

© Ismael Campanario Cabrera

Reservados todos los derechos. Está prohibida la reproducción total o parcial de esta obra por cualquier medio o procedimiento, incluidos la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler o préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

# **Registro telemático de representantes legales**

---

**Ismael Campanario Cabrera**

Ingeniería Técnica en Informática de Gestión  
Consultor: Óscar Escudero Sánchez  
Curso 2011-2012 / 2º semestre  
18 de junio de 2012

---

# Índice de contenidos

ÍNDICE DE CONTENIDOS.....	- 2 -
ÍNDICE DE FIGURAS.....	- 5 -
INTRODUCCIÓN .....	- 6 -
<i>Justificación del proyecto</i> .....	- 6 -
<i>Objetivos del proyecto</i> .....	- 6 -
<i>Enfoque metodológico</i> .....	- 7 -
<i>Planificación del proyecto</i> .....	- 7 -
<i>Diagrama de Gantt</i> .....	- 9 -
<i>Productos obtenidos</i> .....	- 10 -
<i>Descripción del resto de capítulos</i> .....	- 10 -
ANÁLISIS DEL SISTEMA .....	- 11 -
<i>Especificación de requisitos</i> .....	- 11 -
<i>Requisitos funcionales</i> .....	- 11 -
<i>Requisitos no funcionales</i> .....	- 15 -
<i>Requisitos de almacenamiento de información</i> .....	- 16 -
<i>Especificación del sistema</i> .....	- 18 -
<i>Definición de actores y participantes</i> .....	- 18 -
<i>Casos de uso</i> .....	- 22 -
DISEÑO DEL SISTEMA .....	- 30 -
<i>Diseño de la arquitectura</i> .....	- 30 -
<i>Diagramas de clases</i> .....	- 32 -
<i>Clases de dominio</i> .....	- 32 -
<i>Clases de persistencia</i> .....	- 34 -
<i>Componente Authenticator</i> .....	- 36 -
<i>Componente CitizensManager</i> .....	- 37 -
<i>Componente DocumentsManager</i> .....	- 39 -
<i>Componente Signer</i> .....	- 41 -
<i>Diagramas de secuencias</i> .....	- 43 -
<i>Autenticación del ciudadano</i> .....	- 43 -
<i>Nuevo apoderamiento</i> .....	- 44 -

<i>Diagrama de estados</i> .....	- 45 -
<i>Diagrama de estados (apoderado)</i> .....	- 46 -
<i>Diagrama de estados (poderdante)</i> .....	- 46 -
<i>Diagrama de estados (sistema)</i> .....	- 47 -
<i>Diseño del modelo de datos</i> .....	- 48 -
<i>Nomenclatura de tablas y columnas</i> .....	- 48 -
<i>Auditoría</i> .....	- 49 -
<i>Prototipo interfaz gráfica</i> .....	- 50 -
<i>Autenticación de usuario</i> .....	- 50 -
<i>Bandeja de apoderamientos</i> .....	- 51 -
<i>Detalle del apoderamiento</i> .....	- 52 -
<i>Nuevo apoderamiento</i> .....	- 53 -
<i>Firma digital del apoderamiento</i> .....	- 54 -
<b>IMPLEMENTACIÓN</b> .....	- 55 -
<i>Requisitos técnicos y tecnologías usadas</i> .....	- 55 -
<i>Estructura del proyecto</i> .....	- 56 -
<i>Código fuente y recursos</i> .....	- 56 -
<i>Dependencias del proyecto</i> .....	- 56 -
<i>Librerías necesarias</i> .....	- 57 -
<i>Configuración mediante maven</i> .....	- 58 -
<i>Cliente de firma digital</i> .....	- 58 -
<i>Instalación y configuración</i> .....	- 60 -
<i>Instalación del modelo de datos</i> .....	- 60 -
<i>Generación de documentos</i> .....	- 61 -
<i>Despliegue en servidor de aplicaciones</i> .....	- 61 -
<i>Configuración de la aplicación</i> .....	- 62 -
<i>Datasource de conexión a la base de datos</i> .....	- 62 -
<i>Configuración general</i> .....	- 62 -
<i>Etiquetas para la internacionalización</i> .....	- 63 -
<i>Trazabilidad del sistema</i> .....	- 63 -
<i>Configuración de componentes</i> .....	- 63 -
<b>CONCLUSIONES</b> .....	- 64 -
<b>GLOSARIO</b> .....	- 65 -
<b>BIBLIOGRAFÍA</b> .....	- 67 -
<b>ANEXO I. MANUAL DE USUARIO</b> .....	- 68 -
<i>Requisitos del cliente</i> .....	- 68 -
<i>Acceso al sistema</i> .....	- 68 -

<i>Creación de un apoderamiento .....</i>	<i>- 69 -</i>
<i>Acciones sobre los apoderamientos.....</i>	<i>- 72 -</i>
<i>Revocación de apoderamientos.....</i>	<i>- 72 -</i>
<i>Confirmación o rechazo del poder otorgado .....</i>	<i>- 73 -</i>
<i>Renuncia de poderes .....</i>	<i>- 74 -</i>
<i>ANEXO II. PRUEBAS FUNCIONALES .....</i>	<i>- 75 -</i>
<i>ANEXO III. TRAZABILIDAD ENTRE ENTIDADES .....</i>	<i>- 79 -</i>
<i>Trazabilidad entre requisitos funcionales entre sí .....</i>	<i>- 79 -</i>
<i>Trazabilidad entre requisitos funcionales y de almacenamiento .....</i>	<i>- 79 -</i>
<i>Trazabilidad entre requisitos funcionales y casos de uso.....</i>	<i>- 80 -</i>
<i>Trazabilidad entre casos de uso y clases.....</i>	<i>- 80 -</i>
<i>Trazabilidad entre requisitos de almacenamiento y clases.....</i>	<i>- 81 -</i>

## Índice de figuras

Figura 1 - Diagrama de Gantt .....	- 9 -
Figura 2 – Requisitos funcionales.....	- 11 -
Figura 3 – Requisitos no funcionales.....	- 15 -
Figura 4 – Requisitos de almacenamiento .....	- 16 -
Figura 5 – Catálogo de actores.....	- 18 -
Figura 6 – Catálogo de actores de negocio .....	- 20 -
Figura 7 – Casos de uso.....	- 20 -
Figura 8 – Diagrama de arquitectura.....	- 30 -
Figura 9 – Diagrama de clases de dominio.....	- 30 -
Figura 10 – Diagrama de clases de persistencia.....	- 34 -
Figura 11 – Diagrama de clases Authenticator .....	- 36 -
Figura 12 – Diagrama de clases CitizensManager.....	- 37 -
Figura 13 – Diagrama de clases DocumentsManager .....	- 39 -
Figura 14 – Diagrama de clases Signer .....	- 41 -
Figura 15 – Diagrama de secuencias Autenticación de usuario .....	- 41 -
Figura 16 – Diagrama de secuencias Nuevo apoderamiento.....	- 41 -
Figura 17 – Diagrama de estados.....	- 45 -
Figura 18 – Diagrama de estados (apoderado).....	- 46 -
Figura 19 – Diagrama de estados (poderdante).....	- 46 -
Figura 20 – Diagrama de estados (sistema).....	- 47 -
Figura 21 – Diagrama Modelo de datos.....	- 48 -
Figura 22 – Interfaz gráfica Autenticación de usuario .....	- 50 -
Figura 23 – Interfaz gráfica Bandeja de apoderamientos.....	- 51 -
Figura 24 – Interfaz gráfica Detalle del apoderamiento.....	- 52 -
Figura 25 – Interfaz gráfica Nuevo apoderamiento.....	- 53 -
Figura 26 – Interfaz gráfica Firma digital del apoderamiento .....	- 54 -
Figura 27 - Librerías del proyecto .....	- 58 -
Figura 28 - clientsigner.jar.....	- 59 -
Figura 29 - Modelo de datos.....	- 60 -
Figura 30 - Generación PDF.....	- 61 -
Figura 31 - Acceso al sistema .....	- 68 -
Figura 32 - Lista de certificados.....	- 69 -
Figura 33 - Bandeja de apoderamientos.....	- 69 -
Figura 34 - Icono nuevo apoderamiento .....	- 69 -
Figura 35 - Consulta de apoderado.....	- 70 -
Figura 36 - Nuevo apoderamiento.....	- 70 -
Figura 37 - Firmar apoderamiento.....	- 71 -
Figura 38 - Apoderamiento firmado .....	- 71 -
Figura 39 - Informe de firma digital.....	- 72 -
Figura 40 - Revocar apoderamiento .....	- 72 -
Figura 41 - Confirmación revocación apoderamiento.....	- 73 -
Figura 42 - Confirmación o rechazo del poder otorgado.....	- 73 -
Figura 43 - Renuncia de poderes.....	- 74 -

## Introducción

---

El presente documento recoge la memoria del Trabajo Final de Carrera, en adelante TFC, cuyo objetivo es consolidar los conocimientos adquiridos a lo largo de estos años mediante el estudio de las asignaturas que componen la Ingeniería Técnica en Informática de Gestión.

En este documento se describe desde la planificación del proyecto, las herramientas y tecnologías utilizadas para llevarlo a cabo, así como la funcionalidad y el comportamiento de la aplicación para que el usuario tenga una visión general de cómo será y qué podrá realizar con el aplicativo.

## Justificación del proyecto

---

La gestión telemática del registro de representantes legales o apoderados facilita la labor a los organismos u organizaciones que lo implanten y supone un paso más hacia la Administración Electrónica.

Hoy en día cualquier organismo público o privado que permita realizar trámites, lleva a cabo la gestión de las representaciones de forma presencial obligando a los ciudadanos (tanto al poderdante como al apoderado) a personarse físicamente para realizar el trámite. El desarrollo de esta aplicación web permitirá realizar esta gestión desde Internet mediante el uso del certificado digital de usuario.

Debido a la gran variedad de clientes que podrán implantar la aplicación, ésta será configurable y orientada a componentes para dar soporte a las distintas estructuras en las que se implante. Además soportará distintos idiomas.

## Objetivos del proyecto

---

El objetivo principal del TFC es adquirir nuevos conocimientos y experiencia con la tecnología JEE, además de comprender el ciclo de vida del desarrollo de software realizando todas las fases necesarias hasta conseguir el producto final.

El desarrollo del TFC además permite afianzar conocimientos de distintas tecnologías que se he ido usando durante varios años en mi labor profesional como programador así como los conocimientos adquiridos durante la Ingeniería Técnica en Informática de Gestión.

Concretando en la aplicación web, el objetivo de la misma es el registro y la gestión de las representaciones legales o apoderamientos entre ciudadanos para unos trámites dentro de un organismo u organización.

La implantación de esta aplicación web en un organismo permite aumentar la eficiencia y eficacia y se convertiría en una pieza más de la Administración Electrónica del mismo. Además facilitar este trámite por Internet a los ciudadanos permitirá ahorrar costes a ambas partes.

## Enfoque metodológico

Para llevar a cabo la elaboración del TFC he seguido las fases de cualquier proyecto de desarrollo de software. Para ello el punto de partida es realizar una planificación del proyecto y organizar el tiempo, recursos y el alcance del mismo.

Una vez obtenido el calendario con los hitos más importantes que cumplir, comencé el análisis y diseño del proyecto del cuál obtuve la especificación de requisitos y un diseño previo de qué haría la aplicación y cómo sería. Además durante este hito tuve que recopilar información de la tecnología JEE y los distintos frameworks existentes para desarrollar el proyecto.

El hito más importante, en cuanto a tiempo de dedicación ha sido la codificación o implementación de los requisitos obtenidos en las fases de análisis y diseño. Durante la realización de este hito es dónde se obtiene mayor experiencia que posteriormente será muy útil en nuestra vida laboral.

Finalizada la codificación, el próximo hito era el desarrollo de la documentación, que incluye esta memoria y la presentación en diapositivas del proyecto.

## Planificación del proyecto

En la siguiente tabla se recogen las tareas y la duración prevista de las mismas para el desarrollo del proyecto.

<i>Tarea</i>	<i>Días</i>	<i>Inicio</i>	<i>Fin</i>
<b>Análisis previo</b>	<b>13</b>	<b>02/03/2012</b>	<b>14/03/2012</b>
<i>Lectura documentación y Plan Docente</i>	<i>2</i>	<i>02/03/2012</i>	<i>03/03/2012</i>
<i>Estudio tecnologías a usar</i>	<i>3</i>	<i>04/03/2012</i>	<i>05/03/2012</i>
<i>Creación Plan de Trabajo</i>	<i>8</i>	<i>06/03/2012</i>	<i>14/03/2012</i>
<i>Entrega Plan de Trabajo (PEC 1)</i>	<i>0</i>	<i>14/03/2012</i>	<i>14/03/2012</i>
<b>Análisis de requisitos</b>	<b>19</b>	<b>15/03/2012</b>	<b>03/04/2012</b>
<i>Análisis de requisitos funcionales</i>	<i>8</i>	<i>15/03/2012</i>	<i>22/03/2012</i>
<i>Análisis de requisitos no funcionales</i>	<i>8</i>	<i>15/03/2012</i>	<i>22/03/2012</i>
<i>Análisis de requisitos de almacenamiento</i>	<i>8</i>	<i>15/03/2012</i>	<i>22/03/2012</i>
<i>Casos de uso</i>	<i>7</i>	<i>23/03/2012</i>	<i>29/03/2012</i>
<i>Elaboración documento de Análisis</i>	<i>4</i>	<i>30/03/2012</i>	<i>03/04/2012</i>



<b>Diseño</b>	<b>16</b>	<b>04/04/2012</b>	<b>19/04/2012</b>
<i>Requisitos de diseño</i>	3	04/04/2012	06/04/2012
<i>Diseño de arquitectura</i>	2	07/04/2012	08/04/2012
<i>Montar entorno de trabajo</i>	4	07/04/2012	10/04/2012
<i>Diseño de prototipo gráfico</i>	5	11/04/2012	15/04/2012
<i>Elaboración documento de Diseño</i>	4	16/04/2012	19/04/2012
<i>Entrega PEC 2</i>	0	19/04/2012	19/04/2012
<b>Implementación</b>	<b>46</b>	<b>20/04/2012</b>	<b>04/06/2012</b>
<i>Codificación</i>	39	20/04/2012	28/05/2012
<i>Elaboración de manual de instalación</i>	3	29/05/2012	31/05/2012
<i>Elaboración de manual de usuario</i>	3	01/06/2012	03/06/2012
<i>Empaquetar entregable</i>	1	04/06/2012	04/06/2012
<i>Entrega PEC 3</i>	0	04/06/2012	04/06/2012
<b>Memoria y presentación final</b>	<b>14</b>	<b>05/06/2012</b>	<b>18/06/2012</b>
<i>Pruebas de integración</i>	2	05/06/2012	06/06/2012
<i>Elaboración de la memoria</i>	10	07/06/2012	16/06/2012
<i>Elaboración de la presentación</i>	4	15/06/2012	18/06/2012
<i>Entrega memoria, presentación y producto final</i>	0	18/06/2012	18/06/2012

A continuación se muestra la planificación del proyecto mediante un diagrama de Gantt que permite ver la duración de las tareas de forma gráfica a lo largo del tiempo.

# Diagrama de Gantt

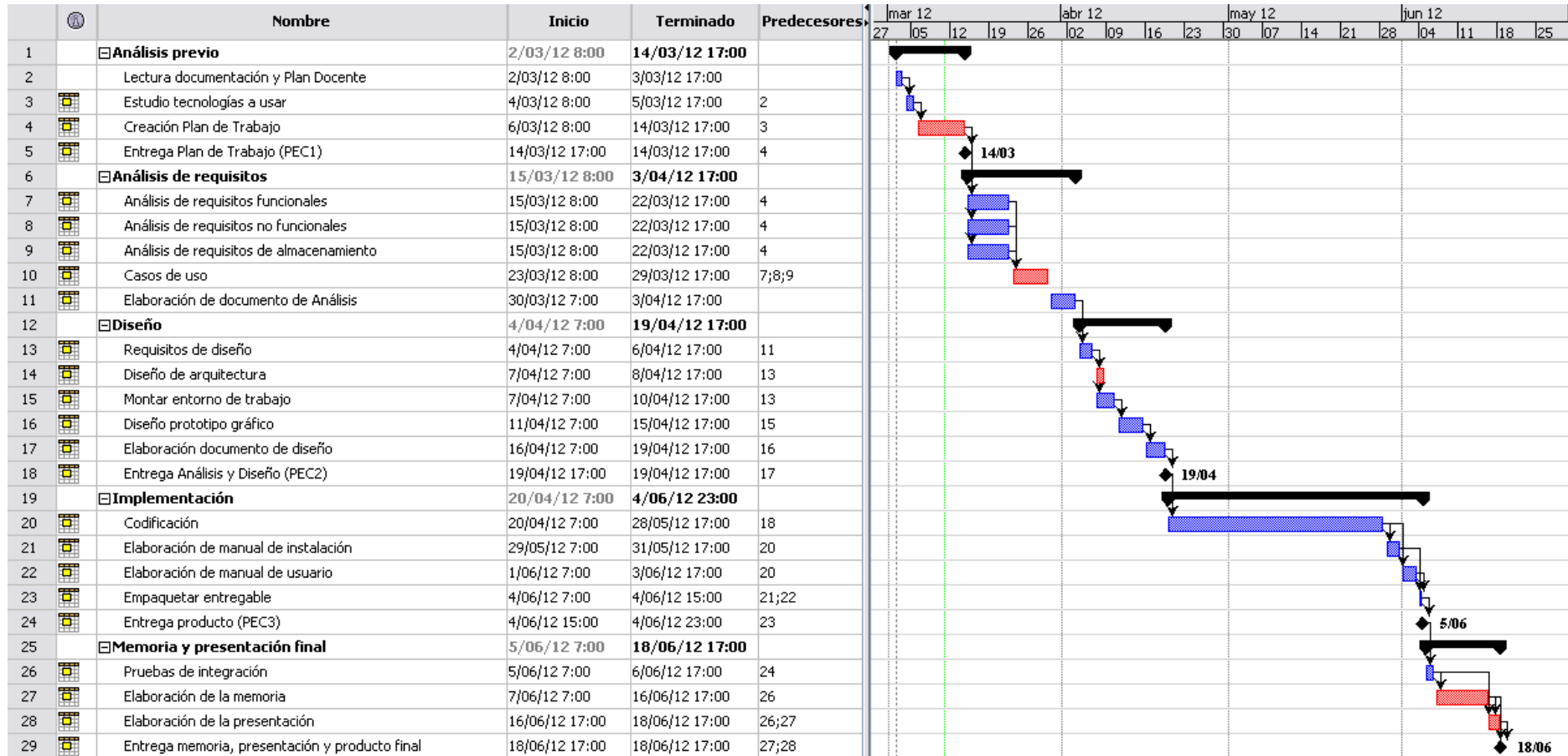


Figura 1 - Diagrama de Gantt

# Productos obtenidos

---

Una vez finalizado el TFC, los productos obtenidos son:

- Memoria del TFC. Documento que recoge los aspectos más relevantes del proyecto.
- Presentación virtual. Presentación que sintetiza en diapositivas el trabajo realizado.
- Software desarrollado. Incluye el código fuente, scripts de base de datos, javadoc, war, en definitiva todo lo necesario para desarrollar el proyecto.

# Descripción del resto de capítulos

---

Además de los capítulos anteriores, la memoria dispone de los siguientes capítulos:

- **Especificación de requisitos.** Recoge el catálogo de requisitos funcionales, no funcionales y de almacenamiento del sistema.
- **Especificación del sistema.** Recoge el catálogo de actores y participantes en el sistema, así como los casos de uso.
- **Diseño de la arquitectura.** Diagrama del diseño de la arquitectura de la aplicación.
- **Diagramas de clases.** Recoge los distintos diagramas de clases de la aplicación.
- **Diagramas de secuencias.** Muestra la funcionalidad del sistema mediante los diagramas de secuencias.
- **Diagrama de estados.** Recoge los distintos estados por los que puede pasar un apoderamiento. Además permite conocer los cambios de estados permitidos para cada actor.
- **Diseño del modelo de datos.** Recoge el diseño del modelo de datos que dará soporte a la persistencia de la información de la aplicación.
- **Prototipo de la interfaz gráfica.** Recoge el diseño inicial de las pantallas de la aplicación.
- **Requisitos técnicos y tecnologías usadas.** Indica los requisitos técnicos, tecnologías así como las herramientas usadas para llevar a cabo el desarrollo del proyecto.
- **Estructura del proyecto.** Indica los distintos recursos que componen el proyecto.
- **Instalación y configuración.** Recoge el proceso de instalación y posterior configuración de los distintos componentes.
- **Anexo I. Manual de usuario.** Permite a los usuarios obtener ayuda de la distinta funcionalidad que ofrece la aplicación.
- **Anexo II. Pruebas funcionales.** Recoge un plan de pruebas de la funcionalidad más relevante de la aplicación.
- **Anexo III. Trazabilidad entre entidades.** Recoge la trazabilidad entre las distintas entidades para garantizar que todos los objetivos del proyecto quedan recogidos en el catálogo de requisitos.

## Análisis del sistema

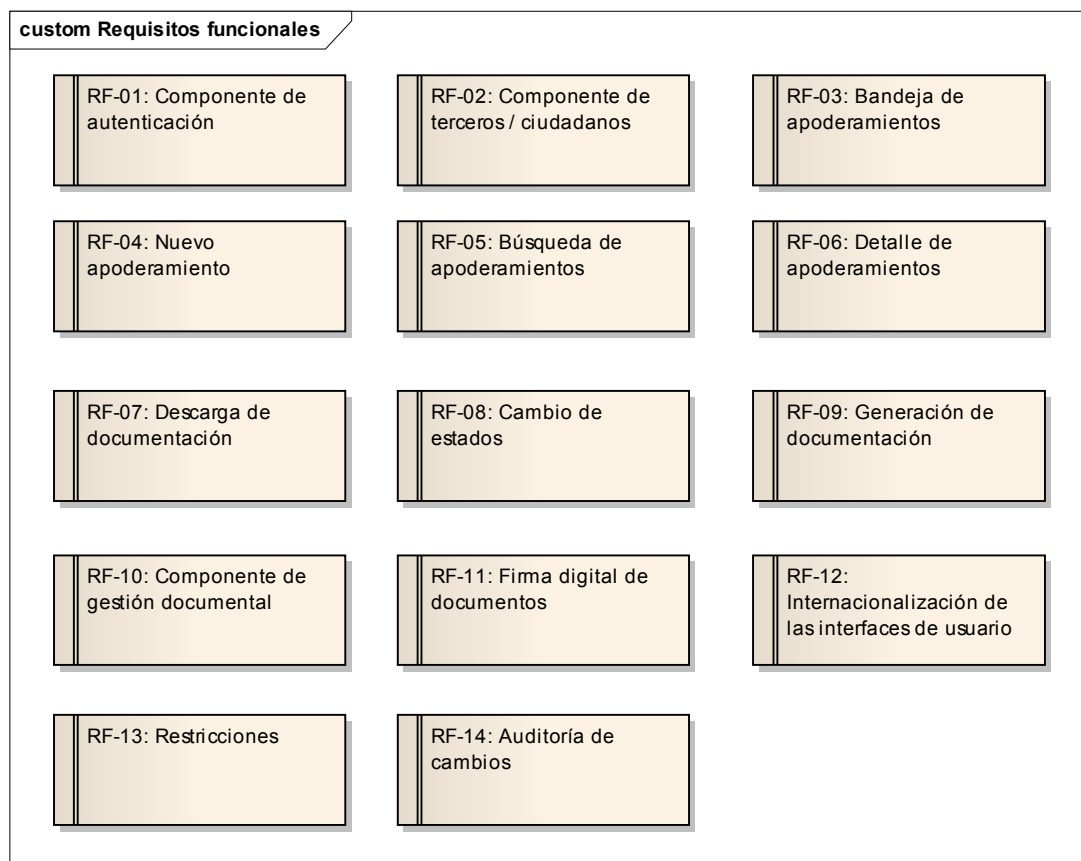
A continuación se recoge el análisis del sistema de información que se va a desarrollar. Entre otros se recogen los requerimientos funcionales, no funcionales y de almacenamiento que deberá cumplir el aplicativo.

Además se realiza el estudio de los casos de uso más representativos del sistema.

## Especificación de requisitos

El catálogo de requisitos recoge la funcionalidad y características que dispondrá el sistema que se va a desarrollar. Entre ellos se recogen los requisitos funcionales, no funcionales y de almacenamiento.

### Requisitos funcionales



**Figura 2 – Requisitos funcionales**

## RF-01

<b>Código:</b>	RF-01
<b>Nombre:</b>	Componente de autenticación
<b>Descripción:</b>	La autenticación se llevará a cabo mediante un componente que seguirá el patrón <i>Factory</i> para crear la implementación según la configuración. La implementación por defecto que se desarrollará será mediante el uso del certificado digital de usuario que disponga el ciudadano y se validará usando el cliente de firma digital de @firma.

## RF-02

<b>Código:</b>	RF-02
<b>Nombre:</b>	Componente de terceros / ciudadanos
<b>Descripción:</b>	Se desarrollará un componente que seguirá el patrón <i>Factory</i> para que permita cambiar mediante configuración la implementación a usar. Este componente permitirá obtener los datos de los ciudadanos desde distintas fuentes de datos. La implementación por defecto que se desarrollará obtendrá los mismos de una base de datos.

## RF-03

<b>Código:</b>	RF-03
<b>Nombre:</b>	Bandeja de apoderamientos
<b>Descripción:</b>	El ciudadano tendrá acceso a una bandeja de apoderamientos en los que está involucrado, ya sea como poderdante (ha dado el poder) o como apoderado (ha recibido el poder).

## RF-04

<b>Código:</b>	RF-04
<b>Nombre:</b>	Nuevo apoderamiento
<b>Descripción:</b>	El ciudadano podrá crear un nuevo registro de apoderamiento mediante el cuál dará el poder a otro ciudadano para un procedimiento concreto para que realice los trámites en su nombre.

## RF-05

<b>Código:</b>	RF-05
<b>Nombre:</b>	Búsqueda de apoderamientos
<b>Descripción:</b>	El sistema dispondrá de una utilidad de búsqueda entre los apoderamientos que tiene el ciudadano.

## RF-06

<b>Código:</b>	RF-06
<b>Nombre:</b>	Detalle de apoderamientos
<b>Descripción:</b>	El ciudadano podrá consultar el detalle de cualquier registro de apoderamiento con el que esté vinculado.

## RF-07

<b>Código:</b>	RF-07
<b>Nombre:</b>	Descarga de documentación
<b>Descripción:</b>	El ciudadano podrá descargar la documentación asociada a un registro de apoderamiento desde el detalle del mismo.

## RF-08

<b>Código:</b>	RF-08
<b>Nombre:</b>	Cambio de estados
<b>Descripción:</b>	Se podrán realizar los cambios de estado del apoderamiento en función de los estados existentes. Para más detalles consultar el diagrama de estados y los cambios posibles entre estos.

## RF-09

<b>Código:</b>	RF-09
<b>Nombre:</b>	Generación de documentación
<b>Descripción:</b>	A partir de unas plantillas OpenOffice con variables se generarán los PDF relacionados con los apoderamientos. Esta generación se llevará a cabo mediante el uso del servicio de OpenOffice en servidor.

## RF-10

<b>Código:</b>	RF-10
<b>Nombre:</b>	Componente de gestión documental
<b>Descripción:</b>	Mediante un componente implementado siguiendo el patrón <i>Factory</i> se podrá definir si los documentos generados por la aplicación son almacenados en algún gestor documental externo, por ejemplo en <i>Alfresco</i> . Como ejemplo se realizará una implementación que almacenará los documentos en el sistema de ficheros del Sistema Operativo para que se pueda comprobar la versatilidad que tiene.

## RF-11

<b>Código:</b>	RF-11
<b>Nombre:</b>	Firma digital de documentos
<b>Descripción:</b>	La confirmación de la creación del apoderamiento, así como algunos cambios de estados se validarán con la firma digital del ciudadano mediante el uso del certificado digital de usuario y el cliente de firma digital de @firma.

## RF-12

<b>Código:</b>	RF-12
<b>Nombre:</b>	Internacionalización de las interfaces de usuario
<b>Descripción:</b>	La aplicación se desarrollará siguiendo las normas de internacionalización para que las interfaces de usuario aparezcan en el idioma seleccionado por el usuario.

## RF-13

<b>Código:</b>	RF-13
<b>Nombre:</b>	Restricciones
<b>Descripción:</b>	<p>Algunas de las restricciones que se deben tener en cuenta son:</p> <ul style="list-style-type: none"> <li>• El inicio de un nuevo registro de apoderamiento mediante medios telemáticos sólo la puede iniciar el poderdante, pero nunca el apoderado.</li> <li>• Si el tipo de otorgamiento es Global o sólo Notificaciones, es decir que tenga algo relacionado con notificaciones, se omite el valor de confirmar apoderamiento ya que pasa directamente al estado Confirmado.</li> </ul>

## RF-14

<b>Código:</b>	RF-14
<b>Nombre:</b>	Auditoría de cambios
<b>Descripción:</b>	Se debe llevar a cabo una auditoría de los datos que gestione el sistema.

## Requisitos no funcionales

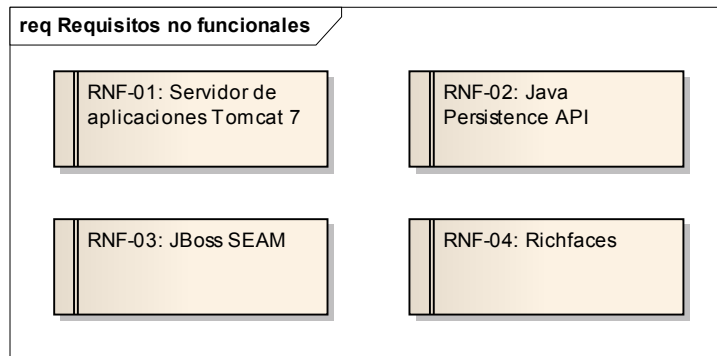


Figura 3 – Requisitos no funcionales

### RNF-01

<b>Código:</b>	RNF-01
<b>Nombre:</b>	Servidor de aplicaciones Tomcat 7
<b>Descripción:</b>	Una vez empaquetado el sistema será desplegado en el servidor de aplicaciones Apache Tomcat 7. El desarrollo de la aplicación seguirá los estándares JEE para que sea compatible con otros servidores de aplicaciones como JBoss, Glassfish, OC4J, etc.

### RNF-02

<b>Código:</b>	RNF-02
<b>Nombre:</b>	Java Persistence API
<b>Descripción:</b>	Las clases de dominio se desarrollarán siguiendo la especificación JPA mediante la implementación ofrecida por Hibernate.

### RNF-03

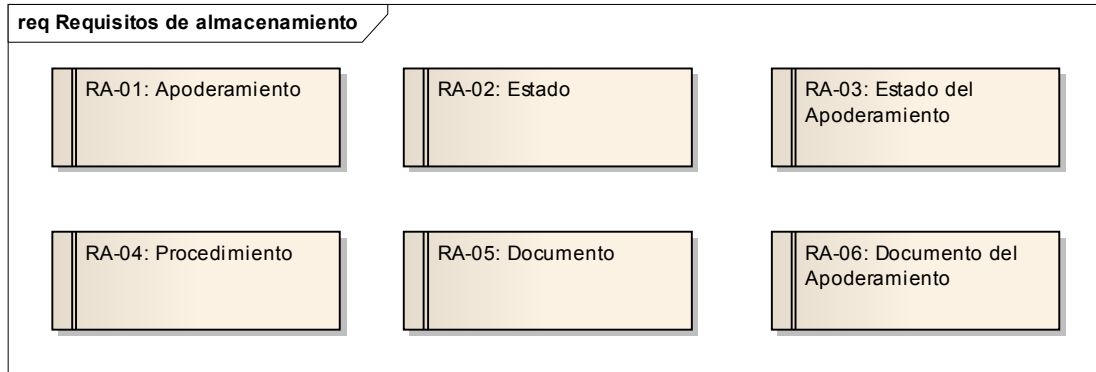
<b>Código:</b>	RNF-03
<b>Nombre:</b>	JBoss SEAM 2.1
<b>Descripción:</b>	La capa de negocio será gestionada con el Framework JBoss SEAM.

### RNF-04

<b>Código:</b>	RNF-04
<b>Nombre:</b>	RichFaces
<b>Descripción:</b>	Para la capa de presentación se hará uso de RichFaces 3.3 ya que permite utilizar componentes gráficos JSF con AJAX, facelets, etc.



## Requisitos de almacenamiento de información



**Figura 4 – Requisitos de almacenamiento**

### RA-01

<b>Código:</b>	RA-01
<b>Nombre:</b>	Apoderamiento
<b>Descripción:</b>	El sistema tendrá almacenada la información referente a los apoderamientos entre ciudadanos.

### RA-02

<b>Código:</b>	RA-02
<b>Nombre:</b>	Estado
<b>Descripción:</b>	El sistema recogerá los distintos estados por los que podrá pasar un apoderamiento. Estos estados serán Pendiente de confirmación, Confirmado, Rechazado, Renunciado y Revocado. Además existe un estado Expirado que se controlará mediante la fecha final de vigencia del apoderamiento.

### RA-03

<b>Código:</b>	RA-03
<b>Nombre:</b>	Estado del Apoderamiento
<b>Descripción:</b>	El sistema almacenará las relaciones entre un apoderamiento y los distintos estados por los que pueda pasar. Esto permitirá obtener un histórico de la evolución del apoderamiento por los distintos estados.

**RA-04**

<b>Código:</b>	RA-04
<b>Nombre:</b>	Procedimiento
<b>Descripción:</b>	El sistema tendrá almacenada la información referente a los procedimientos en los que se podrán crear apoderamientos.

**RA-05**

<b>Código:</b>	RA-05
<b>Nombre:</b>	Documento
<b>Descripción:</b>	El sistema almacenará los documentos aportados o generados para los apoderamientos.

**RA-06**

<b>Código:</b>	RA-06
<b>Nombre:</b>	Documento de Apoderamiento
<b>Descripción:</b>	Recogerá la asociación entre los documentos y el apoderamiento al que pertenece.

## Especificación del sistema

### Definición de actores y participantes

Se registran a continuación todos los actores y usuarios que de alguna manera interactuarán con el sistema, en un completo catálogo que incluya personas, entidades, otros sistemas, etc.

### Catálogo de actores

Conjunto de actores que de algún modo participan en el sistema, ya sea como usuario, administrador o cualquier otra interacción.

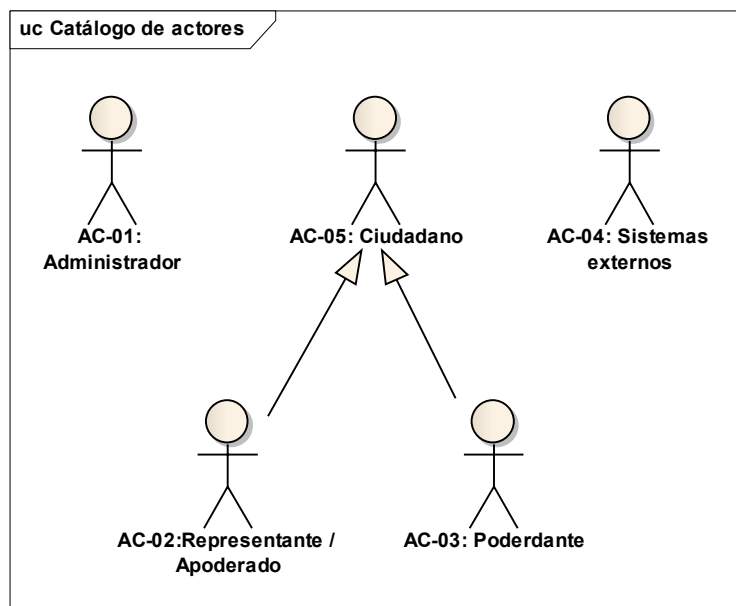


Figura 5 – Catálogo de actores

#### AC-01: Administrador

Administrador	
<b>Código:</b>	AC-01
<b>Descripción:</b>	El administrador del sistema será el encargado de mantener y administrar toda la información necesaria para el mantenimiento del mismo, por ejemplo la administración de tablas diccionario (estados apoderamiento, etc.).

**AC-02: Representante / Apoderado**

<b>Representante / Apoderado</b>	
<b>Código:</b>	AC-02
<b>Descripción:</b>	Ciudadano, usuario del sistema que tiene poderes para actuar en nombre de otro/s poderdante/s con permiso para gestionar sus actuaciones ante la organización.

**AC-03: Poderdante**

<b>Poderdante</b>	
<b>Código:</b>	AC-03
<b>Descripción:</b>	Ciudadano, usuario del sistema que mediante un poder otorga los permisos necesarios a otro ciudadano que actuará en su nombre ante la organización

**AC-04: Sistemas externos**

<b>Sistemas externos</b>	
<b>Código:</b>	AC-04
<b>Descripción:</b>	Terceras aplicaciones o sistemas que harán uso mediante consultas de los datos de apoderamientos para validar las representaciones.

**AC-05: Ciudadano**

<b>Ciudadano</b>	
<b>Código:</b>	AC-05
<b>Descripción:</b>	Generalización de los actores Poderdante y Apoderado.

## Catálogo de actores de negocio

Conjunto de actores relacionados con la lógica de negocio, como puede ser un departamento determinado, una organización, proveedores, otros sistemas, componentes, etc.

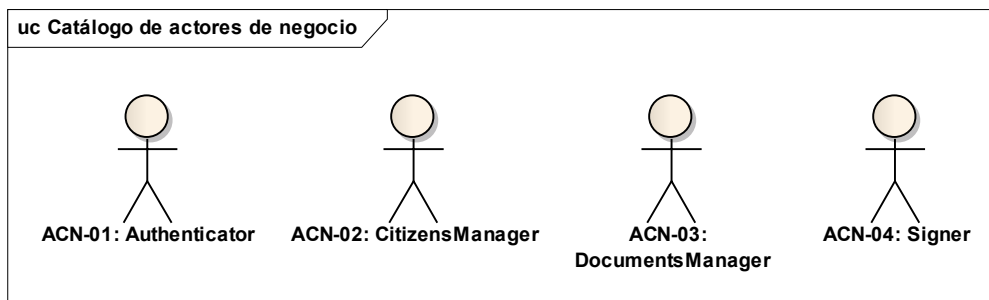


Figura 6 – Catálogo de actores de negocio

### ACN-01: Authenticator

Authenticator	
<b>Código:</b>	ACN-01
<b>Descripción:</b>	Componente que permite la autenticación del usuario en el sistema. Mediante el uso de patrones se permitirá la creación de distintas implementaciones para llevar a cabo la autenticación. La implementación por defecto que se desarrollará será mediante certificado digital usando el cliente de firma de @firma.

### ACN-02: CitizensManager

CitizensManager	
<b>Código:</b>	ACN-02
<b>Descripción:</b>	Componente que permite obtener los datos de los ciudadanos que serán los poderdantes y apoderados del sistema. Mediante el uso de patrones se permitirá la creación de distintas implementaciones para que se puedan obtener los datos de los ciudadanos de distintas fuentes, por ejemplo mediante un webservices del censo, una base de datos, etc.. La implementación por defecto que se desarrollará obtendrá los usuarios de una tabla del modelo de datos.

**ACN-03: DocumentsManager**

DocumentsManager	
<b>Código:</b>	ACN-03
<b>Descripción:</b>	Componente que permite gestionar el almacenamiento de documentos. Mediante el uso de patrones se permitirá la creación de distintas implementaciones que almacenen el documento en distintos sistemas, por ejemplo la integración con un gestor documental como Alfresco, Athento, etc. La implementación por defecto que se desarrollará gestionará los documentos en el sistema de ficheros del sistema operativo.

**ACN-04: Signer**

Signer	
<b>Código:</b>	ACN-04
<b>Descripción:</b>	Componente que permite realizar la firma digital de los documentos para dar validez a los apoderamientos. Mediante el uso de patrones se permitirá crear distintas implementaciones que lleven a cabo la firma digital. La implementación por defecto que se ofrecerá utilizará el certificado digital de usuario mediante el cliente de firma de @firma.

## Casos de uso

En el siguiente capítulo se recogen los diagramas de casos de uso más representativos del funcionamiento del sistema y la relación de cada uno de ellos con los actores y participantes en el sistema.

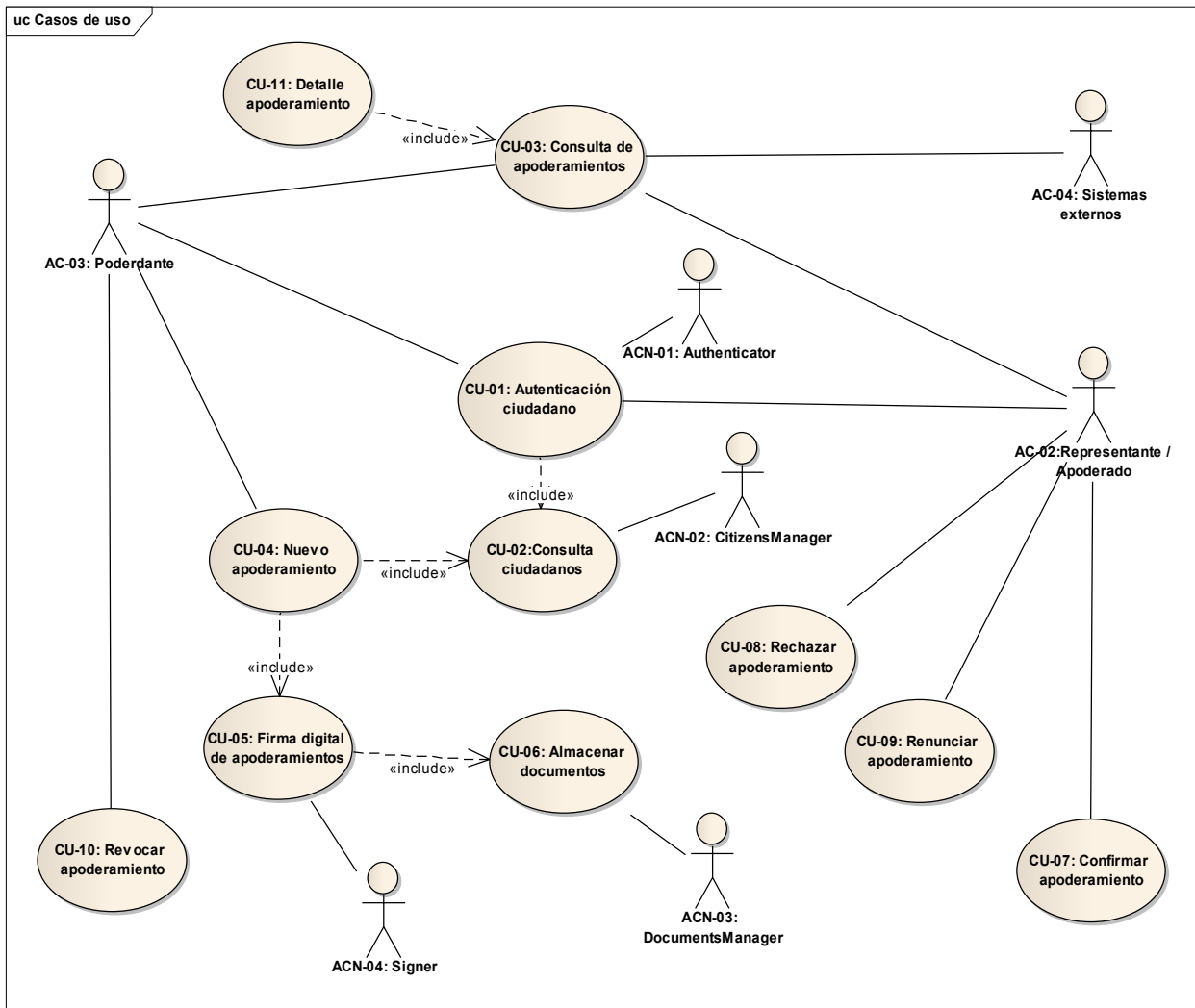


Figura 7 – Casos de uso

### CU-01

Autenticación ciudadano	
<b>Código:</b>	CU-01
<b>Descripción:</b>	El usuario se autentica en el sistema.
<b>Cubre requisitos:</b>	los RF-01: Componente de autenticación
<b>Actores:</b>	Poderdante, Apoderado y Authenticator
<b>Precondición:</b>	El ciudadano debe disponer de certificado digital de usuario.

Flujo normal	
<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de login del aplicativo.</li> <li>2. El sistema muestra los certificados digitales de usuario disponibles.</li> <li>3. El ciudadano selecciona su certificado digital de usuario.</li> <li>4. El sistema realiza el proceso de autenticación mediante el componente Authenticator.</li> <li>5. El usuario accede al sistema y se le muestra la bandeja de apoderamientos, en las que es poderdante o apoderado.</li> </ol>	
Flujo alternativo	
2.1 No existen certificados digitales disponibles 4.1 No se puede llevar a cabo el proceso de autenticación, ya sea porque el certificado está revocado o cualquier otro motivo. 5.1 La bandeja de apoderamientos se muestra vacía.	
<b>Postcondición:</b>	El usuario se encuentra autenticado en el sistema.

## CU-02

Consulta ciudadanos	
<b>Código:</b>	CU-02
<b>Descripción:</b>	Se obtienen los datos del ciudadano
<b>Cubre los requisitos:</b>	RF-02: Componente de terceros / ciudadanos
<b>Actores:</b>	CitizensManager
<b>Precondición:</b>	-
Flujo normal	
<ol style="list-style-type: none"> <li>1. Se pasan los datos de consulta del ciudadano al componente.</li> <li>2. El componente devuelve los datos del ciudadano.</li> </ol>	
Flujo alternativo	
2.1 El usuario no existe en el sistema de terceros / ciudadanos	
<b>Postcondición:</b>	Los datos del usuario se cargan en el aplicativo.

## CU-03

Consulta apoderamientos	
<b>Código:</b>	CU-03
<b>Descripción:</b>	El sistema muestra un listado de apoderamientos en los que el ciudadano es poderdante o apoderado. Además se permite realizar filtros sobre este listado.
<b>Cubre los requisitos:</b>	RF-03: Bandeja de apoderamientos RF-05: Búsqueda de apoderamientos



<b>Actores:</b>	Poderdante, Apoderado, Sistemas externos
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema
<b>Flujo normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema muestra un listado de apoderamientos en los que el ciudadano es poderdante o apoderado.</li> <li>2. El usuario indica los filtros para realizar la búsqueda de los apoderamientos que tiene en su bandeja.</li> <li>3. Se muestran los resultados que coincidan con los filtros especificados.</li> </ol>	
<b>Flujo alternativo</b>	
3.1 No existen apoderamientos que coincidan con los filtros indicados.	
<b>Postcondición:</b>	-

**CU-04**

<b>Nuevo apoderamiento</b>	
<b>Código:</b>	CU-04
<b>Descripción:</b>	El poderdante creará un nuevo registro de apoderamiento.
<b>Cubre los requisitos:</b>	RF-04: Nuevo apoderamiento
<b>Actores:</b>	Poderdante, CitizensManager
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema
<b>Flujo normal</b>	
<ol style="list-style-type: none"> <li>1. El ciudadano accede a la pantalla de alta de un nuevo apoderamiento.</li> <li>2. El sistema muestra el formulario de apoderamiento que debe rellenar el ciudadano.</li> <li>3. El ciudadano rellena los datos del apoderamiento, indicando el NIF, nombre y apellidos del apoderado, así como el procedimiento y las fechas de vigencia para el que va a dar el poder.</li> <li>4. El sistema comprueba mediante el componente CitizensManager que el apoderado es válido</li> <li>5. El ciudadano confirma los datos del apoderamiento y envía el formulario.</li> <li>6. El sistema valida los datos introducidos.</li> <li>7. Los datos han sido validados y se procede a realizar la firma digital del apoderamiento. Se inicia el proceso de firma descrito en el caso de uso CU-05</li> </ol>	
<b>Flujo alternativo</b>	
<p>4.1 Lo datos del apoderado no coinciden con ningún ciudadano de los que tiene acceso la implementación del componente CitizensManager.</p> <p>7.1 El sistema no puede validar los datos, puede ser que haya algún dato obligatorio sin especificar o haya errores en los ya especificados.</p> <p>7.2 El ciudadano subsana los datos del formulario y lo vuelve a enviar, volviendo al punto 7 del flujo normal.</p>	
<b>Postcondición:</b>	-

## CU-05

Firma digital de apoderamientos	
<b>Código:</b>	CU-05
<b>Descripción:</b>	El ciudadano firma digitalmente el apoderamiento que está dando de alta en el registro de apoderamientos.
<b>Cubre requisitos:</b>	los RF-11: Firma digital de documentos
<b>Actores:</b>	Poderdante, Signer
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El sistema debe haber validado los datos del formulario de alta de un nuevo apoderamiento.
Flujo normal	
<ol style="list-style-type: none"> <li>1. El ciudadano inicia el proceso de firma digital.</li> <li>2. El sistema mediante el uso del componente Signer muestra los certificados digitales disponibles.</li> <li>3. El ciudadano selecciona su certificado digital.</li> <li>4. El sistema inicia el proceso de firma digital.</li> <li>5. El proceso de firma digital finaliza y el sistema muestra el documento de apoderamiento firmado al ciudadano.</li> <li>6. El ciudadano puede volver a la bandeja de apoderamientos.</li> </ol>	
Flujo alternativo	
2.1 No existe ningún certificado digital disponible	
6.1 Se produce algún error durante el proceso de firma digital.	
<b>Postcondición:</b>	El apoderamiento se ha firmado digitalmente.

## CU-06

Almacenar documentos	
<b>Código:</b>	CU-06
<b>Descripción:</b>	Los documentos que genera el sistema se almacenan en el modelo de datos y además se permite configurar un gestor documental externo mediante el componente DocumentsManager.
<b>Cubre requisitos:</b>	los RF-09: Generación de documentación RF-10: Componente de gestión documental
<b>Actores:</b>	DocumentsManager
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El sistema debe haber validado los datos del formulario que generará un documento PDF y será firmado posteriormente.
Flujo normal	
1. El sistema almacena los documentos en el modelo de datos.	

2. El sistema comprueba si se ha definido un gestor documental externo al que enviarle la documentación generada.
3. Si se ha definido, el sistema envía los documentos mediante el uso del componente DocumentsManager.
4. Se actualizan los datos del documento con el identificador del mismo en el gestor documental externo.
<b>Flujo alternativo</b>
3.1 No se ha definido el uso de un gestor documental externo.
4.1 Se ha producido un error al enviar el documento al gestor documental externo.
<b>Postcondición:</b> El documento está disponible en el gestor documental externo.

### CU-07

<b>Confirmar apoderamiento</b>	
<b>Código:</b>	CU-07
<b>Descripción:</b>	El apoderado realiza la confirmación de la recepción del poder otorgando por el poderdante para el procedimiento especificado.
<b>Cubre los requisitos:</b>	RF-08: Cambio de estados RF-09: Generación de documentación RF-10: Componente de gestión documental RF-11: Firma digital de documentos
<b>Actores:</b>	Apoderado, DocumentsManager, Signer
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El poderdante debe haber creado un nuevo registro de apoderamiento que necesite confirmación por parte del apoderado. También puede ser que el estado del apoderamiento sea Rechazado o Renunciado.
<b>Flujo normal</b>	
1. El sistema muestra los apoderamientos del apoderado cuyo estado sea Pendiente de confirmación.	
2. El ciudadano accede al detalle del apoderamiento y selecciona la opción de confirmar el poder otorgado.	
3. El apoderado confirma el apoderamiento mediante la firma digital del mismo.	
4. El apoderamiento pasa al estado Confirmado.	
<b>Flujo alternativo</b>	
1.1 No existen apoderamientos pendientes de confirmar	
2.1 El ciudadano no desea confirmar el apoderamiento	
3.1 Se produce algún error durante la firma digital del apoderamiento	
<b>Postcondición:</b>	El apoderamiento pasa a estado Confirmado

## CU-08

Rechazar apoderamiento	
<b>Código:</b>	CU-08
<b>Descripción:</b>	El apoderado rechaza el poder otorgando por el poderdante para el procedimiento especificado.
<b>Cubre requisitos:</b> los	RF-08: Cambio de estados RF-09: Generación de documentación RF-10: Componente de gestión documental RF-11: Firma digital de documentos
<b>Actores:</b>	Apoderado, DocumentsManager, Signer
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El poderdante debe haber creado un nuevo registro de apoderamiento que necesite confirmación por parte del apoderado.
Flujo normal	
<ol style="list-style-type: none"> <li>1. El sistema muestra los apoderamientos del apoderado cuyo estado sea Pendiente de confirmación.</li> <li>2. El ciudadano accede al detalle del apoderamiento y selecciona la opción de rechazar el poder otorgado.</li> <li>3. El apoderado rechaza el apoderamiento mediante la firma digital del mismo.</li> <li>4. El apoderamiento pasa al estado Rechazado.</li> </ol>	
Flujo alternativo	
1.1 No existen apoderamientos pendientes de confirmar 3.1 Se produce algún error durante la firma digital del rechazo.	
<b>Postcondición:</b>	El apoderamiento pasa a estado Rechazado

## CU-09

Renunciar apoderamiento	
<b>Código:</b>	CU-09
<b>Descripción:</b>	El apoderado renuncia a un poder que había confirmado con anterioridad.
<b>Cubre requisitos:</b> los	RF-08: Cambio de estados RF-09: Generación de documentación RF-10: Componente de gestión documental RF-11: Firma digital de documentos
<b>Actores:</b>	Apoderado, DocumentsManager, Signer
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El apoderamiento debe estar en estado Confirmado
Flujo normal	
<ol style="list-style-type: none"> <li>1. El sistema muestra los apoderamientos del apoderado cuyo estado sea Confirmado.</li> <li>2. El ciudadano accede al detalle del apoderamiento y selecciona la opción de renunciar al poder otorgado.</li> </ol>	

3. El apoderado renuncia el apoderamiento mediante la firma digital del mismo.	
4. El apoderamiento pasa al estado Renunciado.	
<b>Flujo alternativo</b>	
1.1 No existen apoderamientos a los que se pueda renunciar.	
3.1 Se produce algún error durante la firma digital de la renuncia.	
<b>Postcondición:</b>	El apoderamiento pasa a estado Renunciado.

#### CU-10

<b>Revocar apoderamiento</b>	
<b>Código:</b>	CU-10
<b>Descripción:</b>	El poderdante revoca un apoderamiento que se encuentre vigente.
<b>Cubre requisitos:</b> los	RF-08: Cambio de estados RF-09: Generación de documentación RF-10: Componente de gestión documental RF-11: Firma digital de documentos
<b>Actores:</b>	Poderdante, DocumentsManager, Signer
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El apoderamiento debe estar vigente
<b>Flujo normal</b>	
1. El sistema muestra los apoderamientos del ciudadano.	
2. El apoderado accede al detalle del apoderamiento que quiere revocar.	
3. El apoderado selecciona la opción de revocar el apoderamiento y firma digitalmente la revocación.	
4. El apoderamiento pasa a estado Revocado.	
<b>Flujo alternativo</b>	
1.1 No existen apoderamientos que se puedan revocar.	
3.1 Se produce algún error durante la firma digital de la revocación.	
<b>Postcondición:</b>	El apoderamiento pasa a estado Revocado.

#### CU-11

<b>Detalle apoderamiento</b>	
<b>Código:</b>	CU-11
<b>Descripción:</b>	Se muestran los detalles del apoderamiento.
<b>Cubre requisitos:</b> los	RF-06: Detalle de apoderamientos RF-07: Descarga de documentación
<b>Actores:</b>	Poderdante, Apoderado
<b>Precondición:</b>	El usuario debe estar autenticado en el sistema. El ciudadano debe tener se apoderado o poderdante de al menos un apoderamiento.

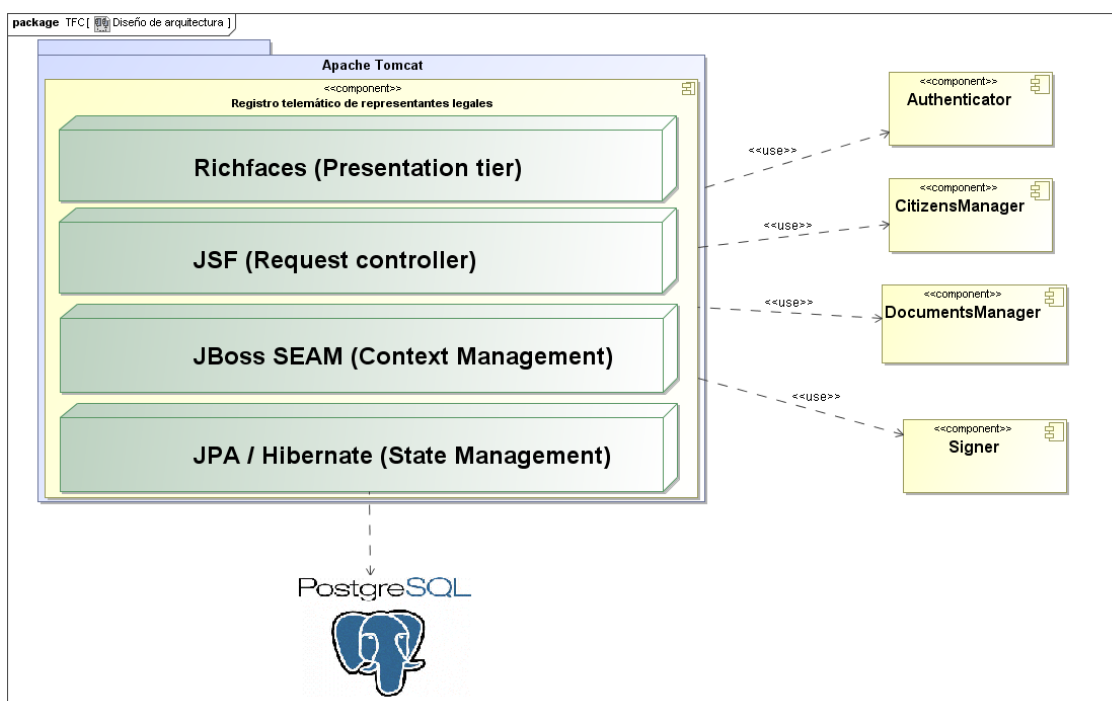
Flujo normal	
1. El sistema muestra los apoderamientos del ciudadano. 2. El ciudadano selecciona un apoderamiento y accede a su detalle. 3. El sistema muestra las acciones que se pueden llevar a cabo sobre el apoderamiento, así como sus datos generales y documentos asociados permitiéndole la descarga de los mismos.	
Flujo alternativo	
1.1 El ciudadano no tiene apoderamientos.	
<b>Postcondición:</b>	-

## Diseño del sistema

A continuación se recoge el diseño del sistema de información. El diseño del mismo recoge una serie de diagramas que facilitarán el desarrollo y comprensión del sistema.

Además el diseño nos permite hacernos una idea de la funcionalidad y la apariencia que tendrá la aplicación.

## Diseño de la arquitectura



**Figura 8 – Diagrama de arquitectura**

Mediante el diagrama de arquitectura se pueden diferenciar las distintas capas en las que se divide la aplicación así como los componentes que usa. La aplicación a desarrollar será una aplicación web cuya arquitectura es Cliente / Servidor.

El funcionamiento de una aplicación Cliente / Servidor se inicia cuando el cliente (navegador web normalmente) realiza una petición al servidor, éste la recoge, realiza las acciones que tenga que realizar y devuelve el resultado al cliente.

El diseño de la arquitectura propuesta define las capas en las que se divide el servidor y concretamente la aplicación que se va a desarrollar para gestionar el registro telemático de representaciones legales entre ciudadanos.

Las capas en las que se divide la aplicación son las siguientes:

- **RichFaces (Presentation tier):** Framework para la capa de presentación, mediante el uso de componentes gráficos RichFaces y Ajax mediante la cual podemos realizar peticiones al servidor y renderizar total o parcialmente la página de resultado que le mostraremos al cliente.
- **JSF (Request controller):** Mediante este Framework realizaremos la gestión de eventos y estados de las páginas que se mostrarán al cliente compuestas de componentes RichFaces.
- **JBoss SEAM (Context Management):** Nos facilita la integración entre las distintas capas de la aplicación mediante el uso de componentes, inyección de los mismos y definición de contextos.
- **JPA / Hibernate (State Management):** Capa de persistencia mediante la cual nos permite mapear las entidades del modelo de datos.

Por otro lado para darle versatilidad a la aplicación se van a desarrollar una serie de componentes que faciliten la integración con nuevos sistemas. Estos componentes serán los encargados de consultar los datos de los ciudadanos, realizar la autenticación en la aplicación, realizar la firma digital e integrar la aplicación con un gestor documental externo.

La implementación de estos componentes se llevará a cabo usando el patrón de diseño *Factory Pattern*, mediante este patrón creacional se podrán definir distintas implementaciones a usar para cada una de las acciones anteriormente descritas.

Los componentes son los siguientes:

- **Authenticator:** Componente que lleva a cabo la autenticación del usuario en la aplicación. Mediante configuración se podrá indicar la clase que lleva a cabo esta acción. La implementación por defecto que se incluirá en la aplicación realizará esta acción mediante el certificado digital de usuario validándolo con el cliente de firma digital de @firma.
- **CitizensManager:** Componente mediante el cual tendremos acceso desde la aplicación a los datos de los ciudadanos. Mediante configuración se podrá indicar la clase que lleva a cabo esta acción. La implementación por defecto que se incluirá en la aplicación se basará en una tabla en el modelo de datos.
- **Signer:** Componente que realiza la firma digital de documentos. Mediante configuración se podrá indicar la clase que lleva a cabo esta acción. La implementación por defecto que incluirá la aplicación realizará la firma digital haciendo uso del certificado digital de usuario y del cliente de firma de @firma.
- **DocumentsManager:** Componente para la gestión de documentos en un sistema externo. Mediante este componente se podrán obtener y enviar los documentos que genera la aplicación a un gestor documental externo a la aplicación. Se podrá configurar la clase que lleva a cabo esta acción. La implementación por defecto que llevará la aplicación configurada realizará esta gestión mediante el sistema de ficheros del sistema operativo.



# Diagramas de clases

En el siguiente capítulo se recogen los diagramas de clases más relevantes que tendrá el sistema. Entre ellos los diagramas de clases de los componentes y de las clases de dominio y persistencia.

## Clases de dominio

Clases de entidad o de dominio que recogerá la información del sistema.

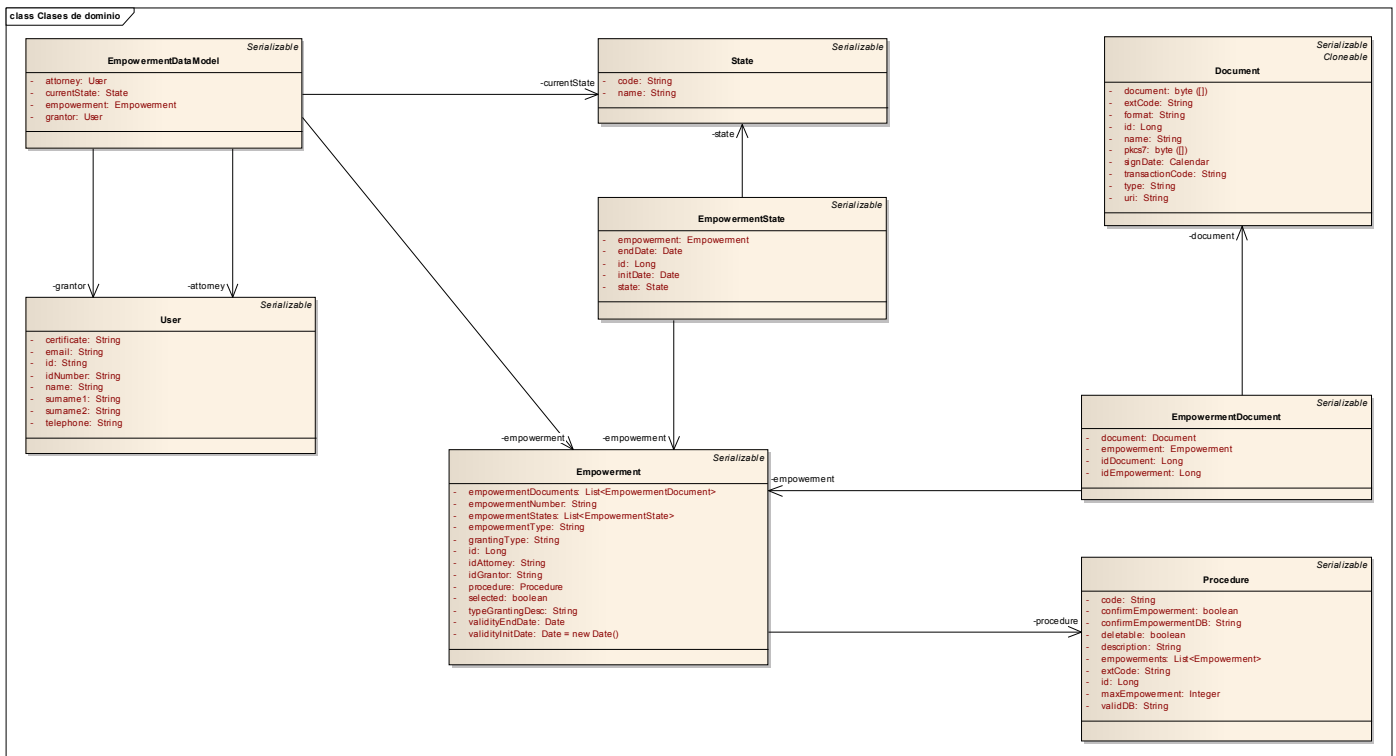


Figura 9 – Diagrama de clases de dominio

### CLD-01: User

<b>User</b>	
<b>Código:</b>	CLD-01
<b>Descripción:</b>	Representa un usuario del sistema, concretamente un ciudadano que será apoderado o poderdante.

### CLD-02: Document

<b>Document</b>	
<b>Código:</b>	CLD-02
<b>Descripción:</b>	Recoge los datos de los documentos que generará la aplicación, así como los datos de la firma digital.

**CLD-03: Procedure**

<b>Procedure</b>	
<b>Código:</b>	CLD-03
<b>Descripción:</b>	Representa un procedimiento administrativo de un organismo u organización para el que se podrán crear apoderamientos entre ciudadanos.

**CLD-04: State**

<b>State</b>	
<b>Código:</b>	CLD-04
<b>Descripción:</b>	Representa el estado en el que se puede encontrar o por el que haya pasado un apoderamiento. Los estados predefinidos son Pendiente, Confirmado, Revocado, Rechazado, Renunciado y Expirado.

**CLD-05: Empowerment**

<b>Empowerment</b>	
<b>Código:</b>	CLD-05
<b>Descripción:</b>	Representa un apoderamiento entre ciudadanos, es decir, el traspaso de poder de un ciudadano como poderdante a otro ciudadano como apoderado para que éste último realice trámites ante alguna administración concreta en nombre del primero.

**CLD-06: EmpowermentState**

<b>EmpowermentState</b>	
<b>Código:</b>	CLD-06
<b>Descripción:</b>	Recoge los distintos estados por los que ha pasado un apoderamiento concreto.

**CLD-07: EmpowermentDocument**

<b>EmpowermentDocument</b>	
<b>Código:</b>	CLD-07
<b>Descripción:</b>	Recoge la relación de documentos de un apoderamiento concreto.

## Clases de persistencia

Clases que permiten persistir y mantener la información, así como dar acceso a los datos almacenados. Para el desarrollo de las mismas se hace uso de la clase *EntityManager*.

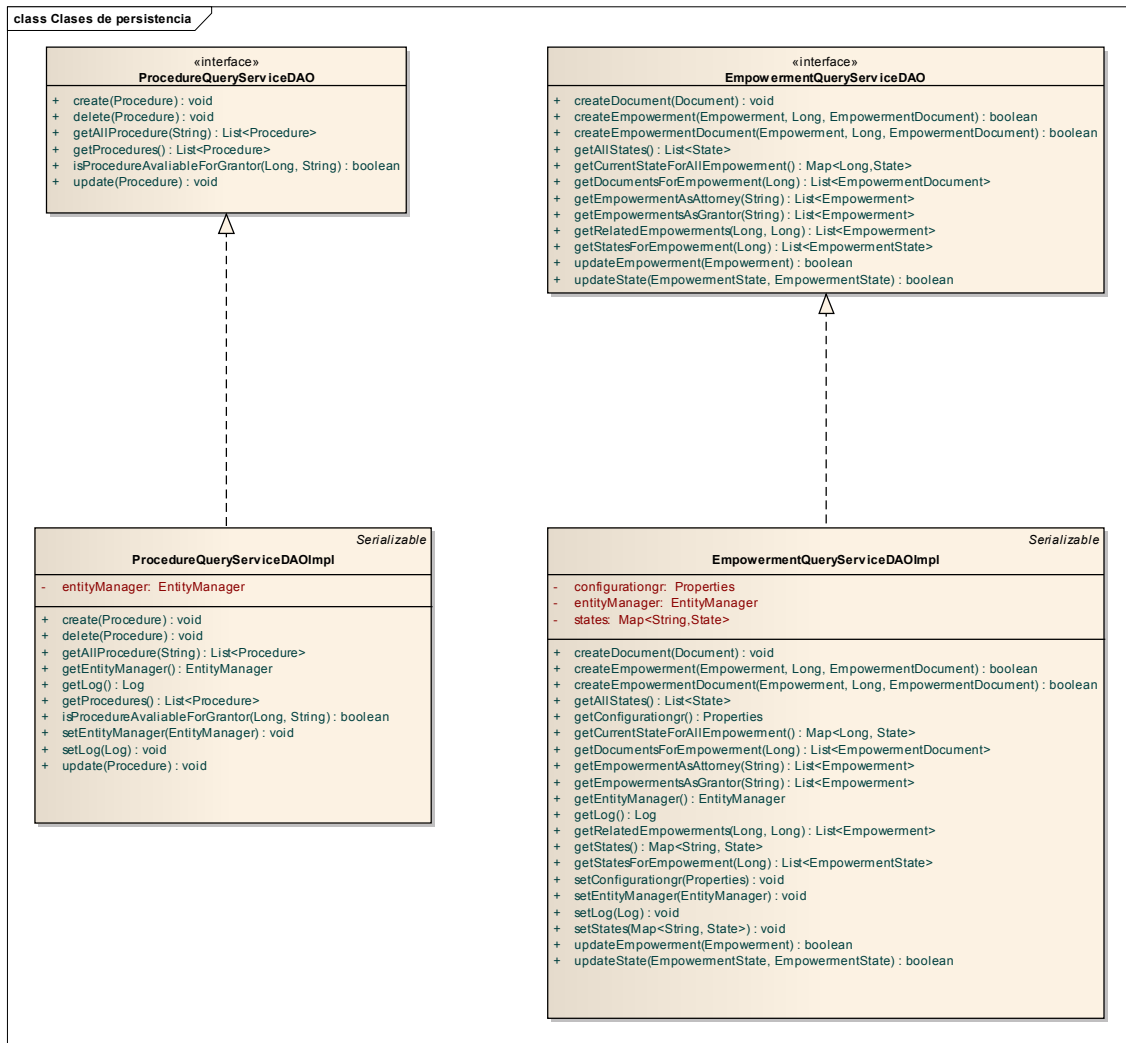


Figura 10 – Diagrama de clases de persistencia

### CLP-01: ProcedureQueryServiceDAO

<b>ProcedureQueryServiceDAO</b>	
<b>Código:</b>	CLP-01
<b>Descripción:</b>	Interfaz que define los métodos para facilitar el acceso a los datos relativos a los procedimientos.

**CLP-02: ProcedureQueryServiceDAOImpl**

<b>ProcedureQueryServiceDAOImpl</b>	
<b>Código:</b>	CLP-02
<b>Descripción:</b>	Implementación de la interfaz CLP-01 que implementa los métodos de acceso a los datos de los procedimientos.

**CLP-03: EmpowermentQueryServiceDAO**

<b>EmpowermentQueryServiceDAO</b>	
<b>Código:</b>	CLP-03
<b>Descripción:</b>	Interfaz que define los métodos para facilitar el acceso a los datos relacionados con los apoderamientos (estados, documentos, etc.)

**CLP-04: EmpowermentQueryServiceDAOImpl**

<b>EmpowermentQueryServiceDAOImpl</b>	
<b>Código:</b>	CLP-04
<b>Descripción:</b>	Implementación de la interfaz CLP-03 que implementa los métodos de acceso a los datos relacionados con los apoderamientos.

## Componente Authenticator

Clases que definen el componente de autenticación.

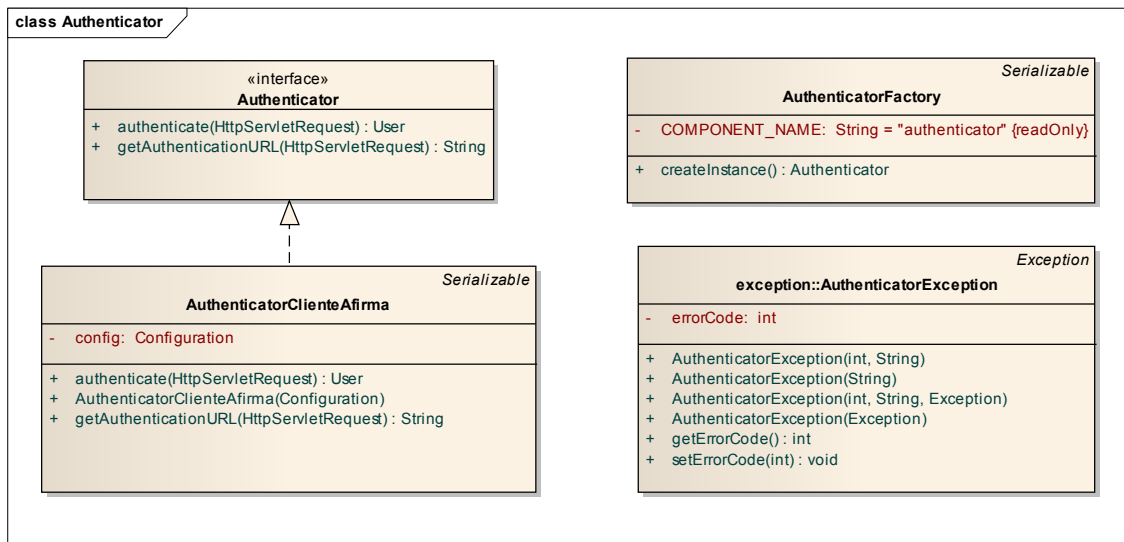


Figura 11 – Diagrama de clases Authenticator

### CLC-01: Authenticator

Authenticator	
<b>Código:</b>	CLC-01
<b>Descripción:</b>	Interfaz del componente que define los métodos para llevar a cabo la autenticación del usuario en el sistema. Los métodos definidos son: <ul style="list-style-type: none"> <li>getAuthenticationURL(): devuelve la url de autenticación.</li> <li>authenticate(): realiza la autenticación del usuario.</li> </ul>

### CLC-02: AuthenticatorFactory

AuthenticatorFactory	
<b>Código:</b>	CLC-02
<b>Descripción:</b>	Clase factoría para crear una instancia de la implementación configurada que implemente la interfaz CLC-01.

### CLC-03: AuthenticatorException

AuthenticatorException	
<b>Código:</b>	CLC-03
<b>Descripción:</b>	Clase que recoge las excepciones del componente de autenticación.

**CLC-04: AuthenticatorClienteAfirma**

AuthenticatorClienteAfirma	
<b>Código:</b>	CLC-04
<b>Descripción:</b>	Implementación de la interfaz <i>Authenticator</i> (CLC-01) que se basa en el cliente de firma de la plataforma @firma para realizar la autenticación de usuario.

**Componente CitizensManager**

Clases que definen el componente de gestión de ciudadanos. Mediante este componente se puede definir la fuente de los datos de los ciudadanos que gestiona el sistema.

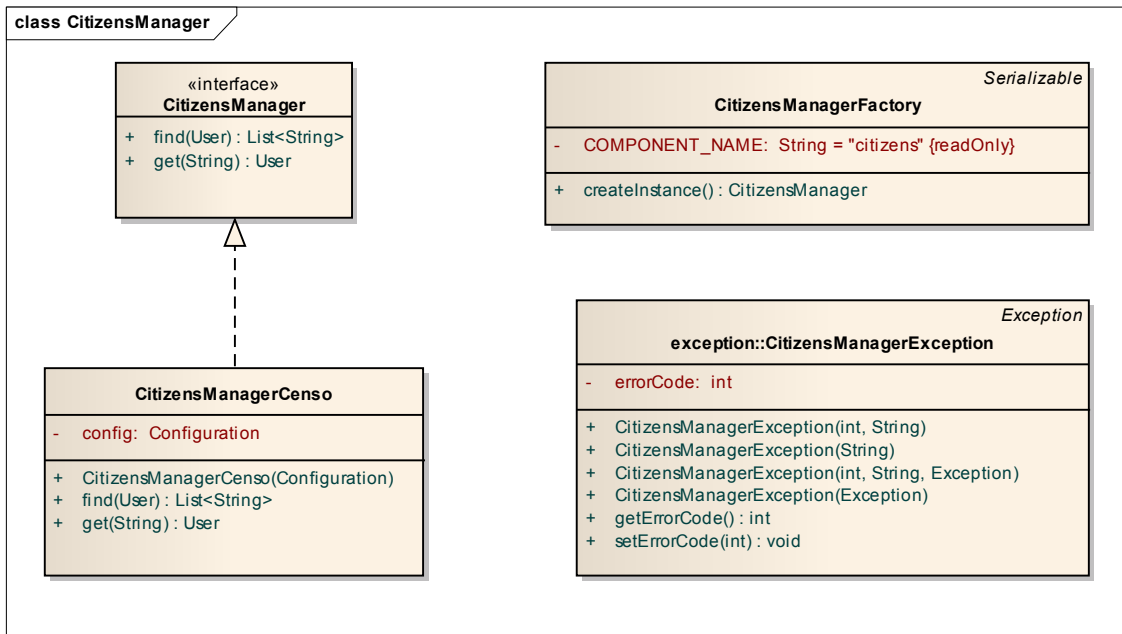


Figura 12 – Diagrama de clases CitizensManager

**CLC-05: CitizensManager**

CitizensManager	
<b>Código:</b>	CLC-05
<b>Descripción:</b>	Interfaz del componente que define los métodos para la obtención de los datos de los ciudadanos. Los métodos definidos son: <ul style="list-style-type: none"> <li>• find(): permite encontrar un ciudadano a partir de sus datos personales.</li> <li>• get(): permite obtener los datos de un ciudadano a partir de su identificador.</li> </ul>

**CLC-06: CitizensManagerFactory**

<b>CitizensManagerFactory</b>	
<b>Código:</b>	CLC-06
<b>Descripción:</b>	Clase factoría para crear una instancia de la implementación configurada que implemente la interfaz <i>CitizensManager</i> .

**CLC-07: CitizensManagerException**

<b>CitizensManagerException</b>	
<b>Código:</b>	CLC-07
<b>Descripción:</b>	Clase que recoge las excepciones del componente de gestión de ciudadanos.

**CLC-08: CitizensManagerCenso**

<b>CitizensManagerCenso</b>	
<b>Código:</b>	CLC-08
<b>Descripción:</b>	Implementación de la interfaz CLC-05 que obtiene los datos de los ciudadanos a partir de una tabla en el modelo de datos, simulando la lectura de un censo de habitantes.

## Componente DocumentsManager

Clases que definen el componente de gestión externa de documentos. Mediante este componente se pueden enviar los documentos que genera la aplicación a un gestor documental externo al sistema.

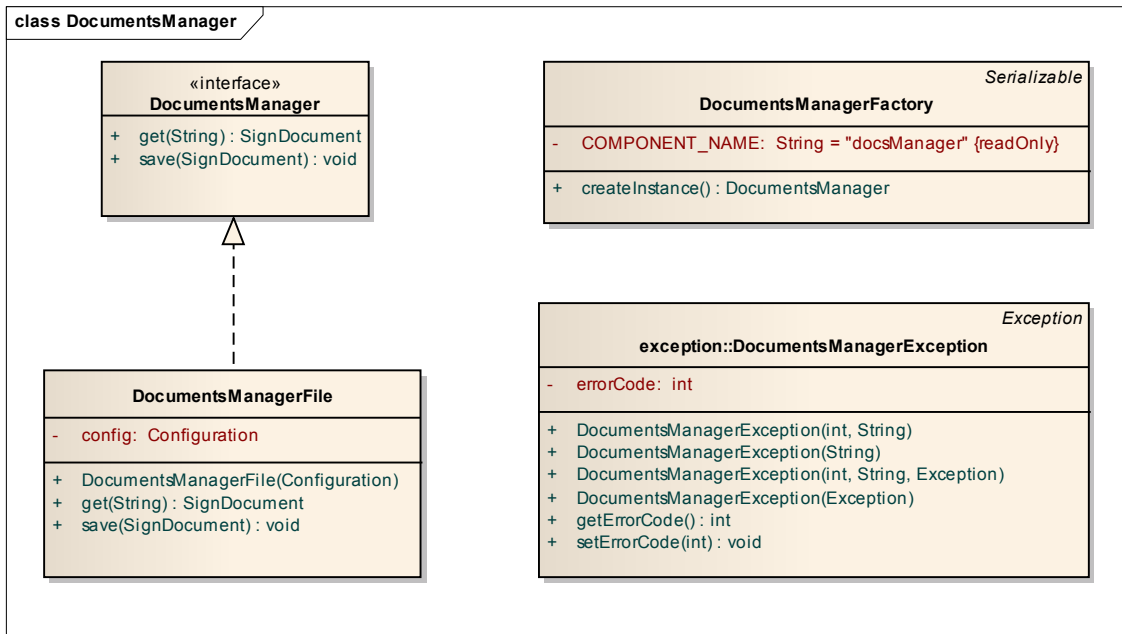


Figura 13 – Diagrama de clases DocumentsManager

### CLC-09: DocumentsManager

<b>DocumentManager</b>	
<b>Código:</b>	CLC-09
<b>Descripción:</b>	Interfaz del componente que define la gestión externa de documentos. Los métodos definidos son: <ul style="list-style-type: none"> <li>• save(): permite almacenar el documento.</li> <li>• get(): permite obtener los datos del documento a partir de su identificador.</li> </ul>

### CLC-10: DocumentsManagerFactory

<b>DocumentManagerFactory</b>	
<b>Código:</b>	CLC-10
<b>Descripción:</b>	Clase factoría para crear una instancia de la implementación configurada que implemente la interfaz <i>DocumentsManager</i> .



**CLC-11: DocumentsManagerException**

<b>DocumentManagerException</b>	
<b>Código:</b>	CLC-11
<b>Descripción:</b>	Clase que recoge las excepciones del componente de gestión externa de documentos.

**CLC-12: DocumentsManagerFile**

<b>DocumentManagerFile</b>	
<b>Código:</b>	CLC-12
<b>Descripción:</b>	Implementación de la interfaz <i>DocumentsManager</i> que permite almacenar y obtener los documentos desde el sistema de ficheros del sistema operativo simulando el funcionamiento de un gestor documental externo a la aplicación.

## Componente Signer

Clases que definen el componente de firma digital de documentos.

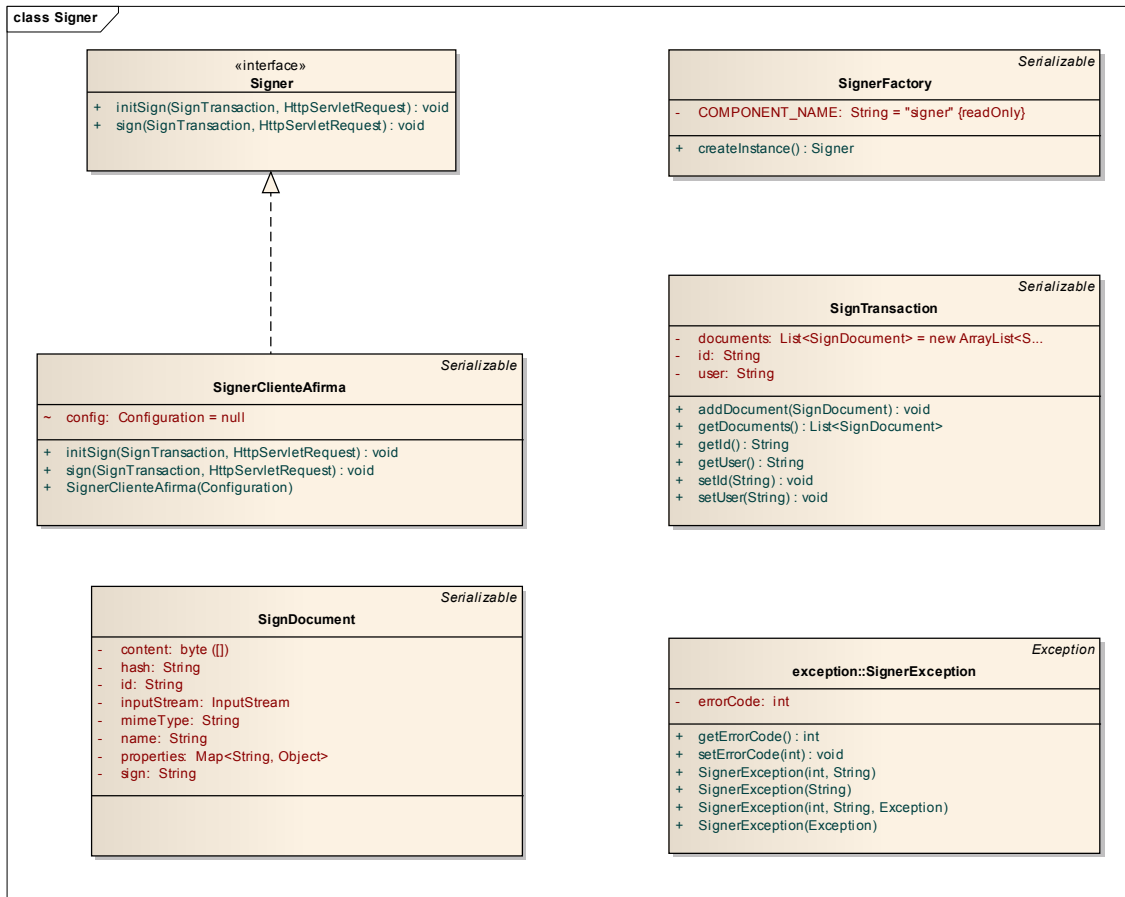


Figura 14 – Diagrama de clases Signer

### CLC-13: Signer

Signer	
<b>Código:</b>	CLC-13
<b>Descripción:</b>	Interfaz del componente que define los métodos para llevar a cabo la firma digital de documentos. Los métodos definidos son: <ul style="list-style-type: none"> <li>• <code>initSign()</code>: inicia el proceso de firma digital.</li> <li>• <code>sign()</code>: finaliza el proceso de firma digital.</li> </ul>

### CLC-14: SignerFactory

SignerFactory	
<b>Código:</b>	CLC-14
<b>Descripción:</b>	Clase factoría para crear una instancia de la implementación configurada que implemente la interfaz <i>Signer</i> .

**CLC-15: SignTransaction**

<b>SignTransaction</b>	
<b>Código:</b>	CLC-15
<b>Descripción:</b>	Clase que recoge los datos necesarios para la firma digital.

**CLC-16: SignDocument**

<b>SignDocument</b>	
<b>Código:</b>	CLC-16
<b>Descripción:</b>	Clase que recoge los documentos que se van a firmar digitalmente en la transacción de firma.

**CLC-17: SignerException**

<b>SignerException</b>	
<b>Código:</b>	CLC-17
<b>Descripción:</b>	Clase que recoge las excepciones del componente de firma digital.

**CLC-18: SignerClienteAfirma**

<b>SignerClienteAfirma</b>	
<b>Código:</b>	CLC-18
<b>Descripción:</b>	Implementación de la interfaz <i>Signer</i> que se basa en el cliente de firma de la plataforma @firma para llevar a cabo la firma digital de los documentos usando el certificado digital del ciudadano.

## Diagramas de secuencias

En este capítulo se recogen los diagramas de secuencias más relevantes del sistema. Mediante estos diagramas se pretende explicar el funcionamiento básico que tendrá el sistema para una autenticación de usuario y para la creación de un nuevo apoderamiento.

### Autenticación del ciudadano

El usuario se autentica en el sistema mediante su certificado digital de usuario. Para llevar a cabo el proceso de autenticación se hace uso del cliente de firma electrónica de la plataforma @firma. Una vez que el usuario se ha autenticado en la aplicación se le muestra la bandeja con sus apoderamientos en los que participa como poderdante o como apoderado.

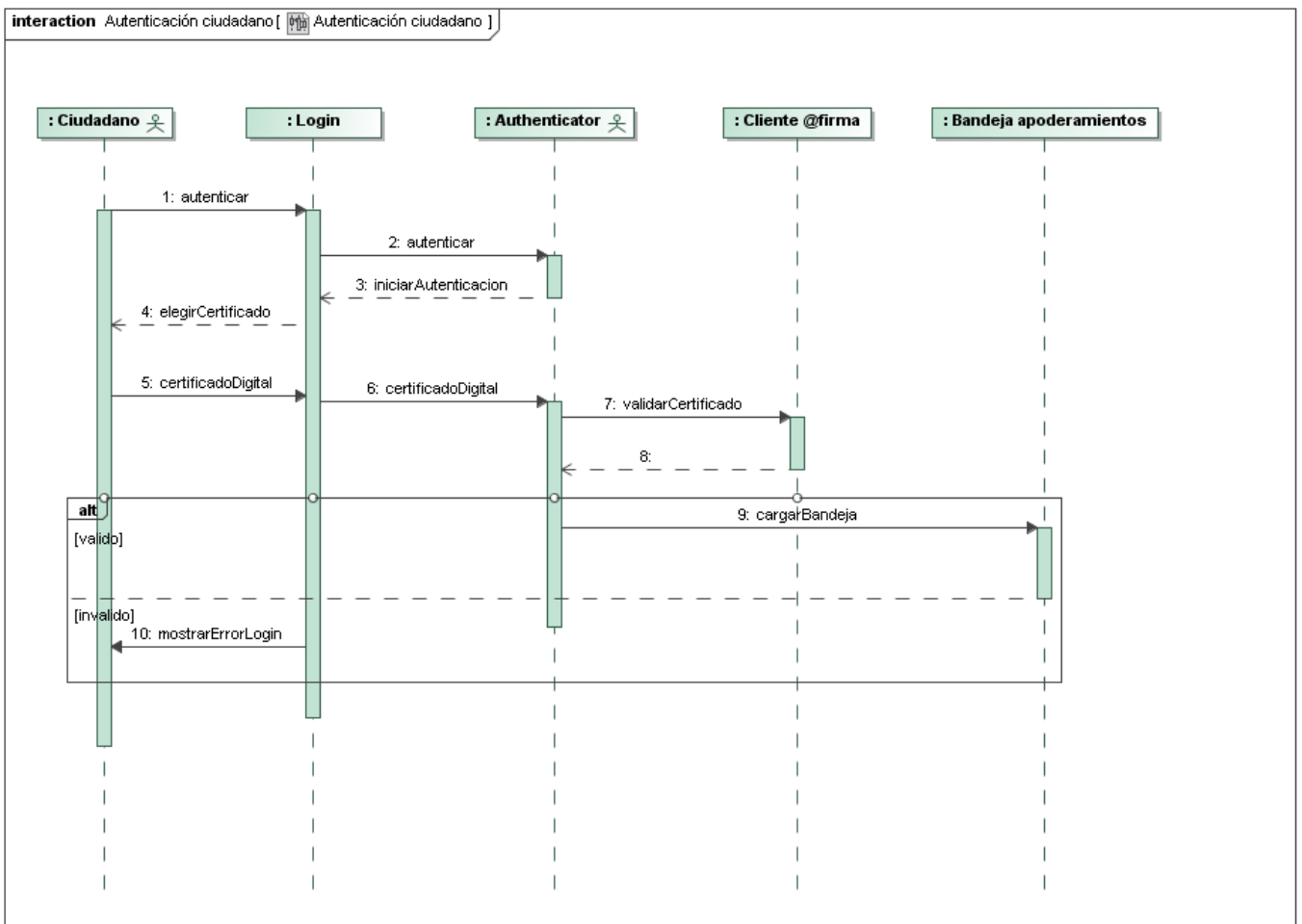


Figura 15 – Diagrama de secuencias Autenticación de usuario

## Nuevo apoderamiento

Mediante este proceso el usuario puede crear un nuevo apoderamiento mediante el cuál le da el poder para que otro usuario (ciudadano) lo represente el uno o varios trámites administrativos.

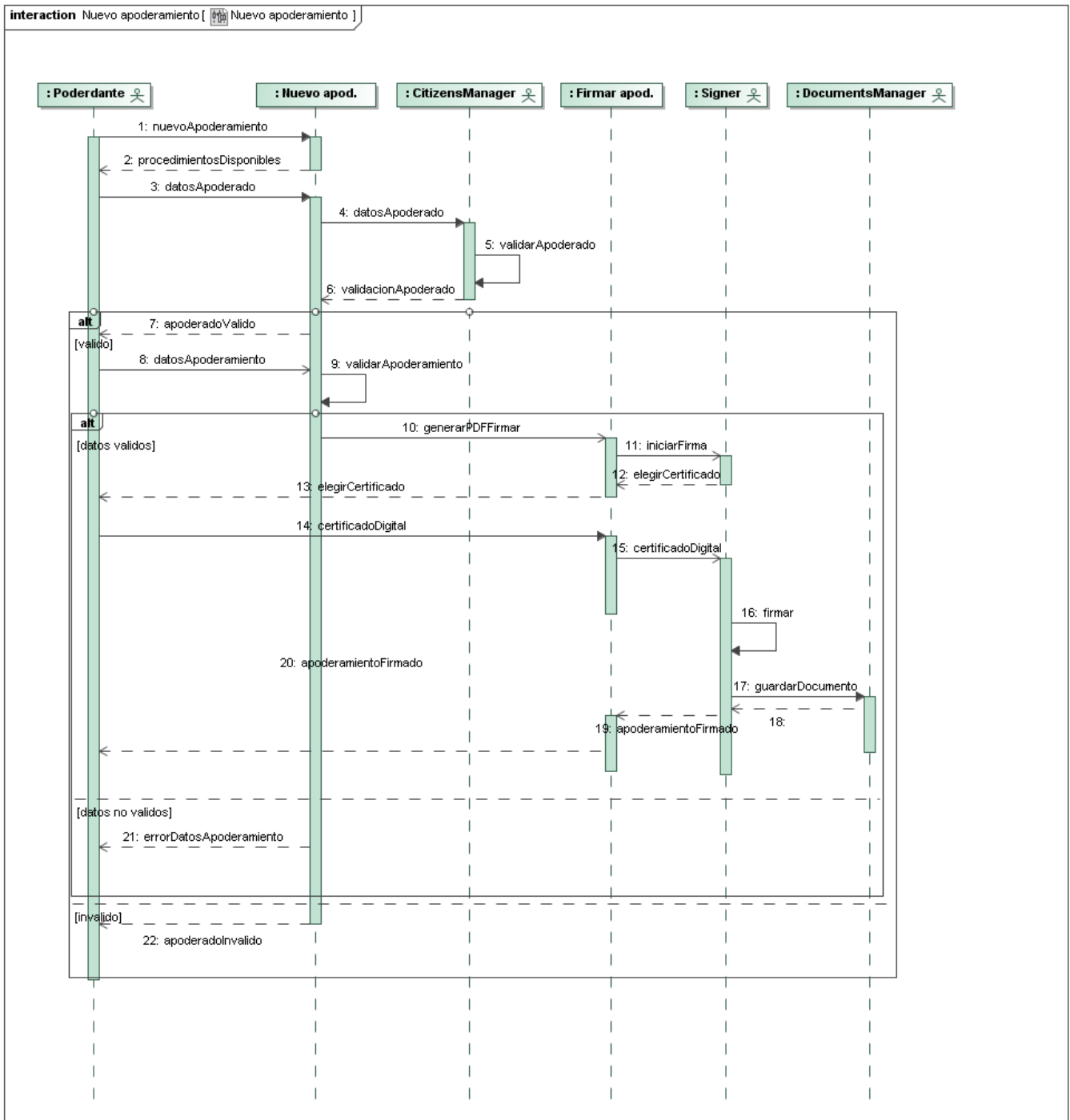
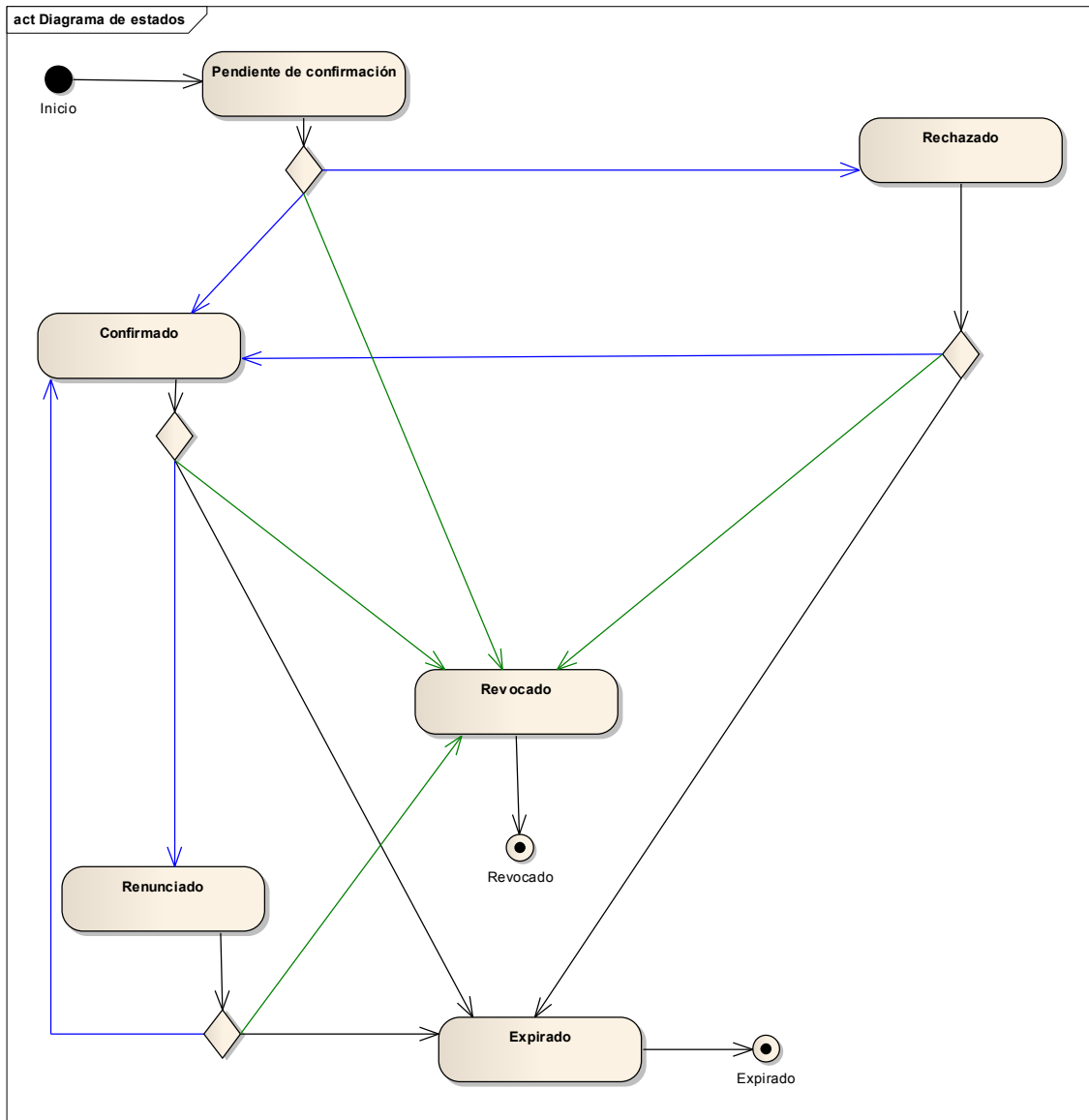


Figura 16 – Diagrama de secuencias Nuevo apoderamiento

## Diagrama de estados

El siguiente diagrama presenta los distintos estados por los que puede pasar un apoderamiento. Además se indica en cada transición entre estados quien puede llevarla a cabo. Estos cambios de estado serán gestionados y garantizados por el sistema.



**Figura 17 – Diagrama de estados**

- Acciones realizadas por el apoderado
- Acciones realizadas por el poderdante
- Acciones realizadas por el sistema al cumplir la vigencia

A continuación se detallan los cambios de estado que puede realizar cada ciudadano en función de su rol en el sistema.

## Diagrama de estados (apoderado)

Mediante este extracto del diagrama de estados se pretende mostrar de manera simplificada los cambios de estados que puede llevar a cabo un apoderado.

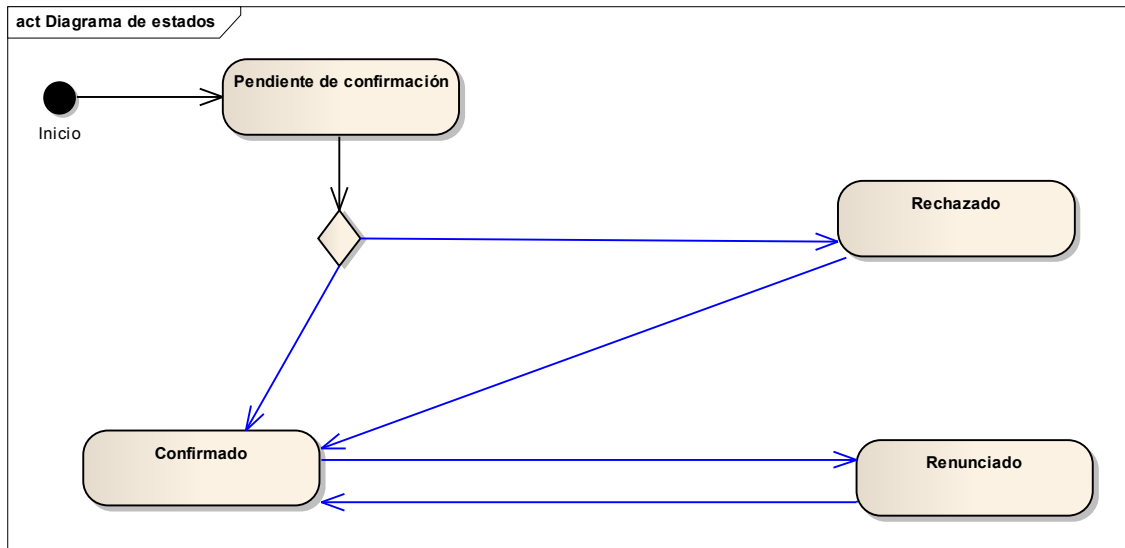


Figura 18 – Diagrama de estados (apoderado)

## Diagrama de estados (poderdante)

Mediante este extracto del diagrama de estados se pretende mostrar de manera simplificada los cambios de estados que puede llevar a cabo un poderdante.

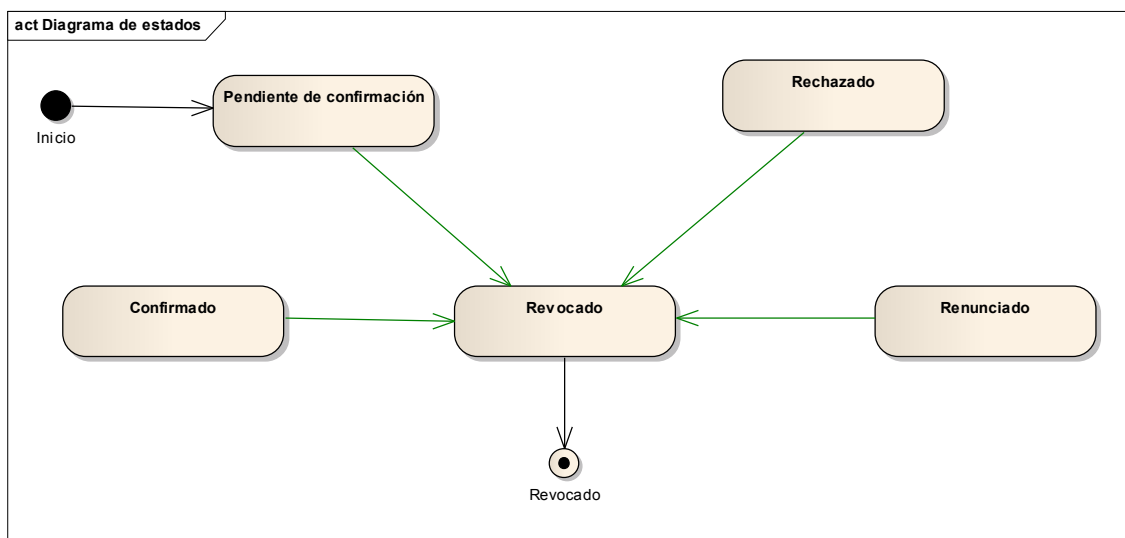


Figura 19 – Diagrama de estados (poderdante)

## Diagrama de estados (sistema)

Mediante este extracto del diagrama de estados se pretende mostrar de manera simplificada los cambios de estados que puede llevar a cabo el sistema cuando se cumplen las fechas de vigencia del apoderamiento.

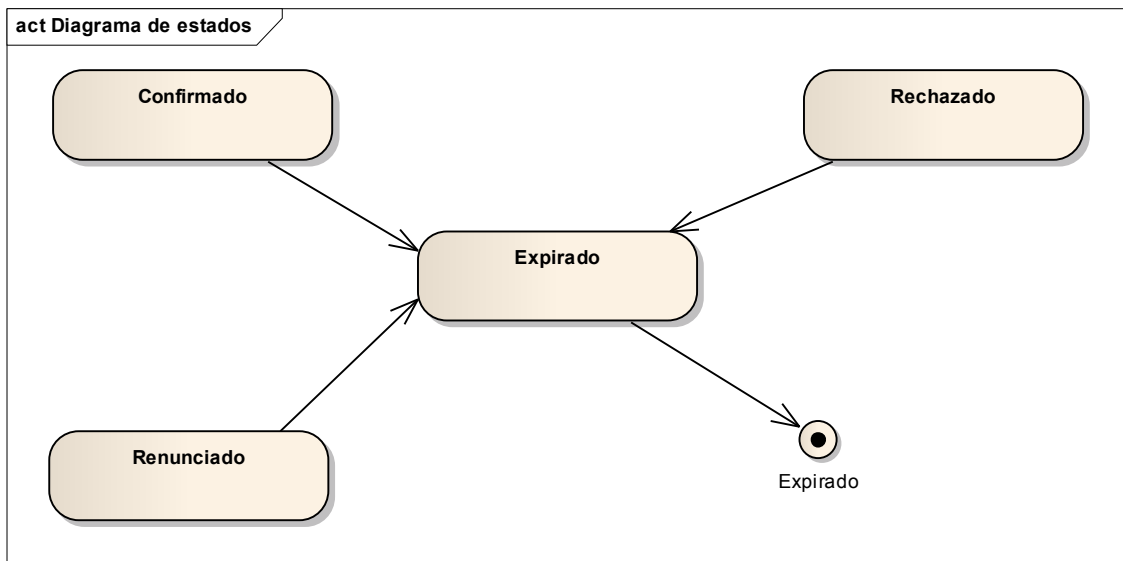


Figura 20 – Diagrama de estados (sistema)



## Diseño del modelo de datos

El diseño del modelo de datos pretende dar soporte a la persistencia de la información que gestionará el sistema. A continuación se muestra un diagrama en el que se recoge el diseño del modelo de datos.

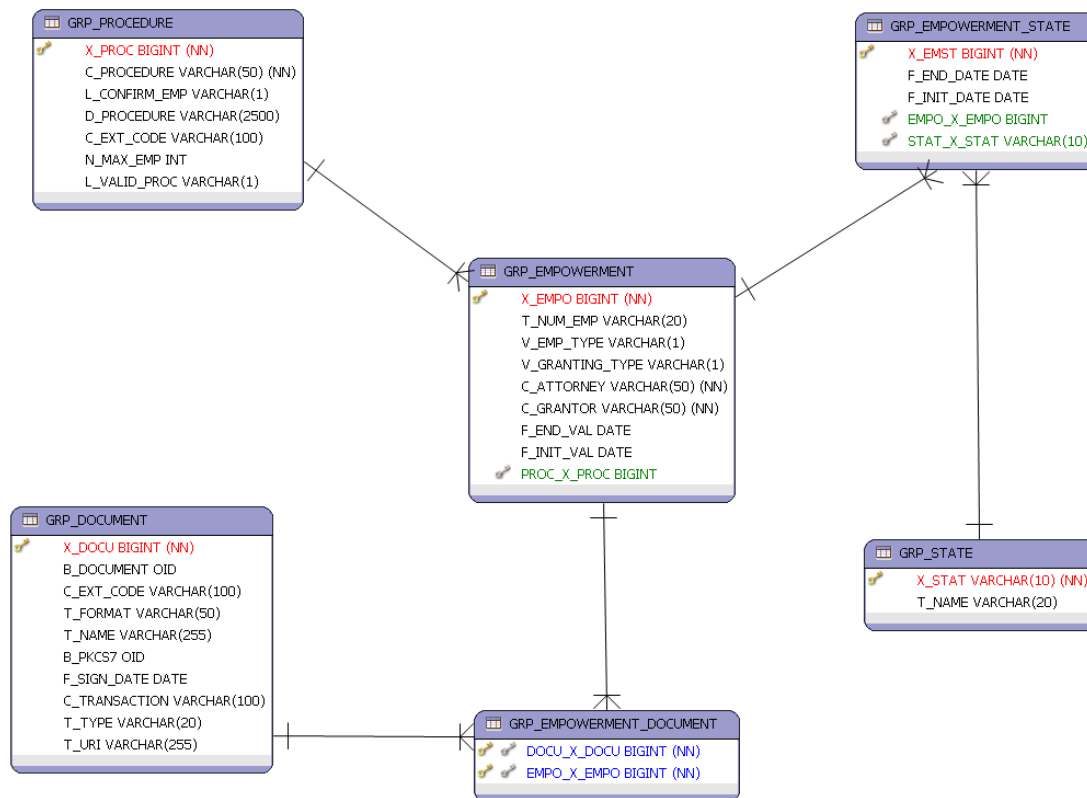


Figura 21 – Diagrama Modelo de datos

(\*) Los tipos de las columnas mostrados en el diagrama son meramente orientativos ya que pueden no reflejar los tipos reales del gestor de base de datos relacional sobre el que se implemente..

## Nomenclatura de tablas y columnas

Para dar nombre a las tablas y columnas se seguirán una serie de indicaciones para facilitar la comprensión de qué es y qué se almacena en ella. Las indicaciones que se seguirán son las siguientes:

- Nombre de tablas y columnas en minúsculas para evitar problemas en la implementación ya que PostgreSQL es “case sensitive” (sensible a mayúsculas y minúsculas).
- Si el nombre de la tabla está compuesto por más de una palabra se separa con el carácter “\_” para que resulta más cómoda su lectura e identificación.

- Se utilizarán distintos prefijos en función del tipo de columna:
  - x\_: para las claves primarias
  - t\_: para columnas de tipo texto
  - c\_: para columnas de tipo texto que almacenen códigos
  - n\_: para columnas de tipo numéricas
  - l\_: para columnas de tipo lógico (true o false)
  - f\_: para columnas de tipo fecha
  - v\_: para columnas con valores variables
  - b\_: para columnas de tipo binario (por ejemplo los documentos)

## Auditoría

Para auditar los datos que almacenará la aplicación haremos uso del módulo que nos ofrece Hibernate, **JBoss Envers**.

JBoss Envers permite auditar / versionar las clases persistentes que maneja la aplicación. Simplemente hay que hacer uso de la anotación `@Audited` en las clases de dominio para que automáticamente se recojan los cambios que se producen en esa entidad.

El sistema se basa en revisiones de los datos para mantener un histórico y permitir la recuperación de versiones anteriores, de forma similar a Subversion. Algunas de las características de JBoss Envers son:

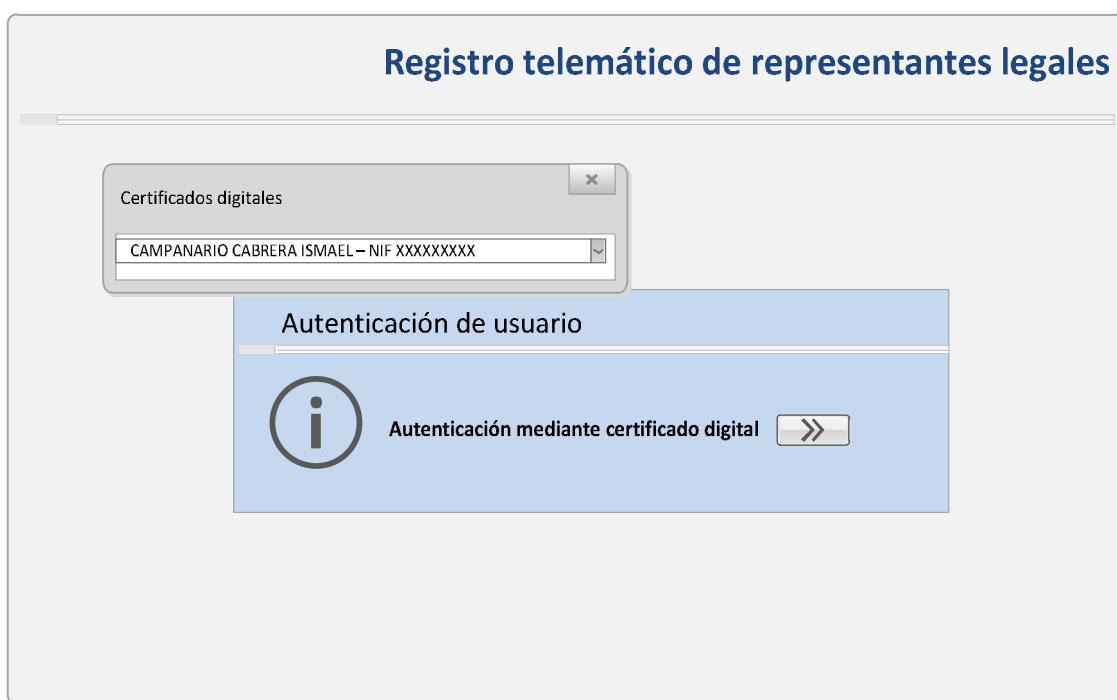
- Auditoría de todas las asignaciones definidas por la especificación JPA.
- Auditoría de algunas asignaciones de Hibernate, que extienden JPA, al igual que tipos personalizados, mapas, etc.
- El registro de los datos para cada revisión mediante una “Entidad de revisión”.
- Consulta de datos históricos.

## Prototipo interfaz gráfica

En el siguiente capítulo se recoge el prototipo de la interfaz gráfica para tener una primera impresión de cómo serán las pantallas a las que tendrá acceso el usuario al usar la aplicación.

El diseño de las pantallas pretende ser un punto de partida básico del funcionamiento que tendrá la aplicación. Se han definido las pantallas cuya funcionalidad se considera más relevante.

### Autenticación de usuario



**Figura 22 – Interfaz gráfica Autenticación de usuario**

Mediante esta pantalla el usuario tendrá la posibilidad de iniciar la autenticación en la aplicación. El sistema le solicitará que seleccione su certificado digital de usuario y una vez comprobado permitirá el acceso al mismo.

## Bandeja de apoderamientos

**Registro telemático de representantes legales**

ISMAEL CAMPANARIO CABRERA 

**Bandeja de apoderamientos**

Apoderamientos como poderdante   

Nº apod.	Apoderado	Procedimiento	Fecha inicio	Fecha fin	Estado
2012/123455	Nombre Apellidos	PROC-01	10/12/2010	-	Pendiente
2011/124323	Nombre Apellidos	PROC-02	03/05/2011	03/05/2012	Confirmado

Apoderamientos como apoderado

Nº apod.	Poderdante	Procedimiento	Fecha inicio	Fecha fin	Estado
2011/234345	Nombre Apellidos	PROC-03	01/05/2011	-	Confirmado
2010/453454	Nombre Apellidos	PROC-04	23/09/2010	01/06/2011	Expirado
2012/003834	Nombre Apellidos	PROC-05	07/11/2011	-	Revocado

**Figura 23 – Interfaz gráfica Bandeja de apoderamientos**

Mediante esta pantalla el ciudadano podrá consultar sus registros de apoderamientos tanto en los que sea poderdante como apoderado. La pantalla mostrará por separado una tabla con los registros dónde actúa como poderdante y otra dónde actúa como apoderado.

Ambas tablas contarán con filtros específicos para cada columna. Las columnas mostrarán los datos del apoderamiento tales como número del apoderamiento, NIF/CIF del poderdante o apoderado, código y descripción del procedimiento, estado, fechas de vigencia, etc.

Otras opciones de las que dispondrá esta pantalla será un campo de texto que permitirá buscar por cualquier campo del apoderamiento y el acceso al alta de un nuevo apoderamiento.

## Detalle del apoderamiento



**Figura 24 – Interfaz gráfica Detalle del apoderamiento**

Permite consultar el detalle de un apoderamiento. Se mostrarán, entre otros, los siguientes datos del apoderamiento: número de referencia, estado, procedimiento, poderdante o apoderado, fechas de vigencia.

Además se mostrará una tabla con el histórico de estados por el que ha pasado el apoderamiento. Se recogerá además un listado de documentos asociados al apoderamiento los cuáles el usuario podrá descargar.

Las acciones que aparecerán para realizar sobre el apoderamiento dependerán del estado del mismo (*Más detalles en el diagrama de estados*).

Si el ciudadano es el **poderdante** podrá revocar el apoderamiento en cualquier momento independientemente del estado en el que se encuentre.

Si el ciudadano es el **apoderado** podrá realizar las siguientes acciones:

- Si el apoderamiento está en estado pendiente de confirmar podrá confirmar o rechazar el apoderamiento.
- Si el apoderamiento está en estado confirmado podrá renunciar al apoderamiento.
- Si el apoderamiento está en estado rechazado podrá confirmar el apoderamiento.
- Si el apoderamiento está en estado renunciado podrá volver a confirmar el apoderamiento deshaciendo la renuncia.

## Nuevo apoderamiento

ISMAEL CAMPANARIO CABRERA

### Nuevo apoderamiento

Poderdante: ISMAEL CAMPANARIO CABRERA

Apoderado:

NIF apoderado:

Procedimientos disponibles

	Código	Fecha inicio	Fecha fin	Descripción	Tipo
<input checked="" type="checkbox"/>	PROC-01	10/03/2012	10/05/2012	Procedimiento 01 de ejemplo	Trámites
<input checked="" type="checkbox"/>	PROC-02	23/03/2012	-	Procedimiento 02 de ejemplo	Notificaciones

**Volver**

**Confirmar**

**Figura 25 – Interfaz gráfica Nuevo apoderamiento**

El ciudadano que actúe como poderdante podrá crear un nuevo registro de apoderamiento introduciendo los datos necesarios en el siguiente formulario.

Los datos que debe facilitar el poderdante son:

- Nombre y apellidos del apoderado
- NIF/CIF del apoderado

Los datos del apoderado serán validados y se comprobará que existen mediante el uso del componente CitizensManager.

Además el poderdante deberá seleccionar el procedimiento o procedimientos a los que otorga el poder de representación. Esta representación puede tener una fecha de vigencia comprendida entre las fechas de inicio y fin que se especifican en el alta o una vigencia indefinida si no se especifica fecha fin.

El tipo apoderamiento podrá ser:

- Trámites: el apoderado sólo podrá realizar trámites como representante del poderdante.
- Notificaciones: el apoderado sólo podrá recibir notificaciones como representante del poderdante.
- Global: el apoderado podrá realizar trámites y recibir las notificaciones como representante del poderdante.

Una vez que se hayan completado todos los datos, se generará el correspondiente documento en pdf que deberá ser firmado digitalmente para completar el alta del apoderamiento.

## Firma digital del apoderamiento



**Figura 26 – Interfaz gráfica Firma digital del apoderamiento**

Mediante esta pantalla el ciudadano confirmará la acción que este llevando a cabo mediante el uso de la firma digital con su certificado digital de usuario.

Esta pantalla permitirá visualizar el documento generado en pdf que recogerá todos los datos del apoderamiento y permitirá iniciar la firma digital del mismo.

Al iniciar el proceso de firma se le solicitará al ciudadano que seleccione su certificado digital entre los que haya disponibles.

Otra opción que dispondrá esta pantalla es la descarga del documento que se va a firmar.

## Implementación

En el siguiente capítulo se indican las herramientas y tecnologías utilizadas durante la fase de implementación del proyecto. Además se dan las pautas a seguir para una correcta instalación y configuración del sistema.

### Requisitos técnicos y tecnologías usadas

A continuación se detallan los requisitos técnicos necesarios para un correcto funcionamiento del sistema, así como información sobre los componentes tecnológicos de los que se hace uso.

Los requisitos técnicos del sistema son:

Requisito	Descripción
Java	Java Runtime Environment 6
Base de datos	Postgresql 8.4
Servidor de aplicaciones	Apache Tomcat 7
Generación de documentos	OpenOffice ejecutado en modo servidor para la generación de plantillas <i>odt</i> en <i>pdf</i> .

Las tecnologías usadas para llevar a cabo el proyecto son:

Tecnología	Descripción
RichFaces 3.3	Para la capa de presentación, permite utilizar componentes gráficos JSF con AJAX, facelets, etc.
JBoss Seam 2	Para la capa de negocio.
JPA – Hibernate	Para la capa de persistencia.

Además de los requisitos técnicos y las tecnologías usadas para desarrollar el proyecto, se ha hecho uso de las siguientes herramientas:

- Eclipse 3.6 Helios, como entorno de desarrollo.
- MagicDraw, para los diagramas UML.
- Microsoft Visio, para los prototipos de las interfaces de usuario.
- OpenProj, para la planificación y construcción del diagrama de Gantt.
- OpenOffice Writer y Microsoft Word, para la edición y generación de documentos.



## Estructura del proyecto

En el presente capítulo se describe la estructura del proyecto, así como las dependencias necesarias para su compilación y ejecución.

### Código fuente y recursos

En la siguiente tabla se describen los directorios en los que se ubican los recursos más importantes de la aplicación.

Directorio	Descripción
<code>src\main\java</code>	Incluye todas las clases java que componen el proyecto.
<code>src\main\resources</code>	Ficheros de configuración de los distintos componentes y para las etiquetas de internacionalización.
<code>src\main\resources\public\form_es</code>	Plantillas de documentos OpenOffice para cada uno de los formularios que se generan en pdf.
<code>src\main\webapp</code>	Páginas para la capa de presentación de la aplicación.
<code>scripts</code>	Scripts de creación del modelo de datos.
<code>doc</code>	Javadoc de las clases
<code>lib</code>	Librerías necesarias para el proyecto, en caso de que no se use maven.
<code>war</code>	Incluye la aplicación empaquetada en un war (sin librerías para que el fichero de la entrega no ocupe demasiado).
<code>pom.xml</code>	Configuración maven del proyecto.

### Dependencias del proyecto

Para gestionar las librerías de terceros de las que depende el proyecto tenemos dos opciones, incluir las librerías directamente como dependencias del proyecto o gestionar las librerías con maven.

Cada opción tiene sus ventajas e inconvenientes, por ejemplo es más costoso incluir las librerías necesarias directamente en el proyecto, pero una vez que las tenemos sólo necesitamos estas. En cambio con maven tenemos la ventaja de que la gestión de las librerías es más fácil y rápida al incluirlas pero se requiere una gestión posterior para que las librerías que se incluyan en el empaquetado final sólo lleve las librerías necesarias y no aquellas dependencias transitivas que incluye maven aunque no las usemos en nuestro proyecto.

## Librerías necesarias

Las librerías necesarias para la compilación y ejecución del proyecto son las siguientes. Para cada una de ellas se indica el enlace de descarga para posteriormente incluirlas como dependencia del proyecto. Para facilitar la tarea de añadir las librerías de las que depende el proyecto, se ha subido a un servidor un fichero zip que incluye todas las librerías necesarias, <http://dl.dropbox.com/u/1504947/lib.zip>.

- RichFaces 3.3.3 (<http://www.jboss.org/richfaces/download/stable.html>)
- JBossSeam 2.2 (<http://seamframework.org/Seam2/Downloads>)
- JBoss Envers (<http://www.jboss.org/envers/downloads>)
- Hibernate (<http://www.hibernate.org/downloads>)
- JDBC Postgresql 8.4 (<http://jdbc.postgresql.org/download.html>)
- Log4j (<http://logging.apache.org/log4j/1.2/download.html>)
- OpenOffice Java (<http://www.openoffice.org/udk/java/man/>)
- JODConverter (<http://sourceforge.net/projects/jodconverter/files/JODConverter>)
- JODReports (<http://sourceforge.net/projects/jodreports/files/JODReports%202.4>)
- JSF 1.2 ([http://java.net/downloads/javaxserverfaces/release/1.2\\_15](http://java.net/downloads/javaxserverfaces/release/1.2_15))
- JTA 1.1 (<http://www.oracle.com/technetwork/java/javaee/jta/index.html>)
- JSON-lib (<http://sourceforge.net/projects/json-lib/files/json-lib/json-lib-2.0/>)
- JSTL (<http://java.sun.com/products/jsp/jstl/downloads/index.html>)
- JDOM (<http://www.jdom.org/downloads/index.html>)
- JavaBeans Activation Framework 1.0.2 ([http://download.oracle.com/otn-pub/java/jaf/1.0.2-fcs/jaf-1\\_0\\_2-upd2.zip](http://download.oracle.com/otn-pub/java/jaf/1.0.2-fcs/jaf-1_0_2-upd2.zip))
- Avalon Framework 4.2.0 (<http://archive.apache.org/dist/avalon/avalon-framework/v4.2.0/avalon-framework-4.2.0.zip>)
- Barcode4j (<http://sourceforge.net/projects/barcode4j/files/barcode4j/Barcode4J%202.0/barcode4j-2.0-bin.zip/download>)
- Bouncy Castle ([http://www.bouncycastle.org/latest\\_releases.html](http://www.bouncycastle.org/latest_releases.html))
- Apache Commons (BeanUtils, Codec, Collections, Configuration, DBCP, Digester, Discovery, Fileupload, IO, Lang, Logging, Pool) (<http://commons.apache.org/>)
- Dom4j (<http://sourceforge.net/projects/dom4j/files/dom4j/1.6.1/dom4j-1.6.1.jar/download>)
- FreeMarker (<http://sourceforge.net/projects/freemarker/files/freemarker/2.3.19/freemarker-2.3.19.tar.gz/download>)
- iText 2.0.8 (<http://itextpdf.com/>)
- EZMorph (<http://sourceforge.net/projects/ezmorph/files/ezmorph/ezmorph-1.0.2/ezmorph-1.0.2.jar/download>)
- Javassist (<http://www.jboss.org/javassist/downloads>)
- el-api (<http://download.java.net/maven/2/javax/el/el-api/1.0/el-api-1.0.jar>)
- el-ri (<https://nexus.sourcesense.com/nexus/content/groups/public/javax/el/el-ri/1.2/el-ri-1.2.jar>)
- clientsigner.jar (<http://dl.dropbox.com/u/1504947/clientsigner.jar>)

Además de los enlaces de descarga facilitados anteriormente, existe un buscador de librerías que es muy útil ([www.findjar.com](http://www.findjar.com)).

En la siguiente imagen se muestran todas las librerías de las que depende el proyecto.

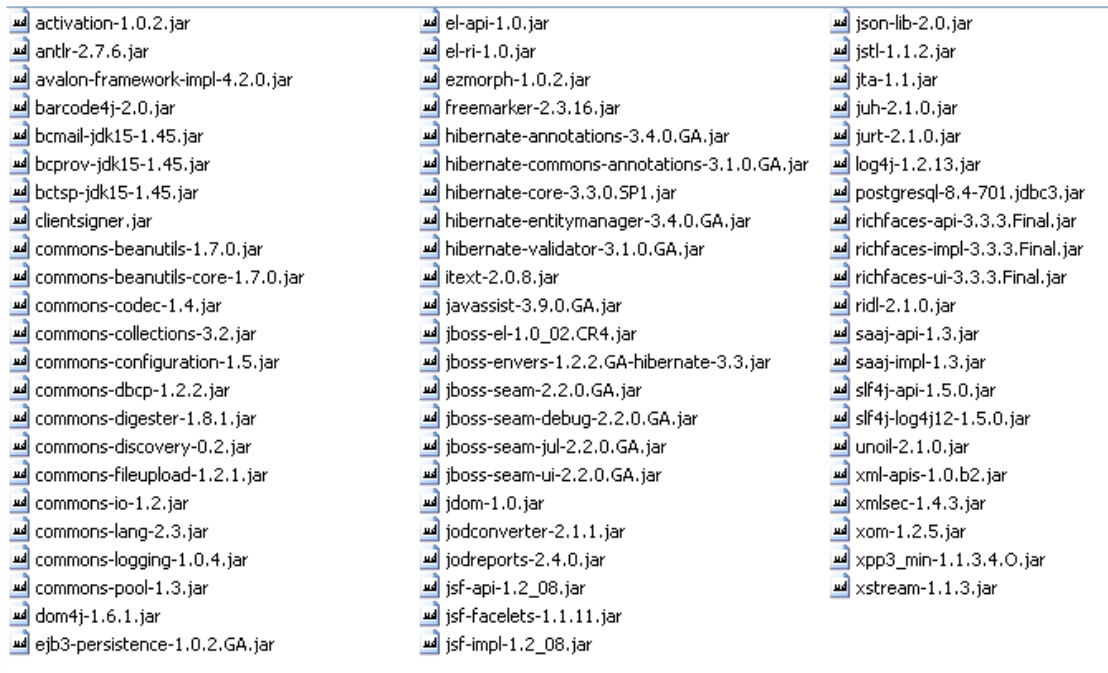


Figura 27 - Librerías del proyecto

## Configuración mediante maven

Para configurar el proyecto mediante maven sólo tenemos que usar el fichero pom.xml que se proporciona con el entregable del proyecto. En este fichero se definen las dependencias necesarias así como los repositorios para obtenerlas.

## Cliente de firma digital

La librería *clientsigner.jar* es un caso particular de dependencia. En este fichero se han empaquetado los recursos para utilizar el cliente de firma digital de la plataforma de @firma de administración electrónica

([http://administracionelectronica.gob.es/?nfpb=true&pageLabel=PAE\\_PG\\_CTT\\_General&langPae=es&iniciativa=138](http://administracionelectronica.gob.es/?nfpb=true&pageLabel=PAE_PG_CTT_General&langPae=es&iniciativa=138)).

Mediante una clase Java que funciona como filtro, definido en el web.xml de la aplicación, estos recursos del fichero *clientsigner.jar* se ponen a disposición de la capa de presentación para ejecutar el cliente de firma electrónica y solicite el certificado digital de usuario.

La estructura del fichero *clientsigner.jar* es la siguiente:

Nombre	Tamaño	Ruta
common-js	0	
authenticator.js	124	common-js\
clientSigner.js	42 061	common-js\
constantes.js	6 899	common-js\
cripto.js	2 669	common-js\
deployJava.js	36 243	common-js\
firma.js	5 279	common-js\
firmaWeb.js	6 370	common-js\
htmlEscape.js	724	common-js\
instalador.js	13 153	common-js\
json2.js	17 384	common-js\
pluginDetect.js	37 808	common-js\
signer.js	1 490	common-js\
styles.js	2 301	common-js\
time.js	1 257	common-js\
utils.js	1 175	common-js\
afirmaBootLoader.jar	45 313	
afirma_5_java_5.jar	307 757	
afirma_5_java_5.jar.pack.gz	89 113	
archDetector.jar	5 637	
COMPLETA_afirma.jnlp	755	
COMPLETA_j5_afirma5_coreV3.jar	2 089 128	
COMPLETA_j5_afirma5_coreV3.jar.pack.gz	779 477	
COMPLETA_j6_afirma5_coreV3.jar	1 584 271	
COMPLETA_j6_afirma5_coreV3.jar.pack.gz	635 695	
mscapi.zip	7 643	
mscapix64.zip	39 896	
msvcr71.zip	181 191	
NOTA.txt	66	
sunmscapi.jar	33 668	
version.properties	51	
xalan.zip	1 205 641	

**Figura 28 - clientsigner.jar**

## Instalación y configuración

La instalación y configuración del sistema consta de tres sencillos pasos: la instalación del modelo de datos, la configuración de los distintos ficheros y el despliegue del war en el servidor de aplicaciones.

### Instalación del modelo de datos

Para instalar el modelo de datos necesitamos una instancia del sistema gestor de base de datos PostgreSQL 8.4. La ejecución de los scripts se pueden llevar a cabo desde la consola de PostgreSQL o usando algún software de gestión del mismo, por ejemplo pgAdmin III que se incluye en la instalación de PostgreSQL 8.4.

La instalación del modelo de datos también se puede llevar a cabo automáticamente al desplegar la aplicación gracias al uso de JPA, pero en nuestro caso la crearemos usando los scripts proporcionados.

Una vez conectado a la base de datos debemos ejecutar el fichero `\scripts\grepresenta_db.sql` el cual creará la base de datos "GREPRESENTA" así como todos los objetos necesarios (tablas, secuencias, etc.).

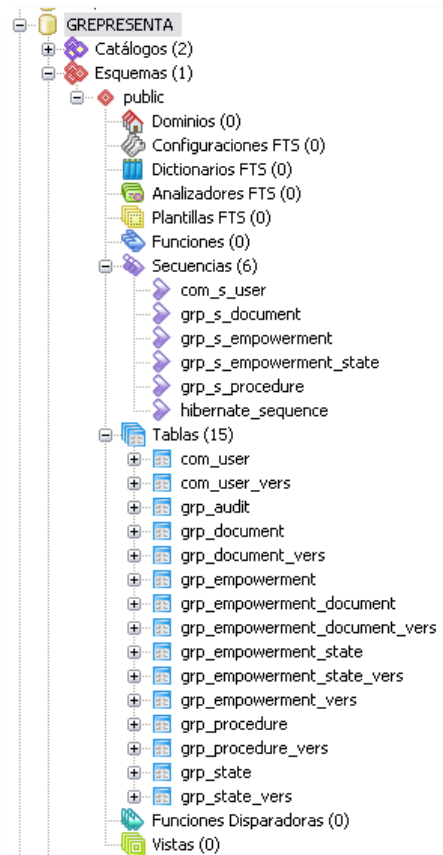


Figura 29 - Modelo de datos

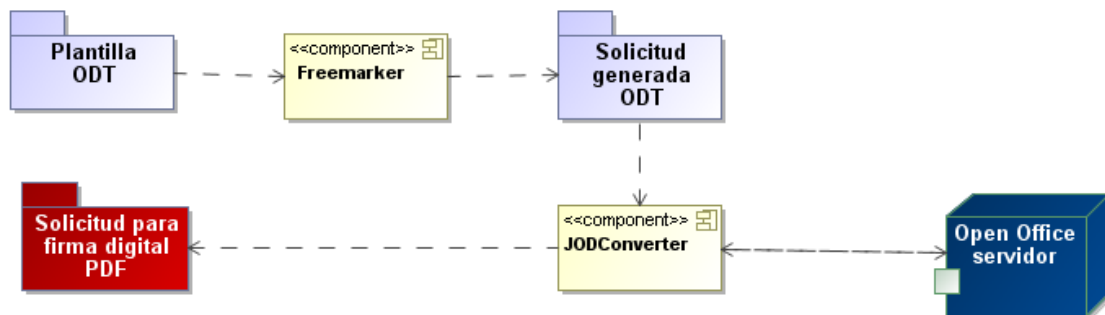
Además se debe ejecutar el fichero `\scripts\grepresenta_datos.sql` para disponer de una carga previa de datos.

## Generación de documentos

La generación de documentos se lleva a cabo en base a unas plantillas Open Office (.odt) que se ubican en el directorio `\WEB-INF\public\form_es`. Existe una plantilla para cada acción que puede realizar el usuario con la aplicación (otorgamiento, rechazo, renuncia, etc.).

Para la firma digital, se genera un documento PDF en base a las plantillas anteriormente mencionadas, previo paso por un proceso de sustitución de variables. La sustitución de variables se lleva a cabo mediante la librería *FreeMarker* y la generación a PDF mediante la librería *JODConverter*.

A continuación se muestra gráficamente el proceso de generación de los documentos PDF que firma digitalmente el usuario.



**Figura 30 - Generación PDF**

Para iniciar el OpenOffice en modo servidor es necesario ejecutar el siguiente comando desde el directorio *program* de la instalación de OpenOffice, `$OPENOFFICE_HOME\program\`

```
soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;" -nofirststartwizard
```

## Despliegue en servidor de aplicaciones

Para llevar a cabo el despliegue de la aplicación en el servidor Apache Tomcat 7 sólo es necesario copiar el fichero **grepresenta.war** en el directorio *webapps*, generalmente (`$TOMCAT_HOME\webapps`). Otra opción es utilizar la aplicación *manager* del Tomcat para llevar a cabo el despliegue. El war ya empaquetado se puede descargar de la siguiente dirección <https://dl.dropbox.com/u/1504947/grepresenta.war>

Una vez desplegada, se habrá creado una carpeta *grepresenta* que contendrá todos los ficheros de la aplicación. Para acceder a ella, suponiendo que la instalación del Tomcat se ha realizado en el puerto por defecto y el navegador desde el que accedemos se encuentra en la misma máquina, la url de acceso será <http://localhost:8080/grepresenta>

## Configuración de la aplicación

La aplicación dispone de distintos ficheros de configuración dependiendo de su finalidad. La configuración básica consta de configuración de datasource de conexión a la base de datos, configuración general, etiquetas para la internacionalización, trazabilidad del sistema y configuración de componentes

### Datasource de conexión a la base de datos

La configuración del datasource de conexión a la base de datos se lleva a cabo en el fichero `\META-INF\context.xml`. El contenido de dicho fichero es:

```
<Context path="/grepresenta" docBase="grepresenta">
  <Resource name="jdbc/grepresenta"
    auth="Container"
    type="javax.sql.DataSource"
    maxActive="20"
    maxIdle="6"
    maxWait="5000"
    username="postgres"
    password="postgres"
    driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://localhost:5432/GREPRESENTA"
    validationQuery="SELECT 1"/>
</Context>
```

Si la instancia de base de datos se encontrase en una máquina y puerto distintos de los configurados por defecto (`localhost:5432`) sólo sería necesario actualizar el valor del atributo `url`. Igualmente si variase el usuario y/o la clave del mismo simplemente sería necesario actualizar los atributos `username` y/o `password`.

### Configuración general

La configuración general de la aplicación se lleva a cabo mediante el fichero `\WEB-INF\classes\grepresenta.properties`. Este fichero de propiedades dispone de las siguientes entradas a configurar:

Propiedad	Descripción
<code>openoffice.url</code>	Máquina dónde se está ejecutando el OpenOffice en servidor. Su valor por defecto es <code>localhost</code> .
<code>openoffice.port</code>	Puerto dónde se está ejecutando el OpenOffice en servidor. Su valor por defecto es <code>8100</code> .
<code>sign.verify.url</code>	Url de verificación de firma que se muestra en el pie de la firma digital de un documento.

<b>sign.endpoint.server</b>	Nombre del servidor de firma digital que se muestra en la información de la firma digital del documento.
<b>sign.endpoint.text</b>	Texto que se muestra en la información de la firma digital del documento.
<b>documentManager.use</b>	Indica si se usa un gestor de documentos externos al del propio sistema. Por defecto su valor es <i>true</i> .

## Etiquetas para la internacionalización

La aplicación **grepresenta.war** está disponible en los siguientes idiomas: castellano, catalán, gallego, inglés y vasco. La configuración de cada etiqueta se lleva a cabo en el fichero correspondiente del idioma, cuyo nombre base es **messages\_XX.properties**.

Los ficheros se encuentran en **\WEB-INF\classes\messages\_XX.properties**.

## Trazabilidad del sistema

La configuración de la trazabilidad del sistema nos permite obtener información de la aplicación para detectar y corregir los distintos problemas que puedan surgir. Para llevar a cabo la trazabilidad se hace uso de la librería log4j y su configuración se realiza en el fichero **\WEB-INF\classes\log4j.xml**.

La ruta que viene configurada por defecto dónde se guardará el log es **\grepresenta.log**.

## Configuración de componentes

Mediante el fichero **\WEB-INF\classes\components.properties** podemos definir la configuración de los distintos componentes que usa el sistema (*Authenticator*, *Signer*, *CitizensManager* y *DocumentsManager*).

Esta configuración mediante componentes permite sustituir la implementación utilizada por el sistema para llevar a cabo la acción, dotando de versatilidad al sistema.



## Conclusiones

---

Durante la realización del presente TFC se ha desarrollado una aplicación que permite realizar el registro de representantes legales entre ciudadanos. Esta es una primera aproximación de lo que puede ser uno de los módulos de Administración Electrónica con los que cuente cualquier organismo o administración que necesite dar soporte a esta casuística.

El principal objetivo del TFC era el desarrollo de las habilidades adquiridas durante estos años de estudio de la Ingeniería Técnica en Informática de Gestión y acumular experiencia en el mundo de la tecnología JEE, objetivo que creo que he conseguido.

Además de afianzar conocimientos teóricos estudiados he logrado enfrentarme a situaciones que se producen en cualquier desarrollo de un producto software con los que nos encontraremos a diario durante nuestra dedicación profesional.

La evolución del desarrollo del producto obtenido como resultado de este TFC puede tomar varias alternativas, pero personalmente creo que entre las más interesantes está el desarrollo de unos servicios web que permitan consultar los apoderamientos y puedan ser consumidos por terceras aplicaciones para consultar los apoderamientos vigentes. Otra alternativa de desarrollo sería un módulo que permita la gestión de los apoderamientos de manera presencial y sea un gestor del organismo en el que se implante el que realice el alta de representación.

## Glosario

---

### @

**@firma:** Plataforma de validación y firma electrónica puesta a disposición de las Administraciones Públicas por el Ministerio de MAP (Ministerio de Administraciones Públicas) que proporciona servicios para la autenticación y firma electrónica.

### A

**ACN-XX:** Código para identificar los Actores de Negocio participantes en el sistema.

**AC-XX:** Código para identificar los Actores participantes en el sistema.

**Alfresco:** Sistema de gestión de contenido empresarial que permite la gestión de documentos.

**Apoderado:** Ciudadano que recibe el poder de representar a un tercero para gestionar sus trámites y/o notificaciones con la Administración.

### C

**Certificado digital de usuario:** Documento digital que contiene, entre otros, los datos identificativos del usuario. Este certificado permite al usuario identificarse en Internet e intercambiar información con otras personas con la garantía de que sólo Ud. y su interlocutor pueden acceder a ella.

**CLC-XX:** Código para identificar las Clases de Componentes del sistema.

**CLD-XX:** Código para identificar las Clases de Dominio del sistema.

**Cliente de firma @firma:** Applet Java que permite realizar la autenticación y firma electrónica en el navegador cliente.

**CLP-XX:** Código para identificar las Clases de Persistencia del sistema.

**CU-XX:** Código para identificar los Casos de Uso del sistema.

## D

**Datasource:** Es un nombre dado a la configuración de la conexión a una base de datos desde un servidor.

## I

**Internacionalización:** Proceso de diseñar y construir software de manera que pueda adaptarse a diferentes idiomas sin necesidad de realizar cambios. Comúnmente conocido por *i18n*.

## M

**Maven:** Herramienta de software para la gestión y construcción de proyectos Java.

## P

**Poder:** Acto o instrumento en que consta la facultad que alguien da a otra persona para que en lugar suyo y representándole pueda ejecutar algo.

**Poderdante:** Ciudadano que otorga el poder a un tercero para que lo represente en sus trámites y/o notificaciones con las Administración.

**PRU-XX:** Código para identificar las Pruebas Funcionales.

## R

**RA-XX:** Código para identificar los Requisitos de Almacenamiento.

**Repositorio:** Servidor centralizado que almacena las distintas librerías que usa el proyecto así como sus dependencias.

**RF-XX:** Código para identificar los Requisitos Funcionales.

**RNF-XX:** Código para identificar los Requisitos No Funcionales.

## Bibliografía

---

- Juntao Yuan, Michael y Heute, Thomas. JBoss® Seam: Simplicity and Power Beyond Java™ EE. Prentice Hall.
- Allen, Dan. Seam in Action. Manning
- JBoss Seam. <http://seamframework.org/>
- Cliente de firma electrónica de @firma. [http://administracionelectronica.gob.es/?\\_nfpb=true&\\_pageLabel=PAE\\_PG\\_CTT\\_General&langPae=es&iniciativa=138](http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=PAE_PG_CTT_General&langPae=es&iniciativa=138)
- Certificado digital de usuario. <http://www.cert.fnmt.es/index.php?cha=cit&sec=3&lang=es>
- RichFaces. <http://www.jboss.org/richfaces>
- RichFaces Live Demo. <http://livedemo.exadel.com/richfaces-demo/index.jsp>
- JBoss Envers. <http://www.jboss.org/envers>
- Bauer, Christian. Java Persistence with Hibernate. Manning.
- JPA e Hibernate. [http://wiki.cetechihuahua.gob.mx/index.php/JPA\\_con\\_Hibernate\\_en\\_JEE\\_5.0#Configurando\\_JPA\\_con\\_Hibernate](http://wiki.cetechihuahua.gob.mx/index.php/JPA_con_Hibernate_en_JEE_5.0#Configurando_JPA_con_Hibernate)
- Eclipse, JPA e Hibernate. <http://java-all-frameworks.blogspot.com.es/2011/06/configuracion-eclipse-jpa-hibernate.html>
- Oracle, JPA e Hibernate. <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>  
<http://www.hibernate.org/>
- Postgresql. <http://www.postgresql.org/docs/8.4/interactive/index.html>
- Apache Tomcat 7. <http://tomcat.apache.org/tomcat-7.0-doc/index.html>
- JODConverter. <http://artofsolving.com/opensource/jodconverter>
- Booch, Grady, Rumbaugh, James y Jacobson, Ivar. UML El lenguaje unificado de modelado. PEARSON Addison Wesley
- Gamma, Erich y otros. Patrones de Diseño. PEARSON Addison Wesley.

## ANEXO I. Manual de usuario

El siguiente anexo recoge una pequeña guía para el usuario de la aplicación, sobre las funcionalidades más relevantes de la misma.

### Requisitos del cliente

A continuación se indican los requisitos que debe cumplir el cliente para un correcto funcionamiento:

- Navegador web compatible con el cliente de firma digital de la plataforma @firma (por ejemplo Mozilla Firefox 8, Internet Explorer 7).
- Certificado digital de usuario.
- JRE. Java Runtime Environment 6 instalada en el navegador.
- Visualizador de documentos PDF, por ejemplo Acrobat Reader.

### Acceso al sistema

La pantalla de acceso al sistema permite al usuario autenticarse mediante su certificado digital de usuario que debe tener instalado en su navegador. Además permite consultar los procedimientos disponibles para los que se pueden crear apoderamientos.

Para llevar a cabo la autenticación, una vez cargada esta pantalla, se deberá seleccionar la opción "Acceso mediante certificado digital".



Figura 31 - Acceso al sistema

El sistema cargará la lista de certificados para que el usuario seleccione aquel con el que autenticarse.

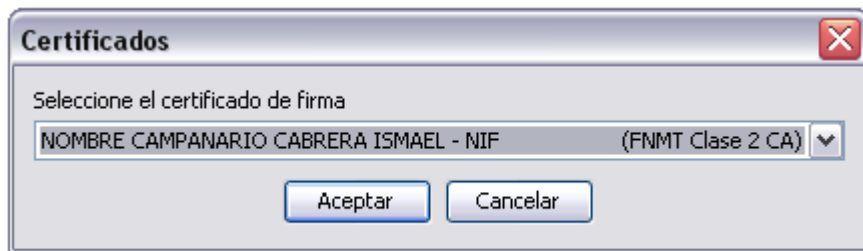


Figura 32 - Lista de certificados

Si la autenticación se lleva a cabo correctamente el usuario podrá consultar la bandeja de apoderamientos en la que se muestran todos los apoderamientos en los que participa como poderdante o apoderado.

**BANDEJA DE APODERAMIENTOS**

12345678Z - ISMAEL CAMPANARIO CABRERA

**ApoDERAMIENTOS como poderdante**

ApoDERADO	Nº Referencia	Procedimiento	Fecha inicio	Fecha fin	Estado
NOMBRE APELLIDO1 APELLIDO2	13/2012	Descripción procedimiento pruebas 03	20/05/2012		Pendiente
NOMBRE APELLIDO1 APELLIDO2	12/2012	Descripción procedimiento pruebas 03	30/04/2012		Pendiente
NOMBRE APELLIDO1 APELLIDO2	11/2012	Descripción procedimiento pruebas 02	27/04/2012		Revocado
NOMBRE APELLIDO1 APELLIDO2	10/2012	Descripción procedimiento pruebas 02	06/04/2012		Revocado

**ApoDERAMIENTOS como apoderado**

Poderdante	Nº Referencia	Procedimiento	Fecha inicio	Fecha fin	Estado
NOMBRE APELLIDO1 APELLIDO2	9/2012	Descripción procedimiento pruebas 01	06/04/2012		Revocado

Registro telemático de representantes legales

Figura 33 - Bandeja de apoderamientos

## Creación de un apoderamiento

Para crear un nuevo apoderamiento el usuario debe seleccionar la opción de “Nuevo apoderamiento” disponible desde la bandeja de apoderamientos.



Figura 34 - Icono nuevo apoderamiento

Una vez en la pantalla de creación del nuevo apoderamiento el usuario deberá buscar el apoderado, para ello deberá introducir su nombre, apellidos y NIF/CIF, para que el sistema valide si son correctos.

Figura 35 - Consulta de apoderado

Una vez seleccionado el apoderado, el usuario deberá seleccionar uno o varios procedimientos sobre los que le va a otorgar el poder. Si junto al código del procedimiento aparece el símbolo © indica que el apoderamiento necesita confirmación por parte del apoderado para que se confirme. Además por cada procedimiento podrá indicar la fecha de inicio, fecha final de vigencia (si se deja vacío el poder tiene carácter indefinido), y el tipo. El tipo puede ser:

- Global: permite la representación total sobre el poderdante.
- Trámites: permite realizar trámites en nombre del poderdante. Este tipo obliga al apoderado a confirmar el poder.
- Notificaciones: permite al apoderado recibir las notificaciones del poderdante.

<input type="checkbox"/>	Código	Fecha inicio	Fecha fin	Descripción	Tipo
<input type="checkbox"/>	PROC-01	jun 4, 2012		Descripción procedimiento pruebas 01	Trámites
<input type="checkbox"/>	PROC-02	jun 4, 2012		Descripción procedimiento pruebas 02	Trámites
<input type="checkbox"/>	PROC-03	jun 4, 2012		Descripción procedimiento pruebas 03	Trámites
<input type="checkbox"/>	PROC-04	jun 4, 2012		Descripción procedimiento pruebas 04	Trámites
<input type="checkbox"/>	PROC-05	jun 4, 2012		Descripción procedimiento pruebas 05	Trámites
<input type="checkbox"/>	PROC-06	jun 4, 2012		Descripción procedimiento pruebas 06	Trámites

Figura 36 - Nuevo apoderamiento

Una vez seleccionados los procedimientos sobre los que se va a otorgar el poder, se debe pulsar “Continuar” y accederemos a la pantalla de firma del apoderamiento.

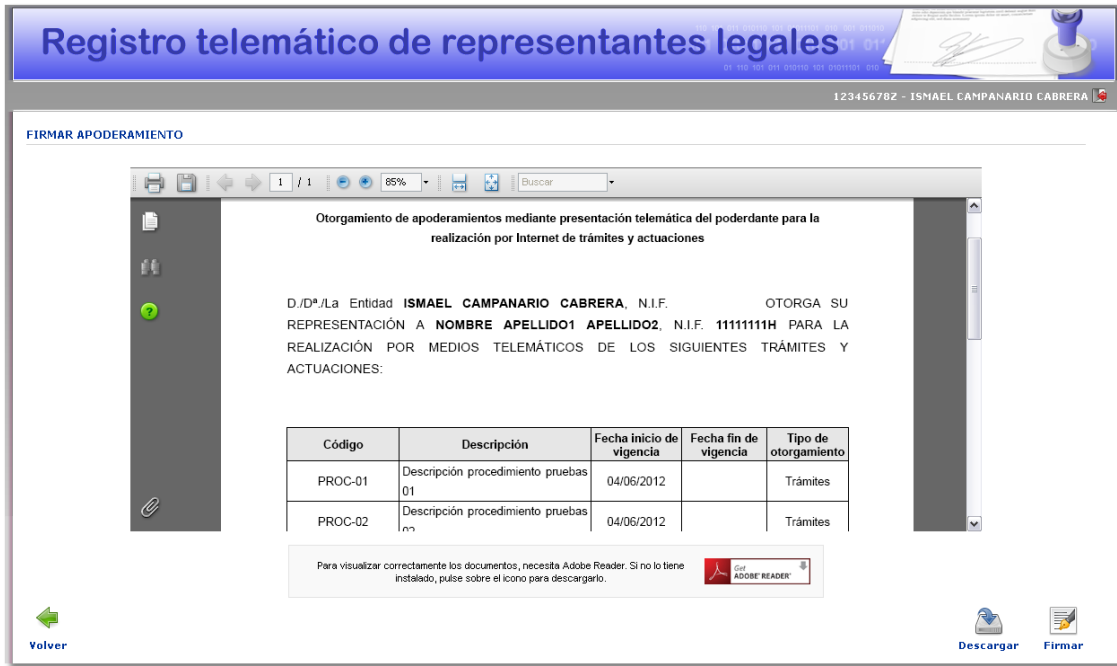


Figura 37 - Firmar apoderamiento

Desde esta pantalla el usuario podrá descargar el formulario mediante el que otorga el poder sobre los procedimientos seleccionados e iniciar el proceso de firma digital para dar validez al apoderamiento.

Al seleccionar la opción “Firmar”, el sistema solicitará el certificado digital de usuario con el que firmar la solicitud (debe ser el mismo que con el que se realizó la autenticación). Si todo el proceso de firma es correcto el usuario podrá visualizar y descargar el formulario firmado junto a su informe de firma.



Figura 38 - Apoderamiento firmado



A continuación se muestra el detalle de un apoderamiento firmado y su informe de firma en el que se recogen los datos de la firma digital. Si el apoderamiento necesita ser confirmado por el apoderado aparecerá con una marca de agua con el texto “Pendiente de confirmar”.



Figura 39 - Informe de firma digital

## Acciones sobre los apoderamientos

Desde la bandeja de apoderamientos, el usuario podrá acceder al detalle de cada uno de ellos y realizar las acciones que le permitan en función del estado en el que se encuentre (Ver *diagrama de estados*).

### Revocación de apoderamientos

El proceso de revocación de un apoderamiento se debe iniciar desde el detalle de un apoderamiento y sólo lo puede iniciar el poderdante. En la parte superior aparece la opción “Revocar”.



Figura 40 - Revocar apoderamiento

Una vez seleccionada esta acción, el sistema pedirá confirmación al usuario para iniciar el proceso de revocación.

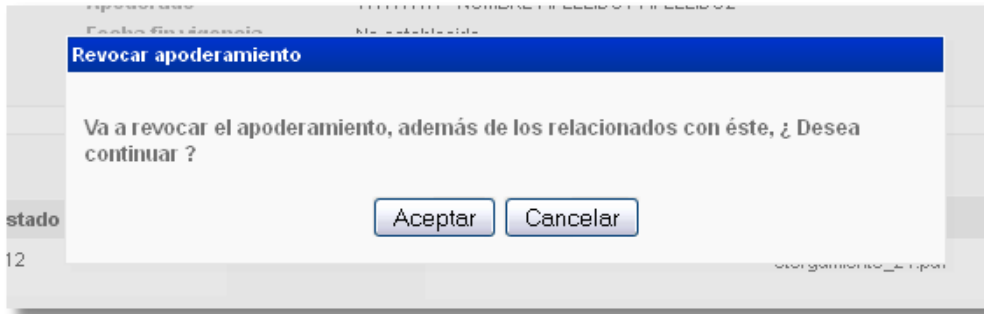


Figura 41 - Confirmación revocación apoderamiento

Si el usuario acepta revocar el procedimiento se iniciará un proceso de firma digital del formulario de revocación del procedimiento que concluirá con la validez del mismo.

## Confirmación o rechazo del poder otorgado

Para llevar a cabo la confirmación o rechazo de un poder otorgado, éste deberá estar en estado “Pendiente”. El apoderado debe decidir si acepta o rechaza el poder otorgado por el poderdante para el procedimiento o procedimientos indicados. Este proceso sólo se lleva a cabo con procedimientos que requieren confirmación.



Figura 42 - Confirmación o rechazo del poder otorgado

Si el usuario selecciona la acción de “Confirmar” o “Rechazar” se iniciará un proceso de firma digital que concluirá con el cambio de estado correspondiente del apoderamiento.

## Renuncia de poderes

La renuncia de poderes se lleva a cabo por parte del apoderado y consiste en desistir de los poderes que le otorgó otro ciudadano y éste aceptó. La diferencia entre rechazar y renunciar un apoderamiento consisten en que el apoderamiento rechazado nunca estuvo vigente ya que no fue confirmado, mientras que un apoderamiento renunciado si llegó a estar vigente.

Para llevar a cabo la renuncia del poder otorgado para uno o varios procedimientos, el usuario deberá seleccionar la acción “Renunciar” desde el detalle del apoderamiento.

The screenshot shows the 'Registro telemático de representantes legales' interface. At the top, there is a header with the title and a user profile '12345678Z - ISMAEL CAMPANARIO CABRERA'. Below the header, the main content area is titled 'DETALLES DE APODERAMIENTO' and features a 'Renunciar' button with a document icon. The page displays the following information:

Nº Referencia	14/2012	Estado actual	Confirmado
---------------	---------	---------------	------------

Procedimiento	PROC-01	Poderante	11111111H - NOMBRE APELLIDO1 APELLIDO2
Fecha inicio vigencia	04/06/2012	Fecha fin vigencia	No establecida
Vía presentación	Telemática		

Below this, there are two tables:

Estados del apoderamiento		Documentos del apoderamiento	
Estado	Fecha del estado	Nombre	Acciones
Confirmado	04/06/2012	rechazo_25.pdf	[Icon]
Rechazado	04/06/2012	otorgamiento_24.pdf	[Icon]
Pendiente	04/06/2012	confirmation_26.pdf	[Icon]

Apoderamientos relacionados			
Nº Referencia	Procedimiento	Fecha inicio	Fecha fin
15/2012	Descripción procedimiento pruebas 02	04/06/2012	

At the bottom left, there is a 'Volver' button with a green arrow icon. The footer of the page contains the text 'Registro telemático de representantes legales'.

**Figura 43 - Renuncia de poderes**

Una vez seleccionada esta acción se iniciará un proceso de firma digital que concluirá con la renuncia del poder otorgado.

Este proceso es reversible, es decir, el apoderado puede confirmar el apoderamiento en cualquier otro momento siempre y cuando el poderdante no lo revoque o expire la vigencia del mismo.

## ANEXO II. Pruebas funcionales

En el siguiente anexo se recogen las pruebas funcionales de cada uno de los casos de uso del sistema. En cada una de ellas se indican los pasos a realizar y el resultado esperado para que la prueba sea satisfactoria.

### PRU-01

PRU-01: Autenticación del ciudadano mediante certificado digital	
<b>Descripción</b>	Se realiza la autenticación del ciudadano mediante su certificado digital de usuario.
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe disponer de certificado digital.</li> <li>2. Acceder a la página de autenticación del sistema a través de la url <a href="http://localhost:8080/grepresenta">http://localhost:8080/grepresenta</a> (máquina y puerto que corresponda).</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. El usuario quedará autenticado en el sistema mostrándose su NIF junto a su nombre y apellidos en la parte superior derecha.</li> <li>2. Se cargarán los apoderamientos en los que el ciudadano es poderdante o apoderado.</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. El usuario seleccionará su certificado digital de entre los disponibles en el navegador y una vez autenticado correctamente accederá a la bandeja de sus apoderamientos.</li> </ol>

### PRU-02

PRU-02: Nuevo apoderamiento	
<b>Descripción</b>	El usuario creará un nuevo apoderamiento dando el poder para uno o varios procedimientos a otro ciudadano.
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> <li>2. Los usuarios poderdante y apoderado que participen en el nuevo apoderamiento deben existir en el sistema del que obtiene los datos el componente CitizensManager.</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. Se creará un nuevo apoderamiento que se mostrará en la bandeja de apoderamientos.</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Acceder al formulario de nuevo apoderamiento situado en la parte superior derecha.</li> <li>2. Introducir los datos del apoderado (NIF, nombre y apellidos) y pulsar en buscar para comprobar que existe.</li> <li>3. Seleccionar uno o varios procedimientos, indicando la fecha de vigencia para cada uno de ellos si es necesario, así como el tipo de otorgamiento.</li> <li>4. Pulsar “Continuar” para llevar a cabo la firma digital.</li> <li>5. Comprobar el documento que se va a firmar y pulsar sobre el icono “Firmar”.</li> </ol>

	<ol style="list-style-type: none"> <li>6. Seleccionar el certificado digital del usuario.</li> <li>7. Finalizada la firma, se muestra el documento firmado con su informe de firma.</li> <li>8. Volver a la bandeja de apoderamientos y comprobar que se ha creado un nuevo registro.</li> </ol>
--	--

**PRU-03**

PRU-03: Buscar apoderamientos	
<b>Descripción</b>	Búsqueda de apoderamientos desde la bandeja de apoderamientos mediante el campo de texto situado en la parte superior derecha.
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> <li>2. El usuario debe tener algún apoderamiento relacionado, ya sea como poderdante o apoderado.</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. La bandeja de apoderamientos mostrará sólo aquellos que cumplan con el filtro introducido en la búsqueda.</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Introducir el texto en el campo de búsqueda de la parte superior derecha.</li> <li>2. Pulsar sobre el icono de búsqueda (lupa).</li> <li>3. Comprobar que los resultados obtenidos cumplen el filtro indicado.</li> </ol>

**PRU-04**

PRU-04: Detalle de un apoderamiento	
<b>Descripción</b>	Consultar el detalle de un apoderamiento y las acciones que se pueden realizar sobre el mismo.
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> <li>2. El usuario debe tener algún apoderamiento relacionado, ya sea como poderdante o apoderado.</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. Se muestra el detalle del apoderamiento (datos generales, histórico de estados, documentos asociados, etc.).</li> <li>2. Se muestran las acciones que se pueden realizar sobre el apoderamiento en función del estado en el que se encuentre (Para más información ver diagrama de estados).</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Pulsar sobre cualquier apoderamiento de la bandeja de apoderamientos.</li> <li>2. Descargar los documentos del apoderamiento.</li> <li>3. Comprobar las acciones permitidas en función del estado, conforme a los definidos en el diagrama de estados.</li> </ol>
<b>Observaciones</b>	Para apoderamientos dónde el ciudadano sea poderdante podrá revocar. Si el ciudadano es apoderado podrá confirmar, renunciar o rechazar.

## PRU-05

PRU-05: Rechazo de un apoderamiento	
<b>Descripción</b>	El usuario que recibe el poder puede rechazar el apoderamiento que se encuentre en estado "Pendiente".
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> <li>2. El usuario debe tener algún apoderamiento en estado "Pendiente".</li> <li>3. El usuario debe acceder al detalle del apoderamiento (PRU-04).</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. El apoderamiento pasa a estado "Rechazado".</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Acceder al detalle de un apoderamiento cuyo estado permita rechazarlo.</li> <li>2. Pulsar sobre el icono "Rechazar".</li> <li>3. Descargar el formulario de rechazo para comprobar los datos.</li> <li>4. Pulsar "Firmar" para iniciar la firma del rechazo del apoderamiento.</li> <li>5. Seleccionar el certificado digital del usuario.</li> <li>6. Se muestra el documento de rechazo del apoderamiento firmado y el informe de la firma digital.</li> <li>7. Volver a la bandeja de apoderamientos y comprobar el estado.</li> </ol>

## PRU-06

PRU-06: Renuncia de un apoderamiento	
<b>Descripción</b>	El usuario que recibe el poder puede renunciar el apoderamiento que se encuentre en estado "Confirmado".
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> <li>2. El usuario debe tener algún apoderamiento en estado "Confirmado".</li> <li>3. El usuario debe acceder al detalle del apoderamiento (PRU-04).</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. El apoderamiento pasa a estado "Renunciado".</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Acceder al detalle de un apoderamiento cuyo estado permita renunciar.</li> <li>2. Pulsar sobre el icono "Renunciar".</li> <li>3. Descargar el formulario de renuncia para comprobar los datos.</li> <li>4. Pulsar "Firmar" para iniciar la firma de la renuncia del apoderamiento.</li> <li>5. Seleccionar el certificado digital del usuario.</li> <li>6. Se muestra el documento de renuncia al apoderamiento firmado y el informe de la firma digital.</li> <li>7. Volver a la bandeja de apoderamientos y comprobar el estado.</li> </ol>

## PRU-07

PRU-07: Revocación de un apoderamiento	
<b>Descripción</b>	El usuario que actúa como poderdante puede revocar el poder del apoderamiento.
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> <li>2. El usuario debe tener algún apoderamiento en algún estado que permite su revocación (Consultar diagrama de estados).</li> <li>3. El usuario debe acceder al detalle del apoderamiento (PRU-04).</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. El apoderamiento pasa a estado “Revocado”.</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Acceder al detalle de un apoderamiento cuyo estado permita revocarlo.</li> <li>2. Pulsar sobre el icono “Revocar”.</li> <li>3. Descargar el formulario de revocación para comprobar los datos.</li> <li>4. Pulsar “Firmar” para iniciar la firma de la revocación del apoderamiento.</li> <li>5. Seleccionar el certificado digital del usuario.</li> <li>6. Se muestra el documento de revocación del apoderamiento firmado y el informe de la firma digital.</li> <li>7. Volver a la bandeja de apoderamientos y comprobar el estado.</li> </ol>

## PRU-08

PRU-08: Cierre de sesión	
<b>Descripción</b>	El usuario cierra la sesión y abandona la aplicación.
<b>Prerrequisitos</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar autenticado (PRU-01).</li> </ol>
<b>Resultado esperado</b>	<ol style="list-style-type: none"> <li>1. Se finaliza la sesión del usuario y se vuelve a la pantalla de autenticación.</li> </ol>
<b>Pasos a realizar</b>	<ol style="list-style-type: none"> <li>1. Pulsar sobre el icono de salir de la aplicación que aparece en la parte superior derecha junto a los datos del usuario autenticado.</li> <li>2. Se muestra la pantalla de autenticación de la aplicación.</li> </ol>

## ANEXO III. Trazabilidad entre entidades

En este capítulo se contrasta que todos los objetivos del proyecto quedan recogidos en el catálogo de requisitos. También se recogen las dependencias entre los diferentes requisitos del sistema.

### Trazabilidad entre requisitos funcionales entre sí

A continuación se detalla la relación de dependencia existente entre los requisitos funcionales definidos en el catálogo.

	RF-01	RF-02	RF-03	RF-04	RF-05	RF-06	RF-07	RF-08	RF-09	RF-10	RF-11	RF-12	RF-13	RF-14
RF-01	-										X			
RF-02		-		X										
RF-03			-	X	X	X						X		
RF-04		X	X	-							X	X	X	X
RF-05			X		-	X						X		
RF-06			X		X	-	X	X				X		
RF-07						X	-		X	X				
RF-08						X		-			X		X	X
RF-09							X		-					
RF-10							X			-				
RF-11	X			X				X			-			
RF-12			X	X	X	X						-		
RF-13				X				X					-	
RF-14				X				X						-

### Trazabilidad entre requisitos funcionales y de almacenamiento

Se relacionan a continuación los requisitos funcionales con los requisitos de almacenamiento para facilitar el seguimiento de su trazabilidad.

	RF-01	RF-02	RF-03	RF-04	RF-05	RF-06	RF-07	RF-08	RF-09	RF-10	RF-11	RF-12	RF-13	RF-14
RA-01		X	X	X	X	X							X	X
RA-02			X			X		X						X
RA-03			X	X	X	X		X					X	X
RA-04				X		X								X
RA-05			X	X		X	X		X	X	X			X
RA-06			X	X		X	X		X	X	X			X



## Trazabilidad entre requisitos funcionales y casos de uso

A continuación se contrastan los requisitos funcionales con los casos de uso existentes en el sistema.

	RF-01	RF-02	RF-03	RF-04	RF-05	RF-06	RF-07	RF-08	RF-09	RF-10	RF-11	RF-12	RF-13	RF-14
CU-01	X	X												
CU-02		X		X										
CU-03			X		X	X						X		
CU-04		X	X	X			X	X	X		X		X	X
CU-05				X					X		X			X
CU-06				X			X		X	X				X
CU-07								X			X		X	X
CU-08								X			X		X	X
CU-09								X			X		X	X
CU-10								X			X		X	X
CU-11		X	X			X				X		X		

## Trazabilidad entre casos de uso y clases

En este apartado se relacionan los distintos casos de uso con las clases del modelo de dominio.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09	CU-10	CU-11
CLD-01	X	X		X							X
CLD-02				X	X	X	X	X	X	X	X
CLD-03				X			X	X	X	X	X
CLD-04				X			X	X	X	X	X
CLD-05			X	X			X	X	X	X	X
CLD-06			X	X			X	X	X	X	X
CLD-07			X	X	X	X					X
CLP-01				X							
CLP-03			X	X	X	X	X	X	X	X	X

## Trazabilidad entre requisitos de almacenamiento y clases

En este apartado se relacionan los requisitos de almacenamiento con las clases del modelo de dominio.

	CLD-01	CLD-02	CLD-03	CLD-04	CLD-05	CLD-06	CLD-07	CLP-01	CLP-03
RA-01	X	X	X	X	X	X	X	X	X
RA-02				X					X
RA-03				X	X	X			X
RA-04			X	X				X	X
RA-05		X		X			X		X
RA-06		X		X			X		X