



UNIVERSIDAD CARLOS III DE MADRID
Ingeniería Técnica en Informática de Gestión

Diseño e implementación de un Sistema de información centralizado del Ciudadano

PROYECTO FIN DE CARRERA

Madrid, Mayo de 2014

Autor: Julio Francisco Vicente Sánchez

Tutor: Javier Fernández Muñoz

Título: Diseño e implementación de un Sistema de información centralizado del Ciudadano

Autor: Julio Francisco Vicente Sánchez

Director: Javier Fernández Muñoz

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 201_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Resumen

Este proyecto describe el desarrollo de un Sistema de información centralizado del Ciudadano en la Administración Pública. Con este Sistema el ciudadano podrá acceder a una aplicación central para consultar el estado en que se encuentran todos los trámites que tiene abiertos con la Administración. El Sistema se diseñará basándose en una Arquitectura Orientada a Servicios (SOA en inglés), existiendo una base de datos central donde se registrará la información suministrada por el resto de Sistemas de Información de la Administración. El presente documento detalla el proceso seguido para la realización de este proyecto.

Índice de contenidos

Introducción.....	9
1.1	Visión general..... 10
1.2	Motivación 12
1.3	Objetivos 13
1.4	Fases del desarrollo 14
1.5	Estructura del documento 16
Estado del arte.....	17
2.1	Evolución de los Sistemas de Gestión en la Administración Pública..... 18
2.2	Ejemplos de utilización de arquitectura SOA 21
2.2.1	Ejemplos de uso de la arquitectura SOA 21
2.2.2	Historia Clínica Digital del Sistema Nacional de Salud (HCDSNS) 22
2.3	Tecnologías utilizada 25
2.3.1	Sistema operativo 25
2.3.2	Herramientas genéricas de desarrollo 29
2.3.3	Herramientas específicas 31
Análisis.....	37
3.1	Alcance del sistema..... 38
3.2	Establecimiento de requisitos 39
3.2.1	Requisitos de usuario 39
3.2.2	Requisitos de software 45
3.2.3	Matriz de trazabilidad..... 50
3.3	Especificación de casos de uso..... 51
3.3.1	Subsistema de Control de Acceso..... 52
3.3.2	Funcionalidades del Portal del Ciudadano..... 54
3.3.3	Subsistema de Administración del sistema..... 58
Diseño.....	61
4.1	Arquitectura del sistema 62
4.2	Diagrama de componentes del sistema 65
4.3	Elaboración del modelo de procesos 70
4.3.1	Acceso al Sistema 70
4.3.2	Validación de XML..... 71
4.3.3	Consulta de asuntos 72
4.3.4	Alta de asunto..... 73
4.3.5	Cambio estado de asunto..... 74
4.3.6	Eliminación de asunto..... 76
4.3.7	Mantenimiento usuarios / aplicación 77
4.4	Diseño de la base de datos 79
4.5	Diagrama de navegación 84
4.5.1	Aplicación de administración 84
4.5.2	Aplicación de gestión..... 85
4.5.3	Portal del ciudadano 88
Implementación y plan de pruebas.....	89
5.1	Implementación 90
5.1.1	Aplicaciones Web 90
5.1.2	Servicio Web 97
5.2	Plan de pruebas del Sistema de Información..... 109

5.2.1	Pruebas de Sistema	109
5.2.2	Pruebas de Integración.....	110
5.2.3	Pruebas Unitarias y de aceptación	111
5.2.4	Matriz de trazabilidad de las pruebas	119
Planificación y presupuesto		121
6.1	Consideraciones iniciales	122
6.2	Planificación	123
6.3	Coste de personal.....	125
6.4	Coste hardware	127
6.5	Coste de software.....	128
6.6	Coste total del proyecto	129
Conclusiones y acciones futuras		131
7.1	Conclusiones del proyecto.....	132
7.2	Conclusiones personales.....	134
7.3	Acciones futuras	135
Bibliografía.....		137
Glosario.....		141
Manual de Usuario del Subsistema de Administración		143
Manual de Usuario del Subsistema de Gestión.....		153

Capítulo 1

Introducción

En este capítulo se detallará el objetivo que pretende cubrir con este proyecto. Se abordarán las razones que lo han motivado y se detallará el proceso que se seguirá a lo largo del desarrollo del proyecto.

1.1 Visión general

El objetivo de este proyecto es el desarrollo de un Sistema de Información centralizado del Ciudadano, que permita a un ciudadano acceder por medios electrónicos (Internet) al estado en que se encuentra cualquier trámite realizado con una Administración Pública (Ayuntamiento, Ministerio,...). Es decir, por ejemplo, poder consultar por internet el estado en que se encuentra una solicitud de una beca, una subvención, una reclamación, etc..

En la ley 11/2007 se reconoce el derecho de los ciudadanos a relacionarse con las Administraciones Públicas por medios electrónicos y regula los aspectos básicos de la utilización de las tecnologías de la información en la actividad administrativa, en las relaciones entre las Administraciones Públicas, así como en las relaciones de los ciudadanos con las mismas con la finalidad de garantizar sus derechos, un tratamiento común ante ellas y la validez y eficacia de la actividad administrativa en condiciones de seguridad jurídica. [1]



Figura 1. Ciudadanos realizando trámites.

Además independientemente de como iniciase la gestión el ciudadano, ya sea en papel o de forma electrónica, debe poder acceder al estado en el que se encuentra su trámite por medios electrónicos.

Por tanto, partimos de un punto inicial en el cuál una Administración Pública que posee de múltiples Sistemas de Información para dar servicio a los ciudadanos en los distintos ámbitos de actuación en los que dicha Administración tiene competencias.

Pero también hay que tener en cuenta que es posible que existan trámites administrativos que aún no tengan detrás un Sistema de Información para gestionarlo. Se cual sea el caso, existe la obligación de adaptarse a esta norma regulatoria.

Partiendo de esta situación inicial debemos abordar el desarrollo de un Sistema de Información que permita ofrecer al ciudadano por medios electrónicos (Internet) el acceso al estado de los distintos trámites que tiene abiertos con la Administración. Y, como decíamos, puede ser posible que alguno de estos trámites no tenga un Sistema de Información que realice la gestión.

Debe ser una solución lo menos costosa posible, tanto a nivel económico como de recursos que debe utilizar la organización.

Hemos escogido como título del proyecto “Sistema de Información centralizado del Ciudadano”, ya que el Sistema debe ser el único punto en el que el ciudadano tenga que consultar la información (y no tener múltiples portales donde intentar encontrar la información que busca).

1.2 Motivación

Como hemos avanzado en el punto anterior, la motivación de este proyecto es adaptarse a los cambios regulatorios que obligan a ofrecer un servicio a los ciudadanos.

Pero sería corto de miras quedarse en este “obligatoriedad” como la motivación de este proyecto. La motivación debe ser más ambiciosa: ofrecer un mejor servicio y de calidad a los ciudadanos, facilitándole la información que necesita de la forma más sencilla y rápida.

Si nos centrásemos sólo en “cumplir el expediente” correríamos el riesgo en realizar un Sistema de Información de mínimos que nos serviría para cumplir con la regulación pero seguramente no fuese el que nos ofreciese la mejor solución tanto a nivel organizativo como en la calidad del servicio a los ciudadanos.

La principal motivación será dar acceso al ciudadano a todos los trámites que tenga abiertos con la Administración. Pero este acceso debe ser lo más simple y centralizado posible, y no limitarnos a dar un punto de acceso a la información para cada Sistema de Información posible. El ciudadano tiene que ser ajeno a la complejidad interna de las administraciones y acceder de la forma más sencilla y rápida a la información que necesita.

Una motivación secundaria, pero igual de importante, es que el Sistema de Información que se desarrolle tiene que permitir el acceso a los datos distribuidos en el resto de Sistemas de los que dispone la administración de la forma más simple posible. Cuando más complicado sea, más trabajo implicará al resto de Sistemas y ello repercutirá en un mayor coste.

1.3 Objetivos

El objetivo del proyecto es crear un Sistema de Información que permita poner a disposición de los ciudadanos un acceso único a la información del estado en que se encuentra cualquier trámite que tenga con la Administración.

Para conseguir este objetivo principal, podemos establecer los siguientes objetivos intermedios:

- Desarrollar un portal web al cuál se conectará el ciudadano para acceder a toda la información disponible. De esta forma el ciudadano tendrá un único punto de acceso a la información de toda nuestra Administración y lo hará más accesible.
- Desarrollo de una interfaz que sea capaz de comunicar al resto de Sistemas de Información de nuestra administración con el **Sistema de Información centralizado del Ciudadano**. El portal del ciudadano, tiene que ser capaz de acceder a la información distribuida en el resto de Sistemas que son realmente los que llevan la gestión de los trámites administrativos. Esta interfaz se realizará siguiendo una arquitectura SOA (Arquitectura Orientada a Servicios), desarrollando un servicio web que al que se conectarán el resto de Sistemas de Información para comunicar la situación de los trámites administrativos que gestionan.

Como se ha mencionado anteriormente, puede suceder que existan trámites administrativos que aún no tengan un Sistema de Información. También puede suceder que si exista un Sistema de Información pero que por falta de soporte por parte de la empresa proveedora o porque sea una tecnología obsoleta, no pueda ofrecer los datos al futuro Sistema que vamos a desarrollar.

Por ello habrá que desarrollar una aplicación web lo más simple que se pueda, que sirva para comunicarse con nuestro **Sistema de Información centralizado del Ciudadano**. Este sistema permitirá registrar los distintos estados por los que va pasando un trámite, solicitando únicamente la información mínima que necesita nuestro Sistema.

1.4 Fases del desarrollo

La primera de estas fases coincide con el estudio de las tecnologías, plataformas y productos existentes con los cuáles podríamos abordar el desarrollo del proyecto. Esta fase en nuestro caso estará condicionada por las políticas internas de la Administración, ya que dispone de una serie de herramientas corporativas que encorsetan nuestra capacidad de decisión.

Pero no sólo debemos adecuarnos a las políticas internas en cuanto a las herramientas a utilizar, sino que también existen unas políticas de desarrollo de aplicaciones. Estas políticas permiten que dentro de una misma organización las aplicaciones desarrolladas sean lo más homogéneas posible, lo cuál redundará en una mejor calidad y más facilidad de mantenimiento con un menor coste. Pero a la vez nos obligan a restringirnos a un entorno tecnológico bastante cerrado.

Una vez seleccionadas las tecnologías a utilizar comienza la fase de toma de requisitos. Para ello deben tenerse en cuenta todos los *stakeholders* posibles.

En la gestión de proyectos, los involucrados o interesados ("stakeholders" en inglés) son todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa.



Figura 2. Stakeholders

Una buena planificación de proyectos debe involucrar la identificación y clasificación de los interesados, así como el estudio y la determinación de sus necesidades y expectativas. [2]

Los stakeholders no tiene porque ser sólo personas, sino también pueden ser normativas, regulaciones... que condicionan el Sistema Informático a desarrollar.

Una vez llegado a este punto se puede comenzar la fase del diseño de la arquitectura del sistema. Es decir, que componentes integran cada el sistema, que requisitos se cubren con cada componente, etc...

Terminado el diseño de la arquitectura comenzaría la implementación de la solución propiamente dicha. Durante la fase de implementación se irán realizando las pruebas unitarias necesarias para comprobar el correcto comportamiento.

Una vez terminada la implementación, se continuaría con la fase de pruebas globales del sistema para garantizar que se cubren los objetivos que se pretendían. Durante esta fase es posible que aparezcan errores que provoquen modificaciones en pasos previos (análisis, diseño..) con lo cuál se realizarían ciclos iterativos.

1.5 Estructura del documento

Para facilitar la lectura de este documento se detalla a continuación la estructura de los distintos capítulos que lo componen.

Capítulo 1. En este capítulo se detallará el objetivo que pretende cubrir con este proyecto. Se abordarán las razones que lo han motivado y se detallará el proceso que se seguirá a lo largo del desarrollo del proyecto.

Capítulo 2. En este capítulo se abordarán posibles soluciones al problema para determinar cuál de ellas es la solución que mejor se adapta a los requisitos específicos de este proyecto.

Capítulo 3. En este capítulo se tratarán las fases de toma de requisitos y análisis de los distintos subsistemas de los que se compone en Sistema de Información.

Capítulo 4. En este capítulo se abordará el diseño de cada uno de los subsistemas que componen el Sistema de Información.

Capítulo 5. En este capítulo se abordará el enfoque seguido en la implementación del Sistema de Información. Así mismo se detallará el plan de pruebas diseñado para asegurar el correcto funcionamiento.

Capítulo 6. En este capítulo se detallará la planificación seguida durante el proyecto y se realizará una estimación del presupuesto del proyecto en función de dicha planificación.

Capítulo 7. En este capítulo se detallarán las conclusiones extraídas de la realización de este proyecto y se estudiarán posibles acciones futuras que podrían ser de interés para mejorar el Sistema de Información.

Capítulo 2

Estado del arte

En este capítulo se abordarán posibles soluciones al problema para determinar cuál de ellas es la solución que mejor se adapta a los requisitos específicos de este proyecto.

2.1 Evolución de los Sistemas de Gestión en la Administración Pública

Durante la última década se ha producido una reforma dentro del sector de la Administración Pública centrada en mejorar el servicio al ciudadano aprovechándose de la evolución tecnológica que se ha producido en los últimos años.

Esta reforma se ha impulsado fundamentalmente en:

- **Normativas.** Como comentamos al principio la Ley para el Acceso Electrónico de los Ciudadanos a los Servicios hace inevitable que se aborde esta reforma.
- **Evolución de la tecnología.** El desarrollo de la sociedad de la información y el grado de penetración en los hogares españoles, hace posible que se puedan ofrecer servicios que hace relativamente pocos años eran impensables.
- **Demanda de los ciudadanos.** Los ciudadanos cada vez son más exigentes en la calidad el servicio que esperan por parte del sector privado y, también, público.

La utilización de las nuevas tecnologías para mejorar la calidad del servicio que se ofrecer a los ciudadanos, también reporta en grandes beneficios para las administraciones:

- Aumentar la **eficiencia** reduciendo las tareas mecánicas al automatizar los procesos de gestión.
- Aumentar el nivel en la **calidad de la gestión** de la información centrándose en dar valor añadido al eliminarse las tareas mecánicas.
- Mayor **calidad en el servicio al ciudadano.** La relación de los ciudadanos con las administraciones son más rápidas, sencillas y cercanas.
- **Reducción de costes.** Una mejora en la eficiencia y calidad de la gestión, repercuten inevitablemente en una reducción de costes.

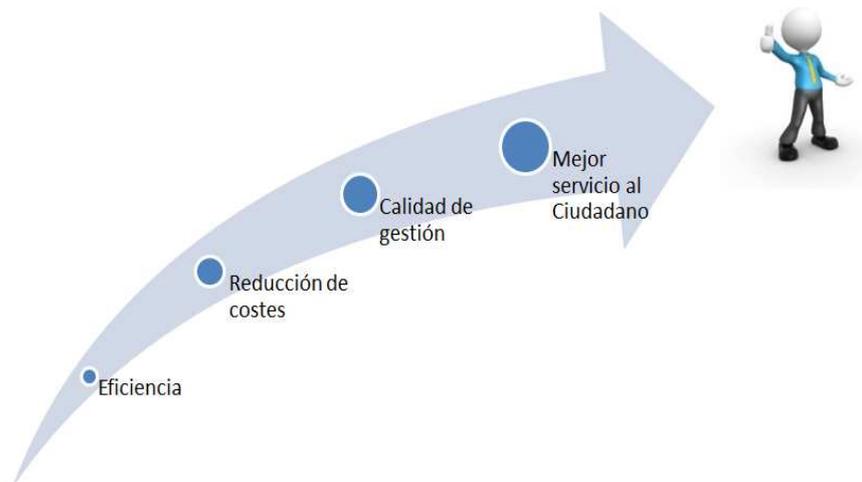


Figura 3. Beneficio TIC en las AAPP

La implantación de los múltiples sistemas de información que se han ido desarrollando para dar servicio a los ciudadanos ha provocado que la complejidad sea tan grande que sea necesaria la incorporación de nuevas herramientas que faciliten la gestión.

En este punto la Arquitectura Orientada a Servicios (SOA en sus siglas en inglés) se ha posicionado como una arquitectura capaz de integrar todas las iniciativas y sistemas existentes para la prestación de todos los servicios de la Administración Electrónica.

Además, la implantación de esta arquitectura permite la interoperabilidad entre distintos Sistemas de Información de una misma organización y, así mismo, con los Sistemas de Información de otras Administraciones Públicas o del sector privado.

Facilitar la interoperabilidad repercute también en aumentar la eficiencia y la reducción de costes. De este modo se eliminan duplicidades de Sistemas similares que se desarrollan en múltiples administraciones. Un ejemplo de ello es por ejemplo la plataforma de validación @firma.

@firma una solución tecnológica desarrollada por el Ministerio de Hacienda y Administraciones Públicas. Es una plataforma de validación de certificados digitales y firma electrónica que se pone a disposición de las Administraciones Públicas, proporcionando servicios para implementar la autenticación y firma electrónica avanzada de una forma rápida y efectiva.

Esta solución permite que cualquier Sistema de Información de cualquier administración pública pueda utilizar este servicio delegando la validación de certificados, firmas... en este sistema de @firma. Ello supone no sólo el ahorro en el desarrollo de un sistema similar, sino más importante aún, en el ahorro en el mantenimiento y actualización del sistema.

2.2 Ejemplos de utilización de arquitectura SOA

A continuación detallaremos algunos ejemplos de Organizaciones que utilizan la arquitectura SOA para desarrollar sus Sistemas de Información. Contemplaremos varios ámbitos sectoriales para ver que esta arquitectura puede abarcar cualquier tipo de sector: sanidad, turismo y banca.

Posteriormente detallaremos un Sistema Informático que utiliza una solución similar a la que proponemos en este proyecto. Usaremos de ejemplo el Sistema de Historia Clínica Digital del Sistema Nacional de Salud (HCDSNS).

2.2.1 Ejemplos de uso de la arquitectura SOA

Harvard Medical School

Esta institución y sus hospitales afiliados utilizan una arquitectura SOA para desarrollar sus Sistemas de Información desde hace más de una década.

En 2005 disponían ya de 25 servicios Web que se utilizaban en aplicaciones de más de 400 diferentes departamentos. Estos servicios iban desde servicios orientados al tratamiento de los pacientes, como también a servicios dedicados para su funcionamiento interno (recursos humanos...). Ya por entonces estimaban en 1 millón de dólares de ahorro gracias al uso de la arquitectura SOA. [3]



Starwood Hotels and Resorts Worldwide, Inc.

Esta cadena hotelera implantó una arquitectura SOA para el sistema de reservas. De forma centralizada desarrollaron los servicios necesarios para este sistema de reservas, que es utilizado desde todo el grupo. Utilizar este tipo de arquitectura les permite escalar de forma adecuada los sistemas en función de la demanda puntual en los periodos estacionales.



Las estimaciones de utilizar esta arquitectura frente a otras fue que se reducían en 20 millones de dólares los costes operacionales anuales. [4]

BIAN

BIAN es Banking Industry Architecture Network's, un organismo que busca fomentar el uso de estándares en el sector financiero.



BIAN promueve fuertemente la utilización de la arquitectura SOA y su implantación como estándar en la banca mundial. [5]

2.2.2 Historia Clínica Digital del Sistema Nacional de Salud (HCDSNS)

La HCDSNS forma parte del conjunto de proyectos denominado Sanidad en Línea, orientada a contribuir a la generalización del uso de las nuevas tecnologías en el sistema sanitario para mejorar la atención a pacientes y ciudadanos. [6]

Los objetivos de este proyecto es compartir dentro del Sistema Nacional de Salud una serie de documentos clínicos acordados por un grupo de expertos encomendados por el Ministerio de Sanidad Servicios Sociales e Igualdad (MSSSI). Estos documentos podrán ser accedidos tanto por ciudadanos como por profesionales (en este caso siempre bajo demanda del usuario desde un centro sanitario público del SNS). De esta forma se permitirá que a pesar de que el ciudadano se desplace dentro del territorio nacional, cualquier profesional sanitario pueda acceder a la información clínica relevante del ciudadano sea cuál sea la Comunidad en donde se encuentre dicha información.

Debido a la criticidad y naturaleza de los datos de carácter personal que se comparten en este sistema, se establecieron una serie de requisitos que garantizaran que no se producen accesos indebidos a los datos clínicos de los ciudadanos. Uno de estos elementos de seguridad fue incluir dentro de este sistema un registro de accesos a los datos clínicos de los ciudadanos, para que estos puedan saber quién ha accedido a sus datos clínicos y, en caso de considerar que el acceso fue ilícito, poder tomar las acciones que crea oportunas.

Este Sistema de HCDSNS se desarrolló siguiendo una arquitectura SOA. Se estableció un nodo central en el MSSSI el cuál actúa como nodo de intercambio entre los distintos nodos autonómicos que son los que dan finalmente el acceso a los profesionales y ciudadanos.

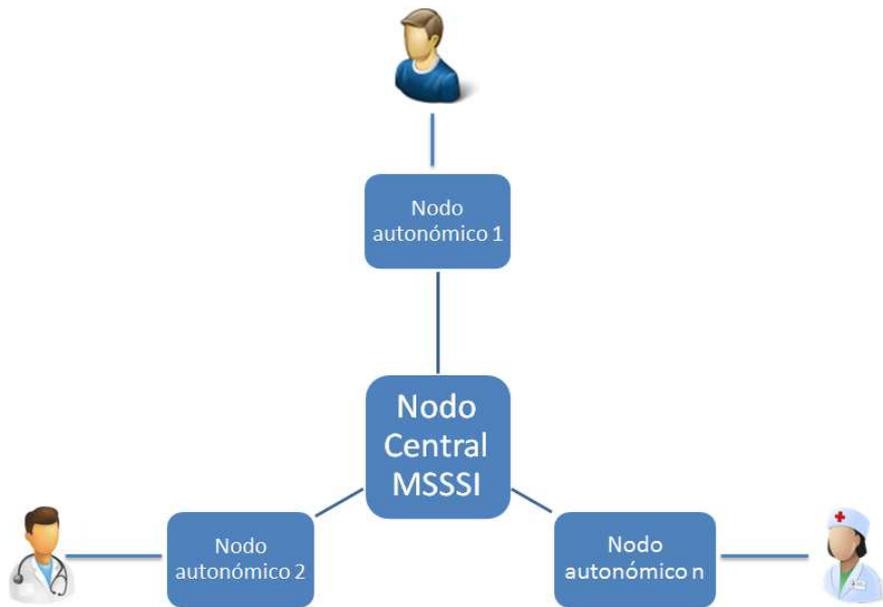


Figura 4. Esquema interoperabilidad HCDSNS

Los ciudadanos y profesionales acceden a través de los nodos autónomos para acceder a la información. Esta información reside en las Comunidades Autónomas que son “dueñas” de la información. Por tanto, cuando se desea consultar un informe clínico que pertenece a una CA distinta se solicita esta información a través del nodo central del MSSSI que a su vez enviará la consulta a la CA en cuestión y una vez tenga la respuesta la reenviará a la CA solicitante. Todas estas comunicaciones se realizan haciendo llamadas a Servicios Web. [7]

Dentro de este Sistema de HCDSNS, nos centraremos en el subsistema que es más similar a la solución que proponemos para nuestro proyecto: el registro de accesos a los datos clínicos de los ciudadanos.

Este subsistema funciona de forma similar a la que proponemos. Existe un registro de todos los accesos producidos al sistema en el nodo central del MSSSI. Cuando un ciudadano desea consultar los accesos producidos a sus datos, debe conectarse a través del nodo autónomo de su CA. Desde este nodo autónomo se enviará una consulta al nodo central del MSSSI (en concreto a un servicio web) para obtener el listado de los accesos producidos.



Figura 5. Consulta del registro de accesos

Se diferencia un poco de la solución que se propone en este proyecto, ya que en este caso no es necesario que los nodos autónomos comuniquen al nodo central que un profesional ha accedido a un documento clínico. Al ser este nodo central el nodo de intercambio entre los nodos autónomos, cuando se acceda a un informe clínico se registrará directamente este hecho sin necesidad de que le sea notificado. Por el contrario, en nuestro caso es necesario que exista una notificación previa para dar de alta los trámites o cambios del estado en el que se encuentra el trámite.

Por lo demás el comportamiento es igual a nuestro Sistema, el ciudadano se conecta a una aplicación que a través de un servicio web consulta a la base de datos central el estado de todos sus trámites.

2.3 Tecnologías utilizada

A continuación se listará y detallará brevemente las tecnologías utilizadas para el desarrollo del proyecto.

2.3.1 Sistema operativo

Un sistema operativo (SO o, frecuentemente, OS —del inglés Operating System—) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes y anteriores próximos y viceversa.

Resumiendo, es un programa que actúa de intermediario entre el hardware y el resto de aplicaciones que utilizan los usuarios.



Figura 6. Capas de abstracción de un ordenador

Los sistemas operativos proporcionan una plataforma de software encima de la cual otros programas, llamados aplicaciones, puedan funcionar. Las aplicaciones se programan para que funcionen encima de un sistema operativo particular, por tanto, la elección del sistema operativo determina en gran medida las aplicaciones que se pueden instalar.

Los sistemas operativos, según sus características, pueden clasificarse en:

- **Multiusuario:** varios usuarios pueden trabajar simultáneamente. Algunos sistemas operativos permiten a centenares o millares de usuarios al mismo tiempo.
- **Multiprocesador:** soporta el abrir un mismo programa en más de una CPU.
- **Multitarea:** varios programas pueden ejecutarse al mismo tiempo.
- **Tiempo Real:** responde a las entradas inmediatamente.

Los sistemas operativos también suelen clasificarse en familias. Las tres “grandes familias” de sistemas operativos más importantes son:

- **Windows:** es el nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft. Microsoft introdujo un entorno operativo denominado Windows el 20 de noviembre de 1985 como un complemento para MS-DOS en respuesta al creciente interés en las interfaces gráficas de usuario (GUI).



Microsoft Windows llegó a dominar el mercado mundial de computadoras personales, con más del 90% de la cuota de mercado, superando a Mac OS, que había sido introducido en 1984.

Las versiones más recientes de Windows son Windows 8.1 y Windows 8 para equipos de escritorio, Windows Server 2012 para servidores y Windows Phone 8 para dispositivos móviles. La primera versión en español fue Windows 3.0.

Windows es un sistema operativo basado en ventanas. La primera versión se lanzó en 1985 y comenzó a utilizarse de forma generalizada gracias a su interfaz gráfica de usuario (GUI, Graphical User Interface). Hasta ese momento, el sistema operativo más extendido era MS-DOS (Microsoft Disk Operating System), y la interfaz consistía en una línea de comandos. [8]

- **Macintosh (Mac):** es la línea de ordenadores personales diseñada, desarrollada y comercializada por Apple Inc.. En sus inicios fue una alternativa económica y doméstica al Lisa, un *avanzado microcomputador empresarial*,

cuya línea de desarrollo fue absorbida por la línea Macintosh. El Mac terminó por convertirse en la línea estándar de desarrollo de los computadores de Apple, al desaparecer la línea evolutiva del Apple II.

La gama de productos Mac en la actualidad varía desde el básico Mac Mini de escritorio hasta los servidores de rango medio como Mac Pro. Los sistemas Mac tienen como objetivo principal de mercado el hogar, la educación y la creatividad profesional. La producción de Mac está basada en un modelo de integración vertical en los que Apple proporciona todos los aspectos de su hardware y crea su propio sistema operativo que viene preinstalado en todos los Mac. Esto contrasta con los PC preinstalados con Microsoft Windows, donde un vendedor proporciona el sistema operativo y múltiples vendedores el hardware. En ambos casos, el hardware permite el funcionamiento de otros sistemas operativos: A partir de 1998, los Mac son capaces de soportar sistemas operativos como Linux, FreeBSD y Windows. [9]



- **Unix (registrado oficialmente como UNIX®):** es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969, por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

Solo los sistemas totalmente compatibles y que se encuentran certificados por la especificación Single UNIX Specification pueden ser denominados



"UNIX®" (otros reciben la denominación "similar a un sistema Unix" o "similar a Unix"). En ocasiones, suele usarse el término "Unix tradicional" para referirse a Unix o a un sistema operativo que cuenta con las características de UNIX Versión 7 o UNIX System V.

Hacia 1991, un estudiante de ciencias de la computación de la Universidad de Helsinki, llamado Linus Torvalds desarrolló un núcleo para computadoras con arquitectura x86 de Intel que emulaba muchas de las funcionalidades de UNIX y lo lanzó en forma de código abierto en 1991, bajo el nombre de Linux. En 1992, el Proyecto GNU comenzó a utilizar el núcleo Linux junto a sus programas. [10]

Algunos ejemplos de estas familias de Sistemas Operativos son:

Familia Windows	Familia Macintosh	Familia UNIX
Windows Vista	Mac OS X	UnixWare
Windows 7	Mac OS 9	Solaris
Windows Mobile	Mac OS 8	System V
Windows CE	Mac OS 7	Minix
Windows Server 2003		Irix
Windows XP		HP-UX

Existen infinidad de sistemas operativos, pero nos hemos centrados en los principales sistemas operativos de ordenadores personales. Si tuviésemos en cuenta, por ejemplo, también sistemas operativos móviles habría que añadir iOS, Android...

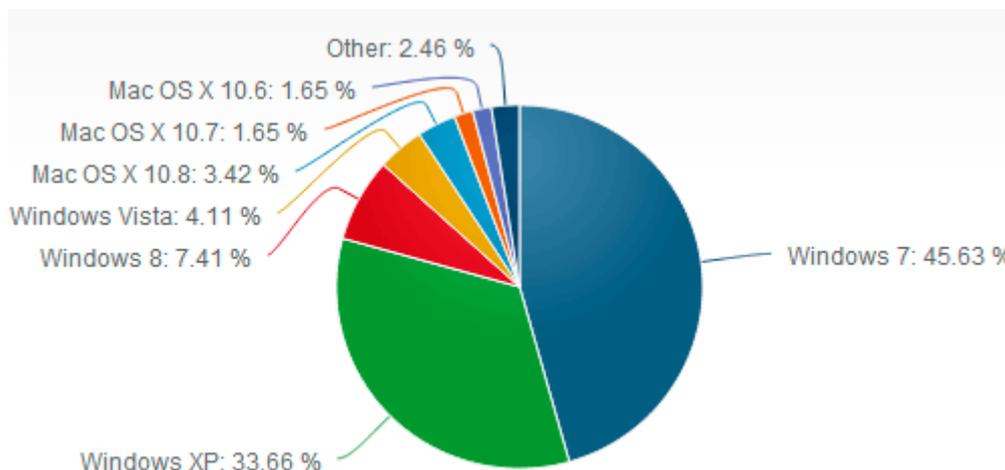


Figura 7. Cuota de uso SO (septiembre 2013) [11]

Habrá que diferenciar entre el entorno de desarrollo y el entorno donde estará instalada la aplicación final.

Para el desarrollo utilizaremos un sistema Windows 7 que es el que viene instalado en el portátil que se utilizará para el desarrollo y además permite la instalación de todas las herramientas necesarias.

En cuanto al sistema operativo del servidor donde estará alojado el sistema de información nos centraremos en los principales SO de servidor:

	Windows Server	Solaris	OS X Server
Compañía	Microsoft	Oracle (antiguamente Sun)	Apple
Año lanzamiento	1993	1992	2001
Ultima versión	2012 R2	Solaris 11	10.8.2
Licencia libre	No	Si (se cobra por soporte)	No

El servidor final donde residirá la aplicación será un sistema Solaris, en su versión 10, ya que es el corporativo de la organización.

2.3.2 Herramientas genéricas de desarrollo

2.3.2.1 Entorno de desarrollo integrado (IDE)

Un Entorno de Desarrollo Integrado, del inglés Integrated Development Environment, es un software que se compone de un conjunto de herramientas de programación que facilitan el trabajo de los desarrolladores. [12]

Para nuestro desarrollo hemos escogido como IDE Eclipse. Es un entorno de desarrollo de código abierto. Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. [13]



Puede decirse que es el IDE estándar de facto ya que es el más extendido. Dispone de infinidad de plugins que permiten extender la funcionalidad que ofrece facilitando las tareas a los desarrolladores.

Utilizaremos en nuestro desarrollo la versión Eclipse Indigo Service Release 2, de 2011, por ser la versión estándar de desarrollo dentro de la organización.

2.3.2.2 Lenguaje de programación

El lenguaje de programación con el que se desarrollará el proyecto es Java. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados. [14]



2.3.2.3 Maven

Maven es una herramienta de software para la gestión y construcción de proyectos Java. Es similar en funcionalidad a Apache Ant pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. [15]



Utilizaremos el plugin m2eclipse de Eclipse para integrar las funcionalidades de Maven. La versión de Maven 3.0.3 por ser la versión estándar dentro de la organización.

2.3.2.4 Subversion

Subversion (SVN) es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local. Dada su flexibilidad, es necesaria la aplicación de buenas prácticas para llevar a cabo una correcta gestión de las versiones del software generado. [16]



Este tipo de herramientas permite llevar el control de todos los cambios que se han producido en todos los archivos. Es especialmente útil en entornos de desarrollo en el que trabajan varios desarrolladores ya que gestiona también los conflictos que se producen cuando varias personas trabajan sobre un mismo archivo. Además permite también crear etiquetas de versiones, las cuáles permiten volver a regenerar el código tal y como se encontraba en ese momento en que se etiquetó.

2.3.2.5 Oxygen XML Editor

Para diseñar los XML de los mensajes que se intercambiarán en un Web Service se necesita un herramienta específica para ello, ya que hacerlo con un editor de texto sería muy laborioso y complejo.

Para ello existen muchas herramientas, tanto de pago como gratuitas, que permiten definir los XSD (XML Schema Definition) para poder definir la estructura y las restricciones del contenido de los XML que se van a intercambiar.

Hemos optado por la herramienta Oxygen XML Editor, que bastante potente y cubre todas nuestras necesidades. Nos permite crear los XSD y también validar que los mensajes XML generados cumplen con el XSD.



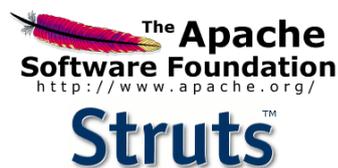
2.3.3 Herramientas específicas

2.3.3.1 Framework desarrollo

Un framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Entre los objetivos principales de un framework tenemos: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework web es un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. [17]

En nuestro caso utilizaremos como framework de desarrollo Struts. Es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition). Su carácter de "software libre" y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible lo convierten en una herramienta altamente disponible.



A pesar de no ser la versión más actual, utilizaremos la versión 1.3 de Struts debido a que es todavía la versión estándar de desarrollo de la organización.

2.3.3.2 Servicios web

Un servicio web (en inglés, Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. [18]

En nuestro caso utilizaremos como protocolo de intercambio SOAP (Simple Object Access Protocol). Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Hemos escogido SOAP frente a otros protocolos como REST debido a que la estructura de los mensajes a intercambiar es compleja y utilizar el protocolo REST puede terminar siendo más difícil y complejo de mantener.

Para el desarrollo de los servicios web se ha escogido Apache Axis2. Axis2 es un motor para servicios web que implementa la especificación JAX-WS del Java Community



Process, WS-Messaging y WS-Security. Dispone de una estructura modular que permitirá ampliar la funcionalidad básica del sistema en el futuro gracias al desarrollo de módulos adicionales (al estilo de los plugins de Eclipse). [19]

2.3.3.3 Servidor de aplicaciones

El concepto de servidor de aplicaciones está relacionado con el concepto de sistema distribuido. Un sistema distribuido, en oposición a un sistema monolítico, permite mejorar tres aspectos fundamentales en una aplicación: la alta disponibilidad, la escalabilidad y el mantenimiento. En un sistema monolítico un cambio en las necesidades del sistema (aumento considerable del número de visitas, aumento del número de aplicaciones, etc.) provoca un colapso y la adaptación a dicho cambio puede resultar catastrófica. Principales características:

- **Alta disponibilidad.** En las aplicaciones empresariales, por norma, la disponibilidad debe ser 24x7 (24 horas al día, 7 días a la semana). Por tanto son necesarias técnicas de balanceo de carga y recuperación ante fallos para poder alcanzar dicha disponibilidad.
- **Escalabilidad.** Capacidad para aumentar la capacidad de servicio ante el aumento de la demanda. Se logra aumentando el número de máquinas que dan servicio.
- **Mantenimiento.** versatilidad a la hora de actualizar, depurar fallos y mantener un sistema. La solución al mantenimiento es la construcción de la lógica de negocio en unidades reusables y modulares.

El estándar J2EE permite el desarrollo de aplicaciones de empresa de una manera sencilla y eficiente. Una aplicación desarrollada con las tecnologías J2EE permite ser desplegada en cualquier servidor de aplicaciones o servidor web que cumpla con el estándar. Un servidor de aplicaciones es una implementación de la especificación J2EE. [20]

Existe multitud de servidores de aplicaciones, tanto de código abierto como propietarios. Entre los más conocidos puede destacarse JBoss (código abierto) o Weblogic (propietario).

A parte también puede incluirse Tomcat 7 ya que a pesar de ser un servidor web, a partir de la versión 7 incluye funcionalidades propias de los servidores de aplicaciones.

A continuación mostramos una tabla resumen comparando las principales características de estos tres servidores de aplicaciones que hemos mencionado.

	Tomcat 7	JBoss	Weblogic
Compañía	Apache Software Foundation	Red Hat	Oracle
Consumo de recursos	Bajo	Alto	Alto
Complejidad administración	Baja	Alta	Alta
Funcionalidad ofrecida	Media	Alta	Alta
Soporta Java	Si	Si	Si
Multiplataforma	Si	Si	Si

En nuestro caso la capacidad de elección no existe debido a que dentro de la organización existen una serie de aplicaciones corporativas dentro de las cuáles está incluida



Oracle Weblogic. Por tanto debemos atenernos a la utilización de este Servidor de Aplicaciones en proyecto.

Utilizaremos la versión Oracle Weblogic 10.3.4, que es la versión oficial actualmente en la organización. Esta decisión repercutirá en el presupuesto del proyecto ya que nos obliga a adquirir una licencia para la instalación de un servidor local donde realizar el desarrollo. Aunque se podría optar por realizar el desarrollo en local utilizando un servidor distinto, por ejemplo JBoss. Pero ello supondría riesgos ya que al no ser el mismo entorno que el entorno final podrían producirse errores que en el entorno local no se producían (por diferencias en versiones de librerías, bugs,...).

Por mucho que JAVA sea una tecnología multiplataforma, siempre existen pequeñas diferencias en archivos específicos de servidores, versiones distintas de librerías...

que provocarían errores que impactarían más en el presupuesto del proyecto que el pagar la licencia de uso.

2.3.3.4 Sistema gestor de base de datos

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.

Los SGBD también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permite presentar la información de la base de datos en variados formatos. [21]

Al igual que en el caso de los Servidores de Aplicaciones existen múltiples sistemas gestores, tanto de código abierto (el más extendido MySQL), como propietarios (SQL Server, Oracle). A continuación mostramos una tabla resumen comparando las principales características de estos tres SGBD que hemos mencionado:

	MySQL	SQLServer	Oracle
Compañía	Oracle (antiguamente Sun Microsystem)	Microsoft	Oracle
Rendimiento	Alto	Alto	Alto
Capacidad	Alto	Alto	Alto
Concurrencia	Si	Si	Si
Licencia Libre	Si	No	No
Multiplataforma	Si	No	SI
Seguridad	Alto	Alto	Alto

Pero también en este aspecto estamos condicionados por las aplicaciones corporativas que impone la organización. Por ello como SGBD utilizaremos Oracle Database en su versión 10.3g, que es la oficial en la actualidad dentro de la organización.



Como en el caso anterior del servidor de aplicaciones, esta decisión repercutirá en el presupuesto del proyecto ya que habrá que adquirir una licencia para la instalación de una BD local para realizar el desarrollo. En este caso también podríamos optar por usar una BD que no necesite licencia, por ejemplo Oracle Database XE (que está pensada para fines educativos, proyectos open source...). Pero preferimos optar por adquirir una licencia de la misma BD que luego será utilizada en el servidor final para evitar cualquier tipo de problema inesperado.

Capítulo 3

Análisis

En este capítulo se tratarán las fases de toma de requisitos y análisis de los distintos subsistemas de los que se compone en Sistema de Información.

3.1 Alcance del sistema

Aunque cuando hablamos de este Sistema lo hacemos como si fuese algo monolítico, realmente constará de, por así decirlo, tres subsistemas de alguna forma independientes.

Los tres subsistemas tendrán en común que utilizarán la misma Base de Datos. Estos sistemas son:

- Portal web de acceso para el ciudadano.
- Servicio web dentro de la intranet de la Administración para que el resto de Sistemas de la organización se conecten para dar de alta la información relevante de la que disponen.
- Aplicación de gestión y administración, para la gestión propia de la aplicación (dar e alta sistemas que pueden utilizarla, dar de alta usuarios,...) y permitir interactuar con el servicio web en casos de procedimientos administrativos que no tengan aplicación de gestión (o aún no se hayan integrado con el Servicio Web).

Las entidades externas que serán los actores del Sistema, o mejor dicho, de cada uno de los subsistemas, serán:

- Los Sistemas de Información (o aplicaciones) que realizan la gestión de los trámites administrativos. Estos serán los que se conecten al Servicio Web para informar del estado del trámite administrativo.
- Los ciudadanos, que se conectarán al portal web para consultar la información referente a sus trámites administrativos.
- Los usuarios de administración y gestión de la aplicación interna.

3.2 Establecimiento de requisitos

Este apartado recoge los requisitos del sistema. Estarán divididos en dos grupos: requisitos de usuario y requisitos de software.

A su vez, dentro de cada grupo, se dividirán los requisitos por subsistemas.

3.2.1 Requisitos de usuario

Cada requisito de usuario se recogerá en una tabla que incluirá la siguiente información:

- **ID:** Identificador único del requisito.
- **Nombre:** Nombre descriptivo del requisito.
- **Descripción:** Descripción detallada del requisito.
- **Importancia:** Grado de necesidad del requisito. Puede tomar los siguientes valores: *PD* (pendiente de definir), *vital*, *importante*, *quedaría bien*.
- **Urgencia:** Prioridad a la hora de implementar el requisito. Puede tomar los siguientes valores: *PD* (pendiente de definir), *inmediatamente*, *hay presión*, *puede esperar*.
- **Estabilidad:** Grado de variabilidad del requisito a lo largo del ciclo de vida del proyecto. Puede tomar los siguientes valores: *PD* (pendiente de definir), *baja*, *media*, *alta*.
- **Comentario:** Cualquier indicación que no pueda registrarse en los campos anteriores.

3.2.1.1 Requisitos de capacidad

Los requisitos de capacidad, indican qué funciones puede o tiene que hacer la aplicación para cumplir los propósitos requeridos para el sistema.

A continuación se listan todos los requisitos clasificándolos por subsistema.

Requisitos del servicio web

FRQ-0001	Información mínima a intercambiar		
Descripción	El conjunto de información mínima que se va a intercambiar, tanto con el ciudadano como con las aplicaciones de gestión, es el ESTADO DE UN ASUNTO. El asunto se corresponde con un trámite particular		
Importancia	Importante	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0002	Un estado de asunto por persona		
Descripción	Cada estado de asunto hará referencia a una sola persona. Si en un asunto intervienen varias personas, la aplicación cliente lo enviará como asuntos independientes.		
Importancia	Importante	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0003	Datos del asunto		
Descripción	<p>El estado de un asunto se compone de:</p> <p>1) Datos identificativos de la persona relacionada:</p> <ul style="list-style-type: none"> - DNI relacionado - CIF relacionado - Nombre y apellidos (todo o parte se utilizará junto con el DNI para permitir el acceso) - Razón social (todo o parte se utilizará junto con el CIF para permitir el acceso) <p>2) Datos identificativos del asunto:</p> <ul style="list-style-type: none"> - Procedimiento administrativo - Área (tabla): Una agrupación de asuntos. Su finalidad es que el ciudadano pueda filtrar fácilmente sus asuntos por un concepto intuitivo. Un área puede tener varios procedimientos. - Identificador único del asunto incluyendo aplicación. Junto con el DNI o CIF, servirá de clave única en el sistema. Si se detectan aplicaciones que no dispongan de identificador único, puede generarse uno en base al momento de llegada. La aplicación cliente no podrá eliminar éste asunto en concreto, pero podrá eliminar todos los asuntos de ese DNI. <p>3) Datos del estado del asunto</p> <ul style="list-style-type: none"> - Estado del asunto - Descripción del estado del asunto - Otra información /ficheros anexos. Información variable con cada procedimiento, que puede ser de utilidad para el usuario. - Fecha de la información o del cambio de estado - Fecha de caducidad. Fecha a partir de la cual la información dejará de estar disponible al ciudadano 		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0004	Servicio web		
Descripción	Existirá un servicio web que permitirá a las aplicaciones dar de alta, eliminar y consultar, asuntos y estados de asuntos del ciudadano. El acceso a este servicio podrá ser con o sin certificado digital.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0005	Nuevo asunto		
Descripción	Para dar de alta un asunto nuevo o realizar un cambio de estado, se invocará al servicio de alta. Cada asunto podrá tener sucesivos cambios de estado, que se corresponderán con sucesivas llamadas a este servicio. Se devolverá un código de estado de asunto. que se podrá utilizar para eliminar ese estado en concreto		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0006	Eliminación de asuntos		
Descripción	Para dar de baja un asunto se invocará al servicio de eliminación. Se podrá borrar un asunto y también todos los asuntos de ese usuario para esa aplicación. El borrado en base de datos será lógico, no físico.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0007	Eliminación de estados de un asunto		
Descripción	Se podrá eliminar, dentro de un asunto que ha pasado por sucesivos estados, un estado en concreto.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0008	Consulta de información de un asunto		
Descripción	Permitirá verificar la información que está disponible para el ciudadano. Puede ser utilizado por la aplicación web para recuperar la información del ciudadano. La consulta se podrá realizar en base a cualquier criterio de búsqueda (según los datos identificativos del ciudadano/empresa o del asunto)		
Importancia	Vital	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

Requisitos aplicación de consulta del ciudadano

FRQ-0009	Acceso con certificado		
Descripción	<p>El ciudadano necesitará un certificado digital /DNI electrónico para conectarse.</p> <p>Se permitirá la conexión con certificado de empresa, en cuyo caso la aplicación ofrecerá dos opciones: Asuntos relacionados con el DNI personal, y asuntos relacionados con el CIF de la empresa a la que representa.</p>		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0010	Listado de asuntos		
Descripción	<p>Se mostrará una lista de asuntos donde conste el estado actual. En caso de no existir asuntos, se mostrará mensaje informando de dicha circunstancia. Cada asunto contendrá dos enlaces:</p> <p>1) Ver estado actual: Llevará a la información detallada del asunto y su estado, donde se muestra entre otras cosas la "Otra información" (adjuntos, etc.)</p> <p>2) Ver histórico: Llevará a una lista de los sucesivos cambios de estados de este asunto. Cada estado de la lista tendrá el correspondiente enlace a su información detallada.</p>		
Importancia	Importante	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0011	Menú		
Descripción	<p>A la izquierda dispondrá en todo momento de una lista de áreas, donde, aunque no tenga asuntos pendientes, podrá ver qué tipo de información puede gestionar esta aplicación.</p> <p>El nombre de las áreas estará activo como enlace en caso de que el usuario tenga asuntos de ese área. A través de dicho enlace se filtrará la lista de asuntos que se muestran, incluyendo sólo los correspondientes a ese área.</p>		
Importancia	Vital	Estabilidad	Media
Urgencia	Puede esperar	Comentarios	Ninguno

Requisitos de la aplicación de administración y gestión

FRQ-0012	Acceso con certificado		
Descripción	El acceso se realizará utilizando certificado digital. El usuario deberá estar registrado para poder acceder.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

FRQ-0013	Gestión de usuarios		
Descripción	<p>El sistema permitirá dar de alta dos tipos de usuarios: de administración y gestión.</p> <p>Los administradores podrán dar de alta aplicaciones, usuarios y modificar tablas maestras.</p> <p>Los gestores podrán dar de alta, baja y consultar asuntos. Esta opción se da para los casos de trámites que no tengan aun una aplicación informática que pueda comunicarse con el servicio web.</p>		
Importancia	Importante	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

FRQ-0014	Gestión de aplicaciones		
Descripción	<p>El sistema permitirá dar de alta aplicaciones para que puedan conectarse al servicio web.</p> <p>Las aplicaciones podrán configurarse para que requieran o no el uso de firma digital.</p> <p>Las aplicaciones podrán ser de: consulta o consulta/actualización.</p> <p>Podrán ser aplicaciones no privilegiadas (sólo consultan los asuntos generados por ellas mismas) o privilegiadas (pueden consultar cualquier asunto)</p>		
Importancia	Importante	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

FRQ-0015	Gestión de tablas maestras		
Descripción	Existirá una opción para realizar el mantenimiento de tablas maestras del sistema: estados y roles.		
Importancia	Quedaría bien	Estabilidad	Media
Urgencia	Puede esperar	Comentarios	Ninguno

FRQ-0016	Gestión de asuntos		
Descripción	Los usuarios con permisos para registrar/consultar asuntos (gestores) podrán dar de alta, consultar y borrar asuntos y estados desde la aplicación.		
Importancia	Importante	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

3.2.1.2 Requisitos de restricción

Los requisitos de restricción, indican qué limitaciones tiene la aplicación.

Requisitos del servicio web

FRQ-0017	Imposibilidad modificar asuntos		
Descripción	Un asunto o estado no podrá ser modificado. En caso de ser errónea la información deberá eliminarse y volverse a dar de alta.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

FRQ-0018	Imposibilidad consultar asuntos de otra aplicación		
Descripción	Una aplicación sólo podrá consultar los asuntos generados por ellas mismas, salvo en el caso de las aplicaciones privilegiadas que podrán consultar cualquier asunto.		
Importancia	Vital	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

Requisitos de la aplicación de administración y gestión

FRQ-0019	Imposibilidad de eliminar roles		
Descripción	En el mantenimiento de las tablas maestras, en el caso de roles sólo se podrán crear nuevos roles. En ningún caso se podrán eliminar.		
Importancia	Vital	Estabilidad	Media
Urgencia	Puede esperar	Comentarios	Ninguno

3.2.2 Requisitos de software

Cada requisito de usuario se recogerá en una tabla que incluirá la siguiente información:

- **ID:** Identificador único del requisito.
- **Nombre:** Nombre descriptivo del requisito.
- **Dependencias:** Indica que requisito de usuario se implementa con dicho requisito. Al menos deberá existir un requisito de software por cada requisito de usuario (en caso contrario ese requisito de usuario quedaría sin cubrir).
- **Descripción:** Descripción detallada del requisito.
- **Importancia:** Grado de necesidad del requisito. Puede tomar los siguientes valores: *PD* (pendiente de definir), *vital*, *importante*, *quedaría bien*.
- **Urgencia:** Prioridad a la hora de implementar el requisito. Puede tomar los siguientes valores: *PD* (pendiente de definir), *inmediatamente*, *hay presión*, *puede esperar*.
- **Estabilidad:** Grado de variabilidad del requisito a lo largo del ciclo de vida del proyecto. Puede tomar los siguientes valores: *PD* (pendiente de definir), *baja*, *media*, *alta*.
- **Comentario:** Cualquier indicación que no pueda registrarse en los campos anteriores.

Requisitos del servicio web

IRQ-0001	Servicio web		
Dependencias	[FRQ-0004] Servicio web [FRQ-0005] Nuevo asunto [FRQ-0006] Eliminación de asuntos [FRQ-0007] Eliminación de estados de un asunto [FRQ-0008] Consulta de información de un asunto [FRQ-0017] Imposibilidad de modificar asuntos [FRQ-0018] Imposibilidad de consultar asuntos de otra aplicación		
Descripción	Se creará un servicio web que contemple las operaciones de alta, borrado y consulta. Para interactuar con el servicio web, se enviarán XML que incluyan además de la información del asunto/estado, el identificador de la aplicación origen del mensaje. Opcionalmente, el mensaje podrá venir firmado digitalmente para asegurar la identidad de la aplicación origen.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

IRQ-0002	Nuevo asunto		
Dependencias	[FRQ-0001] Información mínima a intercambiar [FRQ-0002] Un estado de asunto por persona [FRQ-0003] Datos del asunto [FRQ-0004] Servicio web [FRQ-0005] Nuevo asunto [FRQ-0014] Gestión de aplicaciones		
Descripción	Para dar de alta un nuevo asunto se invocará a la operación de alta de un asunto. Se definirá un XML de alta que permita únicamente dar de alta un asunto para una persona o empresa. Se comprobará que la aplicación origen tiene permisos para dar de alta asuntos. En caso de venir firmado, se comprobará que el certificado es el esperado para dicha aplicación a través del DN almacenado para dicha aplicación.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

IRQ-0003	Eliminación de asuntos y estados		
Dependencias	[FRQ-0003] Datos del asunto [FRQ-0004] Servicio web [FRQ-0006] Eliminación de asuntos [FRQ-0007] Eliminación de estados de un asunto [FRQ-0014] Gestión de aplicaciones [FRQ-0017] Imposibilidad de modificar asuntos		
Descripción	Para borrar un asunto o un estado de un asunto se invocará a la operación de baja. Se definirá un XML que permita: - Eliminar únicamente un estado indicando el identificador de dicho estado-asunto. - Eliminar el asunto (incluyendo sus estados asociados) indicando el identificador del asunto. - Eliminar todos los asuntos (incluyendo sus estados asociados) indicando el DNI del ciudadano o CIF de la empresa. Se comprobará que la aplicación origen tiene permisos para dar de alta asuntos. En caso de venir firmado, se comprobará que el certificado es el esperado para dicha aplicación.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

IRQ-0004	Consulta de información de un asunto		
Dependencias	[FRQ-0003] Datos del asunto [FRQ-0004] Servicio web [FRQ-0008] Consulta de información de un asunto [FRQ-0014] Gestión de aplicaciones [FRQ-0018] Imposibilidad de consultar asuntos de otra aplicación		
Descripción	Se definirá un mensaje XML de consulta que permita consultar:		

	<ul style="list-style-type: none"> - La información de un asunto (incluyendo sus estados asociados) - La información del estado de un asunto. - La información de todos los asuntos (incluyendo sus estados asociados) de un ciudadano o empresa. <p>Se filtrarán para mostrar sólo los asuntos generados por dicha aplicación, salvo que sea una aplicación privilegiada que mostrará todos.</p> <p>En caso de venir firmado, se comprobará que el certificado es el esperado para dicha aplicación.</p>		
Importancia	Vital	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

Requisitos de aplicación de consulta

IRQ-0005	Acceso con certificado		
Dependencias	[FRQ-0009] Acceso con certificado		
Descripción	Se requerirá de un certificado al ciudadano que acceda a la aplicación. Dicho certificado se validará para comprobar que no está caducado o revocado.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Inmediatamente	Comentarios	Ninguno

IRQ-0006	Listado de asuntos		
Dependencias	[FRQ-0009] Acceso con certificado [FRQ-0010] Listado de asuntos		
Descripción	Se listarán los asuntos asociados al ciudadano. La consulta se realizará por DNI, nombre y primer apellido. En caso de ser un certificado de empresa, se permitirá consultar o bien los asuntos asociados al CIF o al DNI del apoderado de la empresa.		
Importancia	Vital	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

IRQ-0007	Menú de filtrado		
Dependencias	[FRQ-0010] Listado de asuntos [FRQ-0011] Menú		
Descripción	En el margen izquierdo de la aplicación se presentará un menú que siempre contendrá un listado de las áreas en las que pueden existir asuntos. Dichas áreas permitirán filtrar los asuntos que se muestran al ciudadano o empresa, para hacer más fácil la búsqueda de un asunto en concreto.		
Importancia	Importante	Estabilidad	Media
Urgencia	Puede esperar	Comentarios	Ninguno

IRQ-0008	Visualización información de un asunto		
Dependencias	[FRQ-0010] Listado de asuntos		
Descripción	Si el usuario selecciona uno de los asuntos, se mostrará una pantalla con la información del estado actual del asunto.		
Importancia	Vital	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

IRQ-0009	Visualización histórico de un asunto		
Dependencias	[FRQ-0010] Listado de asuntos		
Descripción	Si el ciudadano selecciona ver el histórico de un asunto, se mostrará una pantalla con la información de todos los estados por los que ha pasado el asunto.		
Importancia	Vital	Estabilidad	Media
Urgencia	Inmediatamente	Comentarios	Ninguno

Requisitos de la aplicación de administración y gestión

IRQ-0010	Acceso con certificado		
Dependencias	[FRQ-0012] Acceso con certificado		
Descripción	Se requerirá de un certificado al administrador que acceda a la aplicación. Dicho certificado se validará para comprobar que no está caducado o revocado. Una vez comprobado que el certificado es válido se comprobará en BD que el usuario está registrado. La búsqueda se realizará por Nombre, Apellido y DNI.		
Importancia	Vital	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

IRQ-0011	Gestión de usuarios		
Dependencias	[FRQ-0012] Acceso con certificado [FRQ-0013] Gestión de usuarios		
Descripción	Se registrarán los usuarios indicando de forma obligatoria DNI, nombre y apellidos, rol (administrador o gestor) y, en caso de ser gestor, las aplicaciones para las que está capacitado para registrar asuntos. Se validará que el DNI sea correcto. El nombre deberá ser el mismo con el que tiene asociado el certificado digital, de lo contrario no podrá validarse.		
Importancia	Importante	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

IRQ-0012	Gestión de aplicaciones		
Dependencias	[FRQ-0004] Servicio web [FRQ-0014] Gestión de aplicaciones		
Descripción	Las aplicaciones podrán ser de: consulta o consulta/actualización. Podrán ser aplicaciones no privilegiadas (sólo consultan los asuntos generados por ellas mismas) o privilegiadas (pueden consultar cualquier asunto) Las aplicaciones podrán configurarse para que requieran o no el uso de firma digital. En caso de ser requerido el uso de certificado digital, se almacenará el DN del certificado con el que firma en el primer envío que se realice al servicio web.		
Importancia	Importante	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

IRQ-0013	Gestión de tablas maestras		
Dependencias	[FRQ-0019] Imposibilidad de eliminar roles [FRQ-0015] Gestión de tablas maestras		
Descripción	Se podrán mantener a través de la aplicación las siguientes tablas maestras: estados y roles. - Estados: sólo se podrá añadir nuevos estados. No se podrá eliminar debido al histórico asociado. - Roles: se podrán crear nuevos roles a los que le asociarán las funcionalidades que puede realizar (alta asunto, consulta asunto, eliminación asunto, gestión usuarios, gestión aplicaciones, gestión tablas maestras y generación estadísticas).		
Importancia	Quedaría bien	Estabilidad	Media
Urgencia	Puede esperar	Comentarios	Ninguno

IRQ-0014	Gestión de asuntos		
Dependencias	[FRQ-0004] Servicio web [FRQ-0005] Nuevo asunto [FRQ-0006] Eliminación de asuntos [FRQ-0007] Eliminación de estados de un asunto [FRQ-0008] Consulta de información de un asunto [FRQ-0016] Gestión de asuntos [FRQ-0014] Gestión de aplicaciones		
Descripción	A través de la aplicación, los usuarios gestores podrán dar de alta, consultar y eliminar asuntos. Se recogerá la información a través de un formulario y a partir de dicha información se generarán los XML que se enviarán al servicio web. Para ello, los gestores se asociarán a "aplicaciones virtuales" que habrá que registrar previamente.		
Importancia	Importante	Estabilidad	Alta
Urgencia	Puede esperar	Comentarios	Ninguno

3.2.3 Matriz de trazabilidad

A continuación comprobamos que todos los requisitos de usuario están cubiertos por al menos un requisito de software. Para ello utilizamos una matriz de trazabilidad.

Esta matriz cruza los requisitos de software en el eje vertical, con los requisitos de usuario en el eje horizontal.

TRM-0001	IRQ-0001	IRQ-0002	IRQ-0003	IRQ-0004	IRQ-0005	IRQ-0006	IRQ-0007	IRQ-0008	IRQ-0009	IRQ-0010	IRQ-0011	IRQ-0012	IRQ-0013	IRQ-0014
FRQ-0001	.	X
FRQ-0002	.	X
FRQ-0003	.	X	X	X
FRQ-0004	X	X	X	X	X	.	.
FRQ-0005	X	X	X
FRQ-0006	X	.	X	X
FRQ-0007	X	.	X	X
FRQ-0008	X	.	.	X	X
FRQ-0009	X	X
FRQ-0010	X	X	X	X
FRQ-0011	X
FRQ-0012	X	X	.	.	.
FRQ-0013	X	.	.	.
FRQ-0014	X	.	X	X	X	.	.
FRQ-0015	.	X
FRQ-0016	X
FRQ-0017	X
FRQ-0018	X
FRQ-0019	.	X

Gracias a la matriz, se puede comprobar de forma visual que todos los requisitos de usuario están cubiertos.

3.3 Especificación de casos de uso

A continuación se describe el modelo de roles del sistema y la generalización de los distintos actores disponibles en el sistema.

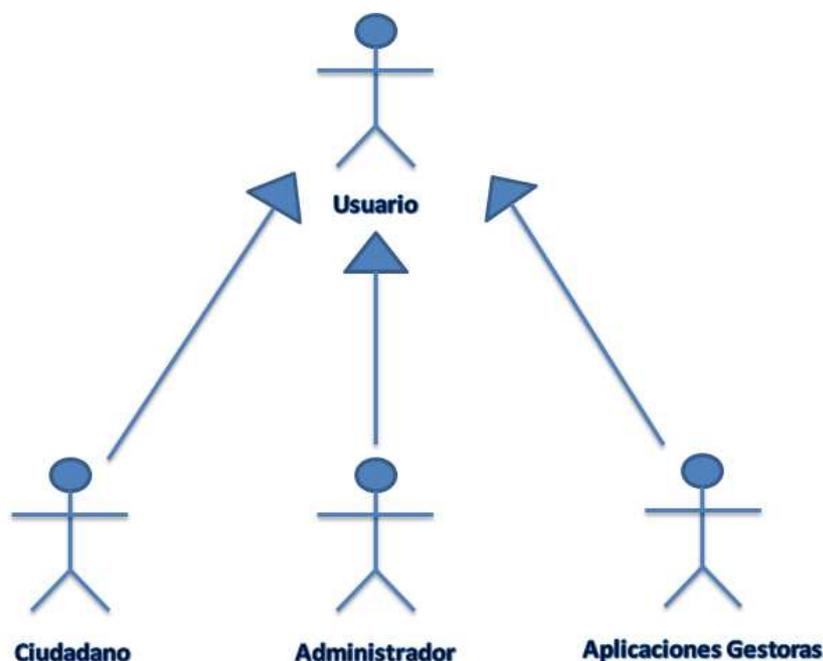


Figura 8. Actores del sistema

Actor	Descripción
Usuario	Es la generalización del Ciudadano, del Administrador y de Aplicaciones Gestoras.
Ciudadano	Usuario acreditado, con certificado digital, que accederá al sistema. Podrá tratarse de una persona física o de una entidad.
Administrador	Usuario registrado con permisos de administración.
Aplicaciones Gestoras	Cada una de las aplicaciones que realizan funciones de gestión y que notifican los estados de los asuntos almacenados en el sistema.

A continuación se detallan los casos de usos del sistema. Cada caso de uso se recogerá en una tabla que incluirá la siguiente información:

- **ID:** Identificador único del caso de uso.
- **Nombre:** Nombre descriptivo del requisito.
- **Descripción:** Descripción detallada del caso de uso.
- **Actores:** Actores externos implicados en el caso de uso.

- **Pre-condición:** Condición necesaria que debe cumplirse previamente para hacer el caso de uso.
- **Post-condición:** Hecho que se cumple si se ejecutan correctamente las acciones del caso de uso.
- **Acciones:** Descripción detallada de los pasos que se realizan en el caso de uso.

3.3.1 Subsistema de Control de Acceso

El acceso al sistema está controlado y limitado por el uso de certificación digital, no permitiéndose realizar ninguna operación a usuarios no acreditados y registrados en el sistema.

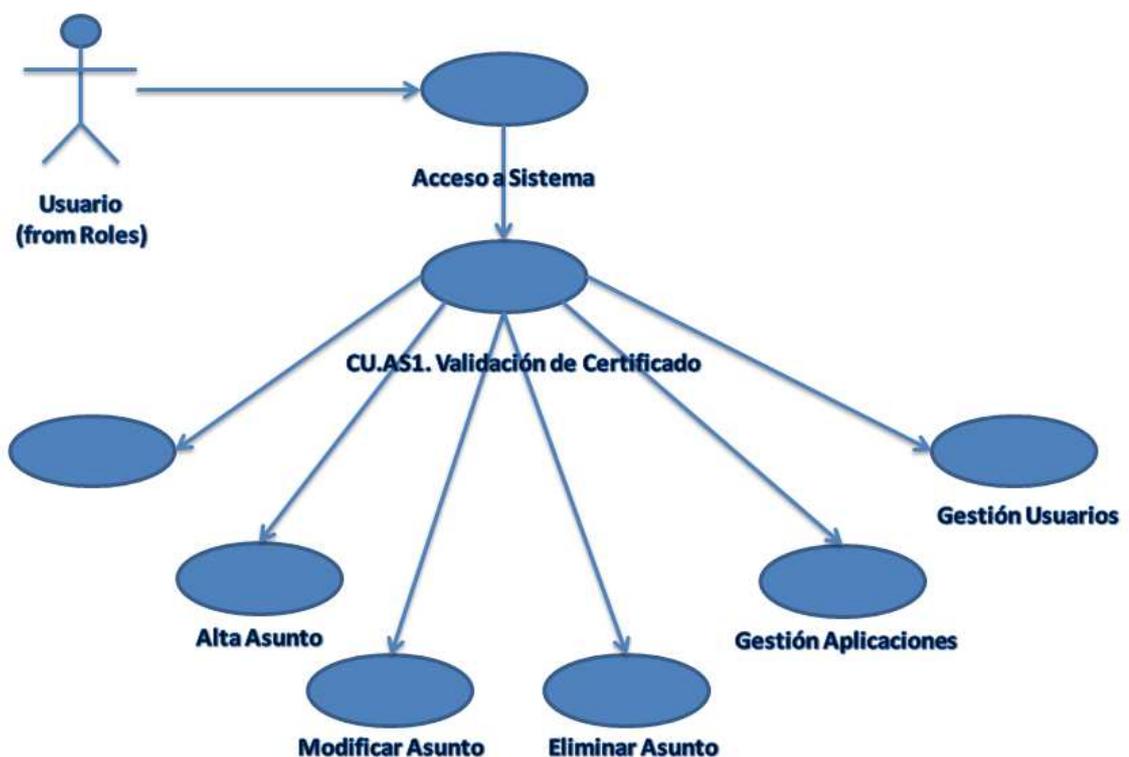


Figura 9. Casos de uso Control de Acceso

CU.AS1	Validación de Certificado	
Descripción	<p>Recogerá el certificado digital del usuario, procediendo a su validación con el Servidor de Certificados.</p> <p>A. - Conexión OK con el servidor de certificados:</p> <p>1.- Si el certificado es correcto: Se comprobará que el usuario está registrado en el sistema:</p> <ul style="list-style-type: none"> -- Si el usuario no está registrado se mostrará pantalla de aviso. -- Si el usuario está registrado se obtendrá toda la información almacenada sobre él mostrando pantalla de acceso al sistema. <p>2.- Si el certificado es incorrecto: Se tratarán cada uno de los posibles errores:</p> <ul style="list-style-type: none"> -- Certificado Revocado -- Certificado Caducado -- Certificado Erróneo <p>B. - Conexión ERROR con el servidor de certificados:</p> <p>El sistema validará temporalmente la existencia de registro del usuario, pudiendo permitir o no su acceso.</p> <p>Todos los accesos deben quedar registrados en un Log.</p>	
Actores	Ciudadano y Aplicaciones Gestoras	
Pre-condición	El usuario debe acceder con un Certificado Digital.	
Post-condición	Se guarda en sesión la información del usuario.	
ESCENARIOS		
#	Actor	# Sistema
1	El usuario accede al sistema	2 Recoge el certificado digital del usuario
		3 Conecta OK con el servidor de certificados
		4 Recoge resultado del servidor de certificados
		5 Si Certificado OK → Comprueba que el usuario está registrado en sistema, y en ese caso obtiene sus datos. En caso de que el usuario no este registrado mostrará la pantalla correspondiente.
		6 Si Certificado NO OK → Mostrar aviso.
		7 No Conecta OK con el servidor de certificados, el sistema validará el usuario en Base de Datos y si ya está registrado posibilitará o no el acceso. Envío de correo avisando del error al personal de sistemas.
		8 Registro en el Log de Acceso.

3.3.2 Funcionalidades del Portal del Ciudadano

El sistema permitirá realizar funcionalidades de entrada, modificación, eliminación y consulta de asuntos pertenecientes a usuarios. Además, existirá la posibilidad de administrar **los datos de usuarios administradores y aplicaciones gestoras que intervienen en el sistema.**

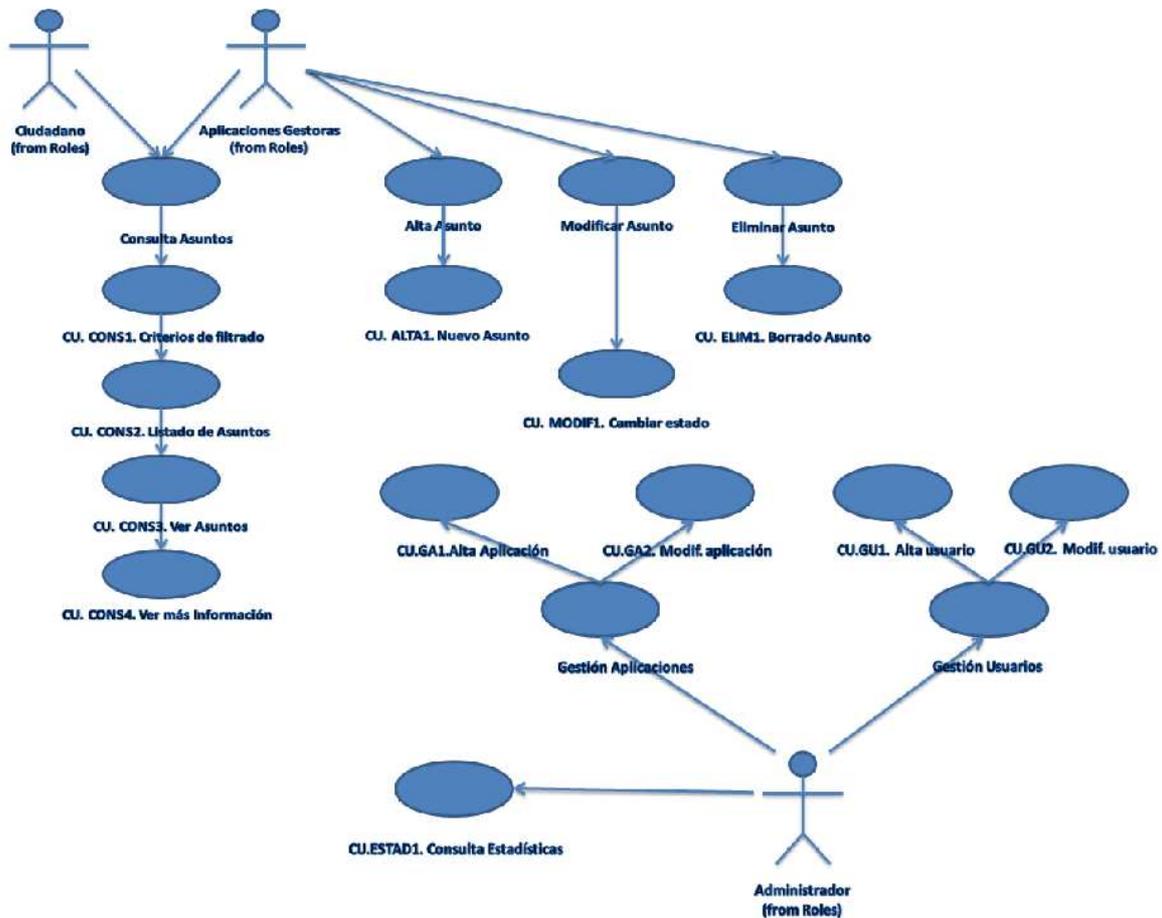


Figura 10. Casos de uso Sistema Información Ciudadano

Consulta de Asuntos

Esta funcionalidad será accesible tanto por ciudadanos en particular como por aplicaciones gestoras de información.

CU.CONSI	Criterios de filtrado		
Descripción	Un usuario desea obtener el listado de asuntos que cumplan una serie de condiciones.		
Actores	Ciudadano y Aplicaciones Gestoras		
Pre-condición			
Post-condición			
ESCENARIOS			
#	Actor	#	Sistema
1	El usuario accede al sistema		
2	Si el usuario que accede se trata de una aplicación gestora, indica que desea realizar una consulta.	3	Dará acceso a la funcionalidad que permite realizar consultas aplicando criterios de filtrado.

CU.CONSI2	Listado de Asuntos		
Descripción	Un usuario indica criterios de filtrado y obtiene el listado de asuntos coincidentes.		
Actores	Ciudadano y Aplicaciones Gestoras		
Pre-condición	El usuario está autenticado en el sistema.		
Post-condición			
ESCENARIOS			
#	Actor	#	Sistema
1	El usuario accede al sistema		
2	Si el usuario que accede se trata de una aplicación gestora, indica que desea realizar una consulta.	3	Dará acceso a la funcionalidad que permite realizar consultas aplicando criterios de filtrado.
4	Se seleccionan los criterios de filtrado.	5	Se aplican los criterios en la búsqueda de registros coincidentes
		6	Se confecciona el listado de coincidentes.

CU.CON3	Ver Asunto		
Descripción	Un usuario selecciona un determinado asunto al que tiene acceso para consultar más detalladamente la información del estado en que se encuentra éste.		
Actores	Ciudadano y Aplicaciones Gestoras		
Pre-condición	El usuario está autenticado en el sistema. Existen asuntos registrados para el usuario.		
Post-condición			
ESCENARIOS			
#	Actor	#	Sistema
1	El usuario accede al sistema		
2	Si el usuario que accede se trata de una aplicación gestora, indica que desea realizar una consulta.	3	Dará acceso a la funcionalidad que permite realizar consultas aplicando criterios de filtrado.
4	Se seleccionan los criterios de filtrado.	5	Se aplican los criterios en la búsqueda de registros coincidentes
		6	Se confecciona el listado de coincidentes.
7	Selecciona el asunto a consultar con mayor detalle.	8	Mostrará toda información del asunto, indicando su estado actual. Dará la posibilidad de acceder al apartado de "Ver más información"

CU.CON4	Ver más Información		
Descripción	Un usuario está consultando un determinado asunto y desea ver el apartado de más información.		
Actores	Ciudadano y Aplicaciones Gestoras		
Pre-condición	El usuario está autenticado en el sistema. Existen asuntos registrados para el usuario.		
Post-condición			
ESCENARIOS			
#	Actor	#	Sistema
1	El usuario accede al sistema		
2	Si el usuario que accede se trata de una aplicación gestora, indica que desea realizar una consulta.	3	Dará acceso a la funcionalidad que permite realizar consultas aplicando criterios de filtrado.
4	Se seleccionan los criterios de filtrado.	5	Se aplican los criterios en la búsqueda de registros coincidentes
		6	Se confecciona el listado de coincidentes.
7	Selecciona el asunto a consultar con mayor detalle.	8	Mostrará toda información del asunto, indicando su estado actual. Dará la posibilidad de acceder al apartado de "Ver más información"
9	Selecciona la opción de "Ver más información"	10	Se muestra la información almacenada en dicho campo.

Alta de Asunto

CU.ALTA1	Nuevo Asunto		
Descripción	Una aplicación nos envía el alta de un nuevo asunto perteneciente a una persona o una entidad.		
Actores	Aplicación Gestora		
Pre-condición	La aplicación está registrada en el sistema.		
Post-condición	Queda registrado en la BD el asunto.		
ESCENARIOS			
#	Actor	#	Sistema
1	La aplicación envía los datos de un nuevo asunto a insertar en el sistema	2	Se recogen los datos del nuevo asunto y se validan para evitar errores.
		3	Se insertan los datos en el sistema
		4	Se comunica del resultado de efectuar el proceso de alta

Modificación de Asunto

CU.MODIF1	Cambio estado		
Descripción	Una aplicación nos envía el cambio de estado de un asunto perteneciente a una persona o una entidad.		
Actores	Aplicación Gestora		
Pre-condición	La aplicación está registrada en el sistema. Existe el asunto al que se va a añadir un estado.		
Post-condición	Queda registrado el nuevo estado en la BD.		
ESCENARIOS			
#	Actor	#	Sistema
1	La aplicación envía los datos de un asunto existente en el sistema, cuyo estado varía	2	Se recogen los datos del asunto y se validan para evitar errores.
		3	Se insertan los datos en el sistema
		4	Se comunica del resultado de efectuar el proceso de alta

Eliminación de Asunto

CU.ELIM1	Borrado asunto		
Descripción	Una aplicación nos envía el borrado de un asunto, o conjunto total de asuntos, perteneciente a una persona o una entidad.		
Actores	Aplicación Gestora		
Pre-condición	La aplicación está registrada en el sistema. La aplicación tiene permisos para eliminar el asunto.		
Post-condición	Queda registrada la fecha de baja en la BD (borrado lógico).		
ESCENARIOS			
#	Actor	#	Sistema
1	La aplicación envía la identificación de un asunto existente en el sistema, o de la persona o entidad a la que pertenecen, para ser eliminado del sistema	2	Se recogen los datos identificativos del asunto y se validan para evitar errores.
		3	Se eliminan de forma lógica los datos en el sistema
		4	Se comunica del resultado de efectuar el proceso de alta

3.3.3 Subsistema de Administración del sistema

CU.ESTAD1	Consulta de estadísticas		
Descripción	Un usuario administrador desea consultar las estadísticas de utilización del sistema.		
Actores	Administrador		
Pre-condición	El usuario está logado en el sistema y tiene permisos para ello.		
Post-condición			
ESCENARIOS			
#	Actor	#	Sistema
1	El administrador solicita datos estadísticos de utilización del sistema.	2	Se efectúa la consulta de accesos a cada una de las funcionalidades del sistema, confeccionando algún tipo de salida.
		3	Se devuelve el resultado obtenido

Gestión de Aplicaciones Gestoras

CU.GA1	Alta aplicación		
Descripción	Un usuario administrador desea dar de alta una nueva aplicación gestora que accederá al sistema.		
Actores	Administrador		
Pre-condición	El usuario está logado en el sistema y tiene permisos para ello.		
Post-condición	Queda registrada en BD la nueva aplicación.		
ESCENARIOS			
#	Actor	#	Sistema
1	El administrador envía los datos de la nueva aplicación gestora que interviene en el sistema.	2	Recogida de datos de la nueva aplicación gestora; registro de la misma y asignación de permisos.
		3	Se devuelve el resultado obtenido

CU.GA2	Modificación aplicación		
Descripción	Un usuario administrador desea modificar los permisos de una aplicación gestora existente en el sistema.		
Actores	Administrador		
Pre-condición	El usuario está logado en el sistema y tiene permisos para ello.		
Post-condición	Queda registrada en BD la modificación.		
ESCENARIOS			
#	Actor	#	Sistema
1	El administrador envía los datos de la aplicación gestora y los permisos a modificar.	2	Recogida de datos de la aplicación gestora y permisos; modificación de permisos a funcionalidades.
		3	Se devuelve el resultado obtenido

Gestión de Usuarios

CU.GU1	Alta usuario		
Descripción	Un usuario administrador desea dar de alta un nuevo usuario, asignándole un rol, el cual le permitirá tener ciertos permisos dentro del sistema.		
Actores	Administrador		
Pre-condición	El usuario está logado en el sistema y tiene permisos para ello.		
Post-condición	Queda registrado en la BD el nuevo usuario.		
ESCENARIOS			
#	Actor	#	Sistema
1	El administrador envía los datos del nuevo usuario que intervendrá en el sistema.	2	Recogida de datos del nuevo usuario; registro del mismo y asignación de rol con permisos.
		3	Se devuelve el resultado obtenido

CU.GU2	Modificación usuario		
Descripción	Un usuario administrador desea modificar los datos de un usuario existente en el sistema.		
Actores	Administrador		
Pre-condición	El usuario está logado en el sistema y tiene permisos para ello.		
Post-condición	Queda registrado en la BD la modificación.		
ESCENARIOS			
#	Actor	#	Sistema
1	El administrador envía los datos del usuario y los permisos a modificar.	2	Recogida de datos del usuario y permisos; modificación de rol y/o permisos a funcionalidades.
		3	Se devuelve el resultado obtenido

Capítulo 4

Diseño

En este capítulo se abordará el diseño de cada uno de los subsistemas que componen el Sistema de Información.

4.1 Arquitectura del sistema

Como se ha mencionado anteriormente, el sistema consta de los siguientes subsistemas:

- **Portal web de información al ciudadano.** El acceso se realiza a través de Internet, identificándose al ciudadano a través de su Certificado Digital personal o jurídico (en caso de que desee consultar los procedimientos administrativos que ha iniciado una entidad jurídica).
- **Aplicación de administración/gestión.** El acceso a la aplicación se realizará por los usuarios autorizados. La identidad de los usuarios autorizados se comprobará mediante el Certificado Digital personal.
- **Web Service de notificación.** Aplicaciones externas se conectan a ellos desde la Intranet para notificar altas, bajas, borrado y también para realizar consultas. Los mensajes que envíen pueden estar firmados o no con su certificado.

Serán por tanto dos aplicaciones web y un servicio web.

Las aplicaciones web seguirán una arquitectura cliente-servidor en tres capas:

- **Nivel 1:** cliente que ofrece al usuario una interfaz para interactuar con el sistema y envía peticiones al servidor. En este caso será el navegador web desde el que se conecta el usuario.
- **Nivel 2:** servidor web que trata las peticiones del sistema.
- **Nivel 3:** base de datos donde se almacenan la información del sistema que necesita el servidor web.

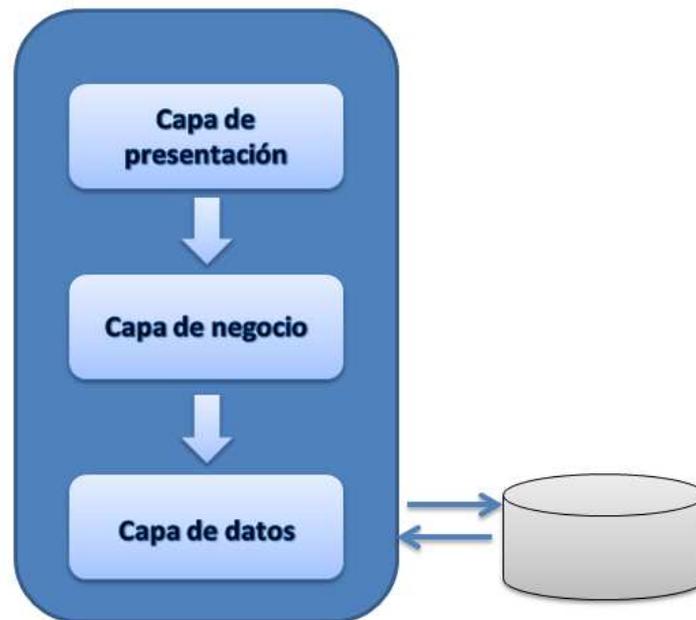


Figura 11. Arquitectura de tres capas

Para estas aplicaciones se utilizará un patrón de diseño Modelo-Vista-Controlador (MVC):

- **Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.
- **Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.
- **Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se

podría decir que el 'controlador' hace de intermediario entre la vista y el modelo.

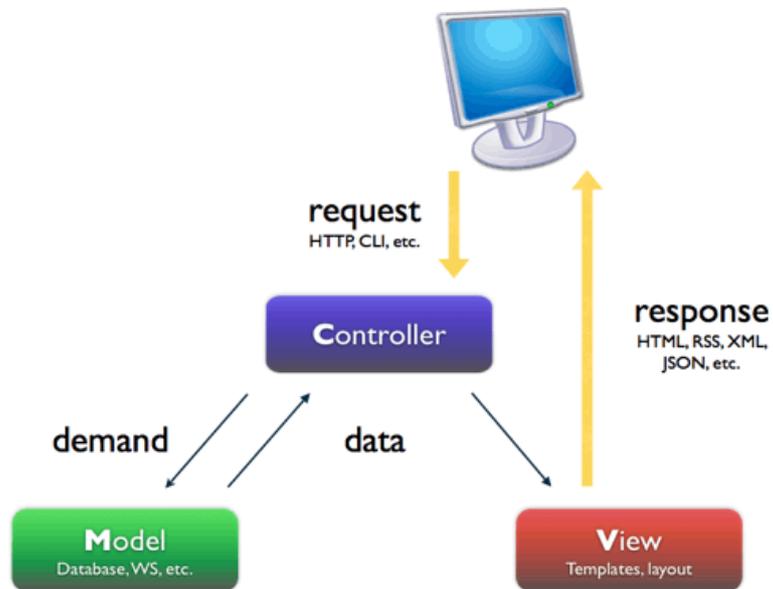


Figura 12. Patrón MVC

El servicio web también seguirá una arquitectura cliente-servidor en tres capas. En este caso el nivel 1 serán los clientes de las aplicaciones que se conectan al servicio web. Una de ellas será la aplicación de administración/gestión.

4.2 Diagrama de componentes del sistema

A continuación se detalla el diagrama de componentes que componen el sistema:

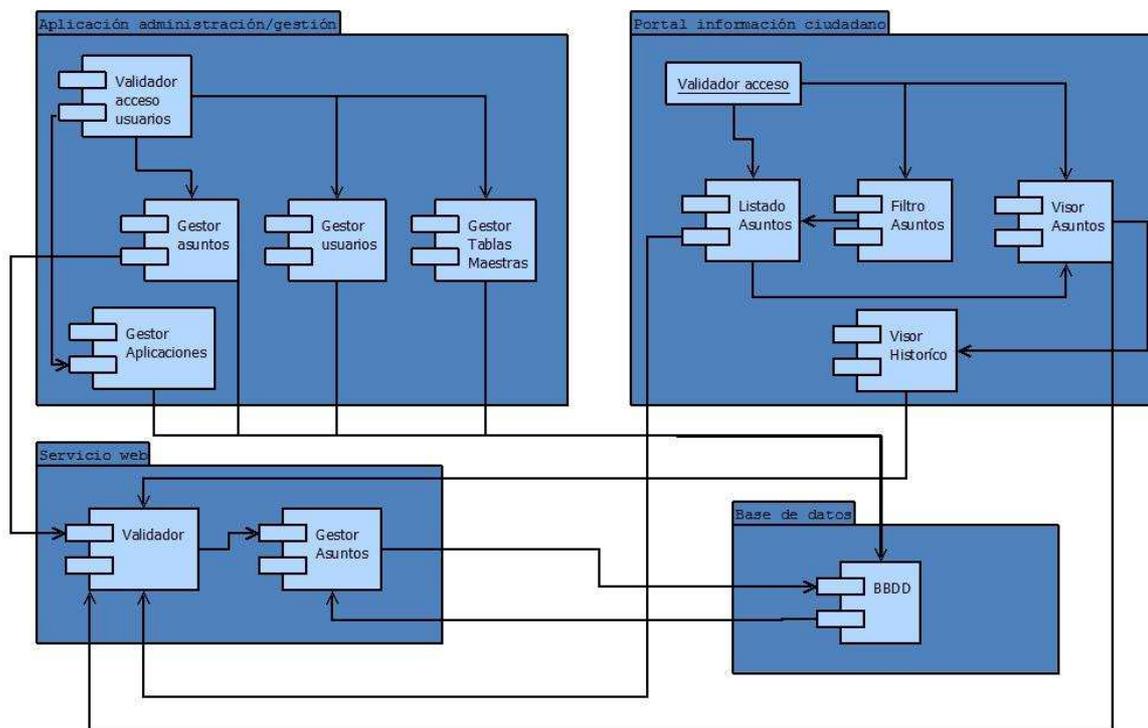


Figura 13. Diagrama de componentes del sistema

A continuación se detallan cada uno de los componentes. Para cada uno de ellos se indicará:

- **ID:** Identificador único del componente.
- **Nombre:** Nombre descriptivo del componente.
- **Descripción:** Descripción general del componente.
- **Función:** Funcionalidades que cumple cada componente.
- **Referencias:** Se indica los requisitos de usuario que se cubren en este componente.

COM.01	Validador acceso usuarios
Descripción	Valida que el usuario tiene permisos de acceso al sistema.
Función	Valida el certificado del usuario Comprueba que está registrado en el sistema Carga la lista de funciones que puede realizar con su perfil.
Referencias	FRQ-0012

COM.02	Gestor asuntos
Descripción	Permite gestionar los asuntos.
Función	Dar de alta un asunto/estado, Eliminar un asunto estado Consulta un asunto/estado
Referencias	FRQ-0001, FRQ-0002, FRQ-0003, FRQ-0004, FRQ-0005, FRQ-0006, FRQ-0007, FRQ-0008, FRQ-0016, FRQ-0017, FRQ-0018

COM.03	Gestor usuario
Descripción	Permite administrar los usuarios del sistema.
Función	Dar de alta un usuario. Eliminar un usuario. Modificar un usuario. Asignarle las aplicaciones con las que puede trabajar.
Referencias	FRQ-0013

COM.04	Gestor tablas maestras
Descripción	Permite administrar las tablas maestras del sistema.
Función	Añadir nuevos estados. Añadir nuevos roles. Asignar o eliminar funcionalidades de cada rol.
Referencias	FRQ-0015, FRQ-0019

COM.05	Gestor aplicaciones
Descripción	Permite administrar las aplicaciones que acceden al sistema.
Función	Añadir nuevas aplicaciones. Eliminar aplicaciones. Modificar aplicaciones.
Referencias	FRQ-0014

COM.06	Validador acceso
Descripción	Valida que el ciudadano puede acceder.
Función	Valida el certificado del usuario
Referencias	FRQ-0009

COM.07	Listado asuntos
Descripción	Muestra la información de los asuntos que tiene el ciudadano.
Función	Lista los asuntos para los que el ciudadano tiene información en el sistema. Muestra un enlace al detalle de cada asunto. Muestra un enlace al histórico en caso de tener varios estados asociados.
Referencias	FRQ-0001, FRQ-0002, FRQ-0003, FRQ-0010

COM.08	Visor asuntos
Descripción	Muestra la información detallada de un asunto.
Función	Muestra el detalle del estado actual del asunto. En caso de tener documentación adjunta, muestra un enlace para ver el adjunto.
Referencias	FRQ-0010

COM.09	Visor histórico
Descripción	Muestra la información del histórico de estados por los que ha pasado un asunto.
Función	Lista los estados por los que ha pasado el asunto.
Referencias	FRQ-0010

COM.10	Filtro de asuntos
Descripción	Permite filtrar los asuntos por las áreas.
Función	Muestra menú con enlaces a las áreas donde tiene asuntos el ciudadano. Filtra sólo los asuntos por esa área.
Referencias	FRQ-0010

COM.11	Validación
Descripción	Valida que la aplicación que está intentado utilizar el servicio web puede hacerlo.
Función	Valida que la aplicación está dada de alta. Valida que la aplicación tiene permisos para realizar la acción. Valida, en caso de utilizar el servicio con firma, que la firma es correcta.
Referencias	FRQ-0004

COM.12	Gestor asuntos
Descripción	Valida que la aplicación que está intentado utilizar el servicio web puede hacerlo.
Función	Valida que la aplicación está dada de alta. Valida que la aplicación tiene permisos para realizar la acción. Valida, en caso de utilizar el servicio con firma, que la firma es correcta.
Referencias	FRQ-0004, FRQ-0005, FRQ-0006, FRQ-0007, FRQ-0008, FRQ-0018

A continuación se muestra la matriz que traza los requisitos de usuario con los componentes del sistema:

TRM-0001	COM-01	COM-02	COM-03	COM-04	COM-05	COM-06	COM-07	COM-08	COM-09	COM-10	COM-11	COM-12
FRQ-0001	-	X	-	-	-	-	X	X	X	-	-	-
FRQ-0002	-	X	-	-	-	-	X	X	X	-	-	-
FRQ-0003	-	X	-	-	-	-	X	X	X	-	-	-
FRQ-0004	-	X	-	-	-	-	-	-	-	-	X	X
FRQ-0005	-	X	-	-	-	-	-	-	-	-	-	X
FRQ-0006	-	X	-	-	-	-	-	-	-	-	-	X
FRQ-0007	-	X	-	-	-	-	-	-	-	-	-	X
FRQ-0008	-	X	-	-	-	-	-	-	-	-	-	X
FRQ-0009	-	-	-	-	-	X	-	-	-	-	-	-
FRQ-0010	-	-	-	-	-	-	X	X	X	X	-	-
FRQ-0011	-	-	-	-	-	-	-	-	-	-	-	-
FRQ-0012	X	-	-	-	-	-	-	-	-	-	-	-
FRQ-0013	-	-	X	-	-	-	-	-	-	-	-	-
FRQ-0014	-	-	-	-	X	-	-	-	-	-	-	-
FRQ-0015	-	-	-	X	-	-	-	-	-	-	-	-
FRQ-0016	-	X	-	-	-	-	-	-	-	-	-	-
FRQ-0017	-	X	-	-	-	-	-	-	-	-	-	-
FRQ-0018	-	X	-	-	-	-	-	-	-	-	-	X
FRQ-0019	-	-	-	X	-	-	-	-	-	-	-	-

Como se puede ver, todos los requisitos de usuario tienen al menos algún componente que lo implementa.

4.3 Elaboración del modelo de procesos

4.3.1 Acceso al Sistema

Todo acceso al sistema requerirá una identificación por medio de certificación digital, comprobándose posteriormente la existencia del usuario entre los registrados y con permiso de acceso.

Descripción

- 1) El usuario accede al sistema y se identifica con un Certificado Digital; este certificado podrá ser personal o de empresa.
- 2) Se validará el certificado digital según la entidad certificadora del mismo, que nos informará de su estado.
- 3) Si el certificado es válido, obtendremos los datos del mismo para comprobar la existencia del usuario registrado en el sistema.
- 4) Si el usuario se encuentra registrado, permitiremos su acceso al sistema.
- 5) Si el certificado no es válido o el usuario no se encontrase registrado, mostraremos un mensaje informativo indicándolo.

Esquema

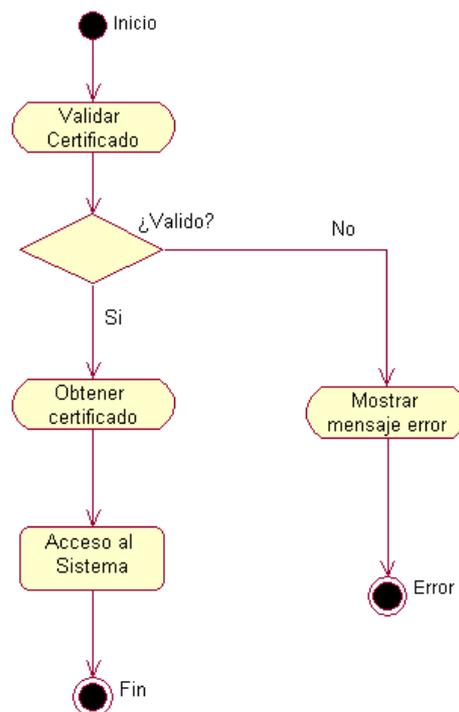


Figura 14. Acceso al sistema

4.3.2 Validación de XML

El sistema recibirá mensajería XML para la realización de las diferentes funcionalidades que se pueden llevar a cabo por la misma. Cualquier aplicación gestora que desea notificar la realización de algún proceso por parte del sistema, generará un mensaje XML con la información necesaria, el cual será enviado para su tratamiento.

Por ello, es necesario realizar una validación de los datos recibidos, antes de llevar a cabo la realización del proceso requerido, notificando a su finalización del resultado del mismo para informar del mismo a la aplicación gestora que lo solicitó.

Descripción

- 1) La aplicación gestora notifica, vía mensajería XML, su deseo de realizar algún tipo de funcionalidad (entrada, modificación, eliminación o consulta de asuntos).
- 2) Se validará el mensaje XML, comprobando que se cumple el formato especificado, así como valores coherentes con lo esperado (ej.: relaciones válidas entre usuario, asunto, procedimiento, etc.).
- 3) Si el mensaje XML es válido, procederemos a la tramitación en el sistema del proceso solicitado.
- 4) Si el mensaje XML no tiene un formato válido o alguno de sus datos presenta incoherencia, se devolverá un mensaje XML indicando el error encontrado.

Esquema

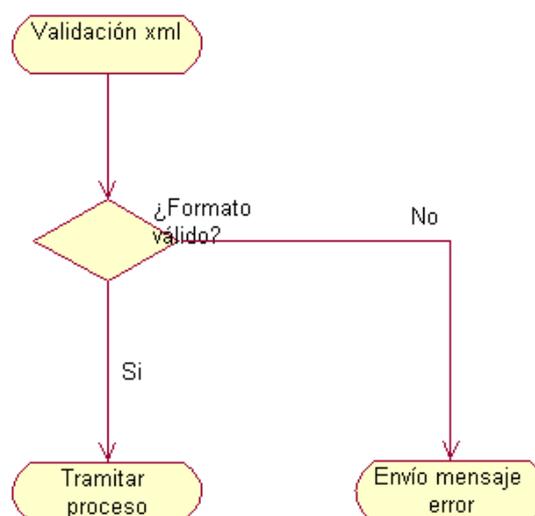


Figura 15. Validación de un XML

4.3.3 Consulta de asuntos

El sistema permitirá la consulta del estado en que se encuentra cada uno de los asuntos registrados.

Dependiendo de quién realiza la consulta, la búsqueda será más o menos restringida:

- Si la consulta la realiza un usuario, éste sólo accederá a la información de sus asuntos que se encuentren activos
- Si la consulta la realiza una aplicación gestora, podrá acceder a todos los asuntos que ésta haya gestionado
- Si la consulta la realiza una aplicación privilegiada, podrá acceder a la totalidad de asuntos existentes en el sistema.

Descripción

- 1) Un usuario o una aplicación gestora notifica, vía mensajería XML, su deseo de realizar una consulta de asuntos, y el XML ha sido validado por el sistema sin encontrarse ningún tipo de error.
- 2) Se comprueba si la consulta es realizada por un usuario determinado, en cuyo caso se comprobará si se trata de una persona física o de una empresa.
- 3) Si se accede con certificación de empresa, se permitirá indicar si la consulta se realiza por la DNI personal o por el CIF de la entidad a la que representa.
- 4) Si la consulta se realiza filtrando por DNI, se obtendrá el listado de asuntos que dicha persona tenga activos en ese momento, y que cumplan los criterios aplicados (si éstos existen).
- 5) Si la consulta se realiza filtrando por CIF, se obtendrá el listado de asuntos de todas las personas pertenecientes a la entidad, que se encuentren activos en ese momento, y que cumplan los criterios aplicados (si éstos existen).
- 6) Si la consulta es realizada desde una aplicación gestora, se comprueba que los criterios de consulta se corresponden con asuntos pertenecientes a la misma, en cuyo caso se obtiene el listado de dichos asuntos.
- 7) Si la consulta es realizada desde una aplicación gestora, que desea acceder a asuntos que no le pertenecen, se comprueba si dicha aplicación gestora es de carácter privilegiada, en cuyo caso se obrará como el punto anterior; en caso contrario, se generará un mensaje de error indicándolo.

- 8) En todos los casos, se generará un mensaje XML de respuesta para notificar de los resultados obtenidos en la consulta realizada. Estos resultado podrán componerse de información de un único asunto consultado, o de un conjunto de asuntos que cumplen con los criterios especificados.

Esquema

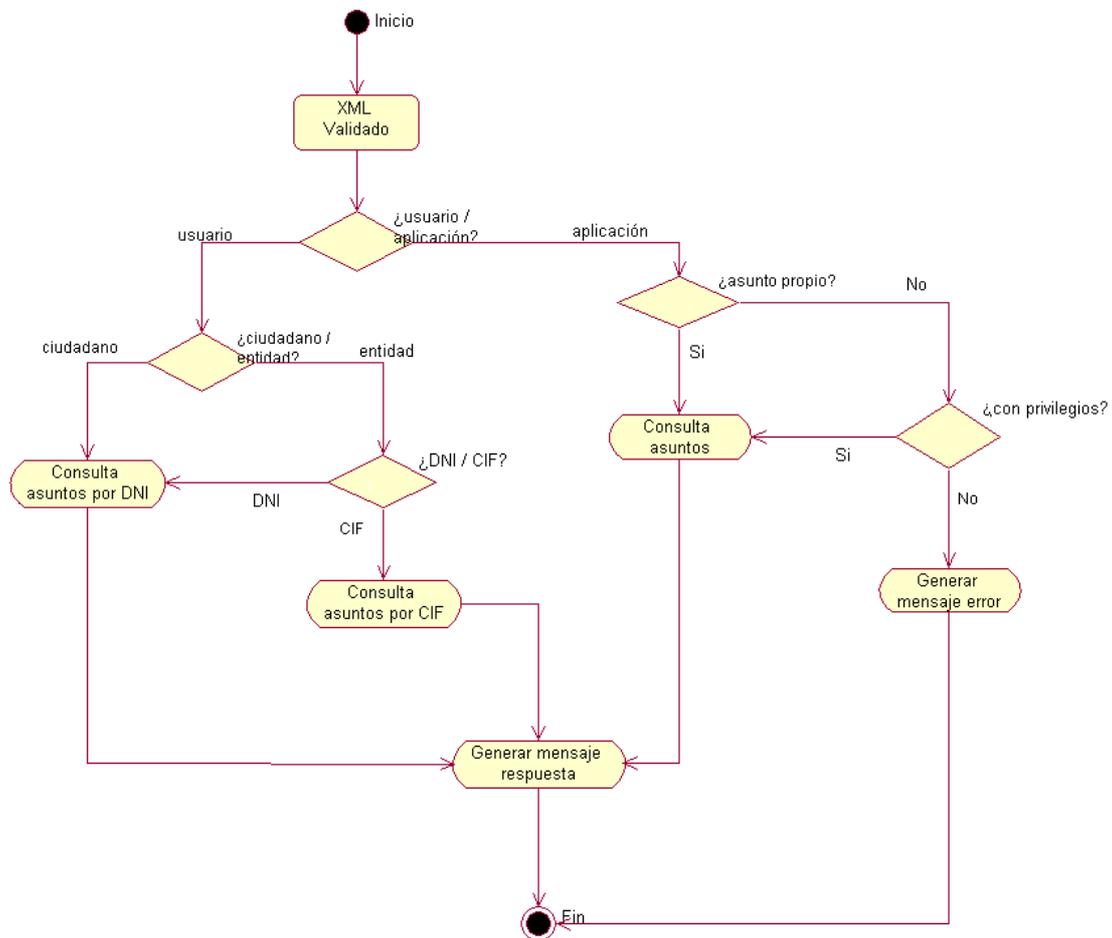


Figura 16. Consulta de un asunto

4.3.4 Alta de asunto

El sistema permitirá la inserción de nuevos asuntos por parte de aplicaciones gestoras. Se permitirá indicar un código identificativo de la misma, o bien si no se especifica éste será generado automáticamente por el sistema y notificado en el mensaje de respuesta para que la aplicación gestora haga uso del mismo en posteriores mensajes en los que desee realizar referencia al mismo.

Descripción

- 1) Una aplicación gestora notifica, vía mensajería XML, su deseo de realizar el alta de un nuevo asunto, y el XML ha sido validado por el sistema sin encontrarse ningún tipo de error.
- 2) Si en los datos especificados no se indica código identificativo, se genera uno de forma automática.
- 3) Se graban los datos del asunto.
- 4) Se generará un mensaje XML de respuesta para notificar de los resultados obtenidos en el proceso de inserción realizado.

Esquema

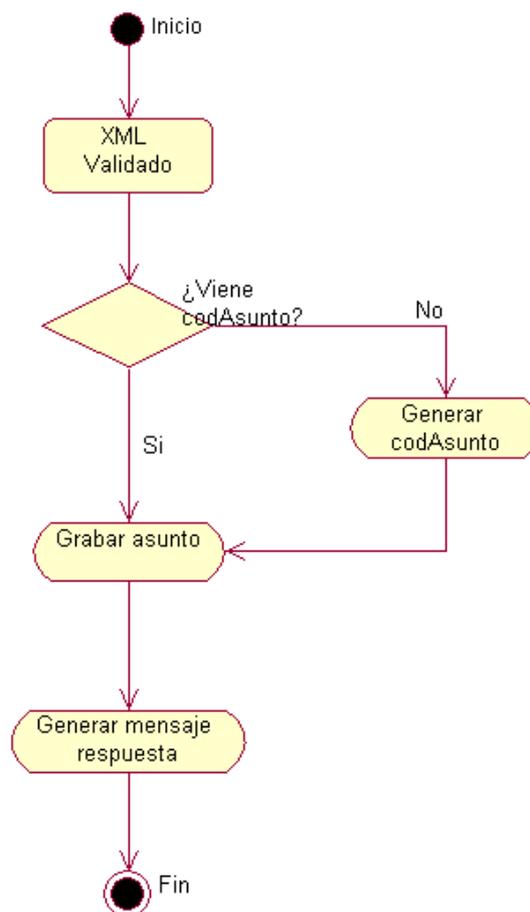


Figura 17. Alta de un asunto

4.3.5 Cambio estado de asunto

El sistema permitirá ir cambiando sucesivamente el estado en el que se encuentra un asunto por parte de aplicaciones gestoras. Con ello, se constituirá un histórico con los

diferentes estados por los que ha ido pasando el asunto, considerando como estado activo el que tenga fecha de notificación más reciente.

Este tipo de proceso obliga siempre a indicar el código identificativo del asunto que se desea tramitar, ya que sino no habría forma de identificarlo.

Descripción

- 1) Una aplicación gestora notifica, vía mensajería XML, su deseo de cambiar el estado de un asunto, y el XML ha sido validado por el sistema sin encontrarse ningún tipo de error.
- 2) Si en los datos especificados no se indica código identificativo, se generará un mensaje de respuesta especificando error.
- 3) Se procede al cambio de estado del asunto especificado.
- 4) Se generará un mensaje XML de respuesta para notificar de los resultados obtenidos en el proceso de modificación realizado.

Esquema

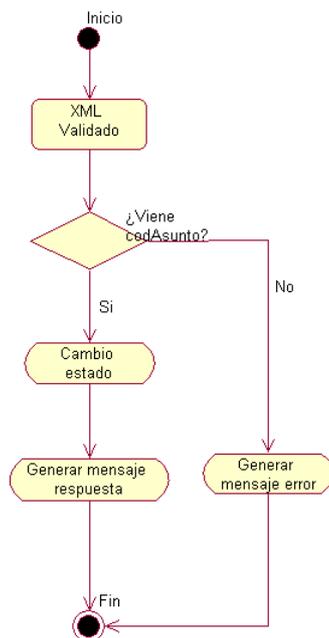


Figura 18. Cambio de estado de un asunto

4.3.6 Eliminación de asunto

El sistema permitirá a las aplicaciones gestoras la eliminación de un asunto, o de la totalidad de asuntos pertenecientes a un ciudadano o entidad.

Este tipo de proceso obliga siempre a indicar el código identificativo del asunto que se desea eliminar, o bien el DNI o CIF de la persona o entidad a la que pertenecen los asuntos a borrar, respectivamente.

Descripción

- 1) Una aplicación gestora notifica, vía mensajería XML, su deseo de eliminar un asunto o conjunto de asuntos pertenecientes a una persona o entidad, y el XML ha sido validado por el sistema sin encontrarse ningún tipo de error.
- 2) Si se especifica código identificativo de asunto, se comprueba si se ha especificado un estado determinado del mismo.
- 3) Si sólo se ha especificado código identificativo de asunto, se procede a la eliminación del asunto especificado del sistema.
- 4) Si se especifica código identificativo de asunto y un código de estado determinado del mismo, se procede a la eliminación del estado de asunto especificado del sistema.
- 5) Si se especifica DNI o CIF, se procede a la eliminación de la totalidad de asuntos pertenecientes a la persona o entidad especificada del sistema.
- 6) Se generará un mensaje XML de respuesta para notificar de los resultados obtenidos en el proceso de modificación realizado.

Esquema

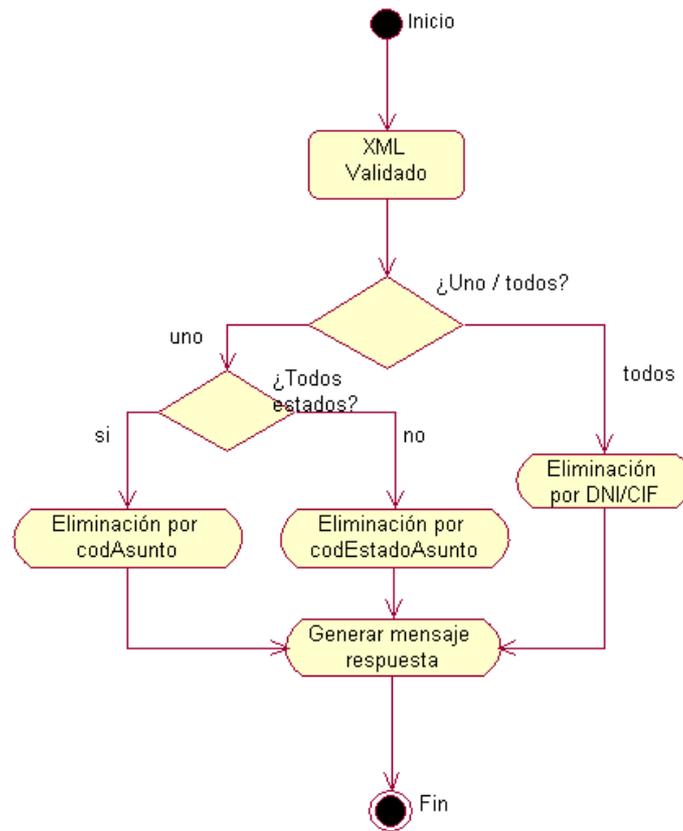


Figura 19. Eliminación de un asunto

4.3.7 Mantenimiento usuarios / aplicación

El sistema dispondrá de una funcionalidad que permita el mantenimiento de datos relativos a usuarios y aplicaciones gestoras que podrán acceder al sistema.

Descripción

- 1) Un usuario administrador accede a las funcionalidades administrativas del sistema.
- 2) Si el usuario administrador dispone de permisos de mantenimiento, se realiza la operación deseada.
- 3) Si el usuario administrador no dispone de permisos de mantenimiento, se mostrará un mensaje informativo indicándolo.

Esquema

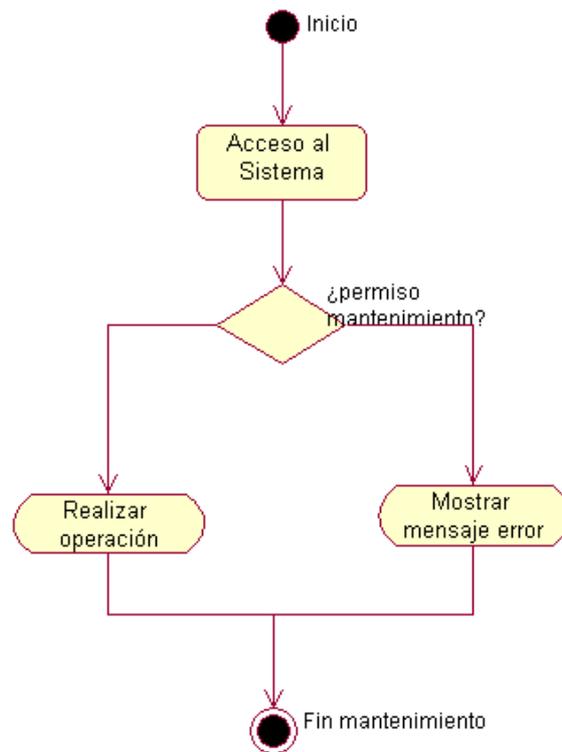


Figura 20. Mantenimiento de usuarios

4.4 Diseño de la base de datos

Existirá una única base de datos que será común para todos los subsistemas. En ella se almacenará la información de los asuntos de los ciudadanos/entidades, así como el resto de datos necesarios para las aplicaciones de gestión y administración.

A continuación se detalla el modelo Entidad-Relación (E/R) que detalla las entidades relevantes de nuestro sistema así como sus interrelaciones y propiedades.

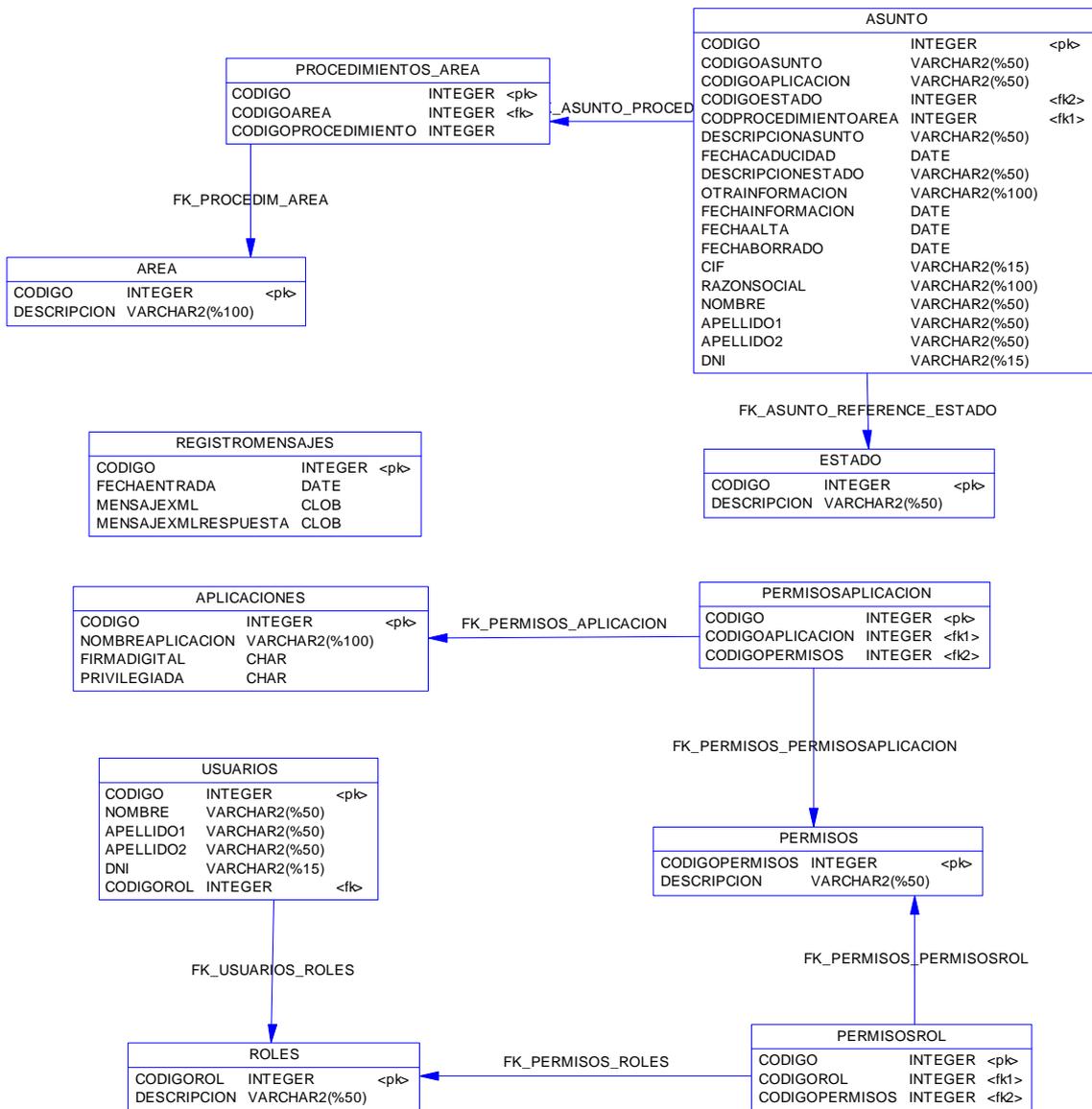


Figura 21. Modelo de BD

A continuación se detalla para cada una de las entidades su significado y el detalle de cada uno de los atributos.

ASUNTO	
Contiene la información de los asuntos que han sido dados de alta para los ciudadanos o entidades.	
Campo	Descripción
CODIGO	Identificador único interno para cada uno de los asuntos.
CODIGOASUNTO	Identificador de las aplicaciones para los asuntos
CODIGOAPLICACION	Identificador único para cada aplicación.
CODIGOESTADO	Identificador único del estado del asunto.
CODIGOPROCEDIMIENTOAREA	Identificador único para cada procedimiento.
DESCRIPCIONASUNTO	Descripción del asunto.
FECHACADUCIDAD	Fecha caducidad del asunto.
DESCRIPCIONESTADO	Descripción del estado.
OTRAINFORMACION	Nombre del fichero adjunto
FECHAINFORMACION	Fecha de la información.
FECHAALTA	Fecha de alta de la información en el sistema.
FECHABORRADO	Fecha de borrado de la información (el borrado es lógico)
CIF	CIF de la entidad a la que está asociado el asunto.
RAZONSOCIAL	Nombre de la entidad a la que está asociado el asunto.
NOMBRE	Nombre del ciudadano al que está asociado el sistema.
APELLIDO1	Primer apellido del ciudadano.
APELLIDO2	Segundo apellido del ciudadano.
DNI	DNI del ciudadano

ESTADO	
Contiene la información de los estados.	
Campo	Descripción
CODIGO	Identificador único del estado.
DESCRIPCION	Descripción del estado.

APLICACIONES	
Contiene la información de las aplicaciones que pueden gestionar asuntos.	
Campo	Descripción
CODIGO	Identificador único para cada aplicación.
NOMBREAPLICACION	Nombre de la aplicación.
FIRMADIGITAL	Indica si es exigida la firma digital.
PRIVILEGIADA	Indica si es una aplicación con privilegios especiales.

PERMISOSAPLICACION	
Contiene la relación de permisos que tiene una aplicación.	
Campo	Descripción
CODIGO	Identificador único para el permiso de la aplicación.
CODIGOAPLICACION	Identificador único de la aplicación.
CODIGOPERMISOS	Identificador único del permiso.

PERMISOS	
Contiene la información de los permisos que una aplicación o profesional puede tener.	
Campo	Descripción
CODIGOPERMISOS	Identificador único para el permiso.
DESCRIPCION	Descripción del permiso.

PERMISOSROL	
Contiene la información de los permisos que tiene asociados un rol.	
Campo	Descripción
CODIGO	Identificador único para la relación.
CODIGOROL	Identificador único del rol.
CODIGOPERMISOS	Identificador único del permiso.

ROLES	
Contiene la información de los roles que puede tener un profesional.	
Campo	Descripción
CODIGOROL	Identificador único del rol.
DESCRIPCION	Descripción del rol.

USUARIOS	
Contiene la información de los usuarios de la aplicación de gestión.	
Campo	Descripción
CODIGO	Identificador único del usuario.
NOMBRE	Nombre del usuario.
APELLIDO1	Primer apellido del usuario.
APELLIDO2	Segundo apellido del usuario.
DNI	DNI del usuario.
CODIGOROL	Identificador único del rol del usuario.

REGISTROMENSAJES	
Contiene la información de los mensajes recibidos por las aplicaciones.	
Campo	Descripción
CODIGO	Identificador único del mensaje.
FECHAENTRADA	Fecha de entrada del mensaje.
MENSAJEXML	Mensaje XML.
MENSAJEXMLRESPUESTA	Mensaje XML de respuesta.

AREA	
Contiene la información de las áreas en las que pueden existir procedimientos	
Campo	Descripción
CODIGO	Identificador único del área.
DESCRIPCION	Descripción del área.

PROCEDIMIENTOS_AREA	
Contiene la información de los procedimientos asociados a un área.	
Campo	Descripción
CODIGO	Identificador único de la relación.
CODIGOAREA	Identificador único del área.
CODIGOPROCEDIMIENTO	Identificador único del procedimiento.

4.5 Diagrama de navegación

A continuación se mostrarán la navegación de pantallas básica para cada subsistema.

4.5.1 Aplicación de administración

En primer lugar se mostrará una página de inicio en la que la parte de arriba contendrá el título de la aplicación, el nombre del usuario que está logueado en la aplicación y la fecha del último acceso. A continuación se muestra un menú básico con las opciones posibles:

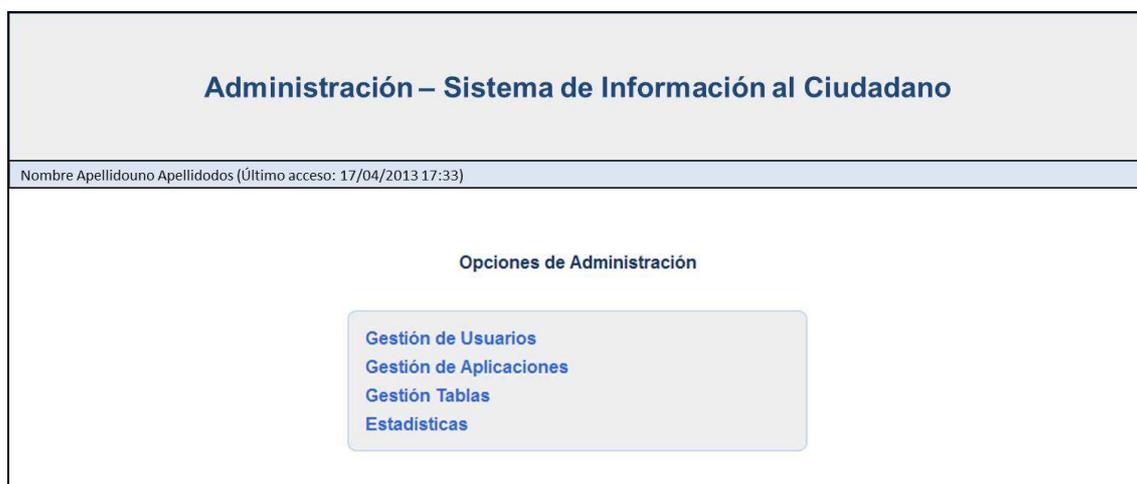


Figura 22. Pantalla de inicio

Cada una de las opciones será un enlace que llevará a la página en cuestión. A modo de ejemplo se muestran algunas pantallas.

En primer lugar una página de consultas de usuarios:

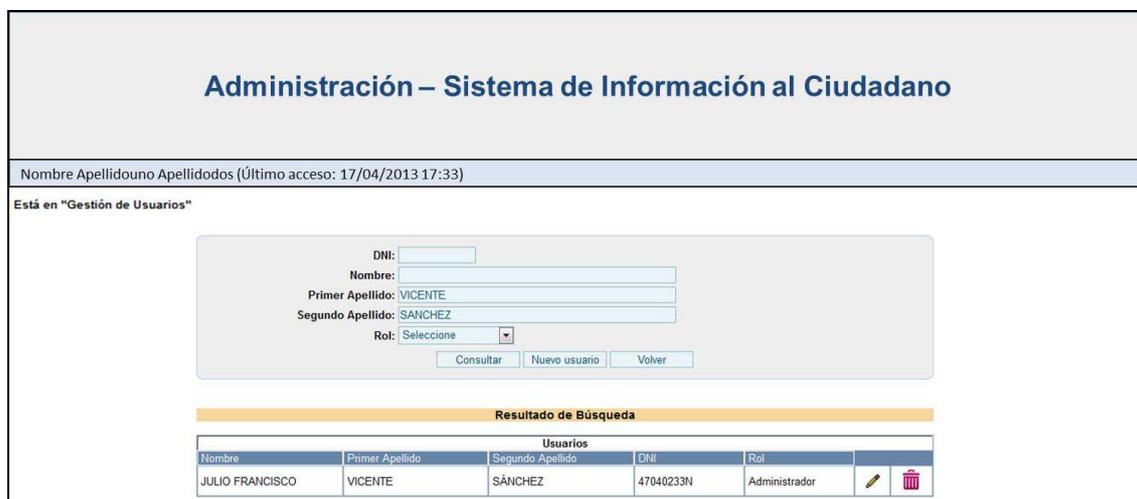


Figura 23. Pantalla de consulta de usuarios

En esta pantalla en función de los criterios de búsqueda en la parte inferior aparecerá un listado con los usuarios que cumplen los criterios de búsqueda. Al lado de cada registro aparecerán dos iconos de las acciones posible sobre el usuario: modificar y borrar.

A continuación se muestra una pantalla de modificación de la información de una aplicación:

Administración – Sistema de Información al Ciudadano

Nombre Apellidoouno Apellidos (Último acceso: 17/04/2013 17:33)

Está en "Gestión de Aplicaciones >> Modificar Aplicación"

Nombre aplicación: NO_PRIVILEG_1

Abreviatura: NO_PRIVILEG_1

Privilegiada:

Utiliza antivirus internamente:

Permitir grabar el DN:

DN: CN="DESCRIPCION PRUEBAS - ENTIDAD - CIF S1527002D", OU=210060112, OU=P

Permiso: Seleccione

ALTA_ASUNTO

CONSULTA_ASUNTO

ELIM_ASUNTO

Añadir Borrar

Grabar Volver

Figura 24. Pantalla modificación de una aplicación

Desde esta pantalla se pueden modificar la información asociada a la aplicación. Se le puede cambiar por ejemplo el nombre, abreviatura, el certificado con el que firma, los permisos que tiene asociados la aplicación...

4.5.2 Aplicación de gestión

La aplicación de gestión seguirá un formato similar. En primer lugar la página de inicio con el menú de acciones posibles:

Gestión – Sistema de Información al Ciudadano

Nombre Apellidoouno Apellidos (Último acceso: 17/04/2013 17:33)

Opciones de Gestión

Nuevo Asunto

Eliminar Asunto

Consultas

Figura 25. Pantalla de inicio

Cada enlace nos llevará a cada una de las posibles acciones que pueden realizar los usuarios de gestión.

A continuación se muestra la pantalla de un nuevo asunto:

The screenshot shows a web application interface with a header 'Gestión - Sistema de Información al Ciudadano' and a user session bar. The main content area is titled 'Nuevo Asunto' and contains a form with the following fields:

- Asunto de:
- DNI:
- Nombre:
- Primer Apellido:
- Segundo Apellido:
- CIF:
- Razón Social:
- Procedimiento:
- Código Asunto:
- Descripción asunto:
- Fecha información:
- Fecha caducidad:
- Estado:
- Descripción estado:
- Otra información: No se ha seleccionado ningún archivo.
- Aplicación:

At the bottom of the form are two buttons: 'Grabar' and 'Volver'.

Figura 26. Pantalla de alta de un asunto

Desde aquí se pueden rellenar los datos del asunto en cuestión. Que podrán ser tanto de ciudadanos como de entidades.

Una vez que se graba el asunto, se muestra una pantalla de confirmación.

The screenshot shows the same application header and session bar. The main content area displays a confirmation message:

Los datos del asunto han sido registrados correctamente en el sistema

Los datos de confirmación obtenidos en el proceso son:
Código Asunto Único: 287
Código Asunto: NO_PRIVILEG_1 - 6

At the bottom of the confirmation box are two buttons: 'Nuevo asunto' and 'Salir'.

Figura 27. Pantalla de confirmación de alta de un asunto

En ella se muestra el mensaje de que se ha registrado correctamente así como los datos del identificado del asunto generado.

A continuación se muestra la pantalla de eliminación de asuntos:

Gestión – Sistema de Información al Ciudadano

Nombre Apellidoouno Apellidos (Último acceso: 17/04/2013 17:33)

Eliminar Asunto

Aplicación: NO_PRIVILEG_1

Asunto de: Seleccione

DNI:

Nombre:

Primer Apellido:

Segundo Apellido:

CIF:

Razón Social:

Código Asunto:

Código Asunto Único: 287

Figura 28. Pantalla de eliminación de asuntos

Rellenando los datos del formulario y eliminará los asuntos que cumplan los criterios indicados. El resultado de los asuntos eliminado se muestra en la pantalla de confirmación:

Gestión – Sistema de Información al Ciudadano

Nombre Apellidoouno Apellidos (Último acceso: 17/04/2013 17:33)

El/Los asunto/s coincidente/s con los criterios indicados ha/n sido eliminado/s correctamente en el sistema

Los asuntos afectados en el proceso son:

Código Asunto Único	Código Asunto
287	NO_PRIVILEG_1 - 6

Figura 29. Pantalla de confirmación de eliminación de un asunto

Se mostrará un listado con la información de los asuntos eliminados.

4.5.3 Portal del ciudadano

En el caso del portal del ciudadano, las funcionalidades son más básicas porque sólo puede consultar. En primer lugar aparecerá una pantalla de inicio con en el listado de todos los asuntos del ciudadano en la parte central. Y en la parte izquierda un menú para filtrar el listado de los asuntos en según las áreas a las que pertenezcan.



Figura 30. Pantalla de inicio

Al lado de cada asunto aparecerán dos iconos que permitirán ver la información detallada del asunto o ver el histórico (en el caso de tenerlo, sino el icono no aparecería). A continuación se muestra un ejemplo de la pantalla del detalle de un asunto

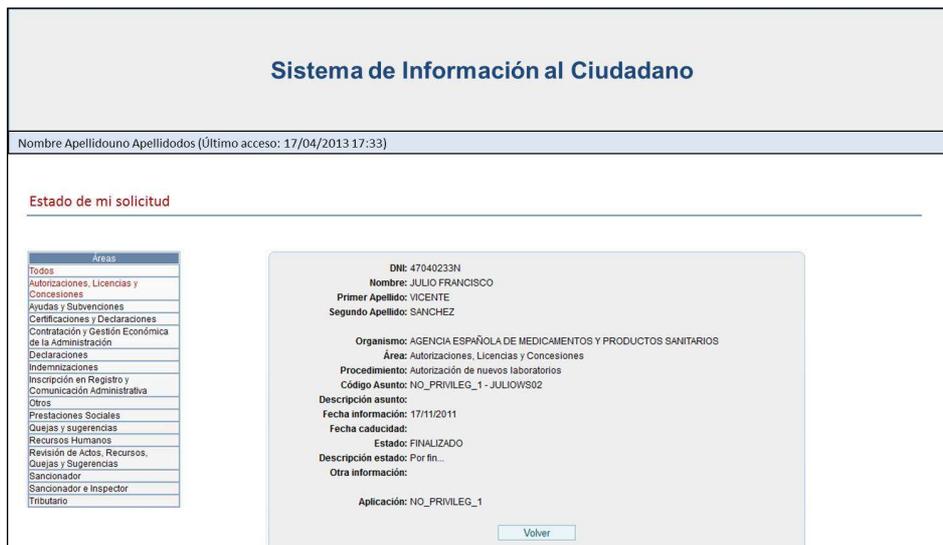


Figura 31. Pantalla de detalle de un asunto

Capítulo 5

Implementación y plan de pruebas.

En este capítulo se abordará el enfoque seguido en la implementación del Sistema de Información. Así mismo se detallará el plan de pruebas diseñado para asegurar el correcto funcionamiento.

5.1 Implementación

El sistema como hemos ido viendo a lo largo del proceso se compone de varios subsistemas: aplicaciones de administración y gestión, y el servicio web.

En primer lugar se abordará el proceso de implementación de las aplicaciones web y segundo lugar el servicio web. Por similitud, se detalla la implementación para un caso específico de la aplicación de administración, por ejemplo el alta de un usuario. Ello cubrirá los aspectos básicos del proceso de implementación.

5.1.1 Aplicaciones Web

Para la implementación se ha utilizado el framework Struts, versión 1.3. No es la versión más actual pero se escoge por ser la estándar en la organización.

Como hemos mencionado nos centraremos en el caso de uso de dar de alta un nuevo usuario en el sistema, ya que sirve de ejemplo al ilustrar los pasos básicos en la implementación de todos los casos de uso.

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC. Ello permite delegar muchos aspectos básicos en el framework despreocupándose de ellos. El controlador de Struts se basa en un archivo de configuración XML en el que se definen las acciones que se pueden realizar y los caminos a los que se conduce desde dichas acciones. También se indica que hacer en caso de producirse algún error y también nos permite realizar validaciones de la información enviada de forma estándar.

A continuación se detalla de forma resumida todos los elementos que intervienen y posteriormente se irán detallando cada uno de los aspectos involucrados.

En el archivo de configuración (struts-config.xml) en primer lugar se declarará el actionForm que será el encargado de capturar la información enviada al servidor desde el cliente:

```
<form-beans>
    ...
<form-bean name="datosUsuarioForm"
type="ciudadano.admin.usuarios.model.DatosUsuarioActionForm" />
    ...
</form-beans>
```

En él se indica el nombre (name) que se le da al actionForm y la clase (type) que lo implementa.

Posteriormente se indicará la acción que tratará el alta de un nuevo usuario:

```
<action input="/admin/usuarios/nuevoUsuario.jsp" name="datosUsuarioForm"
path="/tramitarNuevoUsuarioAction" scope="request"
type="ciudadano.admin.usuarios.action.TramitarNuevoUsuarioAction"
validate="true">
    <forward name="confirmacion"
path="/admin/usuarios/confirmacionNuevoUsuario.jsp" />
</action>
```

En él se indicará el nombre (name) del actionForm que tratará la información, el nombre de la acción (path) que se invocará en el cliente y la clase que implementa la acción indicada (type). Además, se indica a que página se dirigirá en caso de que el alta se haga de forma satisfactoria (en el forward). También se indicará donde se redirigirá en caso de que haya algún error de validación (input).

Para el desarrollo de las páginas se ha utilizado JSP.

JSP nuevoUsuario.jsp

Todo el proceso se inicia en la página que presenta al usuario administrador el formulario para dar de alta un nuevo usuario. En esa página se crea un formulario web para que el administrador registre la información del nuevo usuario. Por simplificar se ha eliminado estilos sólo dejándose el formulario en cuestión:

```

<html:form action="/tramitarNuevoUsuarioAction.do" method="POST">
  <bean:message key="DNI"/>:
  <html:text property="dni" size="10" maxlength="15"></html:text>
  <br/>
  <logic:messagesPresent property="dniError">
    <html:errors property="dniError"/></td>
    <br/>
  </logic:messagesPresent>

  <bean:message key="Nombre"/>:
  <html:text property="nombre" size="50" maxlength="50"></html:text>
  <br/>
  <logic:messagesPresent property="nombreError">
    <html:errors property="nombreError"/>
    <br/>
  </logic:messagesPresent>

  <bean:message key="Apellido1"/>:
  <html:text property="apellido1" size="50" maxlength="50"></html:text>
  <br/>
  <logic:messagesPresent property="apellido1Error">
    <html:errors property="apellido1Error"/>
    <br/>
  </logic:messagesPresent>
  ...
</html:form>

```

Para crear el formulario web se ha utilizado las facilidades que da el framework struts, utilizándose sus etiquetas que posteriormente al compilarse la página en el servidor, ser renderizarán dichas etiquetas a código HTML.

Se ve como en el formulario se ha declarado que la acción que debe tratar el envío es tramitarNuevoUsuarioAction.do, que es lo que estaba mapeado en el fichero struts-config.xml

En el formulario se aprecian mensajes de error. Esos mensajes son los que se mostrarán en el caso de que al enviarse el formulario, al hacer la validación se produzcan errores.

ActionForm DatosUsuarioActionForm

Es una clase que extiende de la clase ActionForm perteneciente a Struts. Lo que vamos a contar es una de las funcionalidades que ofrece el framework y nos hace más simple la implementación.

```

public class DatosUsuarioActionForm extends ActionForm {

    private String dni = "";
    private String nombre = "";
    private String apellido1 = "";
    ...

    public String getDni() {
        return dni;
    }
    public void setDni(String dni) {
        this.dni = dni;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getApellido1() {
        return apellido1;
    }
    public void setApellido1(String apellido1) {
        this.apellido1 = apellido1;
    }
    ...
}

```

Al enviarse el formulario anterior al servidor el controlador de Struts tiene mapeado que el ActionForm que recuperará la información es DatosUsuariosActionForm. Esta clase es un bean que tendrá una serie de atributos que deben coincidir con los nombres de los campos declarados en el formulario de la página. A parte contendrá los métodos getters y setters para dichos campos.

De forma automática se recupera en esa clase la información enviada desde el cliente.

Además, en caso al haberse indicado en el fichero struts-config.xml que esta clase debe validarse (validate = "true") se invocará al método validate() incluido en dicha clase:

```

public ActionErrors validate(ActionMapping actionMapping, HttpServletRequest
HttpServletRequest) {

    ActionErrors errors = new ActionErrors();

    try{
        ValidarFormularios.validarString(errors, "dniError",
Misc.nz(this.getDni()), true, new HashMap(),
Constantes.LONGITUD_MAX_DNI_USUARIO, true);

        ValidarFormularios.validarString(errors, "nombreError",
Misc.nz(this.getNombre()), true, new HashMap(),
Constantes.LONGITUD_MAX_NOMBRE_USUARIO, true);

        ValidarFormularios.validarString(errors, "apellido1Error",
Misc.nz(this.getApellido1()), true, new HashMap(),
Constantes.LONGITUD_MAX_APELLIDO1_USUARIO, true);
        ...
    }
    catch(Exception e){
        errors.add("Error", new ActionMessage("errorAplicacion"));
    }
    return errors;
}

```

Este método se van realizando las distintas validaciones y añadiendo en el ActionError los errores que se detecten junto con el mensaje que se mostrará al usuario. Si hay al menos un error, el controlador de Struts nos redirige a la página que trata los errores. Que en nuestro caso es también nuevoUsuario.jsp. Como vimos anteriormente, en esta página ahora se mostrarán cada uno de los errores de validación que se hayan producido.

Action TramitarNuevoUsuarioAction

Un Action es una clase que extiende de la clase de Struts Action. Esta clase recoge la petición recibida desde el cliente y llama a la lógica de negocio que tiene que tratar la petición.

```

public class TramitarNuevoUsuarioAction extends Action {

    public ActionForward execute(ActionMapping actionMapping, ActionForm
    actionForm, HttpServletRequest httpRequest, HttpServletResponse
    httpResponse) {

        try {

            DatosUsuarioActionForm formulario = (DatosUsuarioActionForm)actionForm;

            AltaUsuario objAltaUsuario = new AltaUsuario();
            Salida objSalida = objAltaUsuario.tramitar(formulario);

            if(!objSalida.getError())
            {
                MensajePageBean confirmacion = new MensajePageBean();
                confirmacion.setDescripcionError(Constants.MSG_ERROR);
                httpRequest.setAttribute("PageBean", confirmacion);
                return actionMapping.findForward("confirmacion");
            }
            else
            {
                return actionMapping.findForward("errorAdministracion");
            }
        }
        catch(java.lang.Throwable e)
        {
            MensajePageBean error = new MensajePageBean();
            error.setDescripcionError(Constants.MSG_ERROR);
            httpRequest.setAttribute("PageBean", error);
            return actionMapping.findForward("errorAdministracion");
        }
    }
}

```

Esta clase recibe la petición. En primer lugar recupera el formulario recibido desde cliente.

A continuación invoca a la clase que tratará el alta del usuario AltaUsuario. Si la operación no produjo ningún error se indica la acción que debe continuar “confirmacion”, la cual está mapeada en el struts-config.xml para que redirija a la página confirmacionNuevoUsuario.jsp

En caso de producirse algún error se redirige a la acción “errorAdministracion”, la cual está mapeada en el struts-config como una acción genérica y nos lleva a la página de genérica de error errorAdmin.jsp

DAO AltaUsuario

Esta clase es la interfaz a través de la cuál se realiza el acceso a la base de datos para registrar el nuevo usuario.

```
public class AltaUsuario {

    public Salida tramitar(DatosUsuarioActionForm formulario){

        Salida objSalida = new Salida();
        String comentarioError = "";

        StringBuffer sbQuery = new StringBuffer();
        HashMap<String,String> parametros = new HashMap<String,String>();
        int cont = 1;

        AccesoBd bd = null;

        try{

            bd = new AccesoBd();

            sbQuery.append("INSERT INTO USUARIOS (NOMBRE, APELLIDO1, ");
            sbQuery.append("APELLIDO2, DNI, CODIGO_ROL, ");
            sbQuery.append("FECHA_ALTA ");
            sbQuery.append("values (?, ?, ?, ?, ?, ?, sysdate)");

            parametros.put(Integer.toString(cont++),
                Misc.nz(formulario.getNombre()).toUpperCase());
            parametros.put(Integer.toString(cont++),
                Misc.nz(formulario.getApellido1()).toUpperCase());
            parametros.put(Integer.toString(cont++),
                Misc.nz(formulario.getApellido2()).toUpperCase());
            parametros.put(Integer.toString(cont++),
                Misc.nz(formulario.getDni()).toUpperCase());
            parametros.put(Integer.toString(cont++),
                Misc.nz(formulario.getCodRol()));

            bd.actualizaConCommit(sbQuery.toString(), parametros);

            objSalida.setError(false);
        }
        catch(Exception e){
            objSalida.setError(true);
        }
        finally{
            bd.cerrar();
        }

        return objSalida;
    }
}
```

En el método en primer lugar se crea la conexión a la base de datos a través de AccesoBD(), que es una clase interna que encapsula la creación de la conexión a la base de datos así como las operaciones de inserción, consulta y eliminación.

A continuación se crea la sentencia SQL INSERT que será con la que se realice la inserción. Se le indicarán con interrogantes “?” los parámetros que posteriormente se pasarán utilizando un objeto HashMap.

Se rellenan el HashMap de parámetros con los valores que relleno el usuario en el formulario y se realiza la inserción con el método actualizaConCommit().

5.1.2 Servicio Web

Para desarrollar el servicio se ha escogido Axis2 como herramienta para generarlo.

Axis2 es un motor de servicios web que, en resumen, es una aplicación web en la que se despliegan los servicios web. Los servicios en sí son clases java que implementan las acciones a realizar empaquetadas en un formato especial. Este formato es un jar que incluye unos archivos XML de configuración que se detallarán más adelante.

Ese archivo empaquetado, como decíamos, se despliega en axis2. Y a través de la consola de administración se obtiene la URL a la que se deberá invocar para acceder al servicio web, así que como el WSDL del servicio. El WSDL es un documento XML que describe el servicio web (operaciones que admite y parámetros esperados).

Con este WSDL cualquier agente interesado debe ser capaz de generar un cliente que ataque a dicho servicio.

Los servicios web que se desarrollan para nuestro caso recibirán mensajes XML con la información de los nuevos asuntos, las peticiones de consulta, eliminación...Estos XML están definidos mediante esquemas XSD. Estos esquemas definen como serán los mensajes XML, que elementos pueden contener, que cardinalidades pueden tener dichos elementos...

Para ayudarnos a manipular los XML que lleguen al servicio web se utilizará la herramienta XMLBeans.

XMLBeans es un framework que permite transformar XML en objetos Java, más fácilmente manipulables. Así mismo permite realizar la operación inversa, transformando objetos Java en XML. Para generar los objetos Java que manipularán los XML, partimos de los XSD que hemos definido previamente.

Definición de mensajes de intercambio

Las operaciones que se pueden realizar sobre el servicio web son: dar de alta un asunto o estado, eliminar un asunto o estado, consultar un asunto o estado.

Por ello son necesarios cinco tipos de mensajes:

- **PC001:** mensaje de alta de un asunto o estado.
- **PC002:** mensaje de eliminación de un asunto o estado.
- **PC003:** mensaje de confirmación de la operación realizada.
- **PC004:** mensaje de consulta de un asunto o estado.
- **PC005:** mensaje de respuesta a una consulta de un asunto o estado.

A continuación se detalla a modo de ejemplo el esquema XSD de definición del mensaje PC001. Utilizaremos una representación gráfica en lugar de el XML de definición por ser más comprensible:

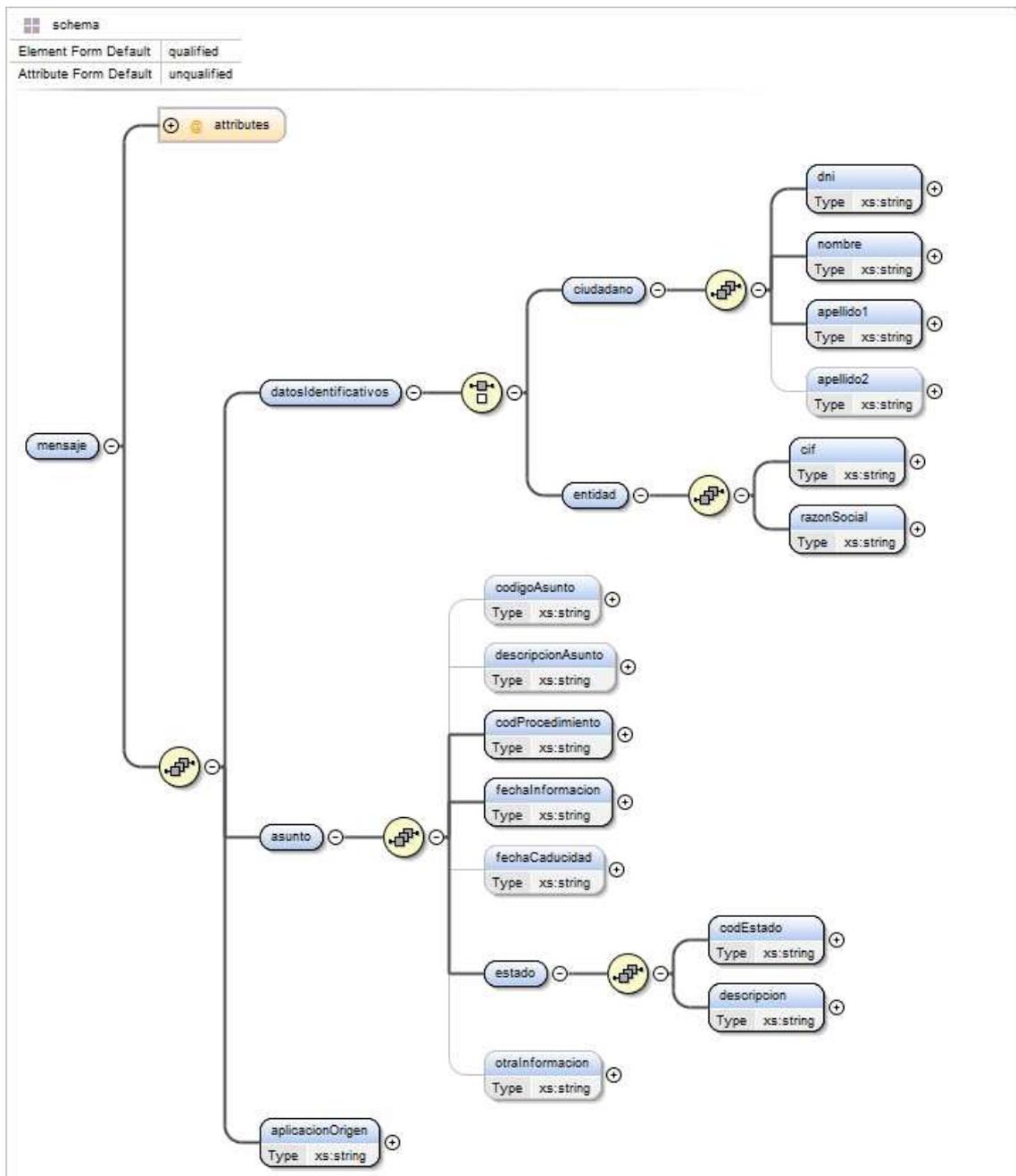


Figura 32. Esquema del mensaje PC001

Esta definición nos permite generar los XML de alta de asuntos o estados.

Por hacer algunas apreciaciones para facilitar la comprensión del esquema, el mensaje se compone de una secuencia de datos identificativos, asunto y aplicación origen.

Lo datos identificativos pueden ser o bien de un ciudadano o una entidad, pero es un elección excluyente. En el caso de que sea un ciudadano se compone de DNI, nombre, apellido 1 y , opcionalmente, apellido 2.

El resto de elementos del esquema siguen una filosofía similar. A continuación se muestra un mensaje XML de alta de un asunto, que cumpliría el esquema que hemos definido:

```
<mensaje id="PC001" key="1264682624226" fecha="2013-06-28-13-43-44">
  <datosIdentificativos>
    <ciudadano>
      <dni>47040233N</dni>
      <nombre>JULIO FRANCISCO</nombre>
      <apellido1>VICENTE</apellido1>
      <apellido2>SÁNCHEZ</apellido2>
    </ciudadano>
  </datosIdentificativos>
  <asunto>
    <codigoAsunto>2010000006</codigoAsunto>
    <codProcedimiento>998237</codProcedimiento>
    <fechaInformacion>28/06/2013</fechaInformacion>
    <estado>
      <codEstado>1</codEstado>
      <descripcion>Recibido</descripcion>
    </estado>
    <otraInformacion>tempCiudadanos4552558203504556136</otraInformacion>
  </asunto>
  <aplicacionOrigen>AplicacionPruebas</aplicacionOrigen>
</mensaje>
```

Es importante señalar que cada mensaje viene con el identificador del tipo de operación que realiza. Este id nos permitirá posteriormente llamar a lógica correspondiente en cada caso.

Transformación del XML en objetos Java con XMLBeans

A partir del esquema de definición del mensaje, utilizando XMLBeans se pueden generar objetos Java que nos permitan trabajar con un XML.

Se generará una librería que permite transformar un XML a los objetos Java que genera; transformar los objetos Java en un XML; recorrer y modificar los nodos del XML a través de los métodos que se generan.

Un extracto de parte de los métodos generados es el siguiente:



Figura 33. Métodos generados con XMLBeans

Generación del Servicio Web

Para generar el servicio web, como vimos anteriormente tenemos que implementar una clase que realice la funcionalidad que queremos.

En nuestro caso, queremos crear un método que de recepción del mensaje XML entrante, llame a la lógica de negocio invocando a un EJB que veremos posteriormente, y devuelva el mensaje de respuesta.

Esta clase luego se empaquetará en un archivo en el que se incluirán unos archivos de configuración XML y se desplegarán en Axis2.

Empezaremos por la clase que nos dará la funcionalidad que queremos:

```
public class WSRecepcion {

    private static final String URI_WS_FIRMA = "WSRecepcionFirma";
    private GestionAsuntosHome home;
    private InicializacionProxyHome inicializacionProxyHome;
    private String PATH_ADJUNTO = "";

    public WSRecepcion() throws Exception{
        try{
            Context jndiContext = new InitialContext();
            Object ref = jndiContext.lookup("ciudadano.ejb.GestionAsuntos");
            home = (GestionAsuntosHome) PortableRemoteObject.narrow(ref,
                GestionAsuntosHome.class);
            ref = jndiContext.lookup("ciudadano.ejb.InicializacionProxy");
            inicializacionProxyHome = (InicializacionProxyHome)
                PortableRemoteObject.narrow(ref, InicializacionProxyHome.class);
        }
        catch(Exception e){
            e.printStackTrace();
            throw e;
        }
    }

    public String recepcion(String xmlEntrada) throws Exception {
        String ubicacionLog = "WSRecepcion.recepcion()";
        GestionAsuntos gestionAsuntos = null;
        InicializacionProxy inicializacionProxy = null;
        try {
            inicializacionProxy = inicializacionProxyHome.create();
            PATH_ADJUNTO =
                Misc.nz(inicializacionProxy.getVariable("PATH_REPOSITORIO")) +
                Misc.nz(inicializacionProxy.getVariable("PATH_ADJUNTO"));
            MessageContext msgContext = MessageContext.getCurrentMessageContext();
            String uriDeEntrada = ((HttpServletRequest)
                msgContext.getProperty(Constants.HTTP_SERVLET_REQUEST)).getRequestURI();
            WSSeguridadBean wSSeguridad = null;
            if(uriDeEntrada.indexOf(URI_WS_FIRMA) != -1){
                WSHelper wSHelper = new WSHelper();
                wSSeguridad = wSHelper.getFirmaCertificado(msgContext, logger);
            }
            else{
                wSSeguridad = new WSSeguridadBean();
            }
            gestionAsuntos = home.create();
            RespuestaAttachmentsBean respuestaBean =
                gestionAsuntos.gestionMensaje(xmlEntrada, wSSeguridad.getFirma(),
                wSSeguridad.getDnAgente(), wSSeguridad.getCertificadoCliente());
            String xmlRespuesta = respuestaBean.getXmlRespuesta();
            String[] idAttachment = respuestaBean.getIdAttachement();
            if(idAttachment!=null && idAttachment.length>0){
                agregarAttachment(PATH_ADJUNTO + idAttachment);
            }
            return xmlRespuesta;
        } catch (Exception e) {
            throw e;
        }
    }
}
```

En primer lugar vemos que hay un constructor WSRecepcion() cuya función es invoca al EJB que tiene la lógica de negocio del alta de un asunto.

A continuación está el método que realmente implementa el servicio recepcion() . Este método recibe como parámetro una cadena de texto que contiene el XML del mensaje que se recibe..

El resto de acciones que se hacen en el método son simples. Se comprueba si se está accediendo con o sin certificado para extraerlo en caso que sea necesario. Y luego se llama al método del EJB que realiza la gestión del mensaje. Cuando obtiene la respuesta, adjunta si es necesario el documento adjunto al mensaje y devuelve el XML de respuesta.

Este método es válido para recibir cualquiera de los mensajes que están definidos. El qué hacer con cada uno de los mensajes, de ello se encarga el EJB que veremos posteriormente.

Esta clase que empaqueta en un jar con la extensión .aar y se le incluye el fichero de configuración service.xml

```
<service name="WSRecepcion">
  <description>Gestion mensajes</description>
  <operation name="recepcion">
    <messageReceiver
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
  </operation>
  <parameter name="ServiceClass"
locked="false">ciudadano.ws.WSRecepcion</parameter>
  <parameter name="enableMTOM">false</parameter>
  <parameter name="enableSwA" locked="false">true</parameter>
  <parameter name="cacheAttachments" locked="false">true</parameter>
  <parameter name="attachmentDIR"
locked="false">/repositorio/CIUDADANOS/ficheros/tmp/</parameter>
  <parameter name="sizeThreshold" locked="false">10000</parameter>
</service>
```

Los valores importantes de este fichero configuración son, el nombre del servicio WSRecepcion que es como se llamará el servicio. Las operaciones que se implementan que se define en el nodo operation. La clase que implementa dicha operación que se define en el parámetro ServiceClass.

A parte se configura el parámetro enableSwA (Soap with Attachements), que sirve para optimizar la transmisión de ficheros adjuntos grandes si sobrecargar el servidor ya que los va cacheando en el directorio configurado en el parámetro attachmentDIR.

En caso de que se utilizase el servicio web que admite la firma con certificado digital habría que incluir en este mismo archivo las políticas de seguridad que cumple el servicio web. Estas políticas de seguridad se implementan con WS-Security, que es un protocolo de seguridad estándar para comunicaciones con servicios web. En nuestro caso se definen las siguientes políticas de seguridad:

```

<wsp:Policy wsu:Id="SigOnly" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsp:ExactlyOne>
    <wsp>All>
      <sp:AsymmetricBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <wsp:Policy>
          <sp:InitiatorToken>
            <wsp:Policy>
              <sp:X509Token
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient">
                <wsp:Policy>
                  <sp:RequireThumbprintReference/>
                  <sp:WssX509V3Token10/>
                </wsp:Policy>
              </sp:X509Token>
            </wsp:Policy>
          </sp:InitiatorToken>
          <sp:RecipientToken>
            <wsp:Policy>
              <sp:X509Token sp:IncludeToken="
http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/Always">
                <wsp:Policy>
                  <sp:RequireThumbprintReference/>
                  <sp:WssX509V3Token10/>
                </wsp:Policy>
              </sp:X509Token>
            </wsp:Policy>
          </sp:RecipientToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:TripleDesRsa15/>
            </wsp:Policy>
          </sp:AlgorithmSuite>
          <sp:Layout>
            <wsp:Policy>
              <sp:Strict/>
            </wsp:Policy>
          </sp:Layout>
          <sp:IncludeTimestamp/>
          <sp:OnlySignEntireHeadersAndBody/>
        </wsp:Policy>
      </sp:AsymmetricBinding>
      <sp:Wss10 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <wsp:Policy>
          <sp:MustSupportRefKeyIdentifier/>
          <sp:MustSupportRefIssuerSerial/>
        </wsp:Policy>
      </sp:Wss10>
    </sp:SignedParts
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
      <sp:Body/>
    </sp:SignedParts>
  </wsp>All>
</wsp:ExactlyOne>
</wsp:Policy>

```

En esa política se define que se requiere SigOnly, es decir, sólo firmado. Podría ser otro tipo de política y exigir también cifrado, pero en nuestro caso a ser un servicio que sólo estará accesible por la intranet, con la firma bastaría.

También se define con que se firmará que sería con certificados que cumplan el estándar x509, que el algoritmo de cifrado es TripleDesRsa15.

Una vez empaquetado el archivo y desplegado en Axis2, podríamos acceder a través de la consola de administración al WSDL que define el servicio web. Con este WSDL y los XSD de definición de los mensajes, se podría crear un cliente que utilizase este servicio web. El WSDL generado sería:

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://ws.ciudadano"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://ws.ciudadano">
  <wsdl:documentation>WSRecepcion</wsdl:documentation>
  <wsdl:types>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://ws.ciudadano">
      <xs:complexType name="Exception">
        <xs:sequence>
          <xs:element minOccurs="0" name="Exception" nillable="true"
type="xs:anyType"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="Exception">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="Exception" nillable="true"
type="ns:Exception"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="recepcion">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="xmlEntrada"
nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="recepcionResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  ...
</wsdl:definitions>
```

EJB con la lógica de negocio

Enterprise Java Beans (EJB) es un API de J2EE para la construcción de aplicaciones. Los EJB permiten centrarse en la lógica de negocio en sí, dejando de lado los problemas de concurrencia, persistencia, seguridad... Se compone de las clases que implementan la lógica de negocio y de unas interfaces que exponen los métodos para que puedan ser utilizados por los clientes.

El EJB sería invocado como vimos anteriormente desde el servicio web, y sería aquí donde se implementaría la lógica para crear, consultar y eliminar asuntos y estados.

```
public RespuestaAttachmentsBean gestionMensaje(String xmlEntrada, String
nombreFichero, String nombreAdjunto, String firma, String dnAgente, byte[]
certificadoCliente) {

    RespuestaAttachmentsBean respuestaBean = new RespuestaAttachmentsBean();
    Salida objSalida = new Salida();
    GestionMensajesHelper gestionMensajes = new GestionMensajesHelper();
    try{
        String wsdl = Inicializacion.URL_WEBSERVICES;
        String idApp = Inicializacion.IDAPP_WEBSERVICES;
        boolean mismoDN;
        HashMap hError = new HashMap();
        String nombreVirus = null;

        if(!Misc.nz(certificadoCliente).equals("")) {
            cap.certs.webservices.generated.CertificadoIdApp result = null;
            cap.certs.cliente.WebServices webServices = new
            cap.certs.cliente.WebServices(wsdl);
            result = webServices.extraerCertificado(certificadoCliente, true,
idApp);
            comprobarDN(dnAgente,gestionMensajes.getIdAplicacion(xmlEntrada));
        }
        if(!Misc.nz(nombreFichero).equals("")) {
            checkVirus(nombreFichero);
        }
        respuestaBean = gestionMensajes.realizarOperacion(xmlEntrada,
nombreFichero, nombreAdjunto, firma, dnAgente);
    }
    catch(Exception e){
        HashMap hError = new HashMap();
        hError.put("codError",Constantes.COD_ERROR_GENERICO);
        try {
            respuestaBean =
            gestionMensajes.generarMensajeError(xmlEntrada, hError,
            firma, nombreFichero);
        } catch (Exception ex) {
            e.printStackTrace();
        }
    }
}
```

Este es el método que habíamos invocado desde el servicio web. Este método recibe el XML como una cadena de texto a demás de otra serie de parámetros como el fichero adjunto (si es que había alguno) y el certificado de firma (si es que hay).

En caso de ser necesario comprueba que el certificado es válido y comprueba que no existan virus. Esto lo hace invocando a otros servicios web que están disponibles para estas tareas y que no se detallan por estar fuera del ámbito del este proyecto. Para nuestro caso es como si fuesen funciones nativas que nos dan dicha funcionalidad.

En caso de no haber error se invoca a la clase que gestiona el alta del asunto. En caso de haber algún error se genera un mensaje XML de respuesta (PC003) indicando del error.

A continuación se muestra como se trataría el caso en el que no hay errores. El método invocado sería:

```
public RespuestaAttachmentsBean realizarOperacion (String xmlEntrada, String
nombreFichero, String nombreAdjunto, String firma, String dnAgente) throws
Exception {

    RespuestaAttachmentsBean respuestaAttachments = new
    RespuestaAttachmentsBean();
    String mensajeRespuesta = "";
    String idMensaje = getIdMensaje(xmlEntrada);
    String codEntrada = registrarXmlEntrada(idMensaje, xmlEntrada, firma);

    if(idMensaje.equals(Constants.ID_PC001)){
        GestionMensajePC001 objGestionMensaje = new GestionMensajePC001();
        mensajeRespuesta = objGestionMensaje.tramitar(xmlEntrada,
        nombreFichero, nombreAdjunto);
        respuestaAttachments.setXmlRespuesta(mensajeRespuesta);
    }
    else if(idMensaje.equals(Constants.ID_PC002)){
        ...
    }

    return respuestaAttachments;
}
```

Extraería del mensaje el identificador del tipo de operación para poder luego saber como tratarlo, lo registraría en BD para tener una auditoría de accesos al servicio web y posteriormente, en función del identificador, llamaría al método que lo trataría. En nuestro caso, al ser un alta de un asunto (PC001) llamaría a ese método.

```

public String tramitar(String xmlEntrada, String nombreFichero, String
nombreAdjunto) throws Exception {
    String xmlRespuesta = "";

    java.io.File[] schemaPath = { new java.io.File("PC001.jar") } ;
    org.apache.xmlbeans.SchemaTypeLoader stl =
    org.apache.xmlbeans.XmlBeans.typeLoaderForResource (
    org.apache.xmlbeans.XmlBeans.resourceLoaderForPath(schemaPath)) ;

    stl = org.apache.xmlbeans.XmlBeans.typeLoaderUnion(new
    org.apache.xmlbeans.SchemaTypeLoader [] {
    ciudadano.schemas.PC001.MensajeDocument.type.getTypeSystem(),
    org.apache.xmlbeans.XmlBeans.getContextTypeLoader()});
    ciudadano.schemas.PC001.MensajeDocument mensajeDoc =
    (ciudadano.schemas.PC001.MensajeDocument )stl.parse ( xmlEntrada ,
    ciudadano.schemas.PC001.MensajeDocument.type, null) ;

    XmlOptions m_validationOptions;
    ArrayList validationErrors = new ArrayList();
    m_validationOptions = new XmlOptions();
    m_validationOptions.setErrorListener(validationErrors);

    boolean esValido = mensajeDoc.validate(m_validationOptions);
    if (esValido){
        DatosMensajeBean objDatosMensaje = obtenerDatosMensaje(mensajeDoc);
        DatosAsuntoBean objDatosAsunto = obtenerDatosAsunto(mensajeDoc);
        comprobarDatosAsunto(objDatosAsunto);
        AltaAsunto objAltaAsunto = new AltaAsunto();
        objSalida = objAltaAsunto.tramitar(objDatosAsunto, nombreFichero,
        nombreAdjunto);
        ciudadano.schemas.PC003.MensajeDocument objPC003 =
        generarMensajeRespuesta(objDatosMensaje, objDatosAsunto);
        xmlRespuesta = objPC003.toString();
    } else {
        hError = (HashMap)objSalida.getValor();
        ciudadano.schemas.PC003.MensajeDocument objPC003 =
        generarMensajeError(objDatosMensaje, objDatosAsunto, hError,
        nombreFichero);
        xmlRespuesta = objPC003.toString();
    }
}

```

En este método se utiliza en primer lugar la librería generada por XMLBeans a partir del esquema de definición del PC001 para transformar la cadena de texto que contiene el mensaje en objetos Java y así poder manipularlos.

A continuación se valida que el mensaje cumple el esquema, sino en caso contrario se generaría un mensaje XML indicando del error.

Si es válido, se recuperan los datos del asunto y se invoca al DAO que realiza la inserción en BD. Este DAO sería similar al que se vio en el caso de las aplicaciones web.

5.2 Plan de pruebas del Sistema de Información

A continuación se detalla el plan de pruebas necesario que constará de pruebas:

- **Unitarias**, de cada uno de los componentes del sistema, para verificar la funcionalidad necesaria y la correcta estructura de cada uno de ellos.
- **De Integración**, para verificar la interacción correcta entre cada uno de los componentes del sistema.
- **De sistema**, garantizando la integración del sistema con el resto de subsistemas o sistemas con lo que se comunica.
- **De aceptación**, validando el cumplimiento de los requisitos de funcionamiento esperado.

Para mostrar la información, se utilizará la siguiente estructura:

- **ID**: Identificador único del tipo de prueba.
- **Nombre**: Nombre descriptivo de la prueba.
- **Objetivo**: Indica que se quiere comprobar.
- **Pruebas**: Descripción detallada de las pruebas que se realizan.

5.2.1 Pruebas de Sistema

PSIS.0001	Servidor Web	
Objetivo	Verificar la correcta instalación del servidor Web	
PRUEBAS		
#	Descripción	Resultado esperado
1	Petición de una página de prueba por HTTPS.	El servidor contesta con la página seleccionada.

PSIS.0002	Servidor de aplicaciones	
Objetivo	Verificar la correcta instalación del servidor de aplicaciones	
PRUEBAS		
#	Descripción	Resultado esperado
1	Petición de una página de prueba por HTTPS.	El servidor contesta con la página solicitada.
2	Petición de una URL mapeada a un Servlet (Action con estructura de STRUTS)	Se ejecuta correctamente el Servlet correspondiente

PSIS.0003	Servidor de bases de datos	
Objetivo	Verificar la correcta instalación del servidor de bases de datos	
PRUEBAS		
#	Descripción	Resultado esperado
1	Conexión con el servidor desde puesto cliente	Hay conexión
2	Creación de una tabla.	Tabla creada.
3	Inserción de registros en tabla de prueba	Registros insertados
4	Selección de registros	Se recuperan los registros deseados.

5.2.2 Pruebas de Integración

PINT.0001	Integración Servidor Web – Servidor de Aplicaciones	
Objetivo	Verificar la correcta configuración de la integración entre el Servidor iPlanet y WebLogic	
PRUEBAS		
#	Descripción	Resultado esperado
1	Petición de una página de prueba por http del servidor de aplicaciones a través del iPlanet.	El servidor contesta con la página seleccionada

PINT.0002	Integración Servidor de Aplicaciones – Servidor Base de Datos	
Objetivo	Verificar la correcta configuración del pool de conexiones entre el servidor de aplicaciones y el servidor de base de datos.	
PRUEBAS		
#	Descripción	Resultado esperado
1	Petición de un Action que realice una conexión a la base de datos.	El Action se conecta correctamente.

5.2.3 Pruebas Unitarias y de aceptación

A continuación se detallan el conjunto de pruebas unitarias y de aceptación que se efectuarán en el sistema.

5.2.3.1 Servicio web

PUNI.0001	Acceso	
Objetivo	Verificar que una aplicación tiene que estar registrada en el sistema para poder realizar peticiones (alta, modificación, borrado, consulta).	
PRUEBAS		
#	Descripción	Resultado esperado
1	Se envía una petición con un identificador de aplicación que no esté registrado.	Se responde con un mensaje de error informando.
2	Se envía una petición con un identificador de aplicación válido.	El sistema responde con un mensaje correcto.
3	Se envía una petición por cada tipo de mensaje.	El sistema responde con el mensaje correcto para cada tipo de mensaje enviado. Se comprueba en BD que el alta y el borrado quedan registradas correctamente.

PUNI.0002	Validación formato de los mensajes	
Objetivo	Verificar la correcta validación del formato de los XML.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se envía una petición con por cada tipo de mensaje con algún error en el formato.	Se responde con un mensaje de error.

PUNI.0003	Verificación permisos de consulta	
Objetivo	Comprobar que una aplicación sólo puede acceder a otros asuntos de otras aplicaciones si es privilegiada.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se envía una petición de consulta para un asunto de otra aplicación siendo una aplicación no privilegiada.	Muestra un mensaje de error de permisos.
2	Se envía una petición de consulta para un asunto de otra aplicación siendo una aplicación privilegiada.	Devuelve un mensaje con la información del asunto.

PUNI.0004	Verificación firma de mensajes	
Objetivo	Comprobar el correcto funcionamiento de la firma de mensajes.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se envía una petición sin firmar al servicio que pide firma	Devuelve un mensaje de error.
2	Se envía una petición firmada al servicio que pide firma, siendo el certificado de firma distinto al que tiene registrado.	Devuelve un mensaje de error

3	Se envía una petición firmada al servicio que pide firma, siendo el certificado de firma el mismo que tiene registrado.	Devuelve el mensaje correcto correspondiente a la petición enviada.
4	Se envía una petición firmada al servicio que no pide firma.	Devuelve un mensaje de error.

5.2.3.2 Aplicación gestión

PUNI.0005	Acceso	
Objetivo	Verificar que un usuario tiene que estar registrado en la aplicación para poder acceder.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Intentar acceder a la aplicación sin estar registrado.	Muestra mensaje indicando que el usuario no tiene permisos para acceder.
2	Intentar acceder a la aplicación una vez registrada en la aplicación.	Se accede correctamente al menú principal de la aplicación

PUNI.0006	Verificar menú	
Objetivo	Verificar el correcto funcionamiento de los links que aparecen en el menú principal.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Pinchar en el link "Nuevo Asunto"	Muestra el formulario para dar de alta un nuevo asunto
2	Pinchar en el link "Eliminar asunto"	Muestra el formulario para eliminar un asunto.
3	Pinchar en el link "Consultas"	Muestra el formulario de consultas.

PUNI.0007	Nuevo asunto	
Objetivo	Verificar el correcto funcionamiento del alta de nuevos asuntos.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se rellena la información para dar de alta un asunto. Se realizarán dos altas, uno para ciudadanos y otro para entidades jurídicas.	Se muestra un mensaje informativo indicando que el alta ha funcionado correctamente. En BD se habrá registrado el alta.
2	Se rellena la información sin rellenar algún campo obligatorio.	El sistema muestra un mensaje indicando que deben rellenarse los campos obligatorios.

PUNI.0008	Eliminar asunto	
Objetivo	Verificar el correcto funcionamiento de la eliminación de asuntos.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se rellena la información para borrar un asunto	Se muestra un mensaje informativo indicando que el borrado ha funcionado correctamente. En BD se habrá registrado el borrado lógico de los asuntos coincidentes con los criterios seleccionados.
2	Se rellena la información sin rellenar algún campo obligatorio.	El sistema muestra un mensaje indicando que deben rellenarse los campos obligatorios.

PUNI.0009	Consultar asunto	
Objetivo	Verificar el correcto funcionamiento de la consulta de asuntos.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se rellena la información para consultar asuntos.	Se muestra un listado con los asuntos coincidentes

PUNI.0010	Comprobar acciones del listado	
Objetivo	Verificar el correcto funcionamiento del alta, borrado y consulta desde el listado de una consulta.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Pinchar en el botón de consulta de un asunto de los que aparecen en el listado.	Se muestra un formulario con la información de ese asunto.
2	Pinchar en el botón de añadir un estado al asunto	Se muestra un formulario con la información de ese asunto. Además se muestran en blanco los campos referentes al nuevo estado que se va a dar de alta. Si se repite la consulta con los mismos criterios deberá aparecer un nuevo estado para ese asunto
3	Pinchar en el botón de eliminar un asunto.	Se muestra un mensaje informando de que se va a eliminar el asunto. Si se acepta, se mostrará un mensaje de que la eliminación ha sido realizada. Se comprueba en la BD que se ha realizado el borrado Si se repite la consulta con los mismos criterios, el asunto eliminado no aparecerá.

5.2.3.3 Portal de información.

PUNI.0011	Acceso	
Objetivo	Verificar que un ciudadano debe acceder al portal con un certificado válido.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se accede al portal sin un certificado.	Se muestra un mensaje de error.
2	Se accede al portal con un certificado no válido (revocado o caducado)	Se muestra un mensaje de error.
3	Se accede al portal con un certificado válido	Se muestra la pantalla principal. Si el ciudadano (o entidad) tiene asuntos asociados, se muestra un listado.

PUNI.0012	Verificar menú	
Objetivo	Verificar que el funcionamiento de los links del menú principal.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se pincha sucesivamente en cada uno de los enlaces que filtran el listado en función del área al que pertenece el asunto.	Se muestra un listado filtrando por el área seleccionada.

PUNI.0013	Verificar consulta	
Objetivo	Verificar la correcta visualización de los datos de un asunto.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se pincha en uno de los asuntos para ver su información.	Se muestra la información relativa al asunto.
2	En la pantalla de consulta de asunto en particular, aparece un link para visualizar el histórico de estados de ese asunto. Se pincha en el enlace	Se muestra la información relativa a todos los estados por los que ha pasado el asunto.
3	Se pincha en el enlace del documento adjunto del asunto.	Se visualiza el documento asociado.

5.2.3.4 Aplicación de administración

PUNI.0014	Acceso	
Objetivo	Verificar que un usuario tiene que estar registrado en la aplicación para poder acceder	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Intentar acceder a la aplicación sin estar registrado.	Muestra mensaje indicando que el usuario no tiene permisos para acceder.

2	Intentar acceder a la aplicación una vez registrada en la aplicación.	Se accede correctamente al menú principal de la aplicación.
---	---	---

PUNI.0015	Verificar menú	
Objetivo	Verificar el correcto funcionamiento de los links que aparecen en el menú principal.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Pinchar en el link "Gestión de Usuario"	Muestra el formulario para consultar o dar de alta usuarios.
2	Pinchar en el link "Gestión de aplicaciones"	Muestra el formulario para consultar o dar de alta aplicaciones.
3	Pinchar en el link "Gestión de tablas"	Muestra un menú para acceder al mantenimiento de tres tablas: estados, permisos y roles
4	Pinchar en el link "Estadísticas"	Muestra un formulario para obtener estadísticas por varios criterios.

PUNI.0016	Gestión usuarios	
Objetivo	Verificar el correcto funcionamiento de la gestión de usuarios.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se rellena el formulario de la página de gestión de usuarios y se pincha al botón de consulta.	Se muestra un listado con los usuarios coincidentes
2	Se rellena el formulario de la página de gestión de usuarios y se pincha al botón de nuevo usuario.	Se da de alta un nuevo usuario.
3	Se pincha en el botón de modificar del listado de una consulta.	Se accede al formulario de modificación

4	Una vez realizado PUNI.0016-3 se modifican los valores de los campos del formulario y se pulsa el botón de grabar.	Se modifica la información del usuario.
5	Se pincha en el botón de eliminar del listado de una consulta	Se elimina el usuario. Si se vuelve a realizar una consulta por los mismos criterios y no aparece el usuario.

PUNI.0017	Gestión aplicaciones	
Objetivo	Verificar el correcto funcionamiento de la gestión de aplicaciones.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se rellena el formulario de la página de gestión de aplicaciones y se pincha al botón de consulta.	Se muestra un listado con las aplicaciones coincidentes.
2	Se rellena el formulario de la página de gestión de aplicaciones y se pincha al botón de nueva aplicación.	Se da de alta una nueva aplicación.
3	Se pincha en el botón de modificar del listado de una consulta.	Se accede al formulario de modificación
4	Una vez realizado PRUNI.0017-3 se modifican los valores de los campos del formulario y se pulsa el botón de grabar.	Se modifica la información de la aplicación.
5	Se pincha en el botón de eliminar del listado de una consulta	Se elimina la aplicación. Si se vuelve a realizar una consulta por los mismos criterios y no aparece la aplicación.

PUNI.0018	Gestión tablas maestras	
Objetivo	Verificar el correcto funcionamiento de la gestión de tablas maestras	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se pincha el link de estados.	Se muestra un listado con los estados y un pequeño formulario para dar de alta nuevos estados.
2	Una vez realizado el PUNI.0018-1 se rellena el formulario para dar de alta un nuevo estado	Se muestra un mensaje informando de que el alta se ha realizado. En el listado de estados aparecerá el nuevo estado.
3	Se pincha en el link de permisos.	Se muestra un listado con los permisos y un pequeño formulario para dar de alta nuevos permisos.
4	Una vez realizado PUNI.0018-3 se rellena el formulario para dar de alta un nuevo permiso.	Se muestra un mensaje informando de que el alta se ha realizado. En el listado de permisos aparecerá el nuevo permiso.
5	Se pincha en el link de roles.	Se muestra un listado con los roles y un pequeño formulario para dar de alta nuevos roles.
6	Una vez realizado PUNI.0018-5 se rellena el formulario para dar de alta un nuevo rol.	Se muestra un mensaje informando de que el alta se ha realizado. En el listado de permisos aparecerá el nuevo permiso.

PUNI.0019	Estadísticas	
Objetivo	Verificar el correcto funcionamiento de las estadísticas.	
PRUEBAS DE SISTEMA		
#	Descripción	Resultado esperado
1	Se rellena el formulario de consulta.	Se muestra un listado con los accesos que cumplen los criterios seleccionados.

5.2.4 Matriz de trazabilidad de las pruebas

A continuación se muestra la matriz de trazabilidad de los requisitos de software con las pruebas unitarias para verificar que todos los requisitos están probados.

	PUNI-0001	PUNI-0002	PUNI-0003	PUNI-0004	PUNI-0005	PUNI-0006	PUNI-0007	PUNI-0008	PUNI-0009	PUNI-0010	PUNI-0011	PUNI-0012	PUNI-0013	PUNI-0014	PUNI-0015	PUNI-0016	PUNI-0017	PUNI-0018	PUNI-0019
IRQ-0001	X	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0002	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0003	X	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0004	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0005	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
IRQ-0006	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
IRQ-0007	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-
IRQ-0008	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
IRQ-0009	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
IRQ-0010	-	-	-	-	X	-	-	-	-	-	-	-	-	X	-	-	-	-	-
IRQ-0011	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
IRQ-0012	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-
IRQ-0013	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
IRQ-0014	-	-	-	-	-	X	X	X	X	-	-	-	-	-	-	-	-	-	-

Todos los requisitos de software están probados en al menos una prueba unitaria.

Capítulo 6

Planificación y presupuesto

En este capítulo se detallará la planificación seguida durante el proyecto y se realizará una estimación del presupuesto del proyecto en función de dicha planificación.

6.1 Consideraciones iniciales

Para la planificación se ha tenido en cuenta que el equipo de proyecto estará formado por tres personas con los siguientes perfiles:

- Jefe de Proyecto
- Analista-programador
- Programador senior

Se ha estimado una jornada laboral de 8 horas diarias, 40 horas semanales y 20 días al mes. No se han incluido vacaciones ni festivos dentro de la planificación.

Para el cálculo de costes del proyecto, se ha estimado el precio por hora (sin incluir el IVA) que se factura por cada perfil se corresponde con:

Perfil	Precio unitario
Jefe de Proyecto	60 €
Analista-Programador	33 €
Programador Senior	25 €

6.2 Planificación

El desglose de horas totales por cada fase del proyecto ha sido el siguiente:

Tarea	Nº de horas
Análisis	200
Diseño	240
Implementación	640
Pruebas	112
Total	1.192

A continuación se desglosan las tareas en un diagrama de GANTT:

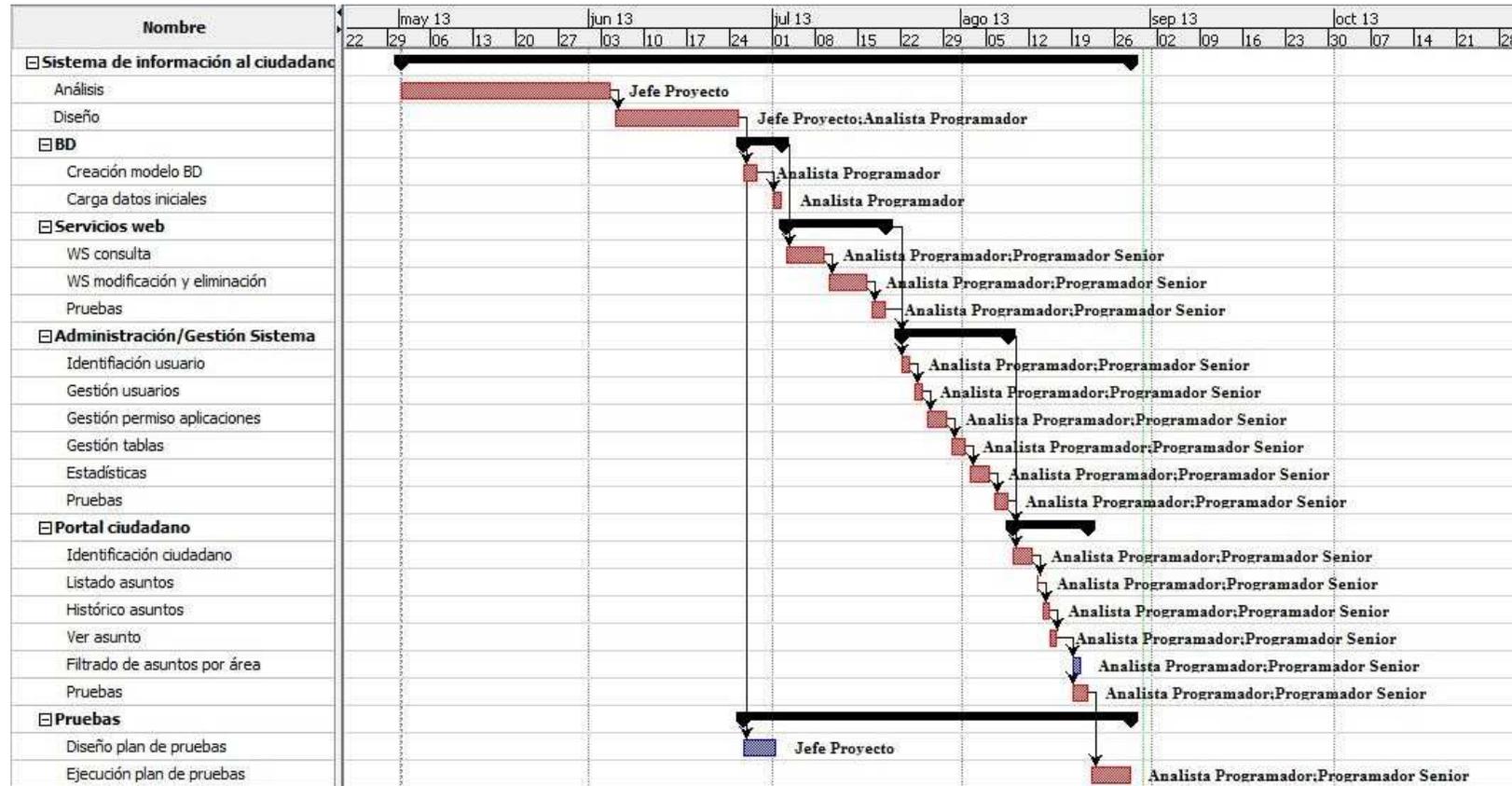


Figura 34. Diagrama de GANTT

NOTA: En rojo están marcadas las tareas críticas, por ser aquellas en las que de haber un retraso, implicarían un retraso en la fecha de entrega del proyecto. Si se aumentase el número de recursos del equipo podrían hacerse algunas tareas de forma concurrente y por ello se podría minimizar los riesgos al estar haber menos tareas críticas

6.3 Coste de personal

Para calcular el coste de personal hay que desglosar las tareas según el perfil. A continuación se presentan, por cada perfil, las tareas y horas dedicadas:

Jefe de Proyecto	Horas	Coste (€)
Diseño plan de pruebas	32	1.920
Diseño	120	7.200
Análisis	200	12.000
Total	352	21.120

Analista Programador	Horas	Coste (€)
Ejecución plan de pruebas	40	1.320
Filtrado de asuntos por área	12	396
Creación modelo BD	24	792
Gestión tablas	24	792
Gestión permiso aplicaciones	16	528
Ver asunto	12	396
Carga datos iniciales	16	528
Listado asuntos	8	264
Pruebas	24	792
Pruebas	24	792
Identificación ciudadano	16	528
Estadísticas	16	528
Gestión usuarios	16	528
WS consulta	40	1.320
Identificación usuario	16	528
Diseño	120	3.960
WS modificación y eliminación	40	1.320
Histórico asuntos	12	396
Pruebas	24	792
Total	500	16.500

Programador Senior	Horas	Coste (€)
WS modificación y eliminación	40	1.000
Filtrado de asuntos por área	12	300
Estadísticas	16	400
Gestión permiso aplicaciones	16	400
Listado asuntos	8	200
Pruebas	24	600
Gestión usuarios	16	400
Pruebas	24	600
Identificación ciudadano	16	400
Ejecución plan de pruebas	40	1.000
Ver asunto	12	300
Pruebas	24	600
Gestión tablas	24	600
WS consulta	40	1.000
Histórico asuntos	12	300
Identificación usuario	16	400
Total	340	8.500

El coste de personal total asciende a **46.120 euros**, IVA no incluido.

6.4 Coste hardware

Para el desarrollo el proyecto ha sido necesaria la adquisición de un portátil para el desarrollo de cada una de las fases del proyecto. El portátil adquirido es un Lenovo G580 I7 3520m 15.5" 8GB RAM 500GB Disco Duro, con un coste de 565 euros IVA no incluido.



Figura 35. Lenovo G580 I7 3520m

Hay que calcular en el coste imputable al proyecto, ya que no será la totalidad el precio del portátil. Para calcular el precio imputable se aplica la siguiente fórmula:

$$\left(\frac{A}{B}\right) * C * D$$

A: nº de meses desde la fecha de facturación en que el equipo es utilizado.

B: periodo de depreciación (60 meses).

C: coste del equipo (sin I.V.A).

D: % del uso que se dedica al proyecto.

Por tanto, aplicando la fórmula, el coste imputable del portátil sería de **133 euros IVA** no incluido.

6.5 Coste de software

La mayor parte del software utilizado ha sido software libre y por ello libre de costes. Aunque si ha sido necesario adquirir diversas licencias para el desarrollo del proyecto.

A continuación se detalla el coste de las licencias adquiridas.

Descripción	Coste (€)
Oracle Database Personal Edition	152,88
Oracle Weblogic Server Standard Edition	667,30
Total	820,18

6.6 Coste total del proyecto

Descripción	Coste (€)
Personal	46.120
Hardware	133
Software	820,18
Total sin IVA	47.073,18
Total con IVA (21%)	56.958,55

El coste del proyecto sin IVA es de 47.073,18 euros, y aplicando el IVA asciende a **56.958,55 euros**.

Capítulo 7

Conclusiones y acciones futuras

En este capítulo se detallarán las conclusiones extraídas de la realización de este proyecto y se estudiarán posibles acciones futuras que podrían ser de interés para mejorar el Sistema de Información.

7.1 Conclusiones del proyecto

El objetivo del proyecto era construir un Sistema de Información centralizado del Ciudadano, que permitiese a los ciudadanos consultar en un único punto el estado de todos los trámites que tiene abiertos con una Administración Pública.

Para conseguir este objetivo se desarrollo un sistema basado en la idea de una arquitectura SOA. De esta forma se pretendía que el desarrollo y la integración con el resto de sistemas de la administración fuese lo más sencillo y rápido posible.

Se han conseguido satisfactoriamente los objetivos marcados al comienzo del proyecto:

- Se ha desarrollado una portal web al que se conectará el ciudadano para acceder a la información sobre todos los trámites abiertos con la administración. Esta interfaz es común, independientemente de que internamente estos trámites sean gestionados por múltiples Sistemas de Información. Con ello se ha conseguido que el acceso por parte del ciudadano sea simple y únicamente en un punto.
- Se ha desarrollado una interfaz para el resto de Sistemas de Información de la administración, que permite comunicar todos los eventos que se produzcan en los distintos trámites que gestionan de los ciudadanos. Esta interfaz desarrollada es un Servicio Web con las operaciones de alta, eliminación y consulta.
- También se ha desarrollado una página web sencilla que permite que aquellos trámites que o bien no estén informatizados, o bien sea imposible integrar con el Servicio Web desarrollado, dar de alta los eventos relevantes que se producen en la gestión de un trámite administrativo de un ciudadano. Esta aplicación se conectará internamente al Servicio Web desarrollado, y así, de esta forma, la información será accesible por el ciudadano a través de su portal de acceso.

Esta aplicación también permite gestionar los permisos para que el resto de Sistemas de la organización puedan conectarse al Servicio Web. Además,

también permite gestionar los usuarios que pueden acceder a la aplicación web que sirve para dar de alta trámites que aún no están informatizados.

Por tanto se han cubierto sobradamente los objetivos marcados cuando se comenzó este proyecto.

7.2 Conclusiones personales

Desde un punto tecnológico el desarrollo de este proyecto me ha llevado a utilizar tecnologías con la que ya estaba familiarizado en mayor menor medida debido a que son comunes en los proyectos que afronto en mi actividad laboral.

Desde que acabé las asignaturas de la carrera hasta la realización de este proyecto han pasado algunos años en los que he estado trabajando en el sector de la consultoría informática.

Por ello el mayor reto que he afrontado con la realización de este proyecto ha sido el poder compatibilizarlo con mi actividad laboral. El número de horas que le podía dedicar no era siempre el que hubiese sido necesario y por ello he pasado por periodos en los que apenas he podido avanzar en él. Esto suponía un esfuerzo extra cada vez que lo retomaba porque exigía volver a realizar una “inmersión” para poder retomarlo en el punto en el que me había quedado.

A pesar de este gran inconveniente, poco a poco el proyecto fue avanzando y ya por fin se puede decir que es un punto y final a mi vida universitaria.

7.3 Acciones futuras

Una vez desarrollado este proyecto se han detectado algunas mejoras que podrían adoptarse para mejorar el Sistema. A continuación detallaremos dos posibles mejoras a realizar en el Sistema de Información. Una de ellas orientada a la implementación y otra orientada a la accesibilidad del Sistema.

Una mejora en la implementación podría ser la utilización de la tecnología JAXB, Java Architecture for XML Binding para transformar los XML en objetos Java.

En nuestro caso hemos utilizado XMLBeans, que como fin tiene el mismo objetivo que JAXB.

XMLBean es parte del proyecto Apache Software Foundation XML y se puede considerar que es un estándar de “facto” debido a su alto grado de utilización. En cambio, JAXB (que realiza la misma función), está integrada en la plataforma Java.

JAXB tiene como ventaja respecto a XMLBeans que consume menos recursos del servidor y ofrece mayor rendimiento. Además, al ser parte de Java no es necesario incluir recursos externos el archivo a desplegar en el servidor de aplicaciones y por tanto tendrá un menor tamaño.

XMLBeans tiene como ventaja que es más potente a la hora de tratar el XML, pero para las necesidades que tenemos JAXB podría ser una mejor opción ya que los XML que se intercambia son bastante pequeños y simples.

Otra mejora posible que podría abordarse podría ser en la línea de mejorar la accesibilidad del Sistema. Se estableció como requisito que el ciudadano accediese con un Certificado Digital. Se utiliza Certificado Digital por la seguridad que ofrece y además para evitar que el ciudadano tenga que registrarse previamente en el Sistema. Pero ello supone una barrera ya que todavía hoy en día el porcentaje de personas familiarizada con el uso de certificados digitales no es alto.

Una posible mejora sería permitir el acceso a través de la utilización de tokens (cadenas numéricas, alfanuméricas...) generados para cada asunto del ciudadano.

Cuando se diese de alta un asunto se generaría un token con el cuál el ciudadano podría acceder al sistema a consultar dicha información.

Para utilizar este mecanismo de autenticación hay que garantizar que para los tokens que se generan sea imposible predecir la secuencia, ya que si eso fuese posible sería fácilmente generable por un ataque de fuerza bruta por ejemplo. Eso puede hacerse generando números realmente aleatorios mediante la utilización de algoritmos existentes para tal efecto.

También, para mayor seguridad, podría combinarse con segundo sistema de autenticación (por ejemplo un apellido) para combinar el token con este segundo sistema de autenticación y reforzar la seguridad.

Bibliografía

- [1] Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos.
www.boe.es/diario_boe/txt.php?id=BOE-A-2007-12352
- [2] Stakeholder.
<http://es.wikipedia.org/wiki/Stakeholder>
- [3] SOA's strong medicine.
www.zdnet.com/blog/service-oriented/soas-strong-medicine/254
- [4] Customer Story: Starwood Hotels and Resorts Worldwide, Inc.
http://www.targetreachmarketing.com/TRM%20Files/Samples/SuccessStory_Systinet_Starwood.pdf
- [5] BIAN.
<https://bian.org>
- [6] Sistema de Historia Clínica Digital del Sistema Nacional de Salud (HCDSNS).
http://www.msssi.gob.es/organizacion/sns/planCalidadSNS/docs/HCDSNS_Castellano.pdf
- [7] Propuesta técnica HCDSNS.
<http://www.msssi.gob.es/organizacion/sns/planCalidadSNS/docs/PT.pdf>
- [8] Microsoft Windows.
http://es.wikipedia.org/wiki/Microsoft_Windows
- [9] Macintosh.
<http://es.wikipedia.org/wiki/Macintosh>
- [10] Unix.
<http://es.wikipedia.org/wiki/Unix>
- [11] Cuota de uso de los SO más usados.
<http://www.marlex.org/windows-gana-terreno-entre-sistemas-operativos-usados/15230/>
- [12] Entorno de desarrollo integrado.
http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado
- [13] Eclipse (software).
http://es.wikipedia.org/wiki/Eclipse_%28software%29
- [14] Java (lenguaje de programación).
http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29
- [15] Maven.
<http://es.wikipedia.org/wiki/Maven>
- [16] Subversion (software).
http://es.wikipedia.org/wiki/Subversion_%28software%29
- [17] ¿Qué es un framework de desarrollo web?.
<http://kokiing.blogspot.com.es/2010/12/que-es-un-framework-de-desarrollo-web.html>
- [18] Servicio web.
http://es.wikipedia.org/wiki/Servicio_web
- [19] Herramientas Java.
<http://www.consultoriajava.com/tools/axis.jsp>

- [20] Tema 1: Introducción a los servidores de aplicaciones.
<http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/sesion1-apuntes.htm>
- [21] Sistema de gestión de bases de datos.
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos

Glosario

API	Application Programming Interface
ASP	Active Server Pages
BIAN	Banking Industry Architecture Network
CIF	Código de Identificación Fiscal
DAO	Data Access Object
DNI	Documento Nacional de Identidad
E/R	Entidad Relación
EJB	Enterprise JavaBeans
Framework	Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IBM	International Business Machines
IDE	Integrated Development Environment
IVA	Impuesto de Valor Añadido
J2EE	Java 2 Platform, Enterprise Edition
JAXB	Java Architecture for XML Binding
JAX-WS	Java API for XML Web Services
JSP	Java Server Page
MVC	Modelo-Vista-Controlador
OASIS	Organization for the Advancement of Structured Information Standards
PHP	PHP Hypertext Pre-processor
Plugin	Componente software que añade capacidades específicas sobre una aplicación software de mayor envergadura
POM	Project Object Model
REST	Representational state transfer
SGBD	Sistema de Gestión de Bases de Datos
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SVN	Subversion
SwA	Soap with Attachements
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	EXtensible Markup Language
XSD	XML Schema Definition

Anexo I

Manual de Usuario del Subsistema de Administración

Pasos previos y requisitos de acceso

La entrada a este módulo de Administración también obliga a que el usuario se identifique; por ello, debemos disponer de un Certificado Digital o DNI electrónico aceptado por el Ministerio de Sanidad y Política Social, y estar registrados en el sistema con un rol de administración.

Certificados digitales

El Profesional que acceda a la aplicación deberá **poseer un Certificado Digital** válido que le identifique en el sistema.

Acceso al sistema

La URL de inicio tendrá el formato:

[https://<HOST>\[:<PUERTO>\]/portalCiudadano/admin/index.html](https://<HOST>[:<PUERTO>]/portalCiudadano/admin/index.html)

La página inicial de este módulo sería:



Figura 36. Pantalla de inicio

Como se puede observar en la imagen, este módulo dispone de una serie de opciones de administración, las cuales se encontrarán habilitadas o no dependiendo de los permisos del usuario que haya accedido.

Gestión de Usuarios

En esta funcionalidad podremos dar de alta usuarios, modificar los datos y/o rol al que pertenecen, consultar un listado de aquellos que cumplan unos criterios, etc. Cada uno de los roles tendrán asociados una serie de funcionalidades permitidas, y serán éstas las que podrá realizar el usuario según el rol que se le asigne.

Nada más acceder a esta funcionalidad, nos encontraremos con una pantalla como la siguiente:

Usuarios					
Nombre	Primer Apellido	Segundo Apellido	DNI	Rol	
JULIO FRANCISCO	VICENTE	SÁNCHEZ	47040233N	Administrador	 

Figura 37. Pantalla de consulta de usuarios

Si deseamos realizar el alta de un nuevo usuario, pulsaremos el botón *Nuevo usuario*, y accederemos a un formulario de entrada como el siguiente:

Añadir Borrar

Grabar Volver

Figura 38. Pantalla de alta de usuarios

Una vez rellenos todos los datos, pulsamos el botón *Grabar* para llevar a cabo el proceso de alta.

Si por el contrario, desde la pantalla inicial de gestión de usuario, deseamos obtener un listado de usuarios que cumplen algunos criterios, rellenaremos los campos de filtrado que deseemos y pulsaremos el botón *Consultar*. Un ejemplo de listado puede ser el siguiente:

Resultado de Búsqueda

Usuarios					
Nombre	Primer Apellido	Segundo Apellido	DNI	Rol	
Arturo	García	Trasobares	48843406m	Administrador	
Maria	Pérez	Muga	48843407Y	Usuario estadísticas	

Figura 39. Pantalla resultado de una consulta de usuarios

Si se deseara dar de baja a un usuario, sólo tendríamos que pulsar en el icono de la papelera; en cambio, si quisiéramos modificar algún dato de un usuario, pulsaremos en el icono del lápiz, dando lugar a una pantalla como la siguiente:

Como se puede observar, el formulario es similar al de entrada, pero en este caso estará relleno con datos. Una vez realizadas las modificaciones, pulsaremos el botón *Grabar* para llevar a cabo dicho proceso.

DNI: 00000000T
 Nombre: NOMBRE
 Primer Apellido: APELLIDOPRIMERO
 Segundo Apellido: APELLIDOSEGUNDO
 Rol: Administrador
 Aplicaciones: Seleccione
 NO_PRIVILEG_1
 PRIVILEGIADA
 Añadir Borrar
 Grabar Volver

Figura 39. Pantalla modificación de usuarios

Gestión de aplicaciones

En esta funcionalidad podremos dar de alta aplicaciones, modificar los datos y/o permisos a funcionalidades de gestión que pueden realizar, consultar un listado de aquellas que cumplan unos criterios, etc.

Nada más acceder a esta funcionalidad, nos encontraremos con una pantalla como la siguiente:



Figura 39. Pantalla de consulta de aplicaciones

Si deseamos realizar el alta de una nueva aplicación, pulsaremos el botón *Nueva aplicación*, y accederemos a un formulario de entrada como el siguiente:

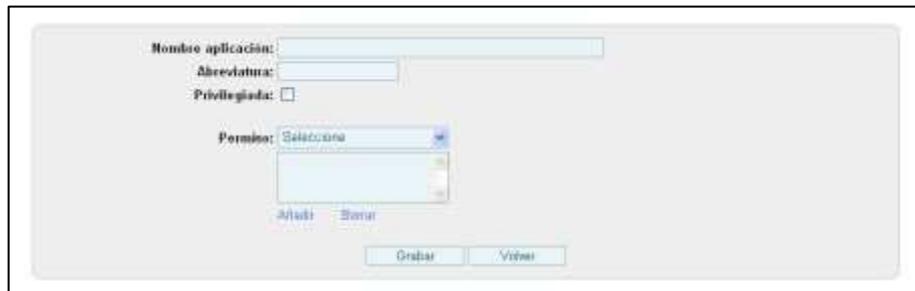


Figura 40. Pantalla de alta de aplicaciones

Una vez rellenos todos los datos, pulsamos el botón *Grabar* para llevar a cabo el proceso de alta.

Si por el contrario deseamos obtener un listado de aplicaciones que cumplen algunos criterios, rellenaremos los campos de filtrado que deseemos y pulsaremos el botón *Consultar*. Un ejemplo de listado puede ser el siguiente:



Figura 41. Pantalla resultado de una consulta de aplicaciones

Si se deseara dar de baja a una aplicación, sólo tendríamos que pulsar en el icono de la papelera; en cambio, si quisiéramos modificar algún dato de una aplicación, pulsaremos en el icono del lápiz, dando lugar a una pantalla como la siguiente:

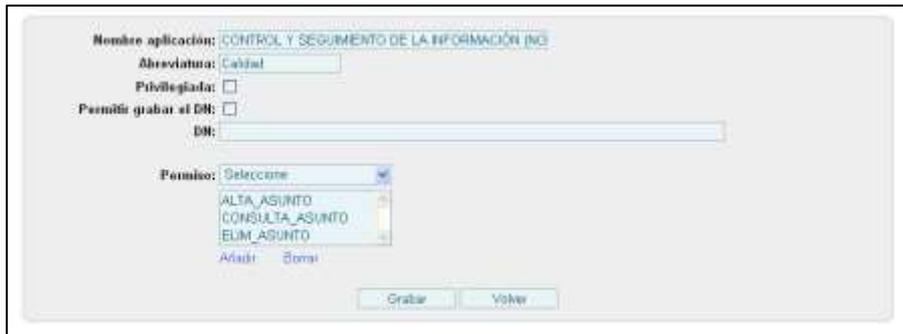


Figura 42. Pantalla de modificación de una aplicación

Como se puede observar, el formulario es similar al de entrada, pero en este caso estará relleno con datos. Una vez realizadas las modificaciones, pulsaremos el botón *Grabar* para llevar a cabo dicho proceso.

Gestión de tablas maestras

Además de las funcionalidades de gestión de usuarios y aplicaciones, existe una funcionalidad de administración de tablas auxiliares empleadas en la aplicación. Esta gestión se ha realizado de forma genérica para cualquier tabla que pudiera ser administrable, por lo que en este apartado sólo se explicará el proceso de principio a fin para una tabla.

En principio, al seleccionar del menú principal la opción de Gestión de Tablas, aparecerá un segundo submenú con el listado de tablas que pueden ser administradas.



Figura 43. Menú de administración de tablas maestras

Por ejemplo, si seleccionamos la administración de la tabla Áreas, nos aparecerá una pantalla como la siguiente, en la que se muestra un listado de registros dados de alta en dicha tabla y en la parte inferior un formulario para la entrada de un nuevo registro.

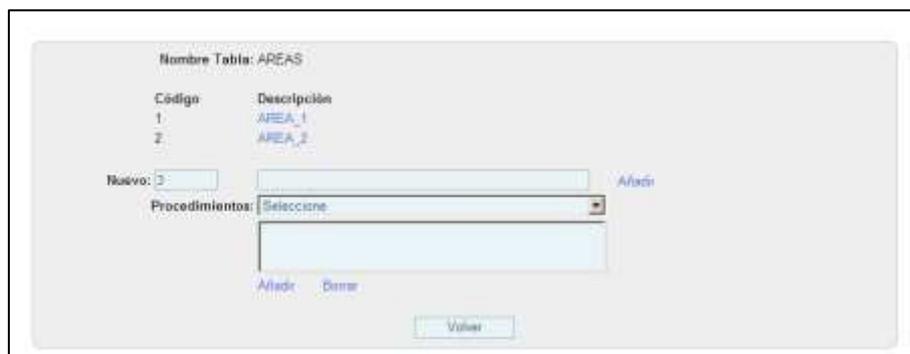


Figura 44. Menú de alta de un área

Si deseamos modificar alguno de los valores de un registro ya existente, lo seleccionaremos y accederemos a un formulario de edición.

Estadísticas

Por último, el módulo de administración nos proporciona una funcionalidad para la obtención de estadísticas, referentes a la realización de procesos de gestión.

La primera pantalla que nos aparecerá nos facilita un formulario de filtrado por una serie de criterios, como se puede observar en la siguiente imagen:



The screenshot shows a web interface for the statistics module. It features a search filter form with three dropdown menus: 'Año' (Year), 'Mes' (Month), and 'Tipo Operación' (Operation Type). Each dropdown menu currently displays 'Seleccione' (Select). Below the dropdowns are two buttons: 'Consultar' (Consult) and 'Volver' (Return).

Figura 45. Pantalla principal del módulo de estadísticas

Tras aplicar los criterios de filtrado deseados, pulsaremos el botón Consultar, obteniéndose un listado de estadísticas con el formato que se puede observar en la siguiente captura de pantalla:



The screenshot shows the results of a search query. At the top, the same search filter form is visible. Below it, a yellow header bar reads 'Resultado de Búsqueda'. Underneath is a table titled 'Estadísticas' with four columns: 'Año', 'Mes', 'Tipo Operación', and 'IP Accesos'. The table contains the following data:

Año	Mes	Tipo Operación	IP Accesos
2009	Mayo	ALTA_ASUNTO	3
2009	Mayo	CONSULTA_ASUNTO	18
2009	Mayo	ELIM_ASUNTO	3
2009	Junio	ALTA_ASUNTO	16
2009	Junio	CONSULTA_ASUNTO	23
2009	Junio	ELIM_ASUNTO	8
2009	Julio	CONSULTA_ASUNTO	3

Figura 46. Pantalla de resultado de las estadísticas

Anexo II

Manual de Usuario del Subsistema de Gestión

Pasos previos y requisitos de acceso

La entrada a este módulo de gestión requiere que el usuario se identifique empleando un Certificado Digital o DNI electrónico aceptado por el Ministerio de Sanidad y Política Social, y estar registrados en el sistema con un rol de gestión. Además, sólo se permitirá el uso de aplicaciones con permisos de gestión, y que éstas tengan permiso a la funcionalidad que queramos llevar a cabo.

1. Certificados digitales

El Profesional que acceda a la aplicación deberá **poseer un Certificado Digital** válido que le identifique en el sistema.

2. Acceso al sistema

La página inicial de este módulo sería:



Figura 47. Pantalla inicial

Como se puede observar en la imagen, este módulo dispone de una serie de opciones de gestión, las cuales se encontrarán habilitadas o no dependiendo de los permisos del usuario que haya accedido.

Nuevo asunto (o estado de procedimiento administrativo)

Una de las funcionalidades de las que disponemos es el alta de estados en los que se encuentra un Procedimiento Administrativo para un determinado usuario o entidad.

Si no disponemos de una aplicación informatizada que almacene el proceso de gestión, o dicha aplicación aún no ha sido adaptada para comunicarse con el WebService, un usuario que tenga permisos de gestión podrá acceder a esta funcionalidad y rellenar el formulario con todos los datos necesarios.

El formulario de entrada tendrá el siguiente formato:

The image shows a web form titled "Nuevo Asunto". It contains the following fields and controls:

- Asunto de: [Selección]
- DNI: [Text input]
- Nombre: [Text input]
- Primer Apellido: [Text input]
- Segundo Apellido: [Text input]
- CIF: [Text input]
- Razón Social: [Text input]
- Área: [Selección]
- Procedimiento: [Selección]
- Código Asunto: [Text input]
- Descripción asunto: [Text input]
- Fecha información: [Text input] (format: dd/mm/yyyy)
- Fecha caducidad: [Text input] (format: dd/mm/yyyy)
- Estado: [Selección]
- Descripción estado: [Text input]
- Otra información: [Text input]
- Aplicación: [Selección]

Buttons: Grabar, Volver, Examinar.

Figura 48. Pantalla de alta de un asunto (estado)

Según se indique o no el valor en el campo Código Asunto, la aplicación considerará si se está insertando un nuevo registro de Procedimiento Administrativo, o si estamos notificando un nuevo estado de un Procedimiento Administrativo ya existente (esto ocurre si hay registrado un asunto con el mismo código, sino también se considera como nuevo).

En la parte inferior del formulario, se deberá indicar la aplicación que realiza la gestión del procedimiento administrativo, es decir, aplicación a la que éste pertenece.

Eliminar asunto (o estado de procedimiento administrativo)

Una segunda funcionalidad es la que permite la eliminación de estados en los que se encuentra un Procedimiento Administrativo para un usuario o entidad. Esto puede ser necesario, por ejemplo, en el caso de registros existentes que ya no tienen sentido mantenerlos para ser consultados, por haberse alcanzado la fecha de caducidad de los mismos, o por existir un histórico extenso de estados del procedimiento y deseemos prescindir de los más antiguos.

Si seleccionamos la opción de Eliminar Asunto del menú principal del Módulo de Gestión, accederemos a un formulario como el que se muestra en la siguiente imagen:



Eliminar Asunto

Asunto de:

DNI:

Nombre:

Primer Apellido:

Segundo Apellido:

CIF:

Razón Social:

Código Asunto:

Código Asunto Único:

Aplicación:

Figura 49. Pantalla de eliminación de un estado (asunto)

Tras indicar los criterios que han de cumplir los registros de estados de Procedimientos Administrativos que se desean borrar, pulsaremos el botón *Eliminar*; una vez se lleve a cabo el proceso de borrado, se mostrará una pantalla de confirmación informando del resultado del mismo.

Consultas

Por último, existe la funcionalidad de consulta del estado en el que se encuentran los Procedimientos Administrativos.

La primera pantalla que se mostrará si accediésemos a esta funcionalidad es la del formulario que nos permite el filtrado aplicando una serie de criterios; este formulario es el que se muestra a continuación:

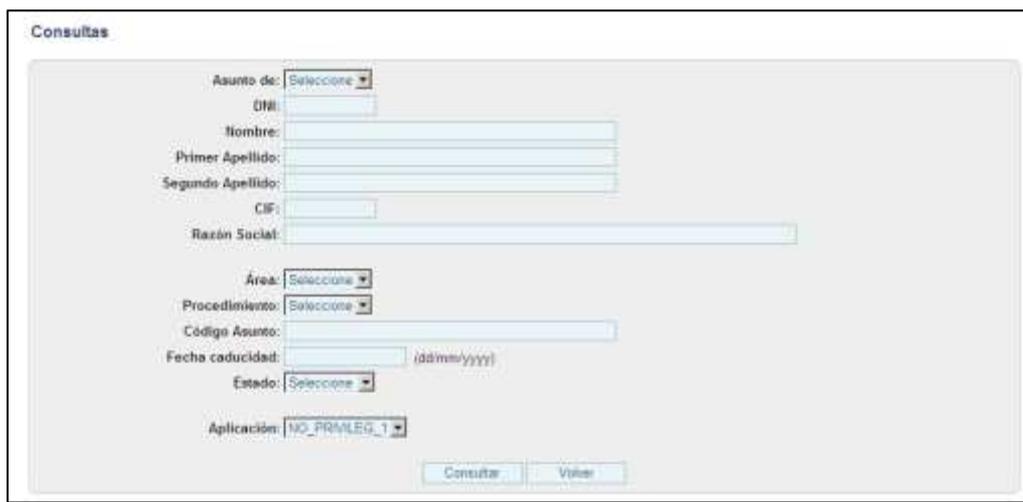


Figura 50. Pantalla de consulta de asuntos

En el formulario de filtrado, rellenaríamos los campos con los criterios que deseásemos aplicar en la búsqueda, y pulsaríamos el botón Consultar. Si existiesen registros coincidentes, se mostraría un listado de los mismos, para una posterior selección de alguno de ellos si se desea consultar con mayor detalle sus datos.

Si por el contrario, la búsqueda no obtuviese registros coincidentes, se mostraría un mensaje informativo indicándolo.

En la siguiente imagen se muestra el resultado de una consulta en la que se han obtenido registros coincidentes:

Consultas

Asunto de:

DNI:

Nombre:

Primer Apellido:

Segundo Apellido:

CIF:

Razón Social:

Organismo:

Área:

Procedimiento:

Código Asunto:

Fecha caducidad:

Estado:

Aplicación:

Resultado de Búsqueda

Asuntos					
Nombre	DNI/CIF	Procedimiento	Área	Estado	
Vicente Sánchez, Julio Francisco	47040233N	Reclamaciones Previa a la vía judicial laboral y civil	Otros	INICIO	  
Vicente Sánchez, Julio Francisco	47040233N	Aprobación de sprays de defensa personal de venta permitida en armerías	Autorizaciones, licencias y concesiones	FINALIZACIÓN	  

Figura 51. Pantalla del resultado de una consulta de estados (asuntos)

El listado de registros coincidentes nos permite tener una visión rápida de los procedimientos administrativos que tiene en proceso un usuario o entidad, además de posibilitar la realización de una de las siguientes acciones:

- Consulta más detallada de alguno de ellos si pulsamos en el icono de la lupa.
- Añadir un nuevo estado al procedimiento administrativo
- Eliminar un asunto (o procedimiento administrativo)

1. Ver datos de un asunto

Un ejemplo de consulta detallada de los datos de un procedimiento administrativo procesado para un usuario, y el estado en el que éste se encuentra, es el reflejado en la siguiente imagen:



Figura 52. Pantalla detalle de un asunto (estado)

En el momento de inserción del estado de un procedimiento administrativo, se puede adjuntar un documento electrónico como valor del campo Otra Información. Si se consultase los datos de uno de estos registros, se muestra un enlace a dicho fichero adjunto.

Al pulsar en el enlace del campo Otra Información, se realizará la descarga del fichero, el cual podrá ser abierto directamente o almacenado en local para su posterior visualización. La siguiente imagen muestra dicha acción:

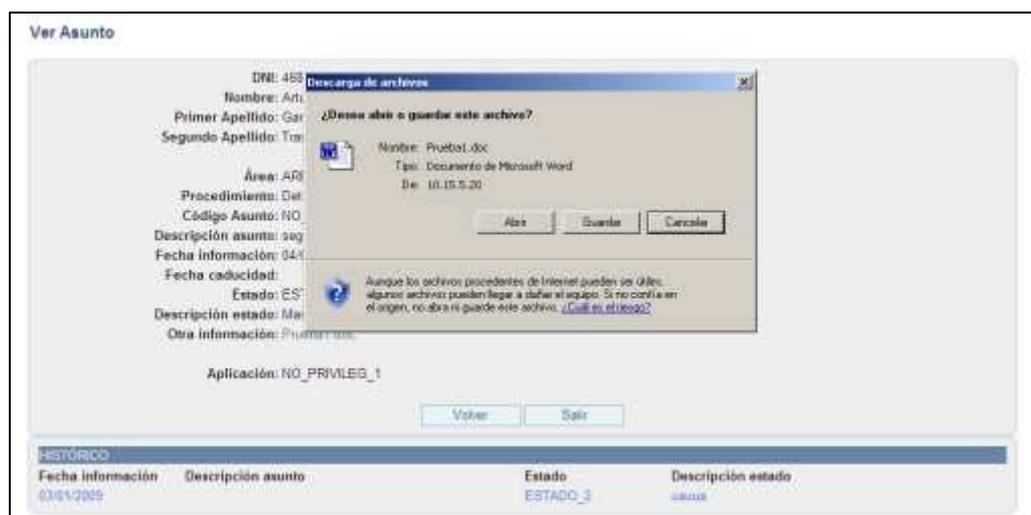


Figura 53. Pantalla de descarga de un adjunto

Además, si el procedimiento administrativo ha pasado por diferentes estados, se mostrará un listado con dicho histórico, el cual nos permitirá poder consultar, por

Añadir Estado

Asunto de: Ciudadano
 DNI: 47040233N
 Nombre: Julio Francisco
 Píname: Vicente
 Apellido: Vicente
 Segundo Apellido: Sánchez
 CIF:
 Razón Social:

Procedimiento: Reclamaciones Previas a la vía judicial laboral y civil
 Código Asunto: ASUNTO_PRUEBA
 Descripción asunto: Prueba de alta de asunto
 Fecha información: 25/11/2009 (dd/mm/yyyy)
 Fecha caducidad: (dd/mm/yyyy)
 Estado: Seleccionar
 Descripción estado:
 Otra información: Examinar...
 Aplicación: Calidad

Figura 55. Pantalla para añadir un nuevo estado

Pulsando el botón de “Grabar” se dará de alta un nuevo estado para este asunto.

3. Eliminar un asunto (o procedimiento administrativo)

Esta opción también es una segunda opción para la eliminación de un asunto. La primera ya fue detallada en el apartado “Eliminar Asunto”

Desde la pantalla de resultado de una consulta, pulsando el icono de la papelera  , se eliminará el asunto en concreto.