de solicitante	Var 10. usuario	Respuesta del Sistema	Resultado de la Prueba	Observaciones
	NA	NA	NA	I	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Segundo apellido no válido"	Satisfactorio	
	NA	NA	NA	NA	I	NA	NA	NA	NA	NA	Muestra el mensaje: "La dirección de correo electrónico no es válida".	Satisfactorio	
	NA	NA	NA	NA	NA	I	NA	NA	NA	NA	Muestra el mensaje: "El nombre de usuario no es válido".	Satisfactorio	
	NA	NA	NA	NA	NA	NA	I	NA	NA	NA	Muestra el mensaje: "El número de cédula no es válido".	Satisfactorio	
	NA	NA	NA	NA	NA	NA	NA	I	NA	NA	Muestra el mensaje: "Debe seleccionar un rol".	Satisfactorio	

# Capítulo 3 Implementación y Prueba

Flujos	Var 1. Primer nombre	Var 2. Segundo nombre	Var 3 .Primer apellido	Var 4. Segundo apellido	Var 5. Correo electrónico	Var 6.Nombre de usuario	Var 7. Numero de cédula	Var 8. Rol	Var 9.Tipo de solicitante	Var 10. usuario	Respuesta del Sistema	Resultado de la Prueba	Observaciones
	NA	NA	NA	NA	NA	NA	NA	NA	I	NA	Muestra el mensaje: "Debe seleccionar un Tipo de solicitante".	Satisfactorio	
Modificar Usuario.	V	V	V	V	V	NA	V	V	V	NA	Modifica los datos del usuario satisfactoriamente	Satisfactorio	
	I	NA	NA	NA	NA	NA	NA	NA	I	NA	Muestra el mensaje: "Primer nombre no válido".	Satisfactorio	
	NA	I	NA	NA	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Segundo nombre no válido".		
	NA	NA	I	NA	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Primer apellido no es válido".	Satisfactorio	
	NA	NA	NA	I	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Segundo apellido no es válido".	Satisfactorio	

# Capítulo 3 Implementación y Prueba

Flujos	Var 1. Primer nombre	Var 2. Segundo nombre	Var 3 .Primer apellido	Var 4. Segundo apellido	Var 5. Correo electrónico	Var 6.Nombre de usuario	Var 7. Numero de cédula	Var 8. Rol	Var 9. Tipo de solicitante	Var 10. usuario	Respuesta del Sistema	Resultado de la Prueba	Observaciones
	NA	NA	NA	NA	I	NA	NA	NA	NA	NA	Muestra el mensaje: "el correo electrónico no es válido".	Satisfactorio	
	NA	NA	NA	NA	NA	NA	I	NA	NA	NA	Muestra el mensaje: "Número de cédula no es válido".	Satisfactorio	
	NA	NA	NA	NA	NA	NA	NA	I	NA	NA	Muestra el mensaje: "Debe seleccionar un rol".	Satisfactorio	
	NA	NA	NA	NA	NA	NA	NA	NA	I	NA	Muestra el mensaje: "Debe seleccionar un tipo de solicitante".	Satisfactorio	

# Capítulo 3 Implementación y Prueba

Flujos	Var 1. Primer nombre	Var 2. Segundo nombre	Var 3. Primer apellido	Var 4. Segundo apellido	Var 5. Correo electrónico	Var 6. Nombre de usuario	Var 7. Numero de cédula	Var 8. Rol	Var 9. Tipo de solicitante	Var 10. usuario	Respuesta del Sistema	Resultado de la Prueba	Observaciones
Eliminar usuario	NA	NA	NA	NA	NA	NA	NA	NA	NA	V	El sistema elimina el usuario seleccionado satisfactoriamente.	Satisfactorio	
	NA	NA	NA	NA	NA	NA	NA	NA	NA	I	Muestra el mensaje: "Debe seleccionar el usuario a eliminar".	Satisfactorio	

Tabla 16 Caso de prueba: Gestionar Usuario.

## 3.6 Resultado de las pruebas:

En consecuencia a las pruebas realizadas se expusieron una serie de no conformidades y errores de las cuales fueron tratadas en una serie de iteraciones hasta llegar a un producto con la calidad suficiente para su puesta en explotación. A continuación se muestran las gráficas (Figura 13 y Figura 144) ilustrando el resultado de las pruebas unitarias y pruebas de caja negra por iteraciones así como el estado de las no conformidades.

## Resultado de las pruebas Unitarias

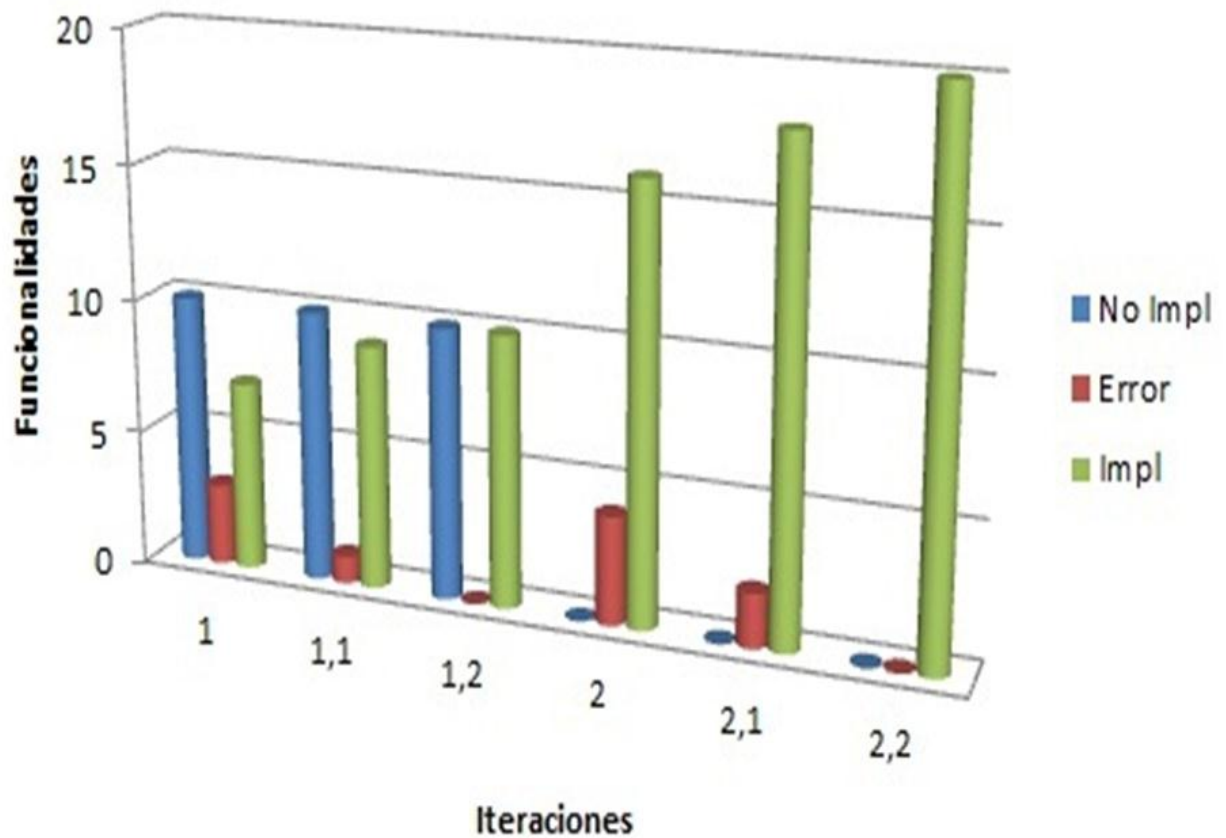


Figura 13 Resultado de las Pruebas Unitarias.

## Resultado de las pruebas de Caja Negra

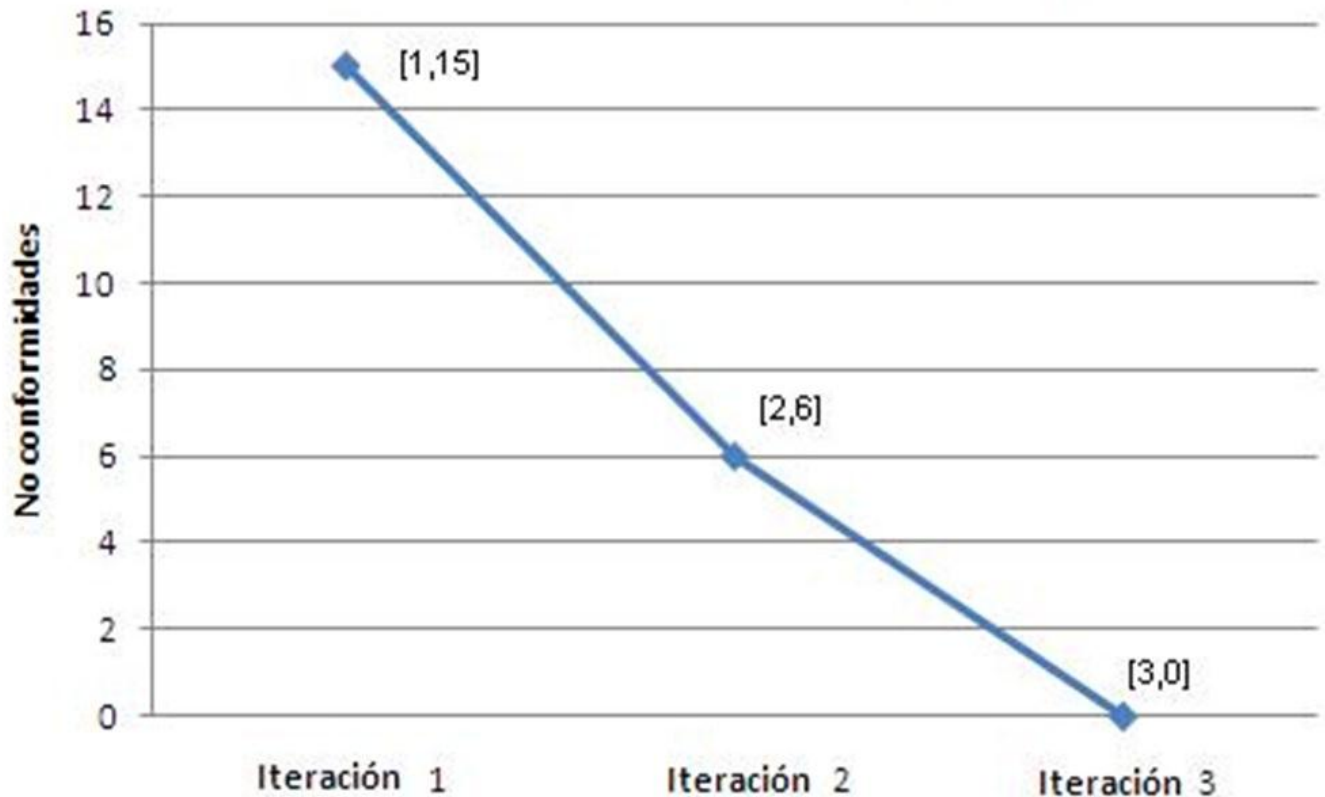


Figura 14 Resultado de las pruebas de Caja Negra.

### 3.7 Conclusiones parciales

A través del diagrama de componentes se mostraron los elementos físicos y las relaciones que existente entre ellos en función del desarrollo de las funcionalidades. Se modeló el diagrama de despliegue donde se evidencia como va a estar desplegado el sistema finalmente.

Al concluir la implementación de las funcionalidades descritas, se le realizaron las pruebas unitarias y de caja negra, obteniéndose en ambas resultados satisfactorios después de una serie de iteraciones,

# Capítulo 3 Implementación y Prueba

comprobando que el sistema posee un correcto funcionamiento y cumple con las funcionalidades definidas por el cliente.



## Conclusiones

En correspondencia con las exigencias del módulo de administración del Sistema de Emisión de Pasaportes Diplomáticos, de Servicio y Acreditaciones y los objetivos trazados en la investigación se concluye:

- ✓ Los sistemas analizados a pesar de brindar soluciones avanzadas no se ajustan a las funcionalidades específicas sistema.
- ✓ Se realizó un estudio de las herramientas, tecnologías y metodología de desarrollo necesarias para el desarrollo de la solución.
- ✓ El desarrollo de la solución permitió la gestión de usuarios, roles y elementos de configuración para el Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones, evidenciándose durante la ejecución exitosa de las pruebas funcionales.

## Recomendaciones

Se recomienda:

- ✓ Desplegar el Sistema de emisión de pasaportes diplomáticos, de servicio y acreditaciones, que comprende las funcionalidades desarrolladas en la propuesta de solución.

## Bibliografía

1. ICAO. 2008. Documentos de viaje de lectura mecánica. 2008. Vol. 1.
2. Reglamento de Pasaportes. Diciembre 1974. Decreto Número 611. Caracas : s.n., Diciembre 1974.
3. GNU. GNU Operating System. . [En línea] 2005. [Citado el: 4 de diciembre de 2010.]  
<http://www.gnu.org/philosophy/free-sw.es.html>
4. Valle, Amaury E. del. Software libre II: Una estrategia decisiva de desarrollo. *Juventud Rebelde*. . 14 de febrero de 2008, pág. pág. Informática
5. Exposito, Ery Delgado. Metodologías de desarrollo de software. [En línea] [Citado el: 20 de noviembre de 2010.] <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml>
6. SGBD. Sistema gestor de base de dato. [En línea] [Citado el: 21 de enero de 2011.]  
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sqbd/>
7. FDD: Feature Driven Development. [En línea] [Citado el: 13 de diciembre de 2010.]  
<http://www.pisis.unalmed.edu.co>
8. Molpeceres, Alberto. Procesos de desarrollo: RUP, XP y FDD. [En línea] [Citado el: 2010 de diciembre de 6.] [http://www.javahispano.org/contenidos/archivo/71/metodos\\_desarrollo.pdf](http://www.javahispano.org/contenidos/archivo/71/metodos_desarrollo.pdf).
9. Calderón, Amaro. Metodologías ágiles. Trujillo : Perú, 2007.
10. Programación, Lenguajes de. Lenguajes de Programación . [En línea] [Citado el: 15 de mayo de 2011.] <http://es.kioskea.net/contents/langages/langages.php3>
11. Adelfa. cad.com.mx. Historia del lenguaje Java. [En línea] 2005. [Citado el: 2010 de diciembre de 18.]  
[http://www.cad.com.mx/historia\\_del\\_lenguaje\\_java.htm](http://www.cad.com.mx/historia_del_lenguaje_java.htm)
12. Gonzalo Álvarez Marañón. Qué es Java? [En línea] 1999. [Citado el: 5 de enero de 2011.]  
<http://www.iec.csic.es/criptonomicon/java/quesjava.html>
13. Modelado de Sistemas con UML. [En línea] [Citado el: 11 de 1 de 2011.]  
<http://www.marquezetelecom.com/LuCAS/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>

14. ProfesorJava. Ambiente de Desarrollo Integrado. [En línea] 15 de abril de 2010. [Citado el: 12 de enero de 2011.] <http://profejavaoramas.blogspot.com/2010/04/ambiente-de-desarrollo-integrado-ide.html>
15. NetBeans. [En línea] [Citado el: 12 de enero de 2011.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html)
16. Visual Paradigm. [En línea] [Citado el: 14 de diciembre de 2010.] <http://www.visual-paradigm.com/product/vpuml/provides/>
17. PostgreSQL. [En línea] [Citado el: 13 de diciembre de 2010.] <http://www.postgresql.org>
18. Framework. [En línea] [Citado el: 15 de diciembre de 2010.] <http://es.wikipedia.org/wiki/Framework>
19. Framework para Java. [En línea] [Citado el: 13 de enero de 2010.] <http://www.santiagoporta.com.ar/site/javaframework.html>
20. Hibernate. [En línea] [Citado el: 30 de noviembre de 2010.] [www.unife.edu.pe/ing/desarrollo.doc](http://www.unife.edu.pe/ing/desarrollo.doc).
21. Convención de Viena sobre Relaciones Diplomáticas. [En línea] [Citado el: 14 de mayo de 2011] <http://www.oas.org/legal/spanish/documentos/convencionviena.htm>
22. MPPRE. 2011. Ministerio del Poder Popular para Relaciones Exteriores. [En línea] 2011. [Citado el: 23 de mayo de 2011.] <http://www.mppre.gob.ve/>.
23. Tutorial de ANT. [En línea] [Citado el: 16 de mayo de 2010.] [http://www.marioalberto.com.mx/articulos/java\\_ant\\_basico.php](http://www.marioalberto.com.mx/articulos/java_ant_basico.php)
24. Desarrollo Web. [En línea] [Citado el: 18 de mayo de 2010.] <http://www.desarrolloweb.com/articulos/1622.php>
25. Ingeniero de Software. [En línea] [Citado el: 19 de mayo de 2010.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
26. RUMBAUGH, J. y BOOCH, I. J. G. El Lenguaje Unificado de Modelado. Manual de Referencia.
27. Java. [En línea] [Citado el: 30 de mayo de 2010.] <http://java.sun.com/docs/codeconv/>
28. Ingeniería de Software. Flujo de trabajo de pruebas. UCI, 2009.
29. Rational RequisitePro. Presentación de Power Point. Clase práctica 3. Ingeniería de Software I. Universidad de las Ciencias Informáticas.
30. CRUD. [En línea] [Citado el: 30 de mayo de 2010.] <http://www.grancomo.com/2006/05/02/patrones-para-el-diseno-de-interfaz/>

# Bibliografía

31. Acreditación. [En línea] [Citado el: 10 de junio de 2010.]  
<http://www.wordreference.com/definicion/acreditaci%C3%B3n>
32. Sánchez, Miguel, Jiménez, Beatriz y Gutiérrez, Francisco L. [En línea] [Citado el: 20 de junio de 2011.]  
<http://www.sistedes.es/sistedes/pdf/2007/SCHA-07-Sanchez-Acceso.pdf>.
33. Shandu Ravi, Ferraiolo David, Kuhn Richard [Citado el: 20 de junio de 2011.] The NIST Model for Role-Based Acces Control.

## Glosario de términos

**Rol:** Es un concepto relacionado con la autorización de los usuarios u operadores del sistema para utilizar las funcionalidades que este brinda, y a través de la administración se le asigna o revoca un permiso en dependencia del perfil de cargo que ocupa.

**MPPRE:** Ministerio del Poder Popular para Relaciones Exteriores de la República Bolivariana de Venezuela.

**Usuario:** Define un concepto administrativo para garantizar seguridad en el sistema, a través de él un funcionario podrá autenticarse y dependiendo de sus roles, podrá efectuar determinadas operaciones en cualquier puesto de trabajo de la oficina, posee un estado de activación y los datos de la persona vinculada.

**JAR (Java Archive):** Tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java.

**Marco de trabajo:** Estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

**HTTP: (Protocolo de Transferencia de Hipertexto):** Protocolo usado en las transacciones de la Web. Define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

**FTP: (*File Transfer Protocol*):** Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

**UML: (*Unified Modeling Language*)** El lenguaje unificado de modelado es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

**IDE: (*Integrated Development Environment*)** Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un

compilador, un depurador y un constructor de interfaz gráfica GUI.

Plug-in: Un Plug-in es un módulo de hardware o software que añade una característica o un servicio específico a un sistema. La idea es que el nuevo componente se agrega simplemente al sistema existente.

XML: (*Extensible Markup Language*): Metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium (W3C)*. Es una manera de definir lenguajes para diferentes necesidades. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya

AXIS2: contenedor de servicios web. Este tipo de soluciones sustituye a las implementaciones nativas de los diferentes contenedores y permitía mejorar la portabilidad de los servicios web.

SOAP: (*Simple Object Access Protocol*): Protocolo de Acceso Simple a Objetos es un protocolo estándar define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

SMTP: Protocolo de comunicaciones que utiliza un servidor de correo para enviar los mensajes hacia otro servidor de correo. Es el protocolo que se utiliza entre el cliente de correo del emisor y su servidor de correo.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), se utiliza a nivel mundial para conectarse a Internet y a los servidores web.

JMS: (*Java Message Service*): Estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma Java2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona.

SGBD: Sistema gestor de base de datos.

ORDBMS: (*objeto-relacional sistema de gestión de base de datos*) es un sistema de gestión de base de datos (DBMS), similar a una base de datos relacional, pero con una orientación a objetos modelo de base de datos: objetos, clases y herencia están directamente apoyado en esquemas de bases de datos y en el

# losario de términos

lenguaje de consulta . Además, se admite la extensión del modelo de datos con la costumbre de tipos de datos y métodos.

BSD: (*Berkeley Software Distribution*) traducido al español, Distribución de Software Berkeley, es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

ORM: (*Object Relational Mapping*) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

JBoss: Servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte.

JAVAE: (*Java Platform, Enterprise Edition*) es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE.



## Anexos

### Anexo 1 Descripción de las Funcionalidades

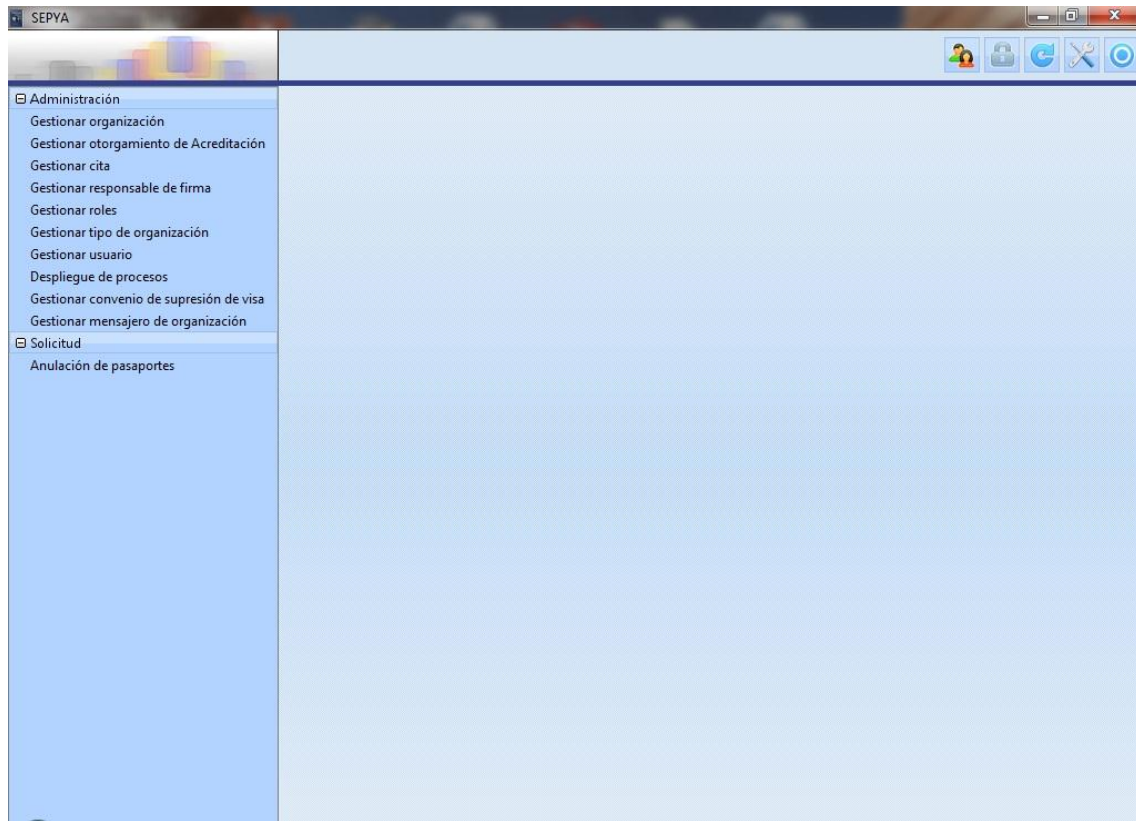


Figura 15 Interfaz Principal módulo de administración.

Tabla 3. Gestionar usuario

Precondiciones	El usuario debe estar autenticado en la aplicación de escritorio con el rol de Administrador.
Funcionalidades asociadas	RF5, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, RF4.
Conceptos tratados	Usuario.  Rol.
Descripción básica	1. El sistema muestra la Interfaz Principal (1) con un vínculo al módulo Administración que contiene las

siguientes opciones:

- Gestionar organización
- Gestionar otorgamiento de Acreditación
- Gestionar Cita
- Gestionar Responsable de firma
- Gestionar roles
- Gestionar tipo de organización
- Gestionar usuario
- Despliegue de proceso
- Gestionar convenio de supresión de visa
- Gestionar mensajero de organización

2. El usuario selecciona la opción Gestionar usuario.

3. El sistema muestra la interfaz Gestión de usuario con un listado de los usuarios existentes en el sistema que contiene los siguientes datos y opciones:

Datos:

- Usuario
- Nombresy apellidos

Opciones:

- Filtrar
- Agregar
- Modificar
- Consultar
- Activo
- Eliminar

- Terminar

3.1 Si el Administrador introduce los nombres y apellidos de la persona, el sistema muestra en el listado de usuarios los que se corresponden con los datos introducidos en el filtro.

3.2 Si el Administrador selecciona la opción Agregar, el sistema muestra la interfaz Crear usuario con los siguientes datos a llenar y las opciones.

Datos:

- Primer nombre
- Segundo nombre
- Primer apellido
- Segundo apellido
- Venezolano
- Extranjero
- Correo electrónico
- Nombre de usuario

Si el usuario es venezolano:

- N° de Cédula

Si el usuario es extranjero:

- Identificador Nacional
- N° de Cédula
- Nacionalidad

Si el usuario tiene el rol solicitante:

Para el Administrador Local el sistema muestra por defecto el tipo de solicitante titular.

- Tipo de solicitante (secretaria, funcionarios,

titular, coordinador)

- Nombre de organización

Listado de organizaciones:

- Nombre de organización
- Tipo de organización

Organización asociada:

- Nombre de organización
- Tipo de organización

Opciones:

- Crear cuenta
- Cancelar
- Agregar
- Filtrar
- Eliminar

3.2.1 Si el usuario introduce el nombre de organización, el sistema filtra los datos y los muestra en el listado de organizaciones.

3.2.2 Si el usuario selecciona la opción Agregar, el sistema actualiza los datos en el listado de organización asociada.

3.2.3 Si el usuario selecciona la opción Eliminar, el sistema elimina la organización del listado de organización asociada.

3.2.4 El usuario introduce los datos y selecciona la opción Crear cuenta. (En caso de seleccionar otra

opción consultar: *Descripción alterna 4 “Cancelar acción”*).

3.2.5 El sistema registra los datos del usuario enviando vía email una notificación que contiene, nombre de usuario y contraseña generada por el sistema, mostrando el siguiente mensaje: “Datos introducidos correctamente” y redirecciona a la interfaz Gestión de usuario si los datos son correctos.

3.3 Si el usuario selecciona la opción Modificar, el sistema muestra los datos editables excepto el nombre de usuario de la persona, en la interfaz Modificar datos de usuario y las opciones Modificar y Cancelar.

3.3.1 El usuario introduce los nuevos datos y presiona Modificar. (En caso de seleccionar otra opción consultar: *Descripción alterna 4 “Cancelar acción”*).

3.3.2 El sistema actualiza los datos del usuario y muestra la interfaz Gestión de usuario.

3.4 Si el usuario selecciona la opción Consultar, el

sistema muestra los datos existentes en la misma interfaz y la opción Cerrar.

3.4.1 El usuario presiona Cerrar.

3.4.2 El sistema muestra la interfaz Gestión de usuario.

3.5 Si el usuario marca la opción Activo, el sistema activa el usuario.

3.6 Si el usuario desmarca la opción Activo, el sistema muestra el siguiente mensaje: “¿Está seguro que desea desactivar el usuario?” con las opciones Aceptar y Cancelar.

3.6.1 El usuario selecciona Aceptar. (En caso de seleccionar otra opción consultar: *Descripción alterna 4 “Cancelar acción”*).

3.6.2 El sistema desactiva el usuario y lo marca en el listado con un color diferente.

3.7 Si el usuario selecciona la opción Eliminar, el sistema muestra el siguiente mensaje de confirmación: “¿Desea eliminar el usuario?” y las opciones Sí o No.

3.7.1 El usuario selecciona Sí. (En caso contrario

consultar: *Descripción alterna 6 “Eliminación cancelada”*).

3.7.2 El sistema elimina el usuario actualizando el listado de usuarios en la interfaz gestión de usuarios.

3.8 Si el usuario selecciona la opción Terminar, el sistema redirecciona a la interfaz Principal.

## Prototipos

The screenshot displays the 'Gestionar usuario' window. On the left is a sidebar with a tree view containing categories like 'Administración' and 'Solicitud'. The main area features a search bar labeled 'Filtrar por Nombres:' and buttons for 'Consultar', 'Agregar', 'Modificar', and 'Eliminar'. Below these is a table with the following data:

#	Usuario	Nombres y apellidos	Activo
1	solicitante	Aarla Beach Blob	<input type="checkbox"/>
2	secretariapsp	Raul Redondo Garcia	<input checked="" type="checkbox"/>
3	secretariacred	Oithium Vasder Meier	<input checked="" type="checkbox"/>
4	coordinadorpsp	Augusto Prendergas Salime	<input checked="" type="checkbox"/>
5	coordinadoracred	Viktor Krum Bolefa	<input checked="" type="checkbox"/>
6	director	Vergara Jusuli Careganton	<input checked="" type="checkbox"/>
7	repcionista	Carlisle Cullen Segundo	<input checked="" type="checkbox"/>
8	captador	Ever Banega Morales	<input checked="" type="checkbox"/>
9	supervisor	Javier Pastore Garcia	<input checked="" type="checkbox"/>
10	operario	Oleg Rasgull Makarov	<input checked="" type="checkbox"/>
11	controlador	Lionel Andres Messi Cuccitini	<input checked="" type="checkbox"/>
12	admin	Florentino DonDinero Mandamas Perez	<input checked="" type="checkbox"/>
13	juancito	Juan Hrtado Gomez	<input checked="" type="checkbox"/>
14	supervisorpsp	Raul Redondo Garcia	<input checked="" type="checkbox"/>

At the bottom right of the window, there is a green button labeled 'Terminar'.

Figura 16 Interfaz principal gestión de usuarios.

**Listado de organizaciones**

#	Nombre de la organización	Tipo de organización
1	MIC	Empresa Informatica
2	UCI	UCI

**Organización asociada**

Nombre de la organización	Tipo de organización
---------------------------	----------------------

Primer nombre:

Segundo nombre:

Primer apellido:

Segundo apellido:

Venezolano  Extranjero

Identificador nacional:

Correo electrónico:

Nº de cedula:

Nombre de usuario:

Nacionalidad:

Rol:

Crear cuenta Cancelar

Terminar

Figura 17 Interfaz agregar usuario.



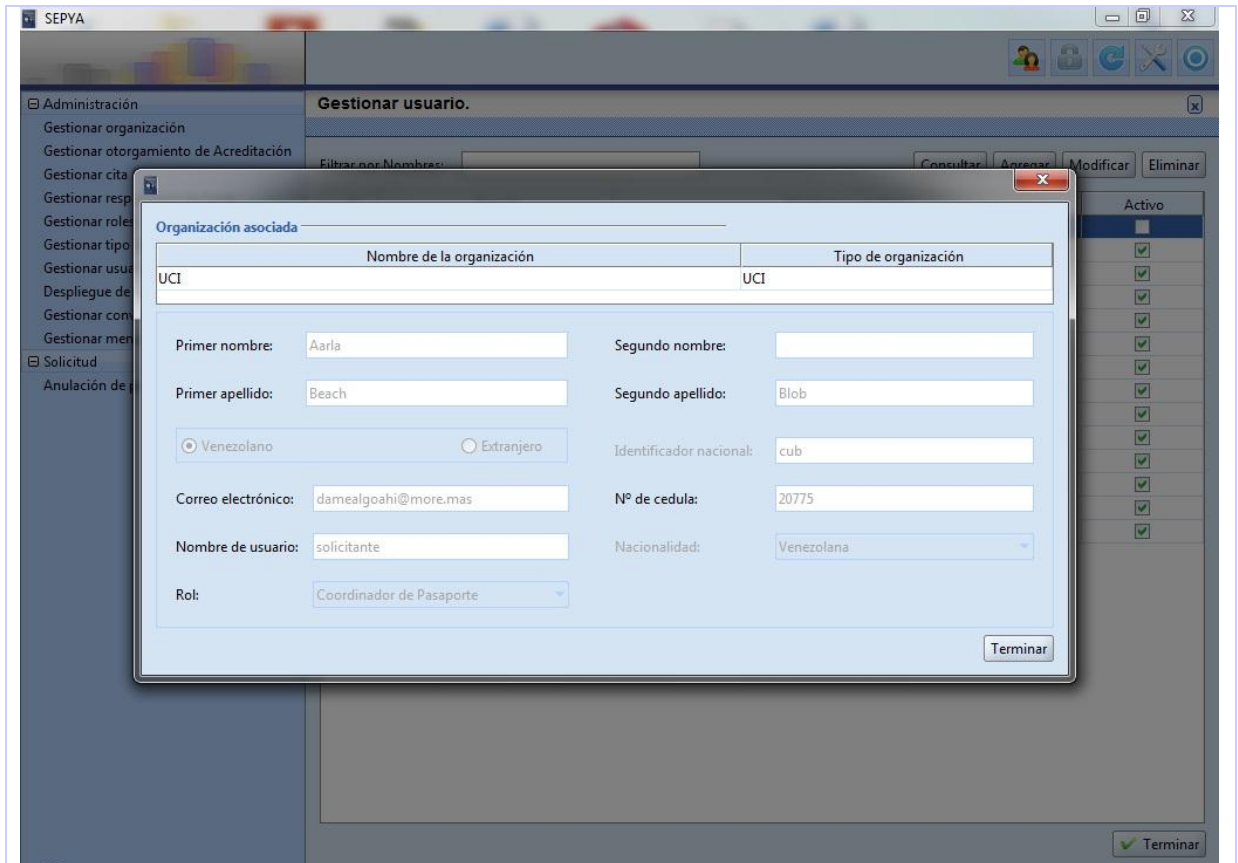


Figura 18 Interfaz consultar usuario.

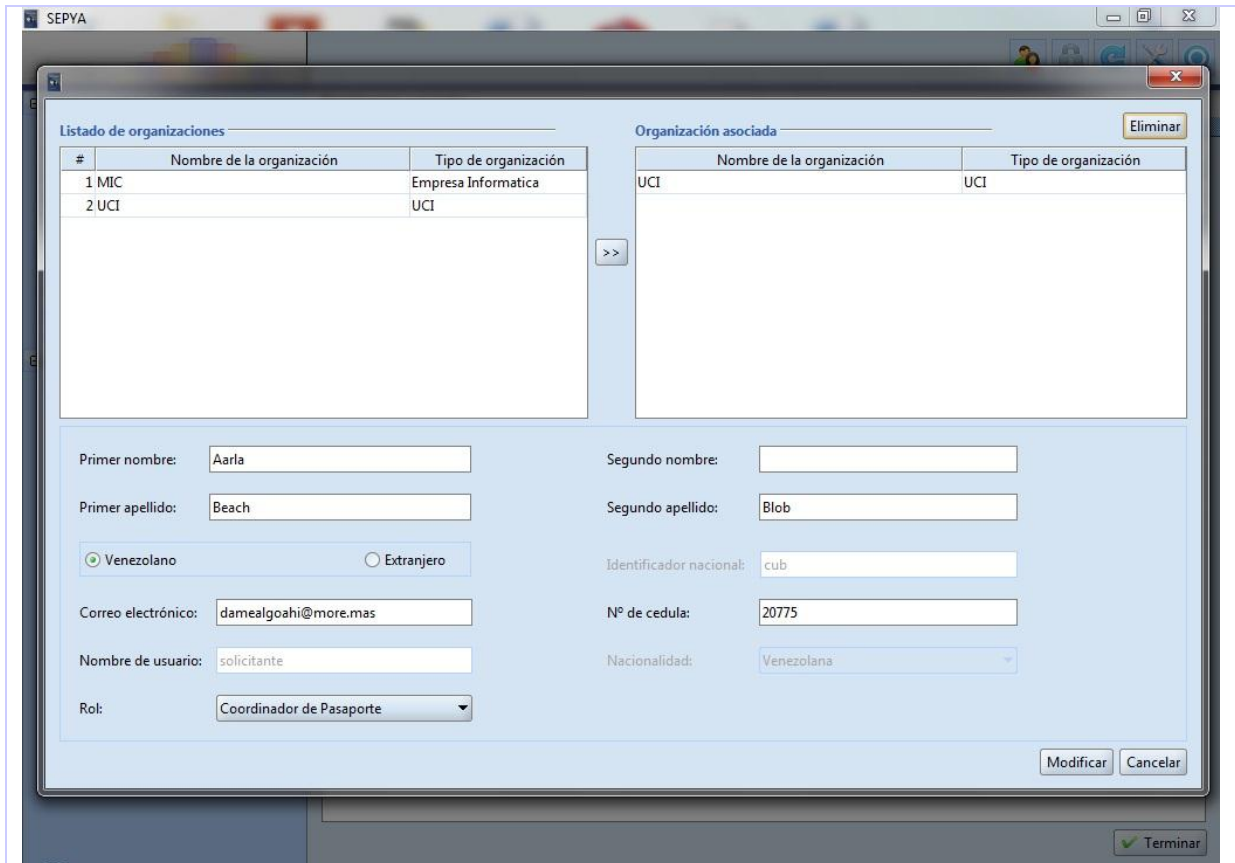


Figura 19 Interfaz modificar usuario.

### Descripción alterna

#### Descripción alterna 4 "Cancelar acción"

- 4. El usuario selecciona la opción Cancelar.
- 4.1 El sistema cancela la acción del usuario y redirecciona a la interfaz Gestión de usuario.

#### Descripción alterna 5 "Datos incorrectos"

	<p>5. El sistema muestra el siguiente mensaje: “Ha introducido los datos incorrectamente” mostrando la interfaz Crear usuario.</p>
<p>Descripción alterna 6 “Eliminación cancelada”</p>	
	<p>6. El usuario selecciona No.</p> <p>6.1 El sistema no realiza ninguna acción de eliminación sobre el usuario.</p>
<p>Prototipos</p> <p>-</p>	
<p>Validaciones</p>	<p>Quando se introducen datos en el sistema debe verificarse que esté completa toda la información.</p> <p>Al registrar un usuario, el mismo no debe existir en la base de datos.</p> <p>Solo se desactiva un usuario siempre que no sea funcional o que se suspenda por un periodo de tiempo el acceso para hacer solicitudes de documentos.</p> <p>Al modificar los datos de un usuario, su nombre de usuario no se puede modificar.</p>
<p>Postcondiciones</p>	<p>Quedan registrados todos los usuarios que pueden acceder al sistema, la modificación y la consulta de sus datos en caso de ser necesaria, así como la</p>

	activación y desactivación de los mismos.
Requisito no funcional	-
Servicios	-
Componente	-

Precondiciones	El usuario debe estar autenticado en la aplicación de escritorio con el rol de Administrador.
Funcionalidades asociadas	
Conceptos tratados	Usuario. Rol.
Descripción básica	<p>1. El sistema muestra la interfaz Principal con un vínculo al módulo Administración que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar organización</li> <li>• Gestionar otorgamiento de Acreditación</li> <li>• Gestionar Cita</li> <li>• Gestionar Responsable de firma</li> <li>• Gestionar roles</li> <li>• Gestionar tipo de organización</li> <li>• Gestionar usuario</li> <li>• Despliegue de proceso</li> <li>• Gestionar convenio de supresión de visa</li> <li>• Gestionar mensajero de organización</li> </ul>

2. El Administrador selecciona la opción Gestionar roles.

3. El sistema muestra la interfaz Gestión de roles con un listado de los roles existentes en el sistema que contiene los siguientes datos y opciones:

Datos:

- Nombre del rol

Opciones:

- Agregar
- Modificar
- Consultar
- Activo
- Eliminar
- Terminar

3.1 Si el Administrador selecciona la opción

Agregar, el sistema muestra la interfaz

Crear rol con los siguientes datos y

opciones:

Datos:

- Nombre del rol
- Descripción
- Listado de funcionalidades por módulos

Opciones:

- Aceptar
- Cancelar

3.1.1 El Administrador introduce los datos en el sistema y presiona Aceptar. (En caso de seleccionar otra opción consultar: *Descripción alterna 4 “Cancelar acción”*).

3.1.2 El sistema registra el rol asignando las funcionalidades por módulos y redirecciona a la interfaz Gestión de roles si los datos son correctos. (En caso contrario consultar: *Descripción alterna 5 “Datos incorrectos”*).

3.2 Si el Administrador selecciona la opción Modificar, el sistema muestra la interfaz Modificar rol con los datos editables y las opciones:

- Modificar
- Cancelar

3.2.1 El Administrador introduce los nuevos datos y presiona Modificar. (En caso de seleccionar otra opción consultar: *Descripción alterna 4 “Cancelar acción”*).

3.2.2 El sistema actualiza los datos del rol y muestra la interfaz Gestión de roles.

3.3 Si el Administrador selecciona la opción

Consultar el sistema muestra los datos existentes del rol en la interfaz Consultar rol y la opción Terminar.

3.3.1 El Administrador del sistema presiona Terminar.

3.3.2 El sistema muestra la interfaz Gestión de roles.

3.4 Si el Administrador marca la opción Activo el sistema activa el rol.

3.5 Si el Administrador desmarca la opción Activo el sistema muestra el siguiente mensaje: "Está a punto de desactivar el rol ¿está seguro que desea hacerlo?" con las opciones Aceptar y Cancelar.

3.5.1 El Administrador selecciona Aceptar.

(En caso de seleccionar otra opción consultar: *Descripción alterna 4 "Cancelar acción"*).

3.5.2 El sistema desactiva el rol.

3.6 Si el Administrador selecciona la opción Eliminar el sistema muestra el siguiente mensaje de confirmación: "¿Desea eliminar el

rol?” y las opciones Sí o No.

3.6.1 El Administrador selecciona Sí. (En caso contrario consultar: *Descripción alterna 6 “Denegar mensaje”*).

3.6.2 El sistema elimina el rol, y actualiza el listado de roles en la interfaz Gestión de roles.

3.7 Si el Administrador selecciona la opción Terminar, el sistema redirecciona a la interfaz Principal.

Prototipos



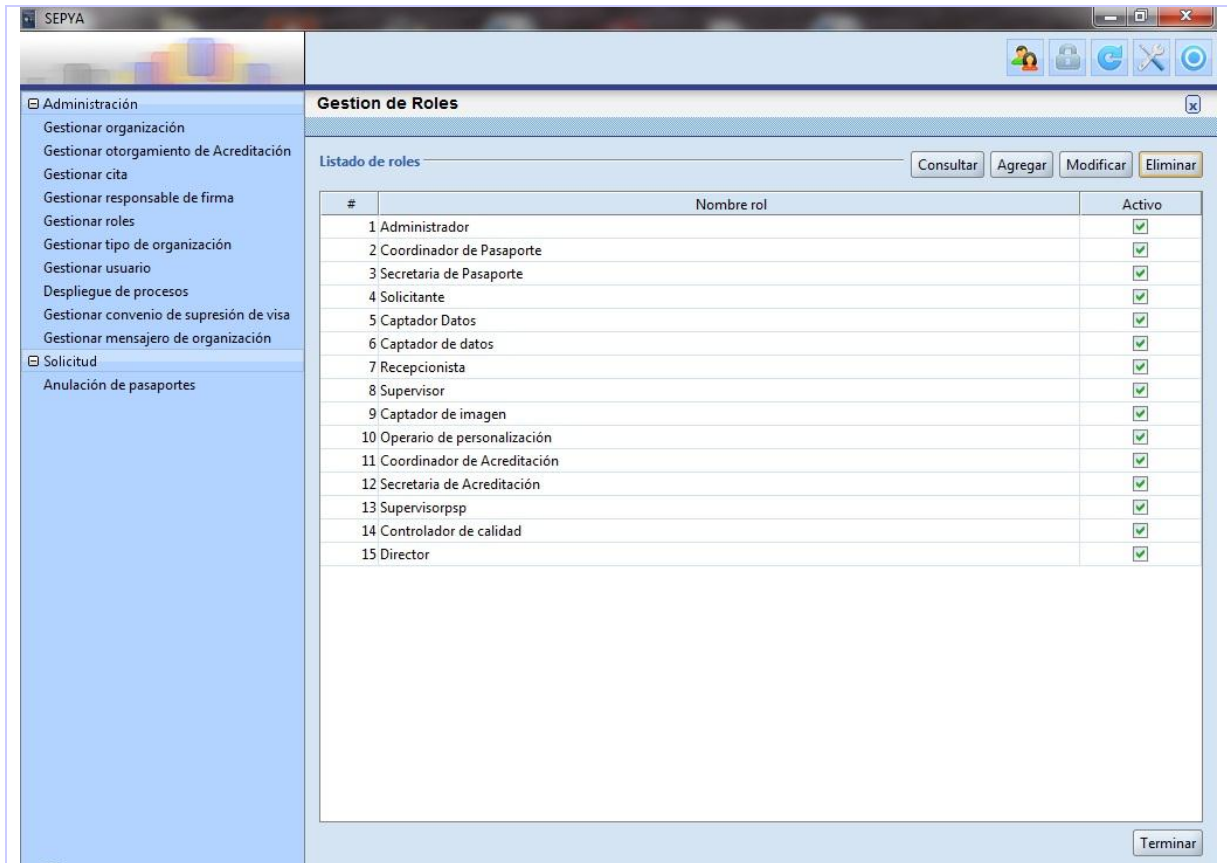


Figura 20 Interfaz principal gestionar roles.

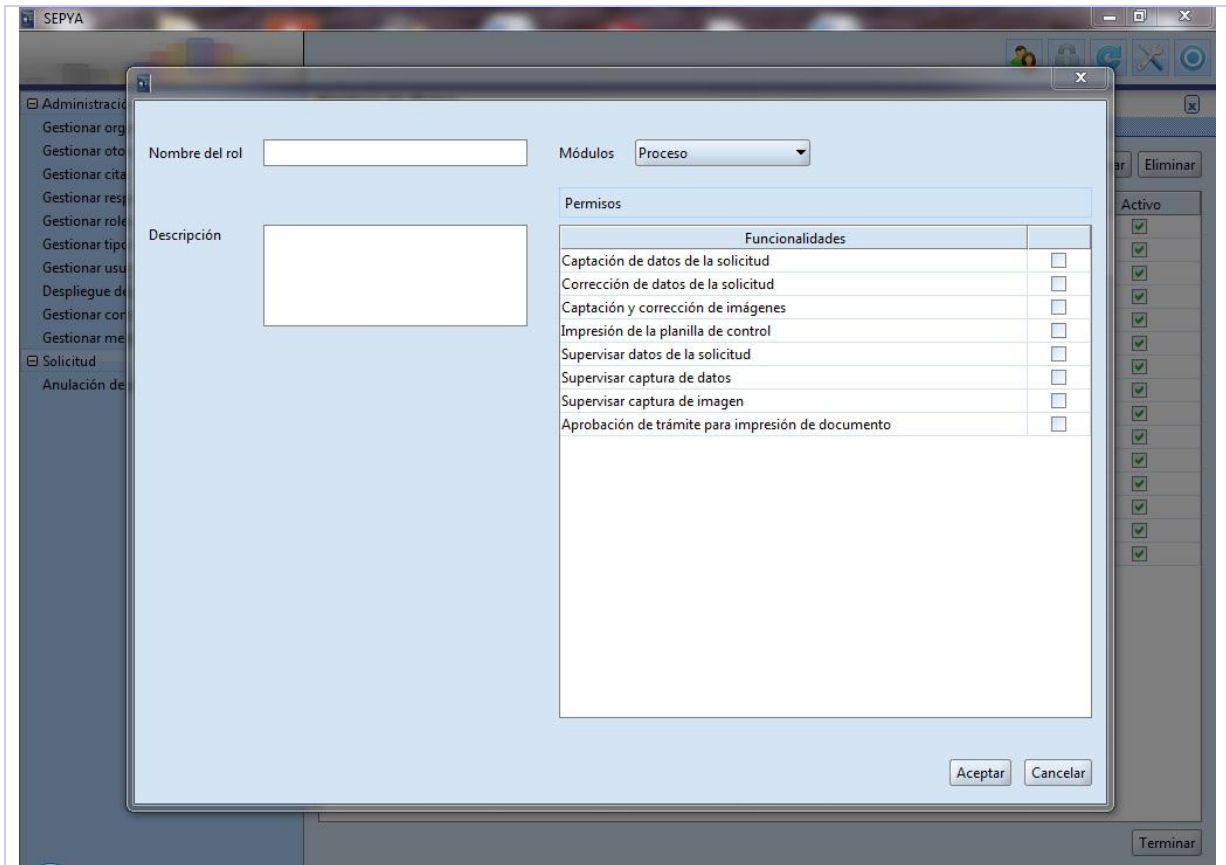


Figura 21 Interfaz agregar roles.

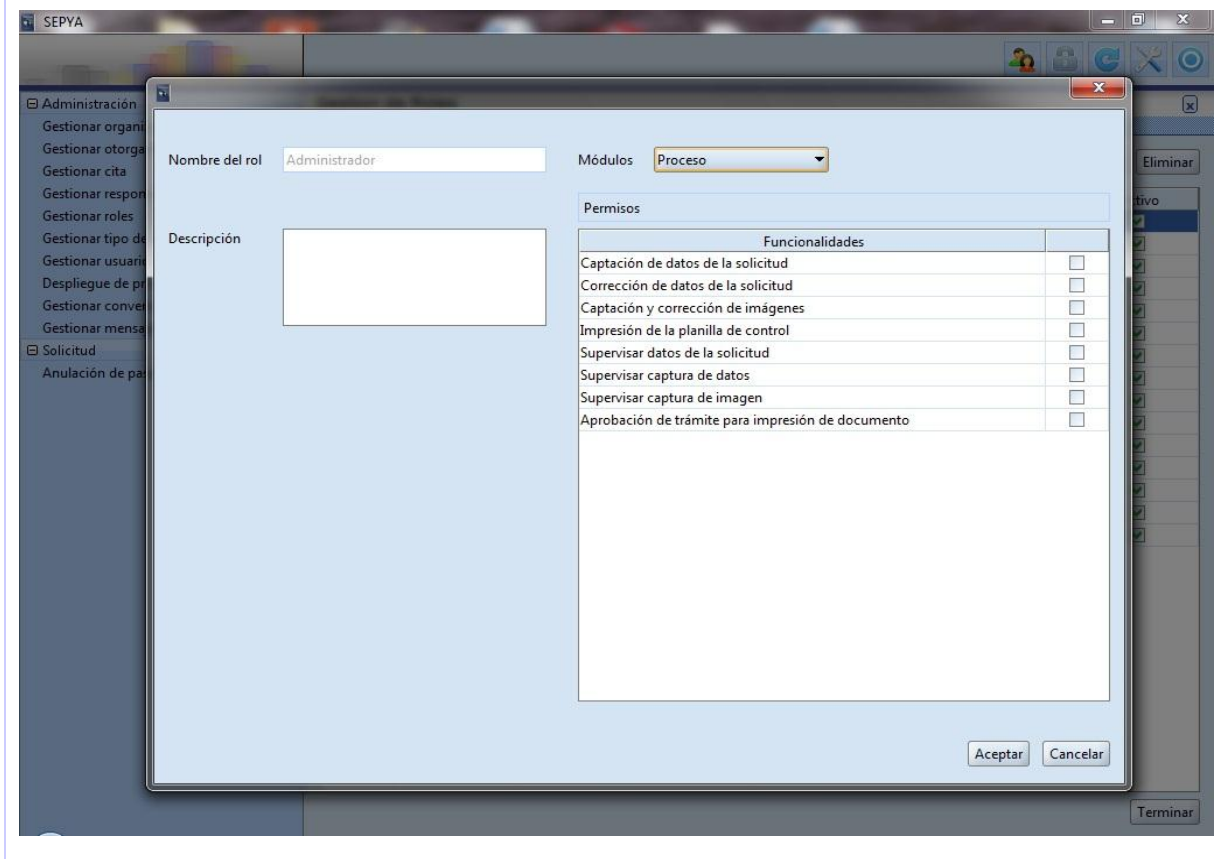


Figura 22 Interfaz modificar roles.

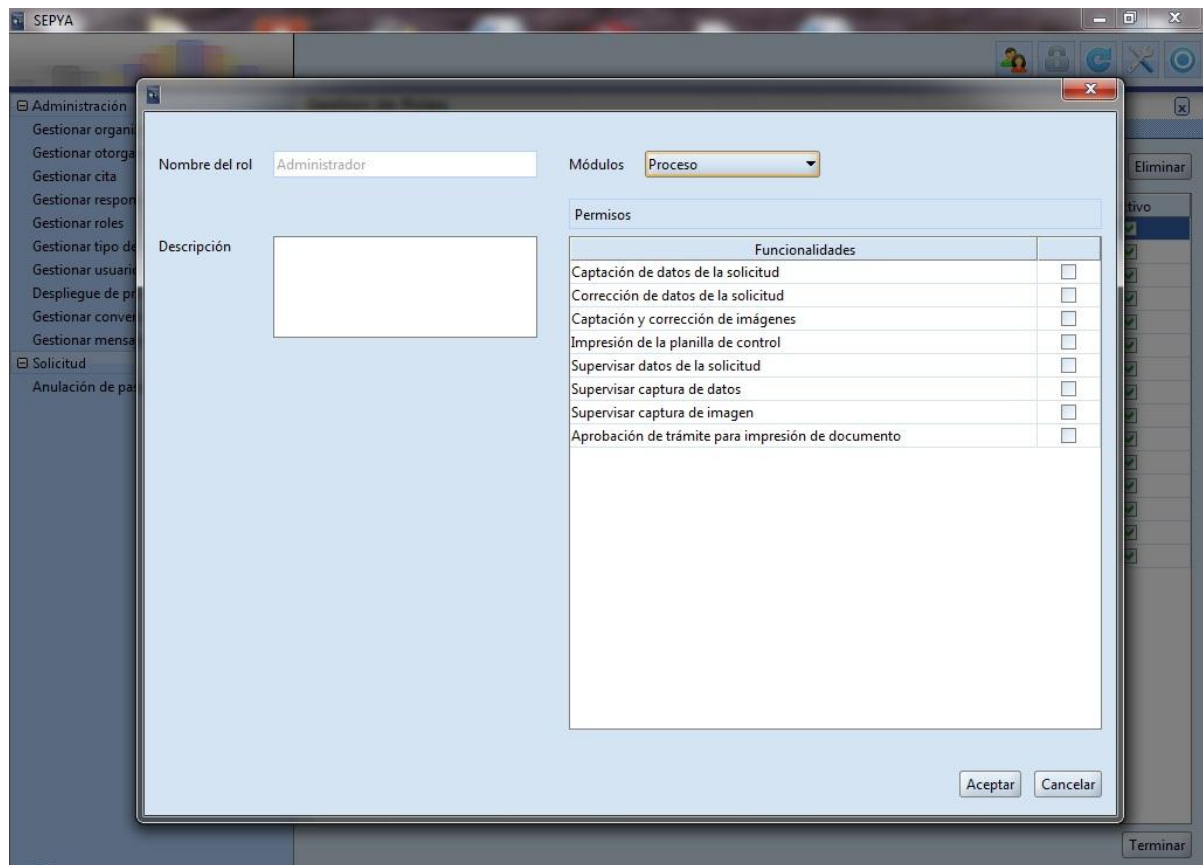


Figura 23 Interfaz consultar roles.

### Descripción alterna

#### Descripción alterna 4 “Cancelar acción”

4. El Administrador presiona Cancelar.
- 4.1 El sistema cancela la acción del usuario mostrando la interfaz Gestión de roles.

#### Descripción alterna 5 “Datos incorrectos”

5. El sistema muestra el siguiente mensaje: “Ha introducido los datos incorrectamente” mostrando la interfaz Crear

	rol.
Descripción alterna 6 “Denegar mensaje”	
	<p>6. El Administrador selecciona No.</p> <p>6.1 El sistema no realiza ninguna acción de eliminación sobre el rol.</p>
Prototipos	
-	
Validaciones	<p>Cuando se introducen datos en el sistema debe verificarse que esté completa toda la información.</p> <p>Al registrar un rol, el mismo no debe existir en la base de datos.</p> <p>Al eliminar el rol, se realiza una eliminación lógica de los datos, dejando la traza de las actividades realizadas por los usuarios que jueguen ese rol.</p>
Postcondiciones	Queda registrado un rol, modificados sus datos, desactivado o activado, así como mostrados sus datos.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla 8. Gestionar convenios de supresión de visa

Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador.
Funcionalidades asociadas	RF8, 8.1, 8.2, 8.3, 8.4, RF4.
Conceptos tratados	<p>Usuario.</p> <p>Convenio de supresión de visa.</p> <p>País.</p>
Descripción básica	<p>1. El sistema muestra la interfaz Principal (1) con un vínculo al módulo Administración que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar organización</li> <li>• Gestionar otorgamiento de Acreditación</li> <li>• Gestionar Cita</li> <li>• Gestionar Responsable de firma</li> <li>• Gestionar roles</li> <li>• Gestionar tipo de organización</li> <li>• Gestionar usuario</li> <li>• Despliegue de proceso</li> <li>• Gestionar convenio de supresión de visa</li> <li>• Gestionar mensajero de organización</li> </ul> <p>2.</p> <p>3. El Administrador selecciona la opción Gestionar</p>

convenio de supresión de visa.

4. El sistema muestra la interfaz Gestión de convenio de supresión de visa con un listado de los países con los convenios existentes que contiene los siguientes datos y opciones:

Datos:

- País

Opciones:

- Filtrar
- Convenio
- Mostrar países con convenio
- Mostrar países sin convenio
- Terminar

3.1 Si el Administrador introduce el país, el sistema muestra en el listado de convenios de supresión los que se corresponden con los datos introducidos en el filtro.

3.2 Si el Administrador selecciona la opción Convenio de un país determinado, el sistema asigna el convenio de supresión de visa al país.

3.3 Si el Administrador desmarca la opción Convenio, el sistema elimina el convenio de supresión de visa del país seleccionado.

3.4 Si el Administrador selecciona la opción Mostrar países con convenio, el sistema actualiza el listado de los países mostrando solamente los que tienen convenio.

3.5 Si el Administrador selecciona la opción Mostrar países sin convenio, el sistema actualiza el listado de los países mostrando solamente los que no tienen convenio.

3.6 Si el Administrador selecciona la opción Terminar, el sistema redirecciona a la interfaz Principal.

Prototipos

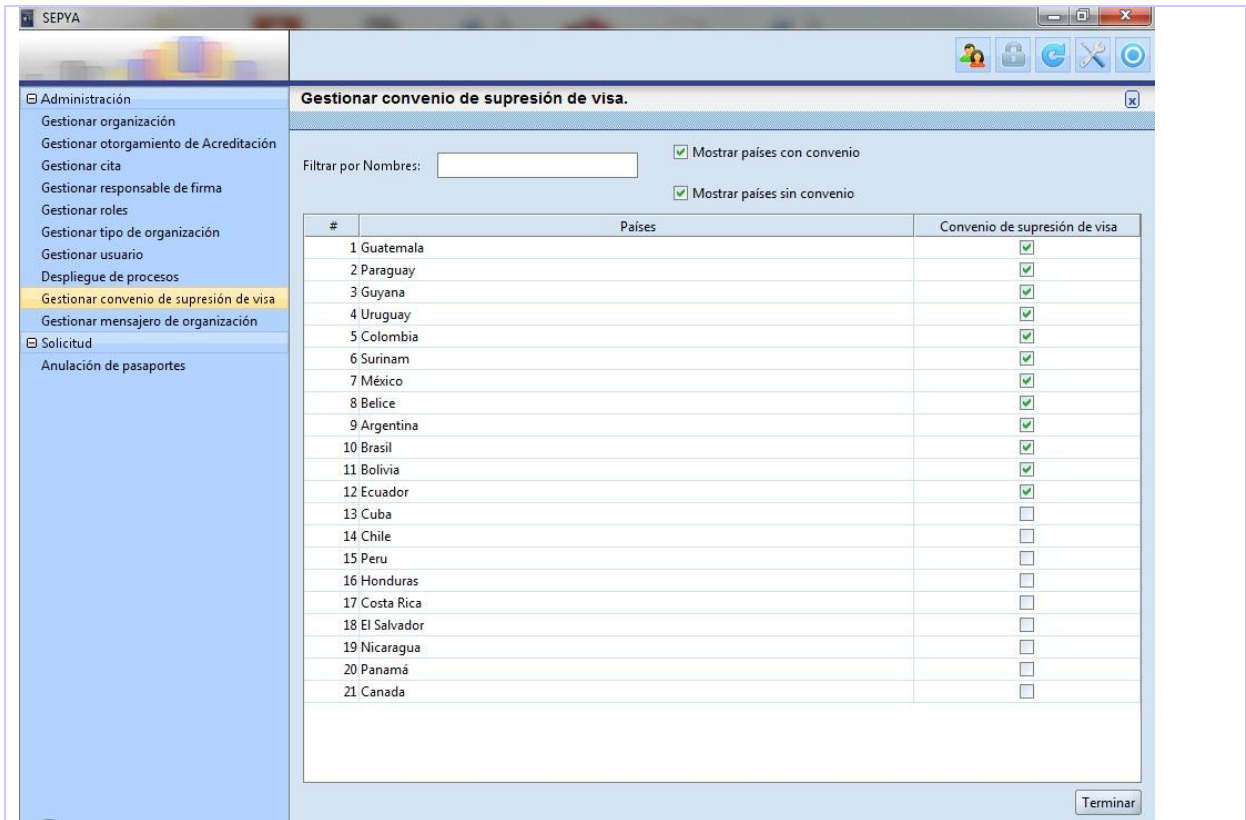


Figura 24 Interfaz principal convenio de supresion de visa.

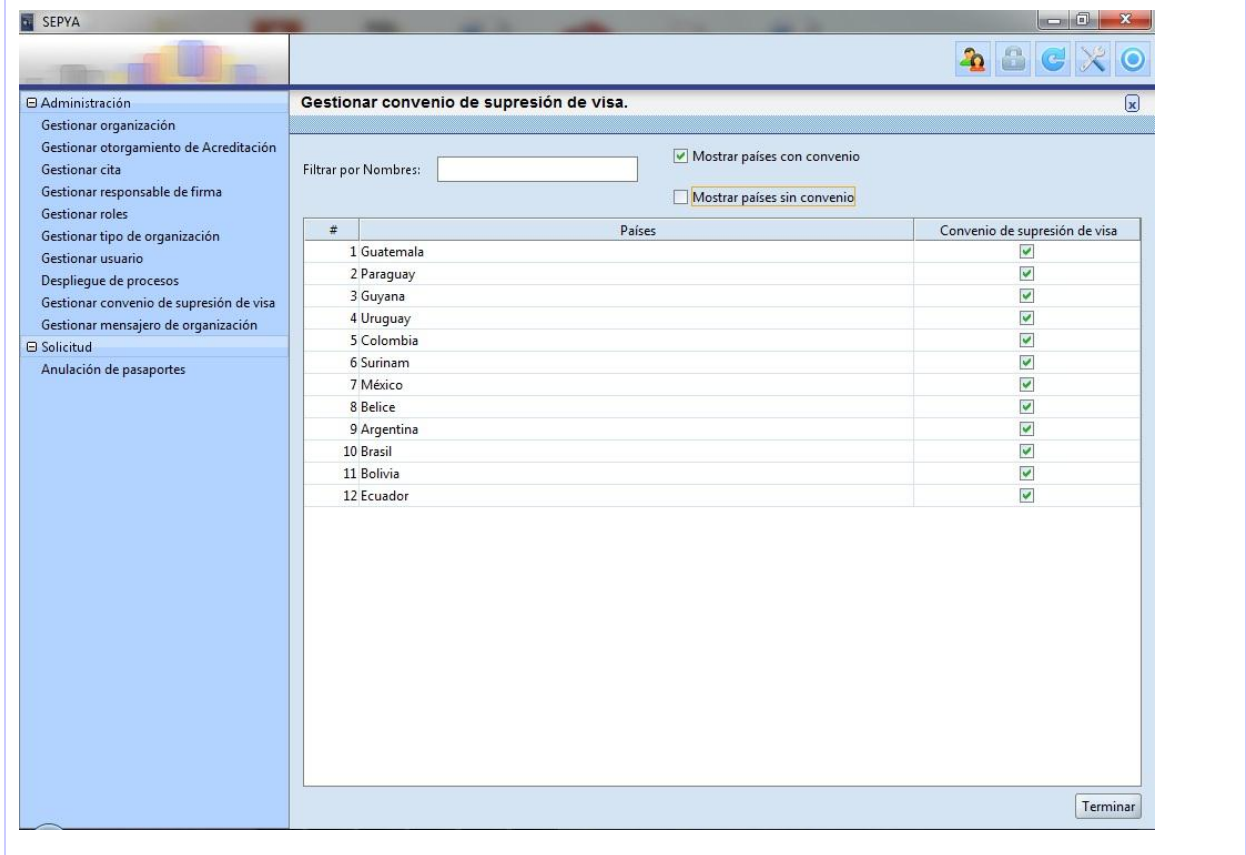




Figura 25 Interfaz países con convenio de supresión de visa.

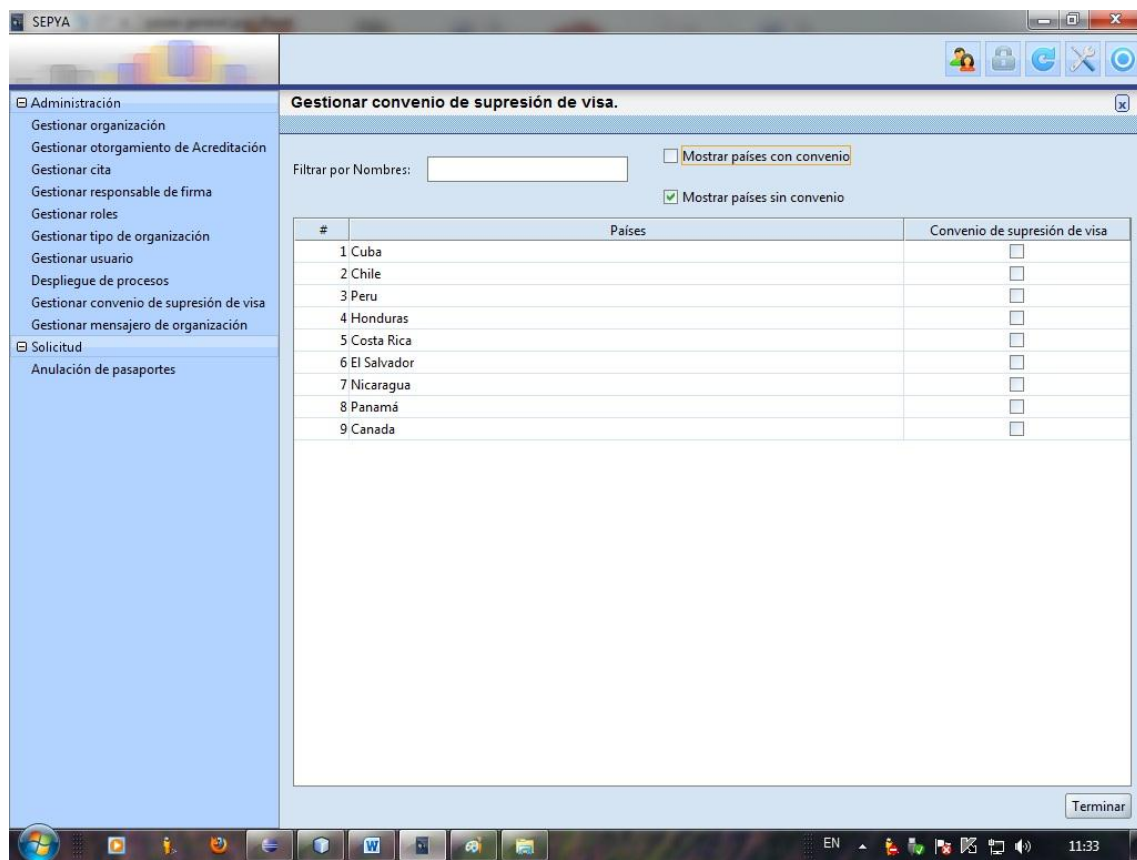


Figura 26 Interfaz países sin convenio de supresión de visa.

Descripción alterna	-
Prototipos	
-	
Validaciones	-
Postcondiciones	Quedan registrados los convenios de supresión de visa, la modificación de sus datos, así como la eliminación de los mismos.
Requisito no funcional	-

Servicios	-
Componente	-

Tabla 10. Gestionar otorgamiento de Acreditación

Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador.
Funcionalidades asociadas	RF35, RF36, RF37, RF38.
Conceptos tratados	Usuario. Acreditación. Carné.
Descripción básica	<p>1. El sistema muestra la interfaz Principal (1) con un vínculo al módulo Administración que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Gestionar organización</li> <li>• Gestionar otorgamiento de Acreditación</li> <li>• Gestionar Cita</li> <li>• Gestionar Responsable de firma</li> <li>• Gestionar roles</li> <li>• Gestionar tipo de organización</li> <li>• Gestionar usuario</li> <li>• Despliegue de proceso</li> <li>• Gestionar convenio de supresión de visa</li> <li>• Gestionar mensajero de organización</li> </ul> <p>2. El Administrador selecciona la opción Gestionar</p>

otorgamiento de Acreditación.

3. El sistema muestra la interfaz Gestión de otorgamiento de Acreditación con un listado de las personas a acreditar que contiene los siguientes datos y opciones:

Datos:

Listado de titulares a acreditar:

- Personas a acreditar

Opciones:

- Carné
- Sello en el Pasaporte
- Funcionario
- Familiar
- Personal de servicio doméstico
- Guardar
- Cancelar

3.1 Si el Administrador selecciona la opción Funcionario, el sistema actualiza el listado de titulares a acreditar con todos los tipos de funcionarios (Funcionario Diplomático, Personal Administrativo y Técnico, Funcionario Internacional y Funcionario Experto en Cooperación Técnica) y la opción Carné seleccionada.

3.2 Si el Administrador Local selecciona la opción Familiar, el sistema actualiza el listado de titulares a

acreditar (cónyuges extranjeros, cónyuges venezolanos, concubinos extranjeros, concubinos venezolanos, acompañantes permanentes extranjeros, acompañantes permanentes venezolanos, hijos extranjeros mayores de 18 años con dependencia económica y cursando estudios, hijos extranjeros menores de 12 años, familiares ascendientes extranjeros con dependencia económica, hijos venezolanos, familiares ascendientes venezolanos) y la opción Carné seleccionada para los casos (cónyuges extranjeros, concubinos extranjeros, acompañantes permanentes extranjeros, hijos extranjeros mayores de 18 años con dependencia económica y cursando estudios, familiares ascendientes extranjeros con dependencia económica).

3.3 Si el Administrador Local selecciona la opción Personal de servicio doméstico, el sistema actualiza el listado de titulares a acreditar (Personal de servicio doméstico extranjero, Personal de servicio doméstico venezolano) y la opción carné seleccionada para el caso de personal de servicio doméstico extranjero.

3.4 Si el Administrador desmarca las opciones Carné o

Sello en el Pasaporte, el sistema elimina la asignación de carné y sello en el Pasaporte para una persona.

3.5 Si el Administrador marca las opciones Carné o Sello en el Pasaporte, el sistema asigna el carné y sello en el Pasaporte para una persona.

3.6 Si el Administrador selecciona la opción Guardar, el sistema registra los cambios realizados.

3.7 Si el Administrador selecciona la opción Cancelar, el sistema no registra ningún cambio y redirecciona a la interfaz principal.

Prototipos

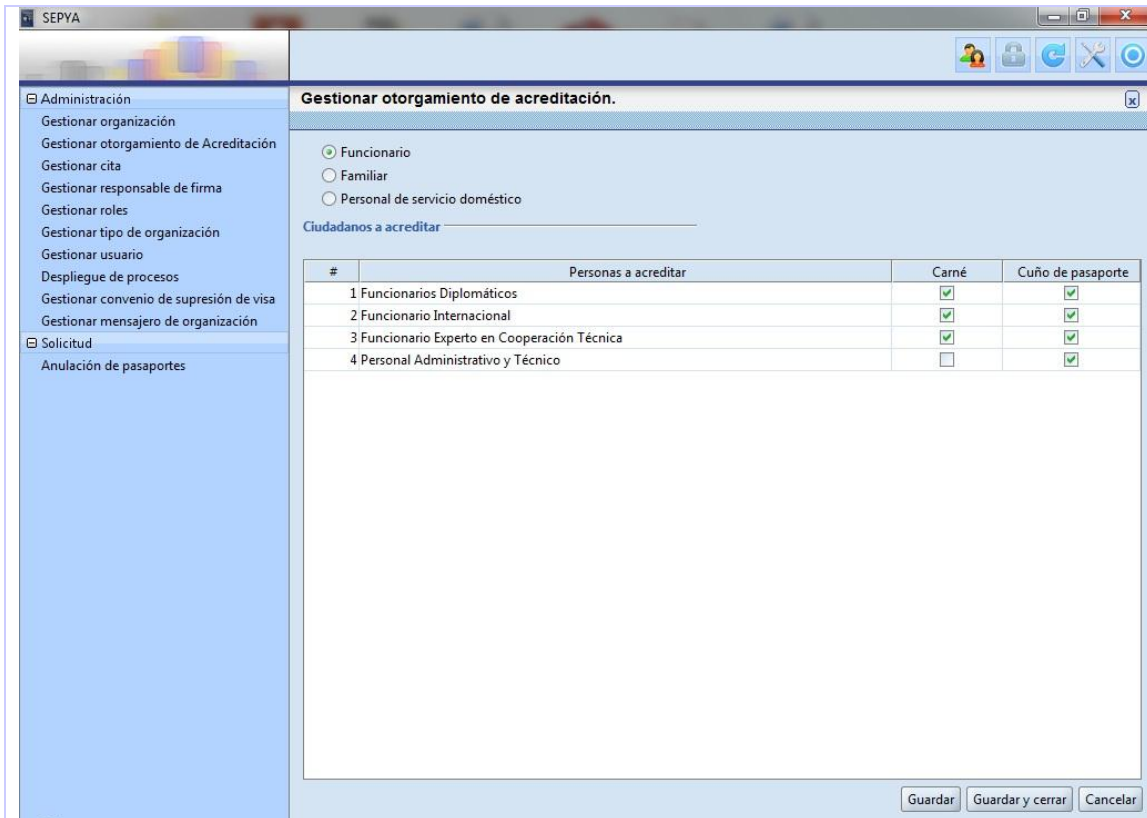


Figura 27 Interfaz acreditación para funcionarios.

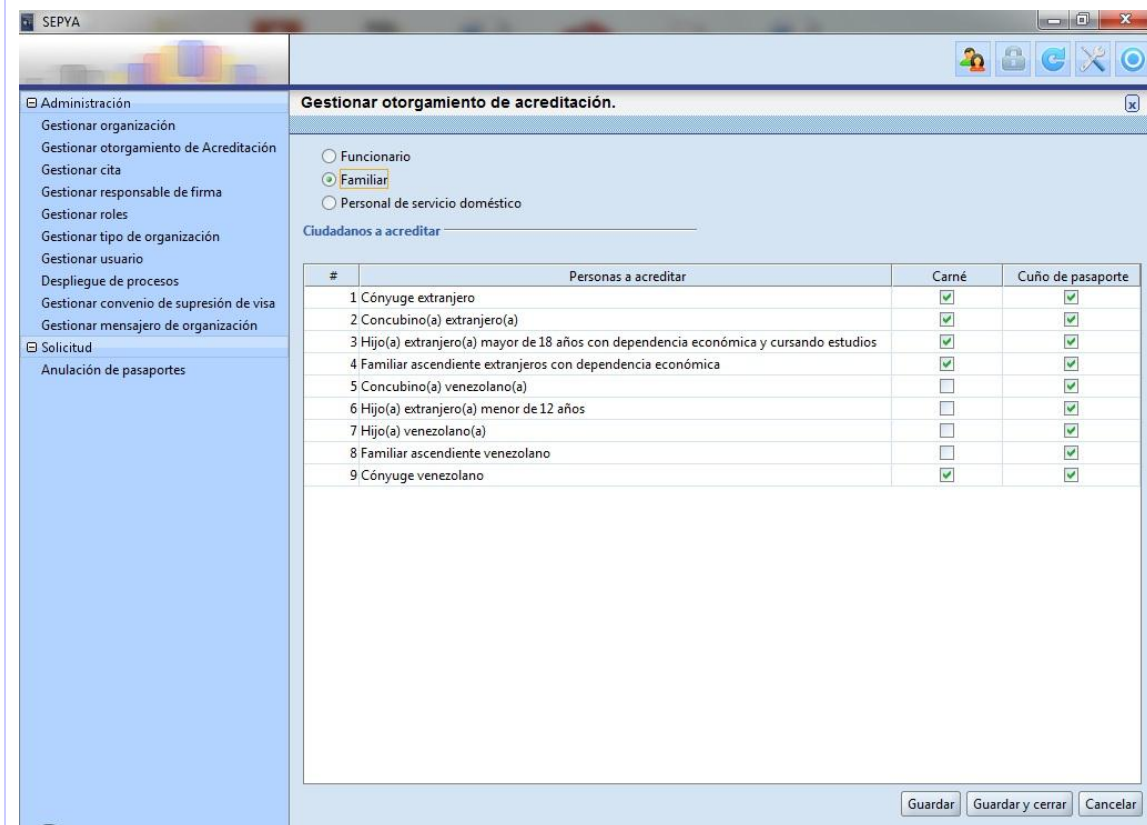


Figura 28 Interfaz acreditación para familiares.

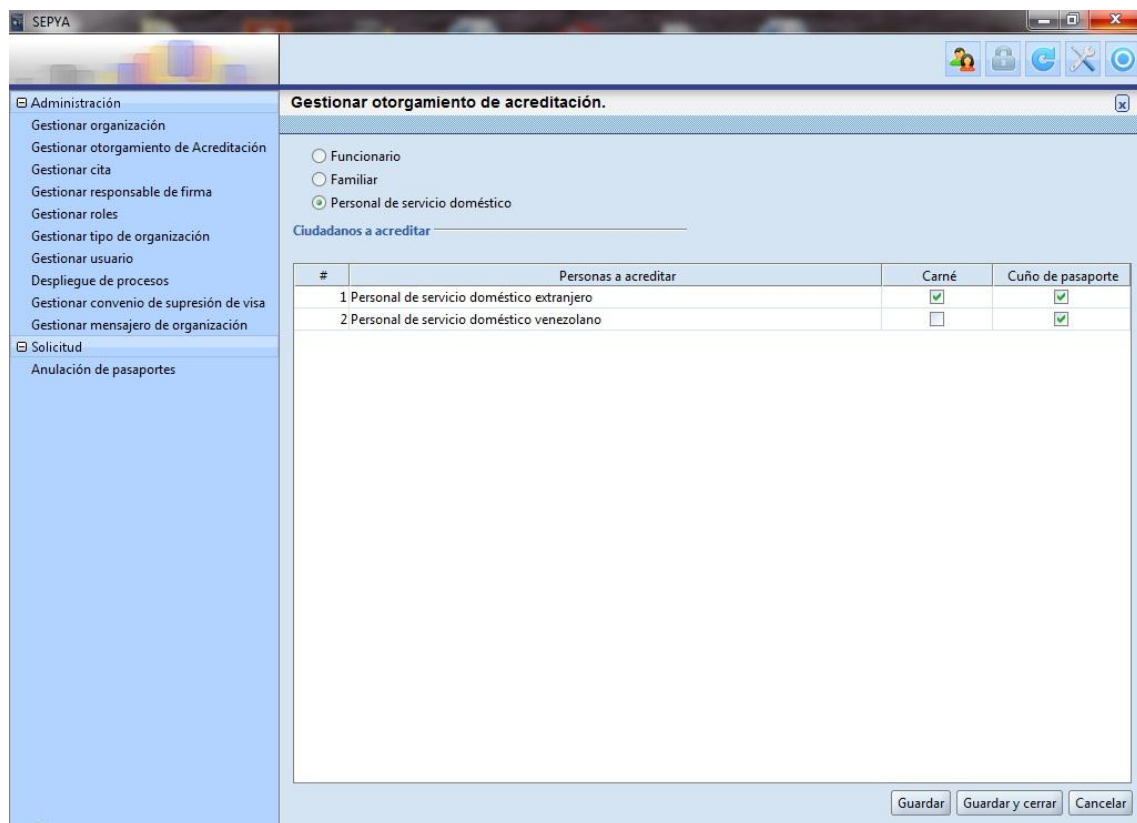


Figura 29 Interfaz acreditación para personal de servicio doméstico.

Descripción alterna	-
Prototipos	
-	
Validaciones	Por defecto la opción sello en el Pasaporte siempre se visualizará marcada.
Postcondiciones	Quedan registradas a qué personas se les otorga carné y sello en el Pasaporte.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla 11. Planificación semanal de citas

Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador.
Funcionalidades asociadas	RF39, RF40.
Conceptos tratados	Usuario.  Planificación.  Cita.
Descripción básica	<p>1. El sistema muestra la interfaz Principal (1) con un vínculo al módulo Administración que contiene la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Gestionar organización</li> <li>• Gestionar otorgamiento de Acreditación</li> <li>• Gestionar Cita</li> <li>• Gestionar Responsable de firma</li> <li>• Gestionar roles</li> <li>• Gestionar tipo de organización</li> <li>• Gestionar usuario</li> <li>• Despliegue de proceso</li> <li>• Gestionar convenio de supresión de visa</li> <li>• Gestionar mensajero de organización</li> </ul> <p>2. El usuario selecciona la opción Gestionar cita.</p> <p>3. El sistema muestra la Interfaz planificación semanal de citas con los siguientes datos y</p>



opciones:

Datos:

- Listado de planificaciones semanales:
  - Semanas (día/mes/año – día/mes/año)

Opciones:

- Consultar
- Modificar
- Agregar
- Terminar

3.1 Si el usuario selecciona la opción Consultar, el sistema muestra en la misma interfaz la planificación de la semana seleccionada, con la opción Finalizar.

3.1.1 El usuario selecciona la opción Finalizar.

3.1.2 El sistema muestra el listado de planificaciones con las opciones.

3.2 Si el usuario selecciona la opción Terminar, el sistema redirecciona a la interfaz principal.

3.3 Si el usuario selecciona la opción Agregar (Consultar tabla 10.1).

3.4 Si el usuario selecciona la opción Modificar (Consultar tabla 10.2).

Prototipos

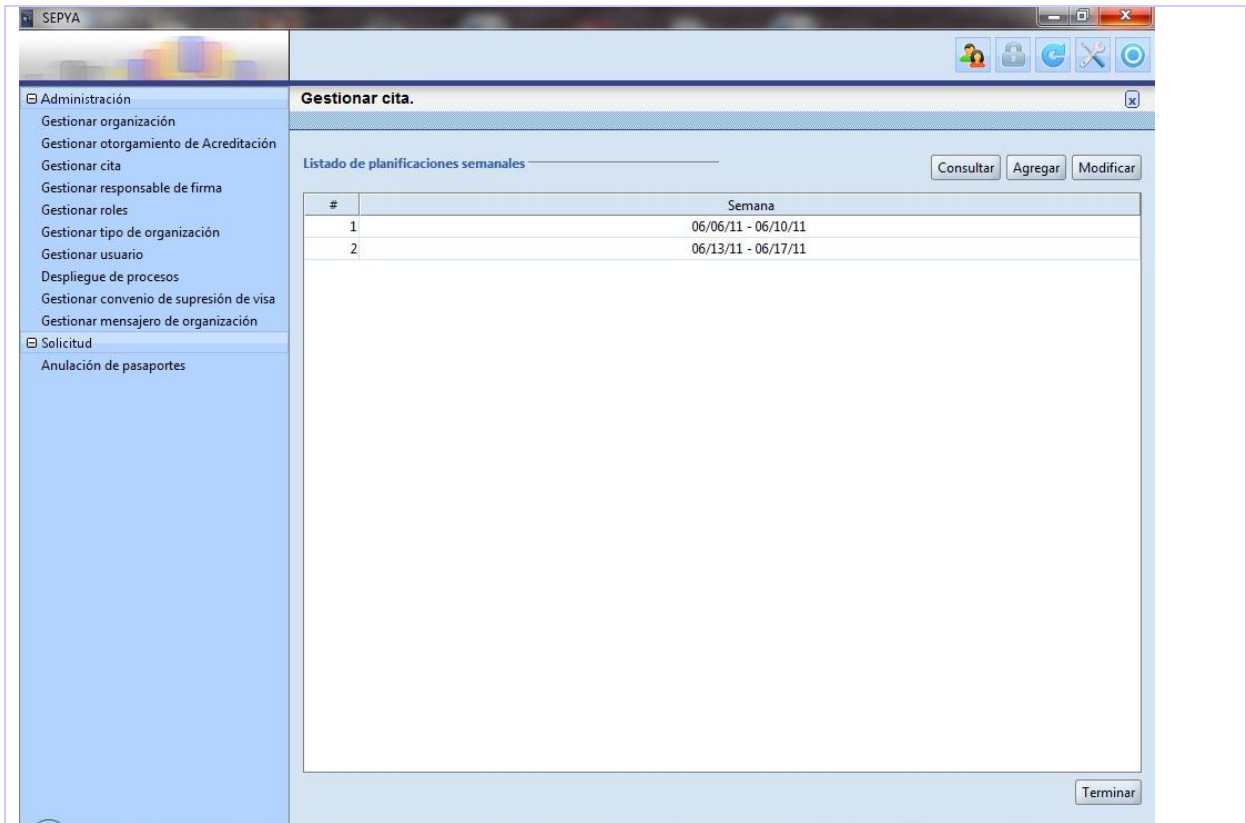


Figura 30 Interfaz pan semanal de citas.

Descripción alterna	-
Prototipos	
-	
Validaciones	Si no hay planificación el listado de planificaciones se muestra vacío, en caso de haberlas contiene las planificaciones de la fecha actual en adelante.
Postcondiciones	
Requisito no funcional	-
Servicios	-
Componente	-

Tabla 11.1. Crear planificación diaria de citas

Precondiciones	El usuario debe estar autenticado en el sistema con el rolAdministrador.
Funcionalidades asociadas	RF41.
Conceptos tratados	Usuario.  Planificación.  Cita.
Descripción básica	<p>1. El sistema muestra la interfaz Crear planificación diaria de citas que contiene los siguientes datos y opciones:</p> <p>Datos:</p> <ul style="list-style-type: none"> <li>• Semanas (día/mes/año) - (día/mes/año)</li> <li>• Tiempo estimado entre turnos (min)</li> <li>• Listado de planificaciones diarias:                         <ul style="list-style-type: none"> <li>○ Días (lunes, martes, miércoles, jueves, viernes)</li> <li>○ Horario mañana</li> <li>○ Horario tarde</li> <li>○ Cupos de Pasaporte</li> <li>○ Cupos de Acreditación</li> </ul> </li> </ul> <p>Opciones:</p> <ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar</li> </ul> <p>2. El usuario selecciona la semana, los días, los</p>

horarios a planificar e introduce el tiempo estimado entre turnos.

3. El sistema por defecto muestra seleccionable los cupos de Pasaporte y Acreditación y permite modificar la asignación de cupos.

4. El usuario selecciona la opción Aceptar. (En caso de seleccionar otra opción consultar: *Descripción alterna 6 "Cancelar acción"*).

5. El sistema guarda la planificación de cupos semanales.

## Prototipos

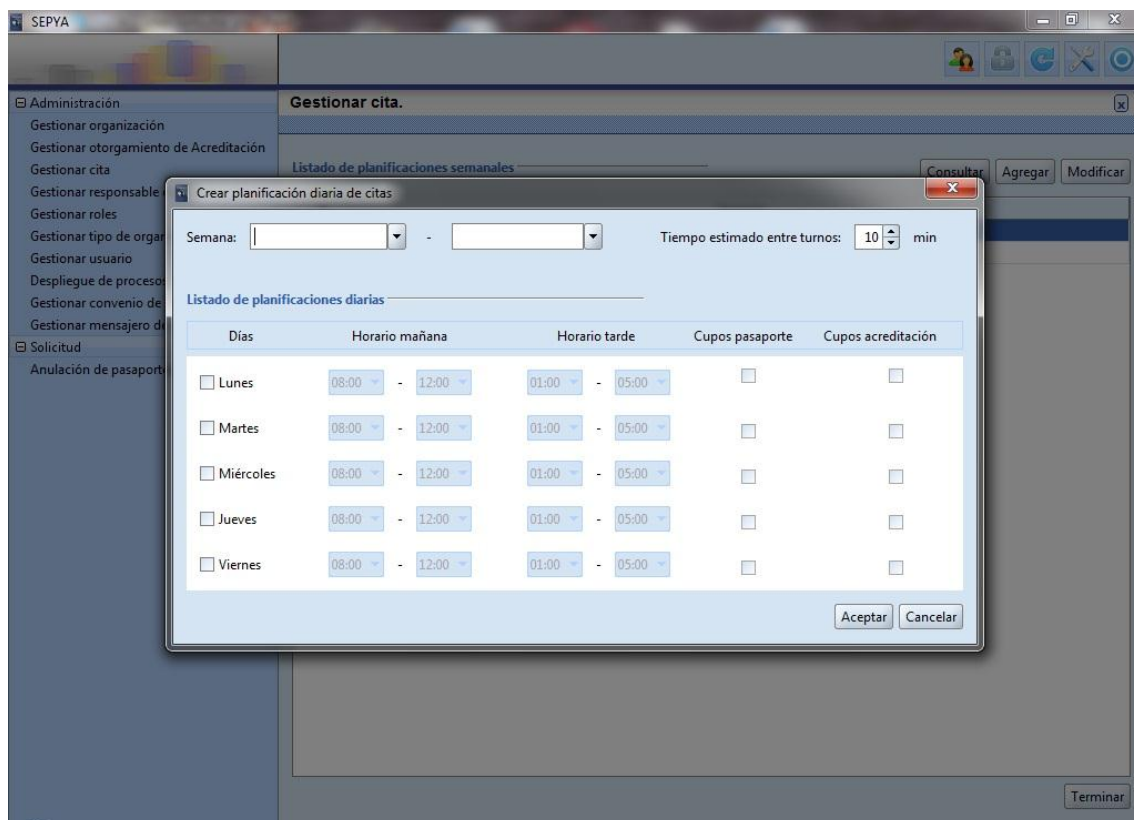


Figura 31 Interfaz agregar plan diario de citas.

Descripción alterna	
Descripción alterna 6 “Cancelar acción”	
	<p>6. El usuario selecciona la opción Cancelar.</p> <p>6.1 El sistema cancela la acción del usuario, redireccionando a la interfaz Planificación semanal de citas.</p>
Prototipos	
-	
Validaciones	Por defecto los días de la semana van a estar deshabilitados, a medida que se seleccionan es que se habilitan los otros campos.
Postcondiciones	Queda creada la planificación diaria de citas.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla 11.2. Modificar planificación diaria de citas

Precondiciones	El usuario debe estar autenticado en el sistema con el rolAdministrador.
Funcionalidades asociadas	RF42.
Conceptos tratados	Usuario.

	<p>Planificación.</p> <p>Cita.</p>
<p>Descripción básica</p>	<p>1. El sistema muestra la interfaz Modificar planificación diaria de citas que contiene los siguientes datos y opciones:</p> <p>Datos:</p> <ul style="list-style-type: none"> <li>• Semanas desde (día/mes/año) hasta (día/mes/año)</li> <li>• Tiempo estimado entre turnos (min)</li> <li>• Listado de planificaciones diarias: <ul style="list-style-type: none"> <li>○ Días (lunes, martes, miércoles, jueves, viernes)</li> <li>○ Horario mañana</li> <li>○ Horario tarde</li> <li>○ Cupos de Pasaporte</li> <li>○ Cupos de Acreditación</li> </ul> </li> </ul> <p>Opciones:</p> <ul style="list-style-type: none"> <li>• Modificar</li> <li>• Cancelar</li> </ul> <p>2. El usuario introduce los nuevos datos y presiona Modificar. (En caso de seleccionar otra opción consultar: <i>Descripción alterna 4 “Cancelar acción”</i>).</p> <p>3. El sistema actualiza la planificación diaria.</p>
<p>Prototipos</p>	

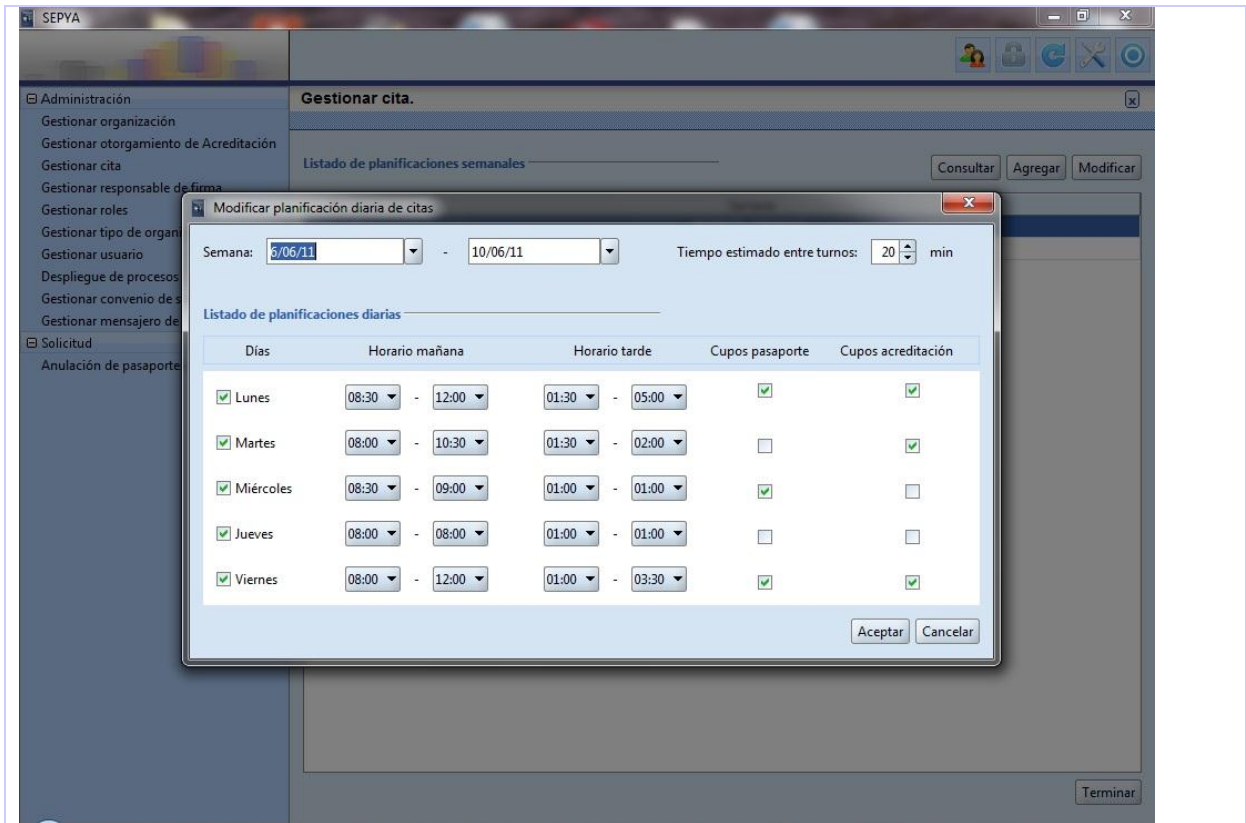


Figura 32 Interfaz modificar plan semanal de citas.

### Descripción alterna

#### Descripción alterna 4 “Cancelar acción”

4. El usuario selecciona la opción Cancelar.
- 4.1 El sistema cancela la acción del usuario, redireccionando a la interfaz Planificación semanal de citas.

### Prototipos

-

### Validaciones

Por defecto los días de la semana van a estar deshabilitados, a medida que se seleccionan es que se habilitan los otros campos.

Postcondiciones	Queda modificada la planificación de citas.
Requisito no funcional	-
Servicios	-
Componente	-



Anexo 2 Diagramas de clases

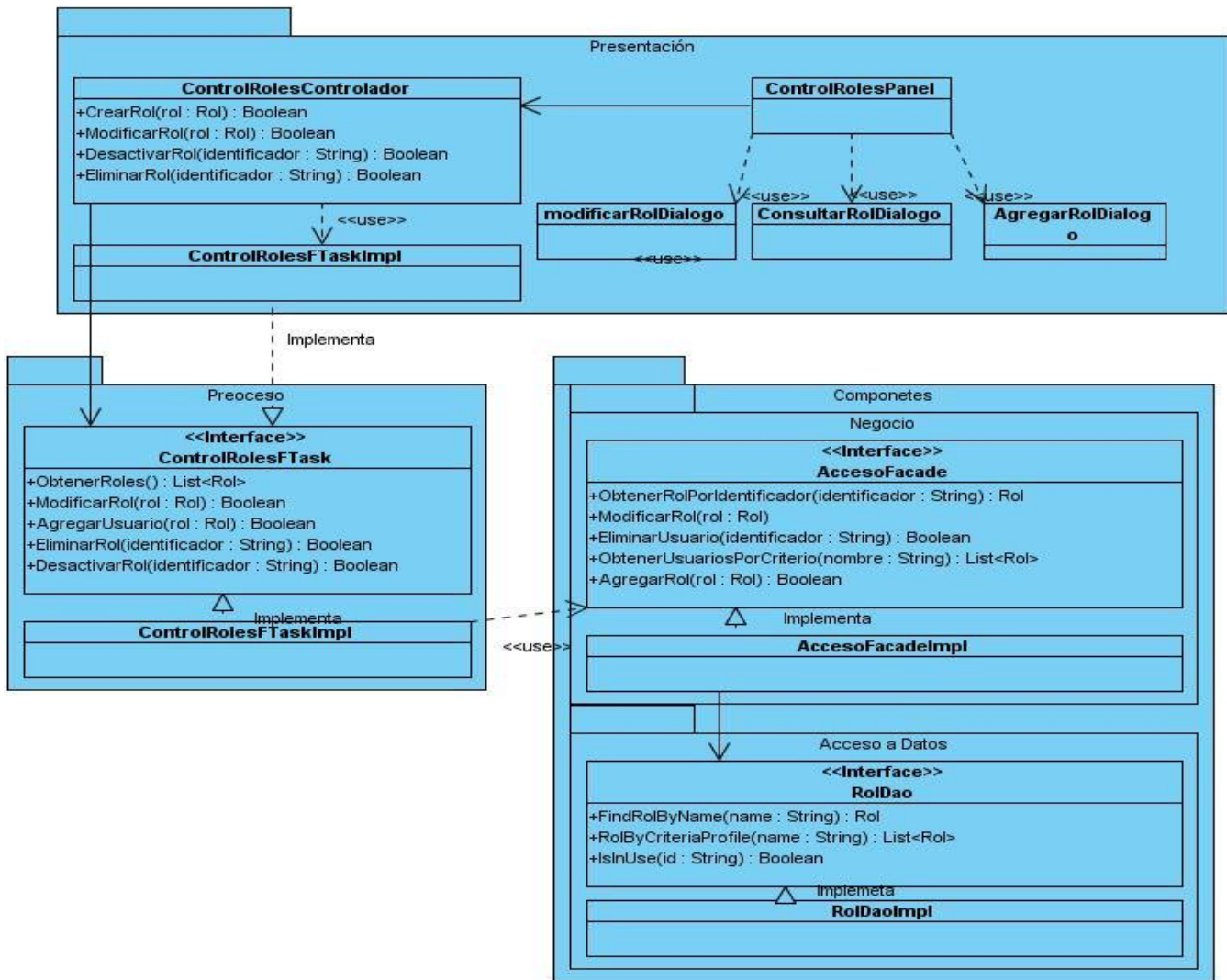


Figura 33 Diagrama de clases para la funcionalidad Gestionar rol.

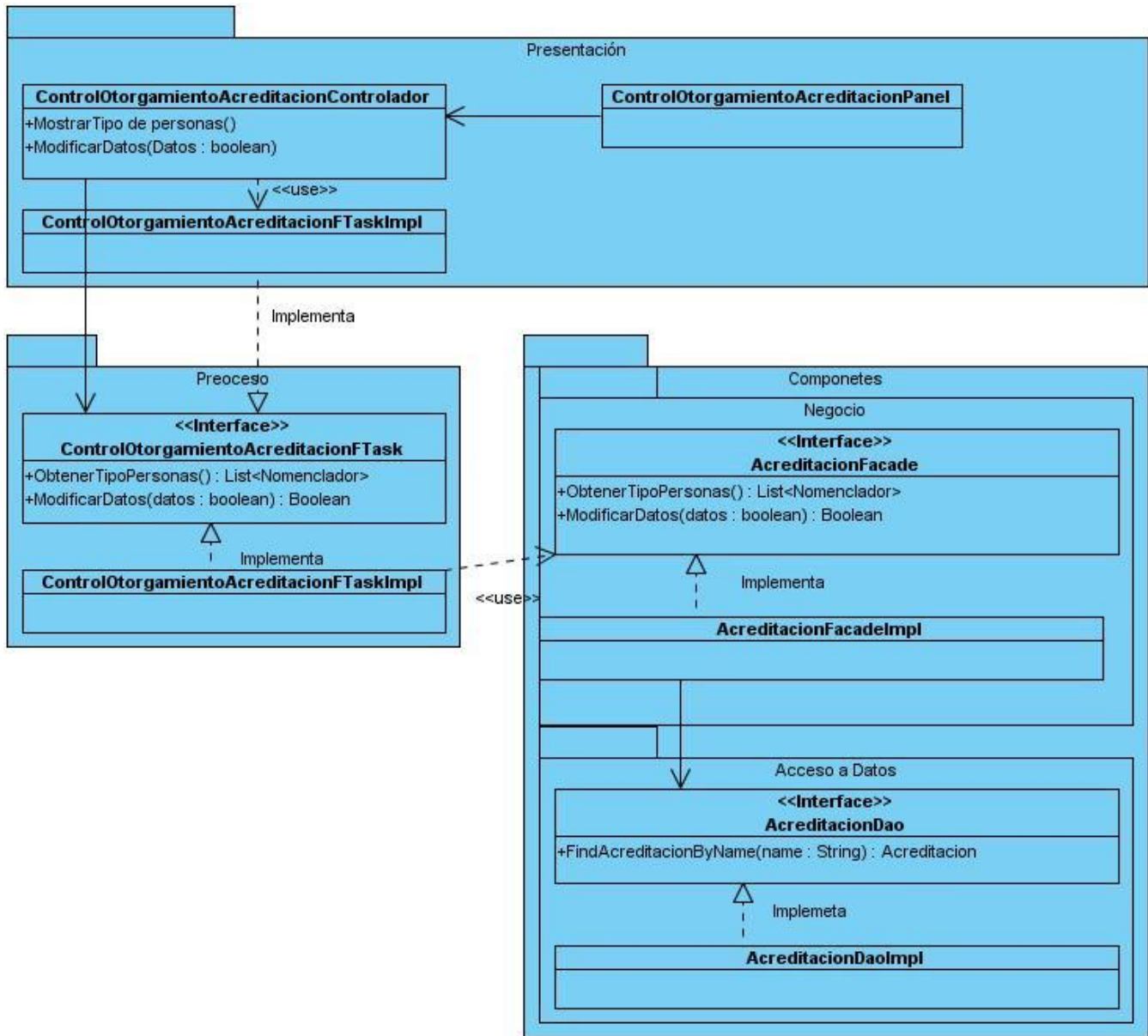


Figura 34 Diagrama de clases para la funcionalidad Definir otorgamiento de acreditación.

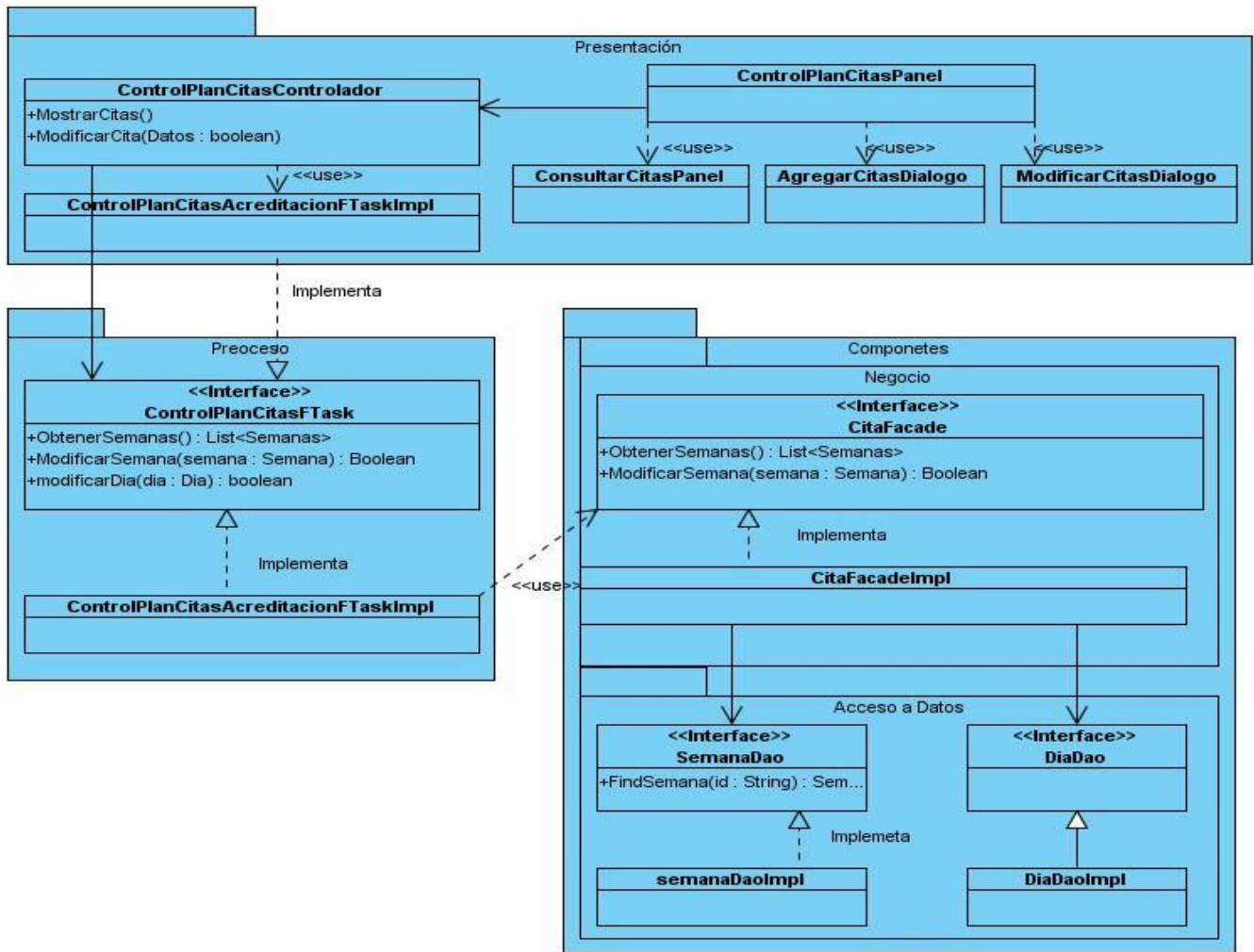


Figura 35 Diagrama de clases para la funcionalidad Gestionar planificación de citas.

Anexo 3 Descripción de clases

Nombre: ControlUsuarioFTask	
Tipo de clase: interfaz	
Está diseñada para la gestión y manejo de todas las acciones básicas que se realizan a los usuarios en la capa de procesos, dándole soporte a la capa de presentación.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	CrearUsuario(AccesoUsuario usuario)
Descripción:	Realiza el proceso de crear un usuario enviando los datos de este a la capa de componentes donde se realizan las acciones necesarias para insertarlo en la base de datos.
Nombre:	ModificarUsuario(AccesoUsuario usuario)
Descripción:	Realiza el proceso de modificar un usuario enviando los datos de la modificación a la capa de componentes donde se realizan las acciones necesarias para ejercer la modificación en la base de datos.
Nombre:	ObtenerUsuarios()
Descripción:	Retorna todos los usuarios del sistema.
Nombre:	eliminarUsuario(String identificador)
Descripción:	Realiza el proceso necesario de eliminar el usuario al cual pertenece el identificador pasado como parámetros.
Nombre:	DesactivarUsuario(String identificador)
Descripción:	Realiza el proceso necesario de eliminar el usuario al cual pertenece el identificador pasado como parámetros.

Tabla 17 Descripción de la clase ControlUsuarioFTaskImpl.

Nombre: ControlUsuarioControlador	
Tipo de clase: Controladora	
Está diseñada para el control de las acciones realizadas por el usuario al interactuar con el sistema.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	agregarUsuario(Rol rol)
Descripción:	Registra (o inserta) los datos del usuario que se le pasa como parámetro en la base de datos.
Nombre:	modificarUsuario(Rol rol)
Descripción:	Realiza la función de modificar los datos de un usuario que se le haya pasado por parámetro.
Nombre:	obtenerUsuarios()
Descripción:	Realiza la función de devolver un listado de todos los usuario.
Nombre:	eliminarUsuario(String identificador)
Descripción:	Elimina el usuario al cual pertenece el identificador pasado como parámetro de la base de datos.
Nombre:	desactivarUsuario(String idRol)
Descripción:	Realiza la operación de desactivar el usuario al cual pertenece el identificador pasado como parámetro de la base de datos.

Tabla 18 Descripción de la clase ControlUsuarioControlador.

Nombre: AccesoFacadeImpl
Tipo de clase: Interfaz
Está diseñada para la gestión y manejo de todas las acciones básicas que se realizan a los usuarios en la

capa de negocio, dándole soporte a la capa de procesos.	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	registrarUsuario(AccesoUsuario accesoUsuario)
Descripción:	Registra (o inserta) el usuario que se le pasa como parámetro en la base de datos.
Nombre:	modificarUsuario(AccesoUsuario accesoUsuario)
Descripción:	Realiza la función de modificar los datos de un usuario que se le haya pasado por parámetro.
Nombre:	obtenerUsuarios()
Descripción:	Realiza la función de devolver un listado de todos los usuarios.
Nombre:	EliminarUsuario(String identificador)
Descripción:	Realiza la función de eliminar el usuario al cual pertenece el identificador pasado como parámetro.
Nombre:	DesactivarUsuario(String identificador)
Descripción:	Desactiva el usuario al cual pertenece el identificador pasado como parámetro.

Tabla 19 Descripción de la clase AccesoFacadeImpl.

#### Anexo 4 Descripción de las tablas de la base de datos

Nombre de la entidad	tbl_acceso_funcionalidad
Definición	Entidad que representa una funcionalidad.

#### *ATRIBUTOS*

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_funcionalidad		CHAR(32)	N	Atributo que representa el identificador de la funcionalidad.
identificador_app		INTEGER	N	Atributo que representa el identificador.
funcionalidad		TEXT	N	Atributo que representa el nombre de la funcionalidad
descripcion		TEXT	S	Atributo que representa la descripción de la funcionalidad.
fk_acceso_modulo		CHAR(32)	N	Atributo que representa el identificador del módulo.

Nombre de la entidad	tbl_acceso_modulo
Definición	Entidad que representa un modulo.

#### *ATRIBUTOS*

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_modulo		CHAR(32)	N	Atributo que representa el identificador del módulo.

modulo	VARCHAR(255 )	N	Atributo que representa nombre del módulo.
descripcion	TEXT	S	Atributo que representa la descripción del módulo.

Nombre de la entidad	tbl_acceso_perfil
Definición	Entidad que representa el perfil de un usuario.

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_perfil		CHAR(32)	N	Atributo que representa el identificador del perfil de un usuario.
cedula		BIGINT	S	Atributo que representa cedula.
letra		CHAR(1)	S	Atributo que representa letra.
identificador_nacional		VARCHAR(30)	S	Atributo que representa identificador nacional.
primer_nombre		VARCHAR(30)	N	Atributo que representa primer nombre.
segundo_nombre		VARCHAR(30)	S	Atributo que representa Segundo nombre.
primer_apellido		VARCHAR(30)	N	Atributo que representa primer apellido.
segundo_apellido		VARCHAR(30)	S	Atributo que representa Segundo apellido.
correo_electronico		VARCHAR(50)	N	Atributo que representa dirección de correo electrónico.



numero_pasaporte	VARCHAR(20)	S	Atributo que representa número de pasaporte.
fk_direccion_pais_nacionalidad	VARCHAR(4)	N	Atributo que representa el identificador de pais.

Nombre de la entidad	tbl_acceso_rol		
Definición	Entidad que representa el rol de un usuario.		

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_rol		CHAR(32)	N	Atributo que representa el identificador del rol de un usuario.
nombre_rol		VARCHAR(50)	N	Atributo que representa el nombre del rol.
rol_habilitado		BIT	N	Atributo que representa el rol habilitado.
descripcion		TEXT	S	Atributo que representa la descripción del rol.

Nombre de la entidad	tbl_acceso_rol_funcionalidad		
Definición	Entidad que representa la relación entre Rol--Funcionalidad.		

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_rol		CHAR(32)	N	Atributo que representa el identificador del rol .
pk_acceso_funcionalidad		CHAR(32)	N	Atributo que representa el identificador de la funcionalidad.

Nombre de la entidad	tbl_acceso_rol_usuario
Definición	Entidad que representa la relación entre Rol--Usuario

### *ATRIBUTOS*

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_usuario		CHAR(32)	N	Atributo que representa el identificador de un usuario.
pk_acceso_rol		CHAR(32)	N	Atributo que representa el identificador del rol.

Nombre de la entidad	tbl_acceso_usuario
Definición	Entidad que representa un usuario.

### *ATRIBUTOS*

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_usuario		CHAR(32)	N	Atributo que representa el identificador de usuario.
existe		BIT	N	Atributo que representa la existencia de usuario.
fk_acceso_perfil		CHAR(32)	N	Atributo que representa el identificador del perfil de usuario.
fk_organismo_organizacion		CHAR(32)	N	Atributo que representa el identificador de organización.

Nombre de la entidad    **tbl\_acceso\_usuario\_eliminado**

Definición                    Entidad que representa a un usuario eliminado.

*ATRIBUTOS*

---

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_usuario		CHAR(32)	N	Atributo que representa el identificador de un usuario.
nombre_usuario		VARCHAR(40)	N	Atributo que representa el nombre de un usuario.

Nombre de la entidad    **tbl\_acceso\_usuario\_vigente**

Definición                    Entidad que representa a un usuario vigente.

*ATRIBUTOS*

---

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acceso_usuario		CHAR(32)	N	Atributo que representa el identificador de un usuario.
nombre_usuario		VARCHAR(40)	N	Atributo que representa el nombre de un usuario.
clave_usuario		VARCHAR(35)	N	Atributo que representa la clave de un usuario.
ip_acceso		VARCHAR(255)	N	Atributo que representa la dirección IP desde la cual se conecta el usuario.

usuario\_activo BIT N Atributo que representa si el usuario está activo.

Nombre de la entidad tbl\_acreditacion\_modelo\_carne

Definición Entidad que representa la acreditación de modelo de carne.

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acreditacion_modelo_carne		CHAR(32)	N	Atributo que representa el identificador de una acreditación de modelo de carne.
modelo_carne		VARCHAR(255)	N	Atributo que representa el modelo de un carne.
siglas		VARCHAR(255)	N	Atributo que representa las siglas.

Nombre de la entidad tbl\_acreditacion\_otorgamiento\_carne

Definición Entidad que representa la acreditación del otorgamiento de un carne

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_acreditacion_otorgamiento_carne		CHAR(32)	N	Atributo que representa el identificador de la acreditación del otorgamiento de un carne.
especificacion		VARCHAR(255)	S	Atributo que representa la especificación.

carne	BIT	N	Atributo que representa el carne.
sello	BIT	N	Atributo que representa el sello.
fk_acreditacion_modelo_carne	CHAR(32)	N	Atributo que representa el identificador de la acreditación del modelo de carne.
fk_persona_acreditar	CHAR(32)	N	Atributo que representa el identificador de la persona a acreditar.

Nombre de la entidad	tbl_adm_dia
Definición	Entidad que representa un día.

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_adm_dia		CHAR(32)	N	Atributo que representa el identificador de día.
dia_semana		INTEGER	N	Atributo que representa el identificador del día de la semana.
numero_dia		INTEGER	N	Atributo que representa el número del día.
manana_laborable		BIT	S	Atributo que representa si la mañana es laborable.
hora_inicio_manana		TIME/DATETIME	S	Atributo que

			representa la hora de inicio de la mañana.
hora_fin_manana	TIME/DATETIME	S	Atributo que representa la hora de fin de la mañana.
tarde_laborable	BIT	S	Atributo que representa si la tarde es laborable.
hora_inicio_tarde	TIME/DATETIME	S	Atributo que representa la hora de inicio de la tarde.
hora_fin_tarde	TIME/DATETIME	S	Atributo que representa la hora de fin de la tarde.
cupos_pasaporte	BIT	N	Atributo que representa si se atenderán solicitudes de pasaportes.
cupos_acreditacion	BIT	N	Atributo que representa si se atenderán solicitudes de acreditación
fk_adm_semana	CHAR(32)	N	Atributo que representa el identificados de la semana a la que pertenece.

Nombre de la entidad	tbl_adm_semana
Definición	Entidad que representa una semana.

### ATRIBUTOS

---

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_adm_semana		CHAR(32)	N	Atributo que representa el identificador de semana.
fecha_inicio		DATE	N	Atributo que representa la hora de inicio de la semana.
fecha_fin		DATE	N	Atributo que representa la hora de fin de la semana.
tiempo_servicio		INTEGER	N	Atributo que representa el tiempo de servicio.

Nombre de la entidad	tbl_convenio_visa
Definición	Entidad que representa el convenio de visa.

### ATRIBUTOS

---

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_convenio_visa		CHAR(32)	N	Atributo que representa el identificador del convenio de visa.
fk_direccion_pais		VARCHAR(4)	N	Atributo que representa el identificador de país.

Nombre de la entidad	tbl_direccion_pais
Definición	Entidad que representa la dirección de país.

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_direccion_pais		VARCHAR(4)	N	Atributo que representa el identificador de país.
valor_pais		VARCHAR(100)	N	Atributo que representa el valor del país.
nacionalidad		VARCHAR(100)	N	Atributo que representa la nacionalidad.
activo		BIT	N	Atributo que representa si es active.

Nombre de la entidad	tbl_nomenclador
Definición	Entidad que representa

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_nomenclador		CHAR(32)	N	Atributo que representa el identificador del nomenclador.
clave_nomenclador		VARCHAR(5)	N	Atributo que



valor_nomenclador	VARCHAR(200)	N	representa la clave del nomenclador. Atributo que representa el valor del nomenclador.
numero	INTEGER	N	Atributo que representa el número.
fk_nomenclador_padre	CHAR(32)	S	Atributo que representa el identificador del nomenclador padre.

Nombre de la entidad	tbl_organismo_organizacion
Definición	.....

### ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_organismo_organizacion		CHAR(32)	N	Atributo que representa el identificador de una organización.
nombre_organizacion		VARCHAR(100)	N	Atributo que representa el nombre de una organización.
funciones_desarrolladas		TEXT	N	Atributo que representa las funciones desarrolladas en la organización.

correo	VARCHAR(50)	N	Atributo que representa el correo electrónico de una organización.
telefono	VARCHAR(50)	N	Atributo que representa el número de teléfono de organización.
organizacion_activa	BIT	N	Atributo que representa si es activa la organización.
existe	BIT	N	Atributo que representa si existe organización.
tipo_solicitante	BIT	N	Atributo que representa el tipo de solicitante.
fk_organismo_tipo_organizacion	CHAR(32)	N	Atributo que representa el identificador de un tipo organización.

Nombre de la entidad	tbl_organismo_tipo_organizacion
Definición	Entidad que representa E tipo de Organizacion.

## ATRIBUTOS

Nombre del atributo	Dominio	Tipo de dato	Nulo	Definición
pk_organismo_tipo_organizacion		CHAR(32)	N	Atributo que representa el identificador de tipo

			organización.
nombre_tipo_organizacion	VARCHAR(200)	N	Atributo que representa el nombre del tipo de organización.
tipo_organizacion_activa	BIT	N	Atributo que representa el tipo de organización activa.
existe	BIT	N	Atributo que representa si existe tipo de organización.

Anexo 5 Diagramas de componentes

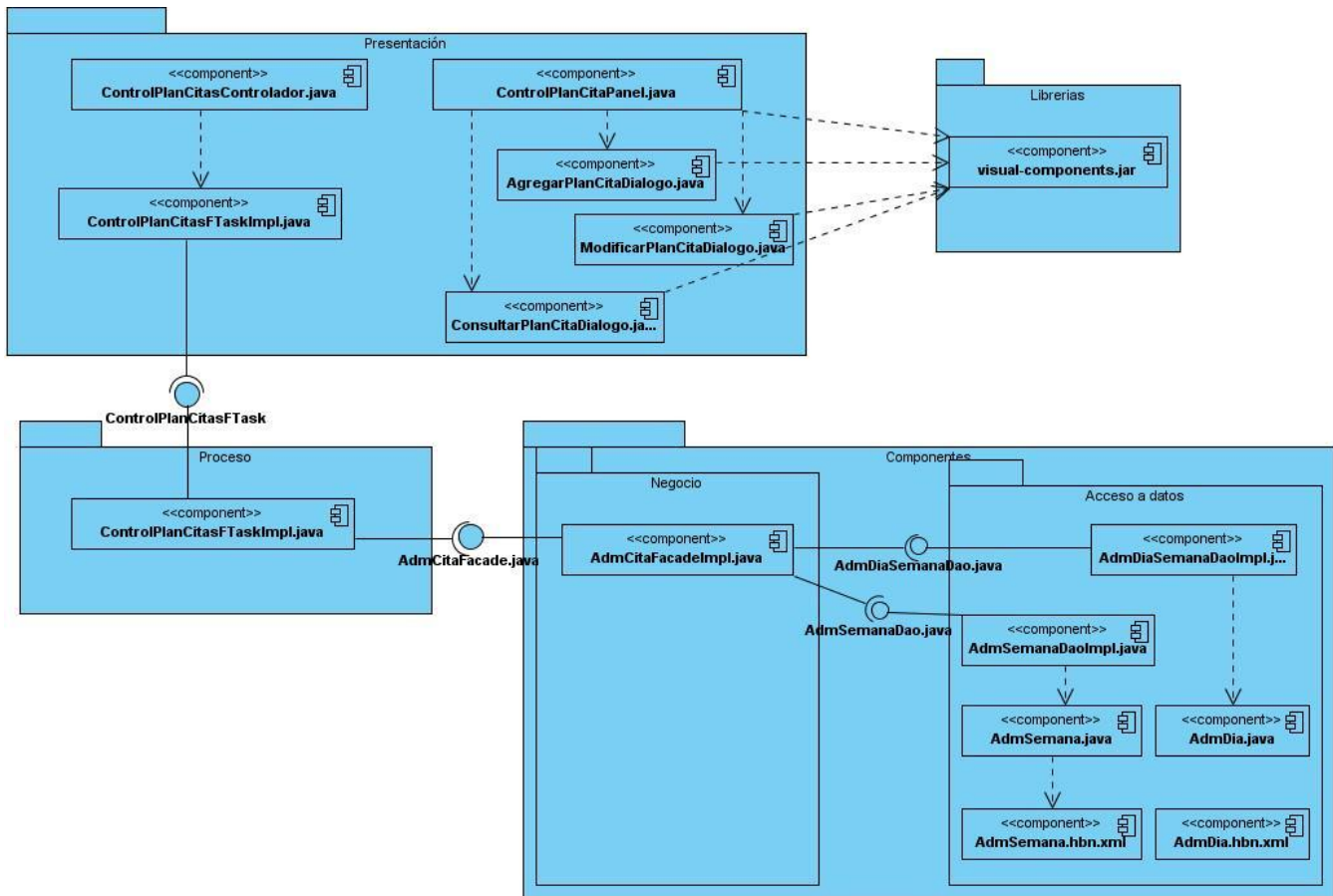


Figura 36 Diagrama de componentes para la funcionalidad gestión de planificación de citas.

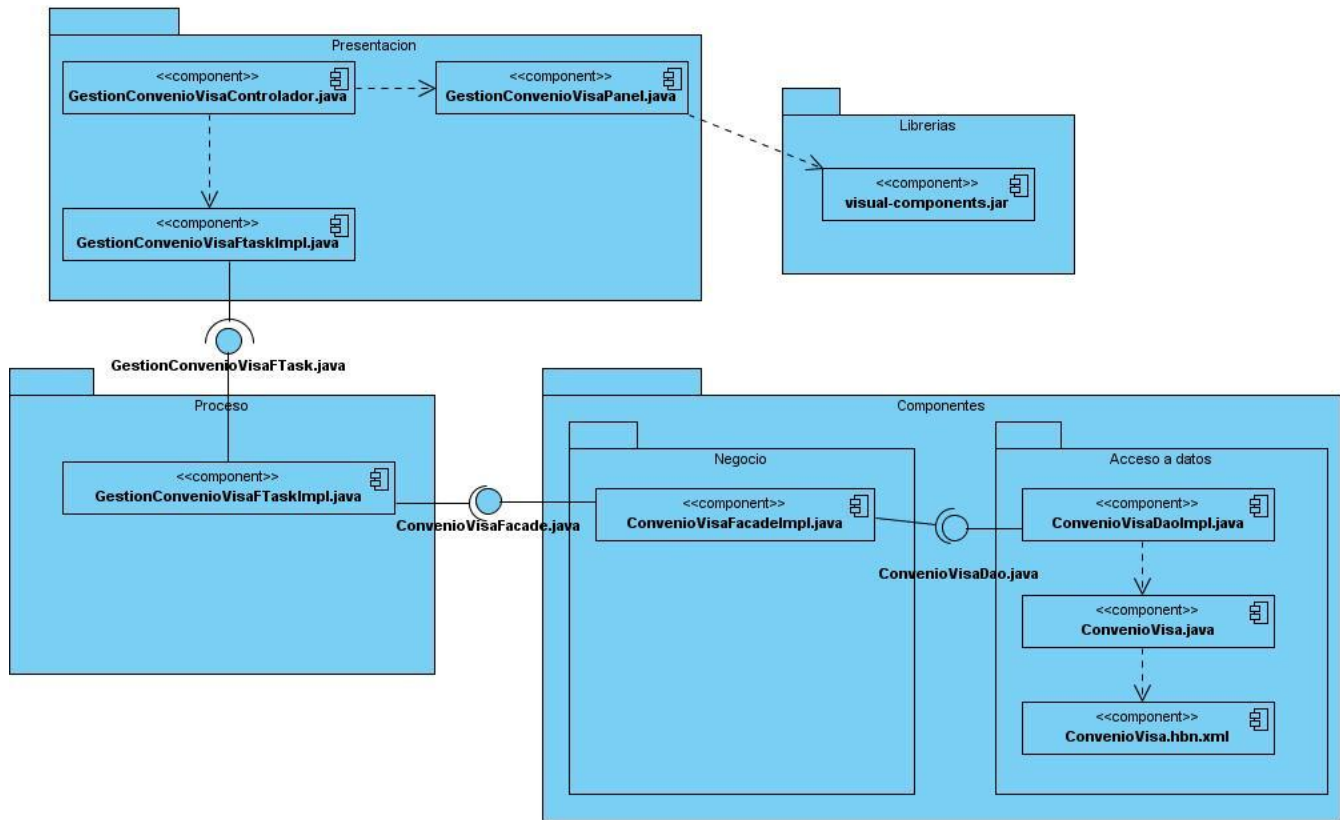


Figura 37 Diagrama de componentes para la funcionalidad definir convenio de supresión de visa

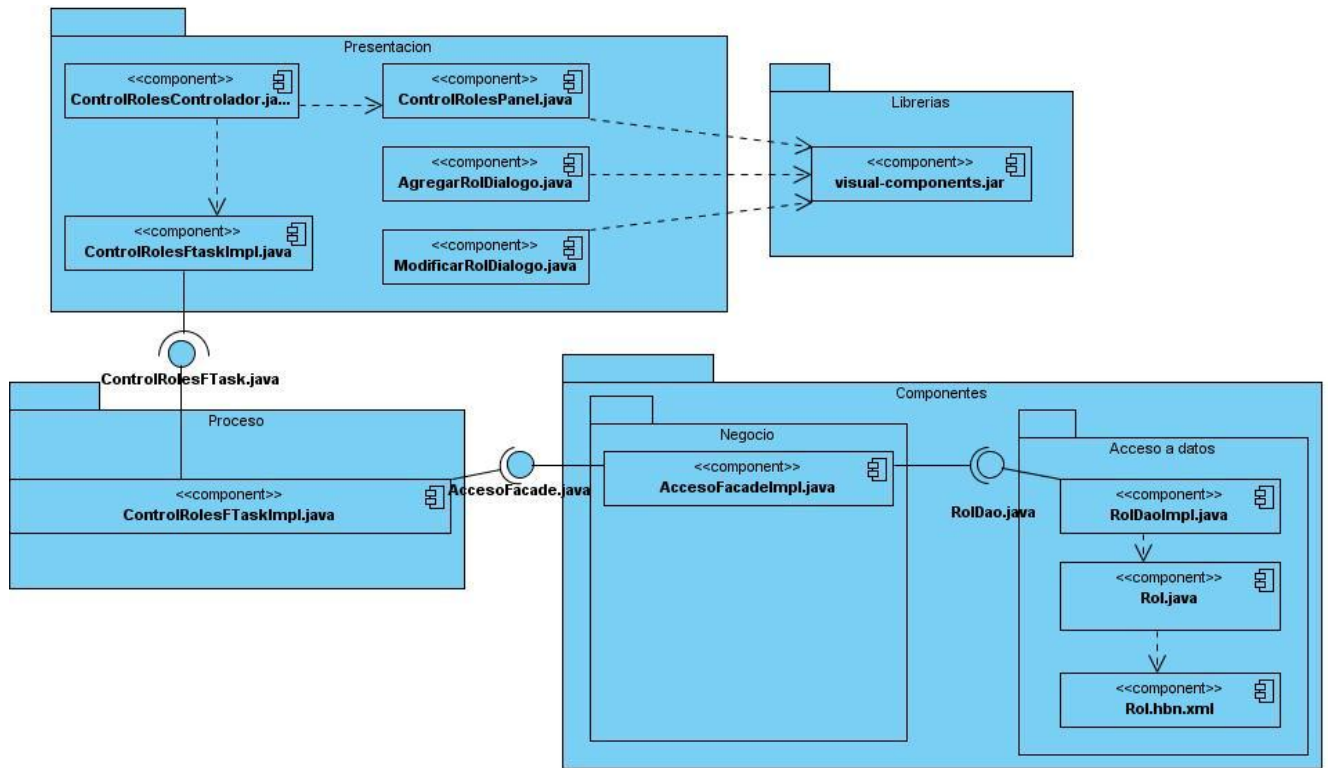


Figura 38 Diagrama de componentes para la funcionalidad Gestionar roles.

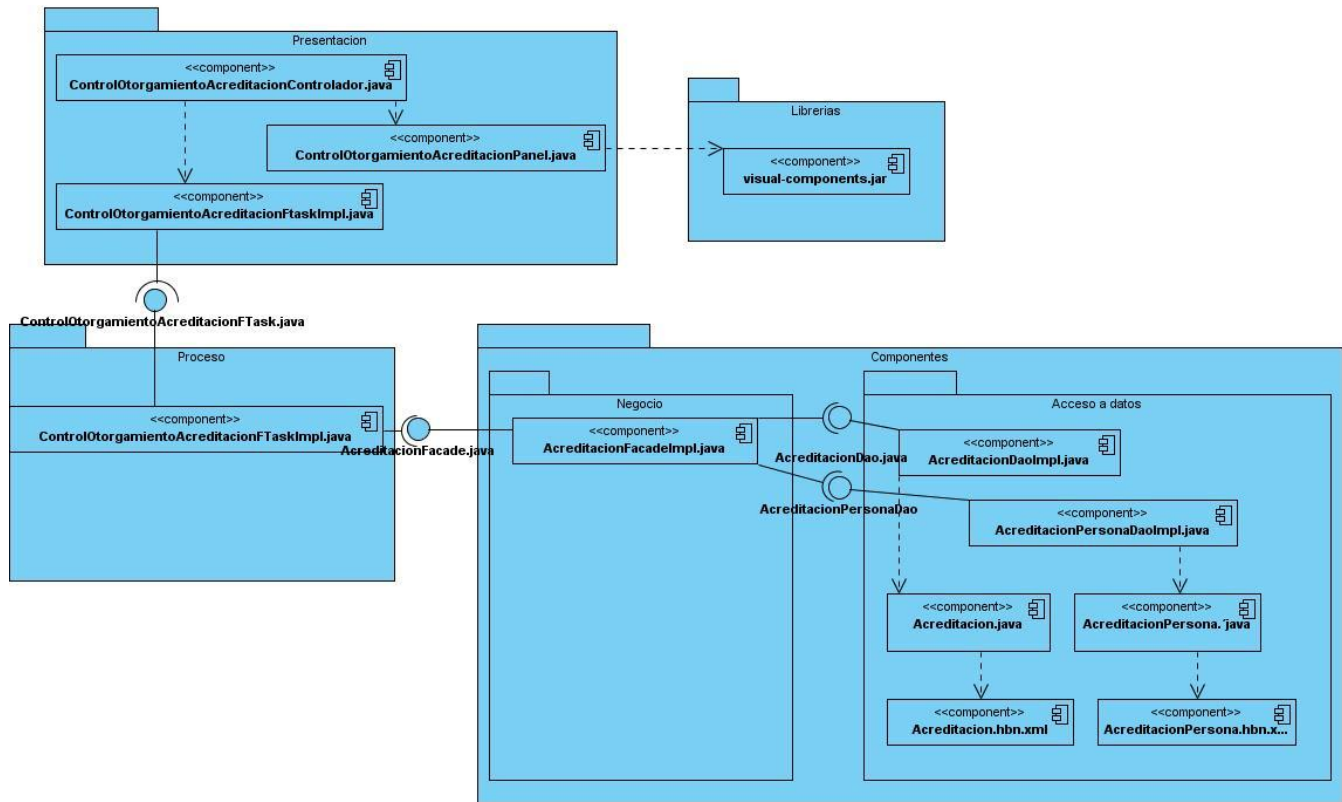


Figura 39 Diagrama de componentes para la funcionalidad definir otorgamiento de acreditación.

Anexo 6 Plantillas de pruebas de caja negra

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Nombre rol	Campo de Texto	No	Debe introducir un texto.
[2]	Módulos	Campo de selección	No	Debe seleccionar una opción.
[3]	Funcionalidades	Campo de selección	No	Debe seleccionar una opción
[4]	Rol	Campo de selección	No	Debe seleccionar una opción

Tabla 20 Descripción de las variables. Caso de prueba: Gestionar rol.

Flujos	Var 1. Nombre del Rol	Var 2. Módulos	Var 3. Funcionalidades	Var 4. Rol	Respuesta del Sistema	Resultado de la Prueba	Observaciones
Crear rol.	V	V	V	N	Crea un nuevo rol con los datos correspondientes a los introducidos.	Satisfactorio	
	I	N	N	N	Muestra el mensaje: "Nombre del rol no válido".	Satisfactorio	



Flujos	Var 1. Nombre del Rol	Var 2. Módulos	Var 3. Funcionalidades	Var 4. Rol	Respuesta del Sistema	Resultado de la Prueba	Observaciones
	N A	I A	N A	N A	Muestra el mensaje: "Debe seleccionar al menos un módulo".	Satisfactorio	
	N A	N A	I A	N A	Muestra el mensaje: "Debe seleccionar al menos una funcionalidad".	Satisfactorio	
Modificar rol.	N A	V A	V A	V A	Modifica los datos del rol satisfactoriamente	Satisfactorio	
	N A	N A	N A	I A	Muestra el mensaje: "Debe seleccionar un rol".	Satisfactorio	
Eliminar rol.	N A	N A	N A	V A	El sistema elimina el rol seleccionado satisfactoriamente.	Satisfactorio	
	N A	N A	N A	I A	Muestra el mensaje: "Debe seleccionar un rol a eliminar".	Satisfactorio	

Tabla 21 Caso de prueba Gestionar rol.