

MEMORIA



Sistema integral de asistencia al empleado público



Trabajo de Fin de Carrera
Ingeniería Técnica Informática de Gestión
Ingeniería del Software. 2º Semestre 2016

Consultor: Oriol Martí Girona

Alumno: Carlos Flores Fernández

Fecha Entrega: 17 de junio 2016

*A mi mujer Lina y a mi hijo Fran
por su renuncia, comprensión y complicidad.*



Trabajo de Fin de Carrera.
Ingeniería del Software.
2º Semestre. 2016.



Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-CompartirIgual
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA GENERAL DEL PRODUCTO

Título	<u>eEmpleadoPublico</u> SISTEMA DE ASISTENCIA AL EMPLEADO PÚBLICO
Consultor	Oriol Martí Girona
Autor	Carlos Flores Fernández
Fecha	Junio de 2016
Área	Ingeniería del Software
Título	Ingeniería Técnica Informática de Gestión
Resumen (máximo 250 palabras)	
<p>El objetivo de este proyecto es la generación de la documentación precisa, en términos de especificación y diseño técnico, para hacer posible el desarrollo de una aplicación que, una vez integrada en una página web, sirva para dar satisfacción a las necesidades de apoyo y asistencia jurídica, para ofrecer materiales que posibiliten y mejoren el conocimiento profesional y, sobremanera, faciliten la gestión de las permutas entre trabajadores públicos de las distintas administraciones públicas que por diversas circunstancias acariciasen la posibilidad de cambiar.</p> <p>La documentación referida habrá de posibilitar, sin perjuicio de revisiones sucesivas, utilizando Java como lenguaje de programación orientado a objetos y la tecnología RMI como sistema de invocación de métodos remotos no sólo la implementación y el mantenimiento sino también, llegado el caso, ampliaciones posteriores.</p> <p>La aplicación final, en todo caso y dado que el target al que va dirigido constituye un variado mosaico de potenciales usuarios finales, con diferentes formaciones, aptitudes y circunstancias personales y profesionales, deberá ofrecer una interfaz sencilla, clara y amigable que facilite la interacción, no sólo con la web que le de soporte, sino también entre ellos.</p>	
Abstract (in English, 250 words or less):	
The objective of this project is the generation of accurate documentation, in terms	

of technical specification and design, to enable the development of an application that once embedded in a web pa'gina, serve to satisfy the needs of support and legal assistance, to provide materials to facilitate and improve professional knowledge and, greatly, to facilitate the management of public exchanges between workers of the various public authorities for various reasons caress the possibility of change.

These documents shall permit, subject to subsequent revisions, using Java as a programming language and object-oriented technology as system RMI remote method invocation, not only the implementation and maintenance but also, where applicable, subsequent extensions .

The final application, in any case, and given that the target to be targeted is a varied mosaic of potential end users, with different backgrounds, skills and personal and professional circumstances, it should provide a simple, clear and friendly interface that facilitates interaction not only with the web that will support, but also between them.

Palabras clave (entre 4 y 8):

Aplicación, cliente-servidor, RMI, objeto, especificación, diseño, Java.

Índice de contenidos

Contenidos.....	Página
1 Preparación del proyecto.....	7
1.1 Introducción.....	7
1.2 Objetivos.....	9
1.2.1 Objetivos Generales.....	10
1.2.2 Objetivos Específicos.....	11
1.3 Metodología y definición de tareas.....	11
1.3.1 Ciclo de vida clásico.....	11
1.3.2 Relación de tareas.....	13
1.3.2.1 Plan de Trabajo.....	13
1.3.2.2 Especificación y requisitos.....	13
1.3.2.3 Diseño técnico.....	14
1.3.2.4 Memoria y presentación.....	14
1.4 Calendario.....	14

2	Análisis de requisitos.....	18
2.1	Consideraciones Previas.....	18
2.2	Actores.....	19
2.3	Funcionalidades generales.....	19
2.3.1	Exigencias funcionales.....	20
2.3.2	Estructura de paquetes.....	20
2.3.2.1	Subsistema de conexión.....	21
2.3.2.2	Subsistema de mantenimiento.....	23
2.3.2.3	Subsistema de facturación.....	25
2.3.2.4	Subsistema de flujo.....	26
2.4	Gestores.....	27
2.5	Funcionalidades específicas.....	30
2.6	Especificación textual.....	33
2.6.1	Diagrama general	33
2.6.2	Descripción general. Casos de Uso.....	34
3	Diseño técnico.....	49
3.1	Fichas CRC.....	49
3.2	Diagramas de Clase.....	61
3.3	Modelo de comportamiento.....	63
3.3.1	Diagramas de actividad	63
3.3.2	Diagramas de secuencia y/o colaboración.....	67
3.4	Persistencia.....	69
3.4.1	Modelo E-R.....	69
3.4.2	Diseño lógico.....	72
4	Valoración económica.....	73
5	Conclusiones.....	74
6	Glosario.....	74
7	Bibliografía.....	75

1. Preparación del proyecto

1.1. Introducción

Cualquier miembro del colectivo que, en principio, fue concebido como un contrapeso y garantía al abuso y la preeminencia, más o menos acusada según el lugar, que el poder ejecutivo parecía tener después de las primeras y tímidas experiencias, con el progresivo tránsito del estado absolutista al estado llamado moderno, de la puesta en práctica de la teoría de la división de poderes, respecto de los otros dos poderes, legislativo y judicial, es el germen de lo que hoy conocemos popularmente como funcionario y generalizando, empleado público.

Este sujeto anónimo, asexuado, desprovisto actualmente de casi cualquier envoltura que pudiera considerarse humana, es percibido por la colectividad, no pocas veces, como un espécimen raro y carente de otros afanes que no sean su propio bienestar, y de otras inquietudes que no sean la contemplación del progreso de su carrera administrativa mediante la aplicación rigurosa del principio liberal y recursivo (y apoyado no pocas veces por esa crítica colectividad), de *“laissez faire, laissez passer”*, y persuadido de que la teoría heliocéntrica es errónea, pues los planetas giran, no alrededor del sol, sino de una parte muy determinada de su anatomía.

En lo tocante al estado español, hay que decir que este estado de cosas se debe a “méritos” tanto propios como ajenos gestados desde tiempos lejanos y en una estructura social muy determinada. Efectivamente, hay abundantes y lejanas referencias documentadas ya por Mariano José de Larra y su célebre obra “Vuelva Ud Mañana” que dan fe de aquella afirmación. En todo caso, nos asomamos a un colectivo con unas características muy determinadas. Intentamos analizarlas a continuación:

- Es **numeroso**. Se sitúa en torno a 2.500.000 efectivos.
- Es **disperso**. La mitad aproximadamente (50%) corresponde a las CC.AA, un 20% a las entidades locales, y algo más del 5% a las universidades de titularidad pública. Por su parte, la administración central “carga” con un porcentaje entre el 20% y el 25%.
- Está inmerso y **sometido** a un proceloso magma de **normas jurídicas** emanadas del amplio mosaico de entes político-territoriales que componen el estado: Administración central, autonomías, diputaciones, municipios, entidades menores, insulares, cabildos, etc,etc.

- A partir de la adscripción enunciada anteriormente, y a pesar de una apariencia homogénea en materia salarial, existen **diferencias** retributivas realmente notables entre individuos de una misma denominación y categoría administrativa, no sólo entre administraciones diferentes. Si hablamos de convenios, el agravio se dispara en materia de horarios, ayudas, etc. Si nos ceñimos a las facilidades de acceso a formación necesaria para actualizar y mejorar su desempeño y/o impulsar su promoción en el escalafón, también las deficiencias pueden ser muy grandes.
- A este galimatías hay que hacerle una mención muy especial. Se trata de los municipios. Hay en todo el estado algo más de 8.000. La mayoría de ellos son entidades que apenas alcanzan los 3000 habitantes y alrededor de media docena de funcionarios de carrera, donde los regidores, no pocas veces, ejercen de sátrapas con total desfachatez, los sindicatos no actúan por complejas razones que ahora no vienen al caso y donde existe un complejo e intrincado entramado de relaciones dispares entre políticos, funcionarios y vecinos en ese ayuntamiento y zona de influencia que hace difícil, cuando no imposible, cualquier queja, reclamación, petición o defensa.
- Así las cosas, y a mayores, circunstancias como motivos de trabajo del cónyuge, estudios de los hijos, oportunidades de promoción, incompatibilidad climática con alguna enfermedad o simplemente hastío de la situación laboral pueden ser sólo algunas de las motivaciones que obliguen al funcionario a acariciar la idea de, si fuera posible, intentar la permuta con otro con igual intención, y con igual o distinta motivación.

Recapitulando, nos encontramos ante un grupo humano numeroso, diseminado en multitud de entidades en unos casos (municipios) o, por el contrario, concentrado en monstruosos edificios administrativos (estado y CC.AA), pero en cualquier caso disperso por su adscripción o por su situación y, sobre todo, inconexo.

Las dos grandes debilidades citadas, dispersión e indefensión van a articular la siguiente propuesta de TFC: Creación de una aplicación informática que, aprovechando las posibilidades que hoy en día ofrece Internet, haga factible una triple finalidad.

- 1) Poner en contacto a potenciales interesados en permutar.**
- 2) Ofrecer asistencia jurídica a quienes por diversos motivos no pudieran acceder a ella de forma cercana, personal y discreta.**
- 3) Posibilitar el acceso a la formación de forma centralizada y continua.**

1.2.-Objetivos

Después de la exposición precedente, el proyecto parece enfrentarse a 2 grandes retos

- Uno interno consistente en relacionar y agrupar una serie muy variada de conceptos aparentemente vulgares, si se quiere pueriles u aparentemente más cercanos a las relaciones laborales o la sociología que de la ingeniería del software, y que, sin duda, van a surgir del intento de materializar en software la problemática referida.
- Otro externo que surge de forma inmediata cuando nos planteamos la necesidad de poner las funcionalidades de ese software a disposición de un grupo humano, numeroso, variado y disperso físicamente.

El primer reto intentaremos solucionarlo mediante lo que se conoce como *paradigma de la orientación a objetos* (POO) que es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación. Proponemos para ello la utilización de un lenguaje de programación basado en este paradigma llamado JAVA



figura 1.- Esquema POO

Para el segundo proponemos, igualmente, otro paradigma, *el de la programación distribuida*, que nos va a permitir repartir o hacer residir en partes llamadas cliente y servidor, de ese software en máquinas diferentes utilizando para ello una tecnología que lo va a hacer conectándolas y sincronizándolas a través de Internet

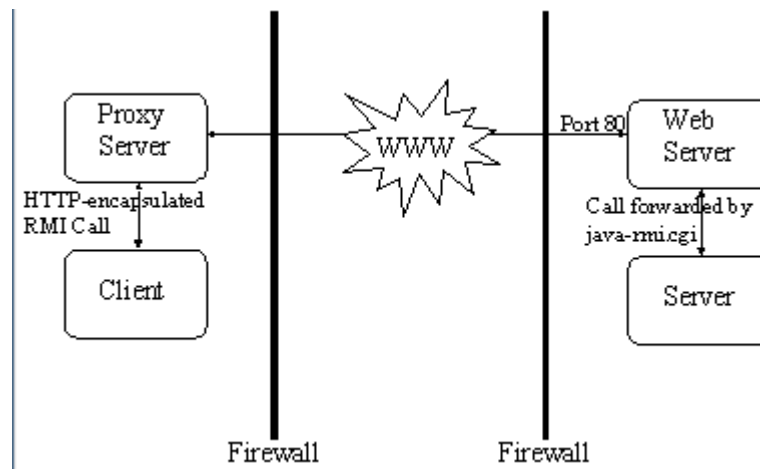


figura2.-Remote Method Invocation

1.2.1.- Objetivos generales.

El proyecto nace de la necesidad de diseñar una aplicación que integre un conjunto de funcionalidades que posibiliten superar y dar satisfacción a las 3 grandes debilidades expuestas en el punto 1.1 de este documento.

Se constituye como objetivo principal, por tanto, de este proyecto dotar al conjunto de los empleados públicos de todo el estado de un sistema que gestione electrónicamente la documentación, la información y el tránsito por el proceloso y variado mundo de las normas del derecho público que rigen las relaciones entre las administraciones y sus servidores.

Así, y con la ejecución del presente proyecto se pretende establecer las bases de un sistema integral que aglutine en su seno las 3 grandes necesidades citadas: *La gestión de permutas, la prestación de asistencia jurídica y la oferta de materiales* didácticos necesarios para su formación, actualización y promoción, utilizando para ello las facilidades que, hoy en día, ofrece la red en particular y las nuevas tecnologías de la información en particular.

1.2.2.-Objetivos específicos.

Con los anteriores precedentes, el objetivo es la creación de la documentación que sirva de base la implementación posterior de de una aplicación informática que, posibilita:

- ✓ Gestionar de forma eficaz las permutas entre empleados públicos que

pudieran plantearse.

- ✓ Ofrecer al cliente asesoramiento, defensa y representación jurídica en cuantos conflictos pudieran afectarle en su vida laboral
- ✓ Poner a disposición de los potenciales clientes de forma estructurada, actualizada y centralizada, materiales didácticos y normativas jurídicas que les fueran de interés para su formación, puesta al día y promoción en su carrera administrativa.

Por tanto, como fin último, se persigue sentar las bases en cuanto a análisis de requisitos y al diseño técnico, de un sistema software de calidad que satisfaga tanto los requerimientos funcionales que se establezcan en las especificaciones correspondientes, como las necesidades de mantenimiento o crecimiento posteriores.

1.3.-Metodología y definición de tareas

Las técnicas de la ingeniería del software señalan *cómo* construir técnicamente el software. “Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento” (Pressman, 2001).

Según Pradel, el método de desarrollo elegido definirá las etapas, es decir, procesos, tareas y actividades que tendrán lugar y, por supuesto interacciones y dependencias entre roles, usuarios y tareas.

1.3.1.--Ciclo de vida clásico o en cascada.

La metodología a seguir para este proyecto, está basada en el ciclo de vida clásico del software, también llamado ciclo de vida en cascada. Este modelo puede ser válido para el caso que nos ocupa, aunque al finalizar la etapa de Especificación y Análisis de Requerimientos y comenzar la etapa de Diseño Técnico pueden surgir modificaciones, ya que tendremos una idea mucho más precisa de los requisitos. En el siguiente gráfico podemos ver un esquema general del proceso de desarrollo de este modelo:

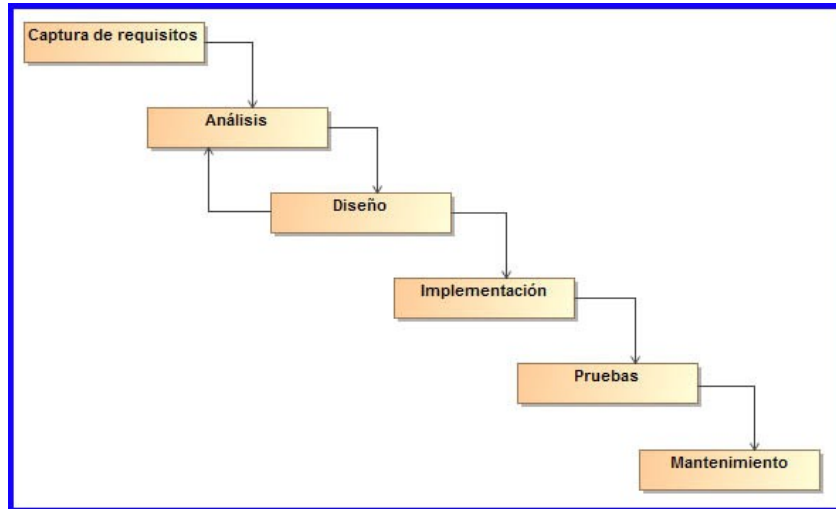


figura3.- Ciclo de vida en cascada

En el caso que nos ocupa, sin embargo, estará limitado a las tres primeras etapas. En el siguiente gráfico podemos ver un esquema general del proceso de desarrollo que se va a llevar a cabo:

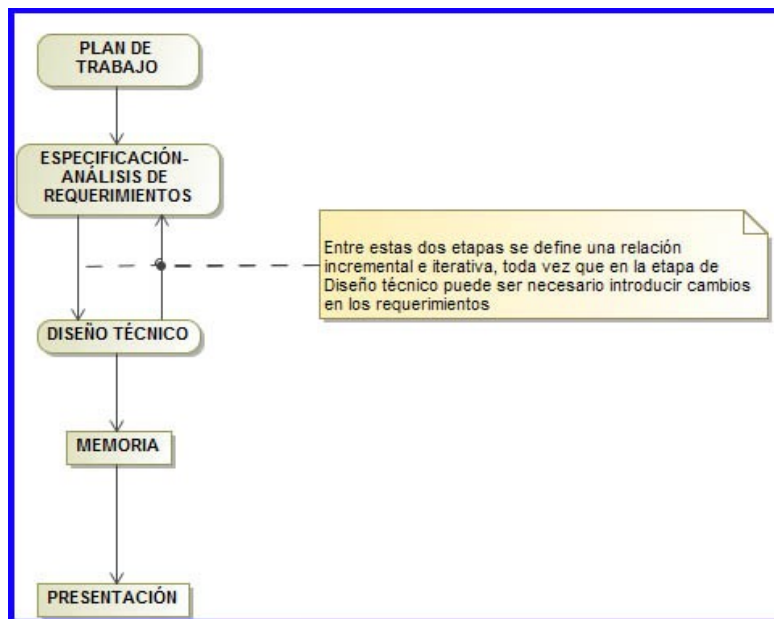


figura 4.- Alcance de eEmpleadoPúblico.

1.3.2.- Relación y estructura de tareas a realizar.

1.3.2.1.-Plan de Trabajo

- Descripción del proyecto
- Metodología y organización de tareas.
- Elaboración del Diagrama de Gantt.
- Confección del documento de entrega del Plan de Trabajo (PEC-1).
- Entrega del Plan de Trabajo (PEC-1)

1.3.2.2.-Especificación y Análisis de Requerimientos

- Definición de actores y subsistemas.
- Funcionalidades por subsistema y actor.
- Estructura de menús principales por actor.
- Diseño de la interfaz de los menús principales.
- Composición de paquetes e interrelaciones.
- Especificación de funcionalidades por subsistema, incluyendo diseño GUI.
- Especificación textual de cada subsistema con diagramas de casos de uso más representativos.
- Composición de paquetes e interrelaciones.
- Especificación de funcionalidades por subsistema.
- Confección documento de entrega.
- Entrega de la PEC-2.

1.3.2.3.- Diseño Técnico

- Diagramas de clases de negocio, gestores, excepciones y pantallas, por subsistema.
- Diagramas de colaboración de los principales casos de uso, por subsistema.
- Clases en notación CRC por subsistema.
- Variaciones del prototipo de interfaz gráfica.
- Diseño de persistencia.
- Relación de clases y su pertenencia a subsistemas,
- Confección del documento de entrega.
- Entrega de la PEC3.

1.3.2.4.- Memoria y presentación

- Confección de la memoria y la presentación
- Entrega.

1.4.- Calendario de actividades

Mediante un Diagrama de Gantt, se ha representado la planificación propuesta en un principio. Dado el peculiar sistema de trabajo del alumno a turnos de mañana, tarde y noche con inclusión de fines de semana y festivos y descansos intercalados, se ha realizado una planificación inclusiva de todos los días de la semana, en función de sus disponibilidades, y sin excesiva precisión. Por razones de legibilidad, se aportan cuatro gráficos. El primero, una aproximación a las tareas a realizar. El segundo, una panorámica general del calendario. El tercero comprensivo de las dos primeras Pec's, y el cuarto de las dos restantes incluyendo también, todos ellos dependencias entre tareas.

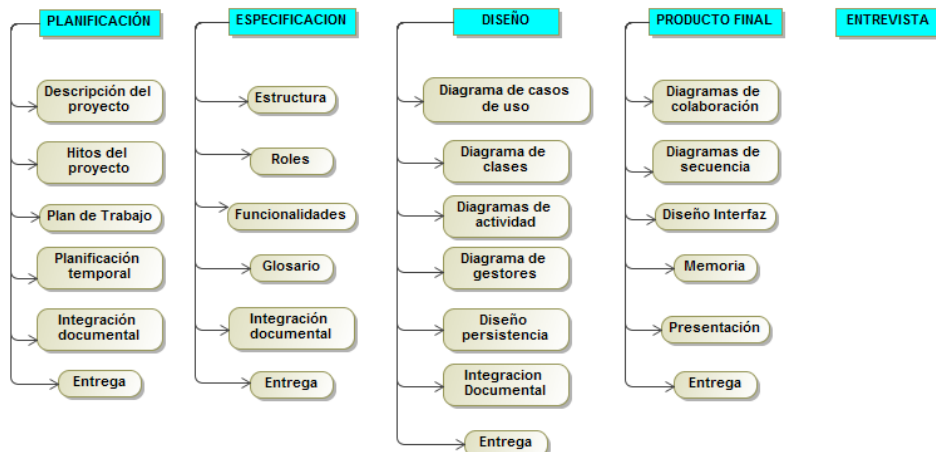


figura5.- Aproximación de tareas a realizar

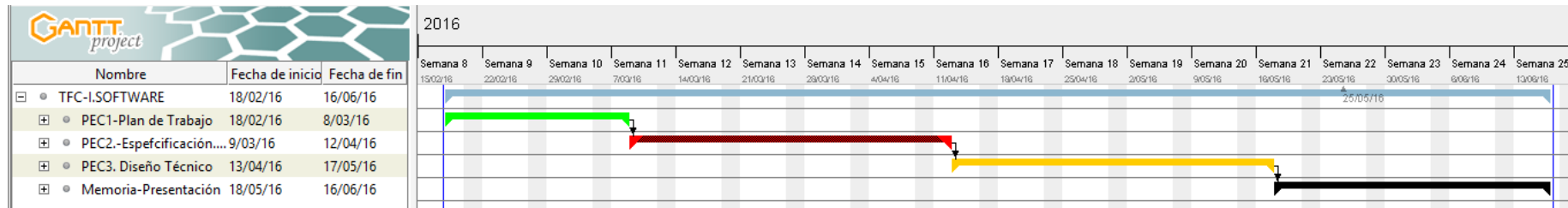


figura 6.- Cronograma general

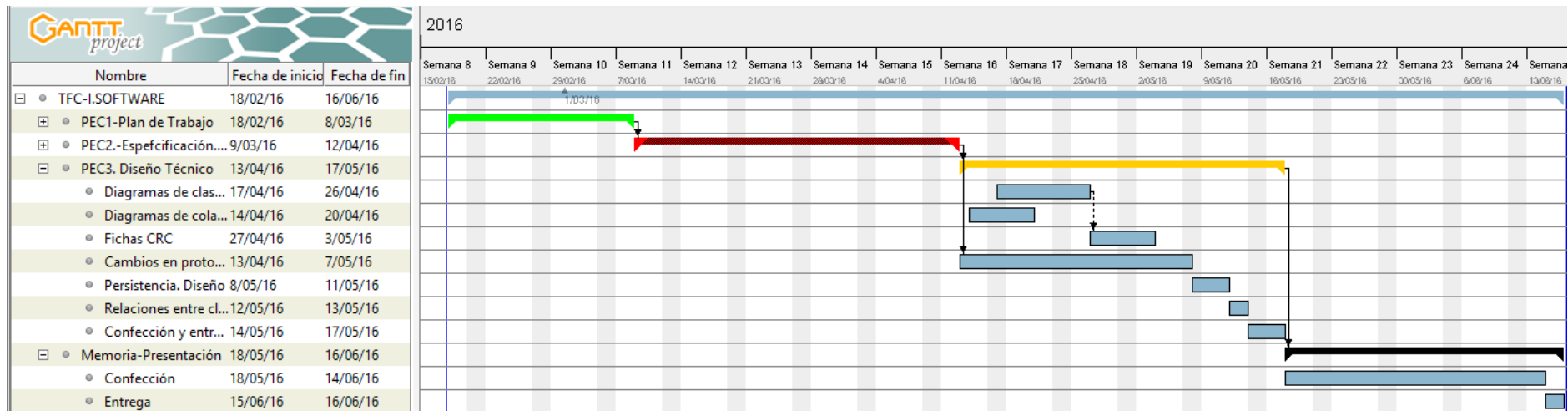


figura 7.- Cronograma PEC1 y PEC2

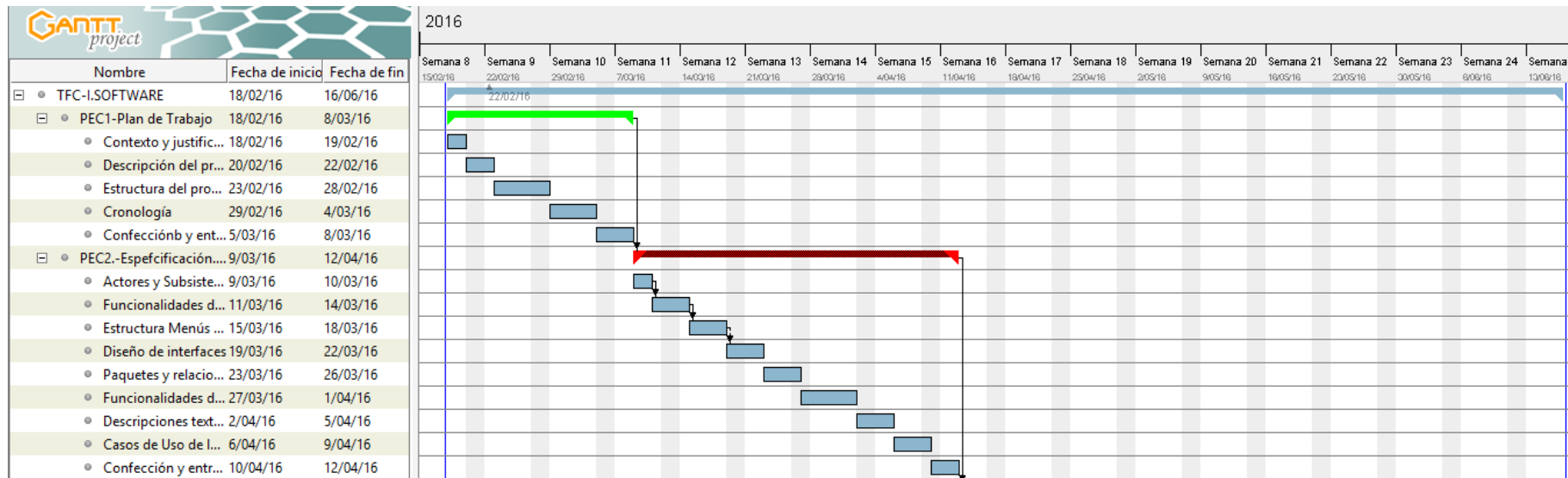


figura 8.- Cronograma PEC3 y PEC4

2. Análisis de requisitos

2.1.-Consideraciones previas

La formulación del problema por parte del cliente que encarga el software suele ser el punto de partida. Sin embargo, muchas veces está incompleta, introduce ambigüedades y está plétórica de inconcreciones por lo que precisa ser completada y refinada hasta que pueda plasmarse en un documento que atesore los requerimientos que deseamos desarrollar.

A partir del enunciado de necesidades planteados en el capítulo anterior, este capítulo se dedicará a desarrollar las especificaciones que van a tener los requerimientos de software a construir y que serán de utilidad en fases posteriores.

Este análisis posibilitará al ingeniero definir las potencialidades operativas del software (función, datos y eficacia), indicará la interacción del software con otros elementos del sistema y fijará las restricciones que debiera cumplir (Pressman, 2001).

Como resulta evidente, este proyecto pertenece a un entorno académico y, por tanto, hipotético. Aunque es el alumno quién define las características del sistema, ello no le quita un ápice de importancia a esta fase, pues, no en vano, consumirá gran cantidad de tiempo decidiendo qué quiere construir, cómo documentarlo y qué herramientas elegir para llevarlo a cabo.

Resumiendo, los aspectos más notables que ha de recoger la especificación del proyecto son:

- Descripción general: Indica los objetivos del sistema a desarrollar, subsistemas de los que dispone y herramientas software y hardware (lenguaje, sistemas a soportar, uso de red, etc.).
- Identificación de actores: Los distintos Roles desarrollados por una persona, un dispositivo hardware u otro subsistema al interactuar con el nuestro.
- Diagrama de paquetes: Donde se indican las relaciones entre los subsistemas y el sistema principal.

- Descripción de las funcionalidades por subsistema.
- Descripción de los casos de uso por subsistema.

Seguidamente se pasa a poner en evidencia los aspectos a desarrollar en el documento.

2.2.-Actores

Siguiendo a Booch, normalmente, un actor representa un rol que es jugado por una persona, un dispositivo o incluso otro sistema al interactuar con nuestro sistema. Como requisito previo es requerido que todos los actores que interactúan con en el sistema deben ser usuarios registrados en el mismo. En el caso que nos ocupa, nos encontramos con los siguientes actores:

- **Administrador.**-Rol de gestión que tendrá control total sobre la aplicación. Permite crear, consultar, modificar y borrar los datos relativos a las diferentes entidades del sistema.
- **Usuario** .- Rol que agrupa a cualquier usuario genérico que tenga acceso a cualquiera de los funcionalidades que ofrece la aplicación (permutas, materiales, asesoría, etc). A su vez, pueden ser *clientes, autores o letrados*.
- **Secretaría.**-Personal administrativo encargado de las tareas de mantenimiento y facturación.

2.3.-Funcionalidades generales

En este apartado se van a incluir una relación de funcionalidades muy generales relativas a los cuatro subsistemas que se citan en el párrafo siguiente inherentes al software a desarrollar con una definición precisa de todas ellas mediante lenguaje literario.

Como se podría intuir de apartados anteriores, nuestra aplicación podría considerarse estructurada en bloques “*especializados*” que a partir de aquí denominaremos subsistemas y que se encargarán de la gestión de parcelas muy determinadas.- Estos serán el subsistema de *Conexión*, el de *Mantenimiento*, el de *Flujo* y el de *Facturación*. Empezaremos centrándonos en la interacción que entre ellos hay, mediante la inclusión de

estas *áreas de gestión* en paquetes de software, para abordar, a continuación, sus respectivas funcionalidades.

2.3.1.- Estructura de paquetes

Es práctica habitual aquella de fraccionar problemas demasiado complejos en partes que puedan ser más digeribles por separado y, a partir de aquí, establecer interacciones entre ellas. Estos serían los paquetes, constituidos, así, como artificios para organizar diferentes elementos en grupos. Estos grupos contendrían entonces, elementos tanto estructurales como grupales, u otros según sus afinidades. Así las cosas, a este nivel, un paquete se identifica con un subsistema.

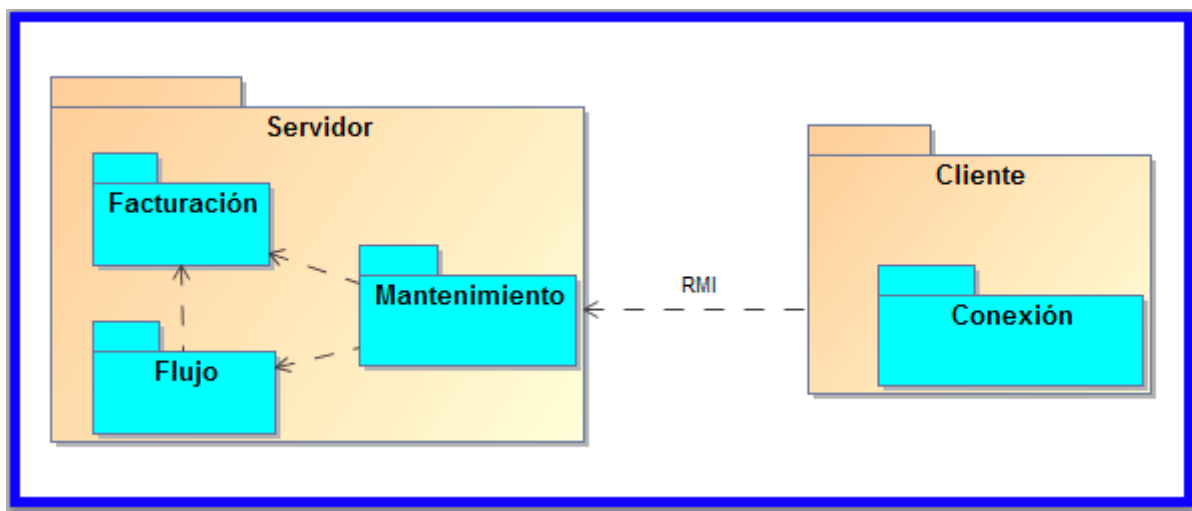


Figura 9.- Diagrama de paquetes de la aplicación

2.3.2.- Exigencias funcionales.

Con esta denominación estamos haciendo alusión al comportamiento que un sistema mantiene ante una determinada situación. Es decir, qué datos maneja, cómo los manipula en sentido amplio, si hace cálculos y cómo los hace, etc.

En el presente documento se va a aprovechar la ocasión de definir las funcionalidades del sistema, para presentar una primera aproximación, espartana por otra parte, del prototipo de la GUI de cada una de esas funcionalidades.

Así las cosas y antes de entrar de lleno en la definición de funcionalidades por cada subsistema, en la figura siguiente se muestra una captura de pantalla de acceso general a la aplicación.



Figura 10.- Pantalla acceso general

2.3.2.1.-Subsistema de Conexión.

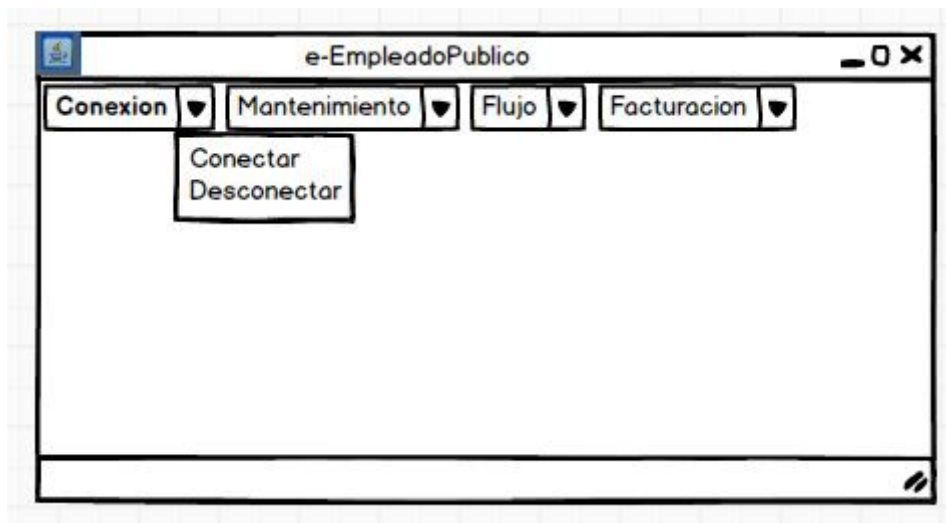


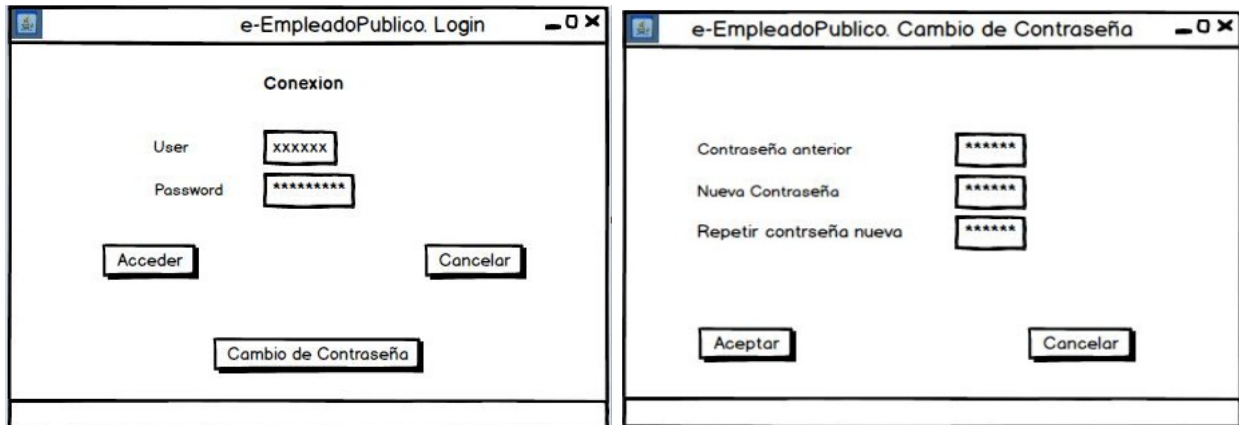
Figura 11.- Menú Conexión

- .Es el subsistema encargado de dar acceso a los usuarios a la aplicación y, en su caso, cambiar la contraseña.
- Los diferentes menús y opciones a ejecutar por los usuarios se mostrarán en función del perfil que estos tienen predefinidos.
- Requisitos:

✓ **PreCondición:** El usuario que accede al sistema

dispone de un código de usuario y contraseña válidos.

- ✓ **Postcondición:** El usuario accede a la aplicación con su perfil una vez introducidas sus credenciales en el sistema. En el caso de que las credenciales informadas no sean válidas se informará al usuario por pantalla.



Figuras 12 y 13.- Login y cambio de contraseña

- Prestaciones.- Como funcionalidades básicas del subsistema de conexión se han identificado las siguientes.

- ✓ **Conexión a la aplicación.** Permite la introducción de las credenciales de los usuarios.

- ✓ **Escenario principal:**

- En la pantalla de inicio de sesión, el usuario introduce el nombre de usuario y la contraseña.
- El usuario pulsa sobre el botón *Acceder*.
- El subsistema valida si las credenciales introducidas son válidas.
- Si los datos introducidos son correctos se inicia el acceso a la aplicación.

- ✓ **Consulta del tipo de Rol que accede.-** Realiza la consulta del nivel de acceso del usuario que está accediendo a la aplicación. Una vez consultado el tipo de Rol, se accede a consultar el nivel de acceso que tiene el rol y poder identificar las funcionalidades a las que puede acceder el usuario.

- ✓ **Roles de la aplicación.** Son los siguientes:

- **Administrador:** Rol de gestión que tendrá

control total sobre la aplicación. Permite crear, consultar, modificar y borrar los datos relativos a las diferentes entidades del sistema.

- **Usuario** .- Rol que agrupa a cualquier usuario genérico que tenga acceso a cualquiera de los funcionalidades que ofrece la aplicación (permutas, materiales, asesoría, etc). A su vez, pueden ser *clientes, autores o letrados*
- **Personal Administración**.- Personal administrativo encargado de las tareas de mantenimiento y facturación.

- ✓ **Nivel de acceso del Rol.** A continuación, mediante la siguiente tabla, se detallan a los subsistemas que puede acceder cada uno de los roles de la aplicación y el nivel de permisos para cada uno de ellos.

Niveles de Acceso por rol y subsistema				
Rol	Conexión	Mantenimiento	Flujo	Facturación
Administrador	Total	Total	Total	Total
P_administración	Parcial	Parcial	Parcial	Total
Usuario	Parcial	Parcial	Parcial	Parcial

Debe entenderse que el acceso denominado *Total* se refiere a la posesión de permisos de lectura y modificación/edición sin restricciones. Por el contrario, el *Parcial* implica la carencia de permisos de lectura y/o edición sobre todas o algunas de las funcionalidades que ofrece el sistema.

Por políticas de seguridad de la aplicación, una vez transcurrido el periodo de días establecido en dicha política la contraseña del usuario pasará a estado caducada debiendo establecer una nueva cuando la aplicación la solicite.

2.3.2.2.-Subsistema de Mantenimiento

- Este subsistema se encargará de la gestión de los Usuarios de la aplicación, las distintas entidades que integran ésta (entendiéndose como los productos generados por la actividad, es decir, materiales actividad jurídica, peticiones permutas y la trascendencia económica

que pudieran tener.

- Sus funcionalidades básicas serán:

✓ *Gestión de Usuarios:*

- Alta de un usuario
- Baja de un usuario
- Consulta/ modificación de un usuario

✓ *Gestión de materiales didácticos:*

- Alta de cada uno de los materiales.
- Baja de un material.
- Consulta/Modificación

- A todas estas funcionalidades se podrá acceder desde el Menú de la Aplicación. Pulsando sobre “Mantenimiento” y la gestión en particular que se desea realizar.

- Requisitos:

- ✓ **Precondición.-** El usuario que accede al sistema dispone de un código de usuario y contraseña válidos y la condición de usuario, personal administrativo o administrador.
- ✓ **PostCondición.** El usuario accede a la aplicación con su perfil una vez introducidas sus credenciales en el sistema y a las funcionalidades que le ofrece este subsistema.

- Prestaciones.- Como funcionalidades básicas del subsistema de mantenimiento se han identificado las siguientes:

-Escenario principal:

1º.- Desde la pantalla principal, el usuario accede al menú mantenimiento

2º.- El usuario elige el submenú correspondiente. El sistema abre una nueva pantalla correspondiente al submenú de mantenimiento con las funcionalidades correspondientes a su rol.

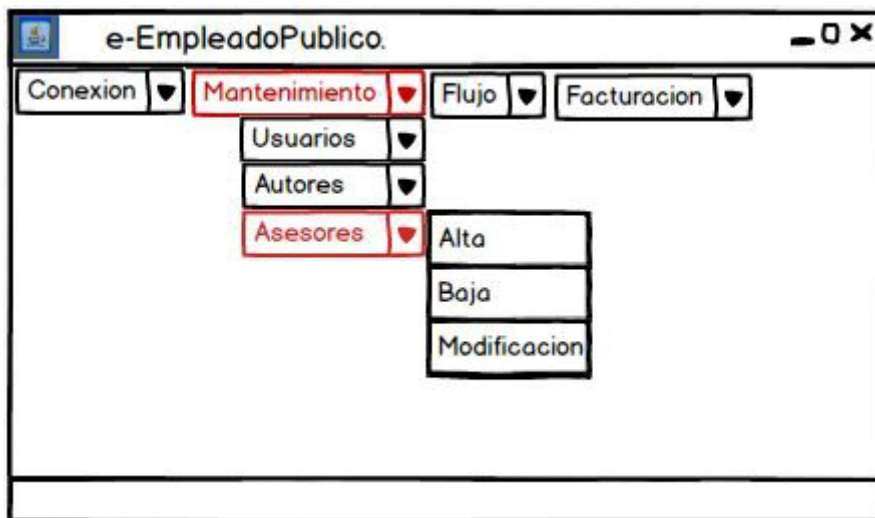


Figura 14.- Menú Mantenimiento

2.3.2.3.- Subsistema de Facturación.

- **Es el subsistema encargado** gestionar las facturas correspondientes a los ingresos económicos derivados de la actividad:
 - ✓ Facturas de correspondientes a gestión de permutas.
 - ✓ Facturas correspondiente a servicios y consultas plasmados en soporte documental y formal.

- **Requisitos:**
 - ✓ **PreCondición:** El usuario que accede al sistema dispone de un código de usuario y contraseña válidos y la condición de usuario, personal administrativo o administrador.
 - ✓ **PostCondición.** El usuario accede a la aplicación con su perfil una vez introducidas sus credenciales en el sistema y a las funcionalidades que le ofrece este subsistema.

- **Prestaciones.**

✓ **Escenario Principal**

- Desde la pantalla principal, el usuario accede al menú facturación
- El usuario elige el submenú correspondiente.
- El sistema abre una nueva pantalla correspondiente al submenú de facturación con las funcionalidades correspondientes a su rol(alta,baja,consulta,modificación).

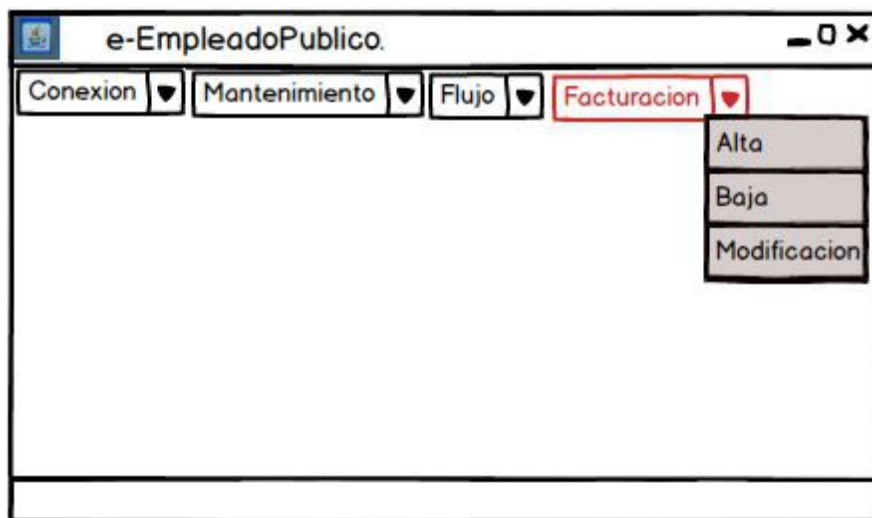


Figura 15.- Menú Facturación

2.3.2.4.-Subsistema de Flujo.-

- Se encargará de gestionar la actividad (alta, baja, consulta/modificación correspondiente de la realización de peticiones,respuestas, permutas, y actividad documental protocolaria procedente de la asesoría.

✓ **Peticiones y respuestas**

✓ **Permutas**

✓ **Respuestas jurídicas** (informes, documentos formales, recursos)

- Requisitos

✓ **Precondición.-** El usuario que accede al sistema dispone de un código de usuario y contraseña válidos y la condición de usuario personal administrativo o administrador.

✓**Postcondición.**-El usuario accede a la aplicación con su perfil una vez introducidas sus credenciales en el sistema y a las funcionalidades que le ofrece este subsistema.

- Prestaciones.-Como funcionalidades básicas del subsistema de Flujo se han identificado las siguientes:

✓**Escenario Principal**

- Desde la pantalla principal, el usuario accede al menú del subsistema de Flujo.
- El usuario elige el submenú correspondiente.
- El sistema abre una nueva pantalla correspondiente al submenú del subsistema de flujo con las funcionalidades correspondientes a su rol.

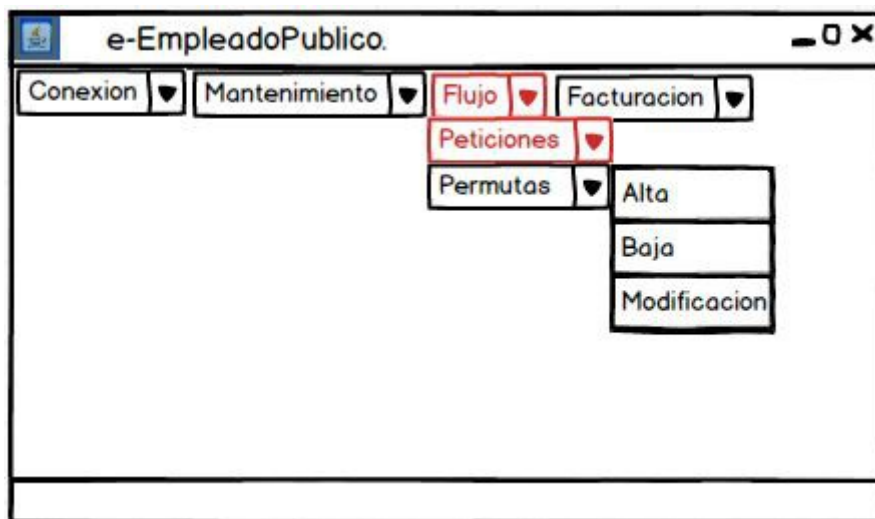


Figura 16.- Menú Flujo

2.4.-Gestores

Hasta ahora hemos visto una panorámica general de las funcionalidades de cada uno de 4 subsistemas en que se ha de estructurar la aplicación. A partir de aquí, esas funcionalidades, de cada uno de ellos, se organizan en una serie de gestores específicos cuyo diagrama se muestra a continuación.

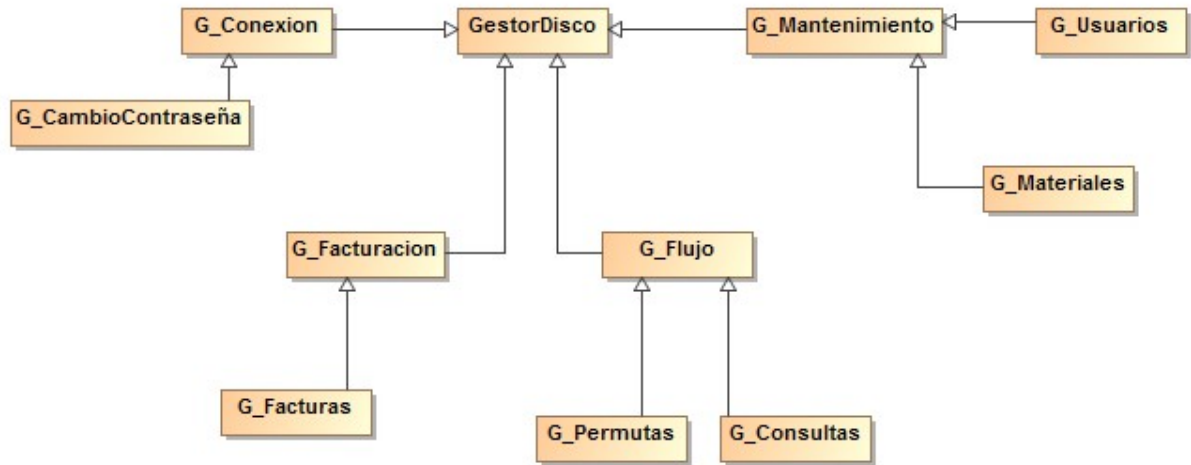
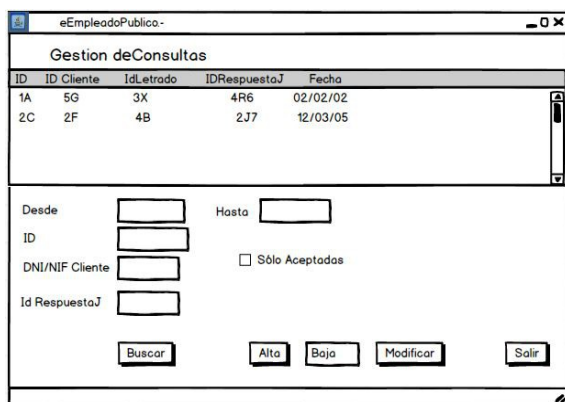


Figura 17.- Diagrama de gestores

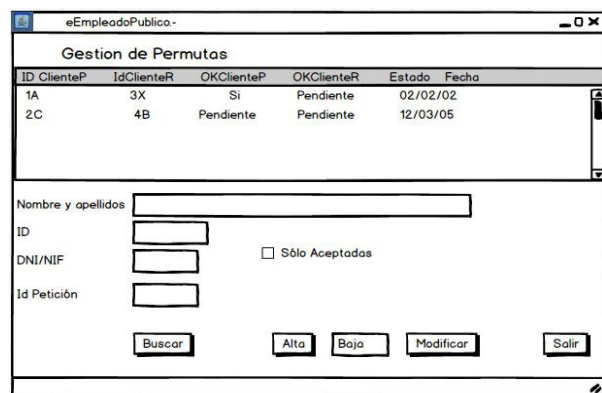
Veamos las pantallas de cada uno de ellos, que como se podrá observar será accedido mediante la elección del menú correspondiente (mantenimiento, facturación, conexión, flujo.), a través de sus opciones de alta, baja o modificación. En cualquier caso se nos abrirá una pantalla con un espacio para albergar un “output” con ocurrencias del concepto que se trate organizadas en una lista y una serie de campos editables que podrán ser utilizados tanto para buscar, como para incorporar un nuevo registro, así como para eliminar o modificar alguno ya existente.

El sistema pedirá confirmación en caso de que la operación no genere error o excepción. En caso contrario informará del impedimento en cuestión.



ID	ID Cliente	IdLetrado	IDRespuestaJ	Fecha
1A	5G	3X	4R6	02/02/02
2C	2F	4B	2J7	12/03/05

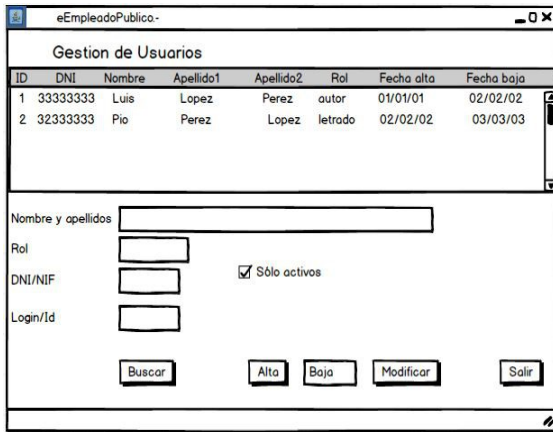
Desde: Hasta:
 ID:
 DNI/NIF Cliente: Sólo Aceptadas
 Id RespuestaJ:
 Botones:



ID ClienteP	IdClienteR	OKClienteP	OKClienteR	Estado	Fecha
1A	3X	Si	Pendiente	02/02/02	
2C	4B	Pendiente	Pendiente	12/03/05	

Nombre y apellidos:
 ID:
 DNI/NIF: Sólo Aceptadas
 Id Petición:
 Botones:

figuras 17 y 18. Gestores de Consultas y Permutas



ID	DNI	Nombre	Apellido1	Apellido2	Rol	Fecha alta	Fecha baja
1	33333333	Luis	Lopez	Perez	autor	01/01/01	02/02/02
2	32333333	Pio	Perez	Lopez	letrado	02/02/02	03/03/03

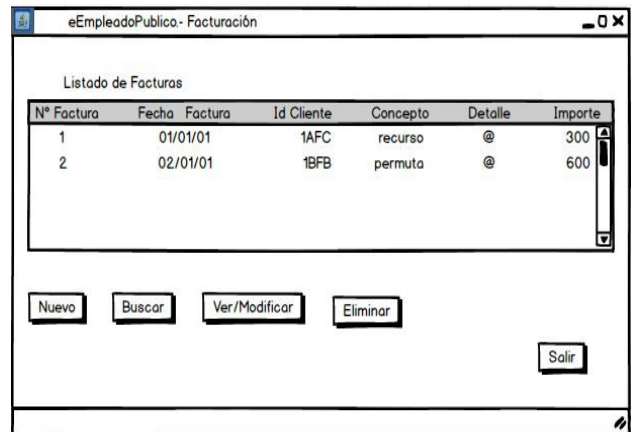
Nombre y apellidos:

Rol:

DNI/NIF: Sólo activos

Login/Id:

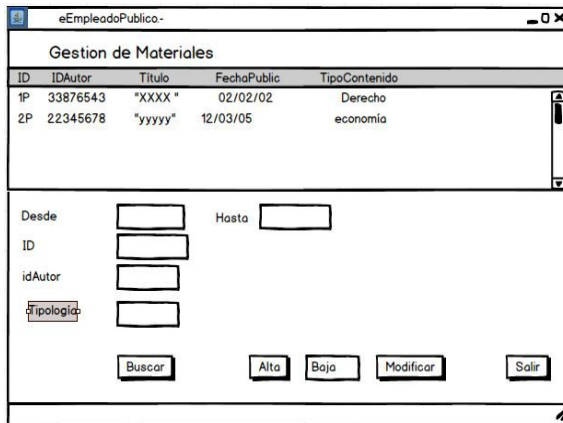
Buttons:



Nº Factura	Fecha	Factura	Id Cliente	Concepto	Detalle	Importe
1	01/01/01	1AFC	recurso	@	300	
2	02/01/01	1BFB	permuta	@	600	

Buttons:

figuras 20 y 21. Gestores de Usuarios y Facturación



ID	IDAutor	Título	FechaPublic	TipoContenido
1P	33876543	"XXXX "	02/02/02	Derecho
2P	22345678	"yyyyy"	12/03/05	economia

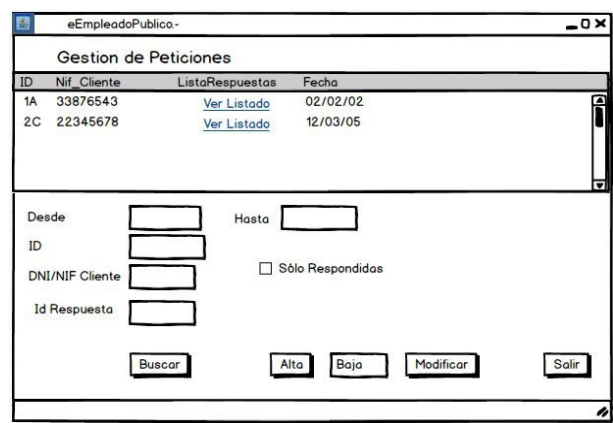
Desde: Hasta:

ID:

idAutor:

Tipología:

Buttons:



ID	Nif_Cliente	ListaRespuestas	Fecha
1A	33876543	Ver Listado	02/02/02
2C	22345678	Ver Listado	12/03/05

Desde: Hasta:

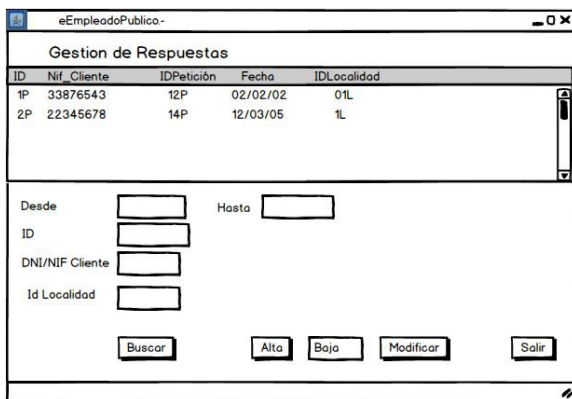
ID:

DNI/NIF Cliente: Sólo Respondidas

Id Respuesta:

Buttons:

figuras 21 y 22. Gestores de Materiales y Peticiones



ID	Nif_Cliente	IDPetición	Fecha	IDLocalidad
1P	33876543	12P	02/02/02	01L
2P	22345678	14P	12/03/05	1L

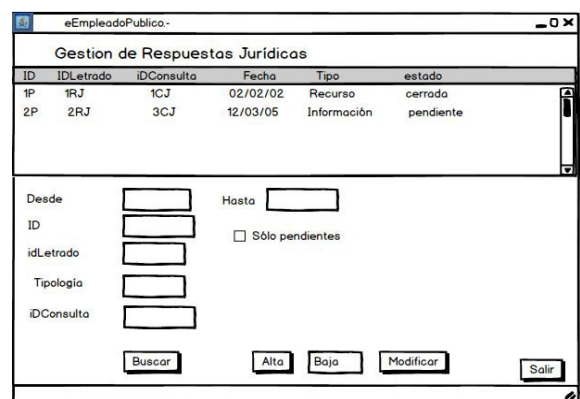
Desde: Hasta:

ID:

DNI/NIF Cliente:

Id Localidad:

Buttons:



ID	IDLetrado	IDConsulta	Fecha	Tipo	estado
1P	1RJ	1CJ	02/02/02	Recurso	cerrada
2P	2RJ	3CJ	12/03/05	Información	pendiente

Desde: Hasta:

ID:

idLetrado:

Tipología:

IDConsulta:

Sólo pendientes

Buttons:

figuras 23 y 24. Gestores de respuestas y Respuestas Jurídicas

2 2.5.-Funcionalidades específicas

Una vez hemos accedido a cualquiera de los gestores enunciados anteriormente y tras elegir cualquiera de las opciones se nos aparecen (alta, baja, eliminación, búsqueda o consulta) se nos aparecerá la pantalla correspondiente. A continuación se ofrece la estructura de pantallas del sistema y el prototipo de una serie de ellas, comprensivo y representativo de cualquiera de dichas opciones:

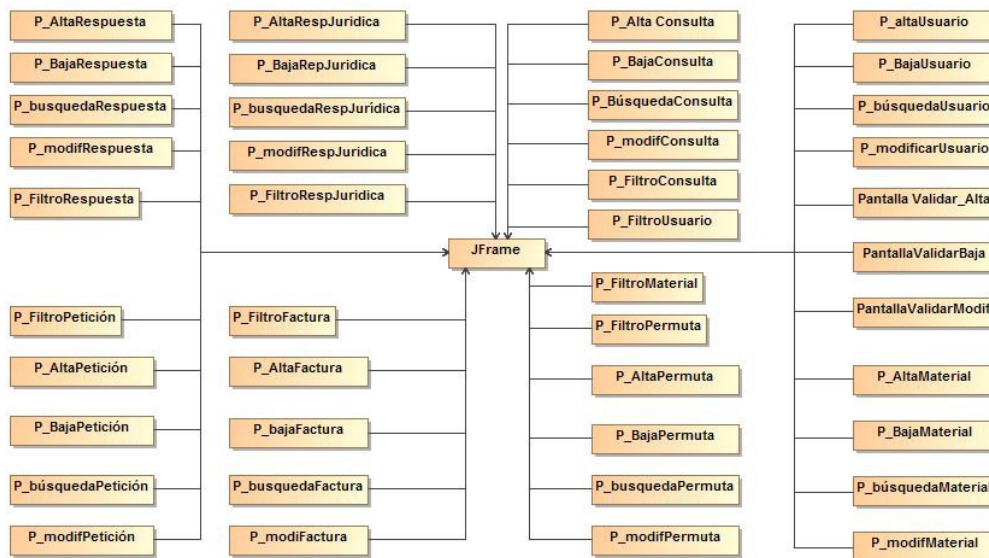
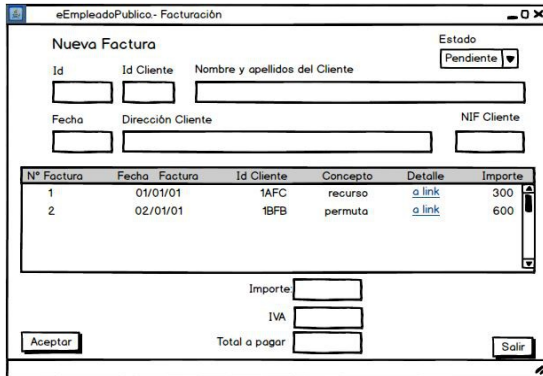
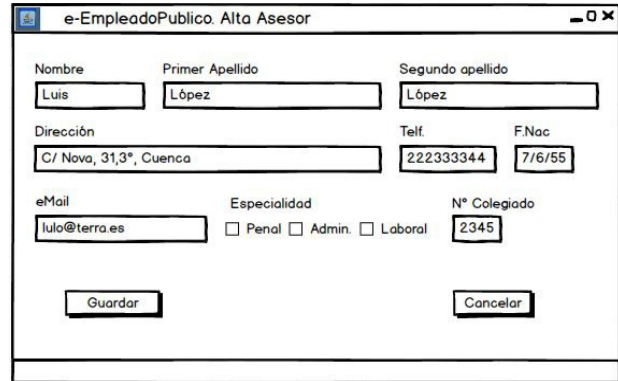


figura 25. Diagrama de Pantallas

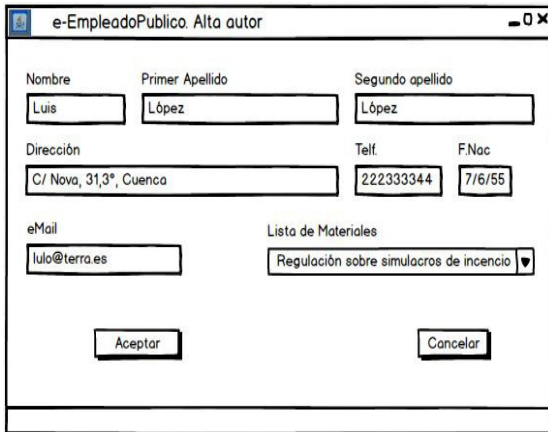
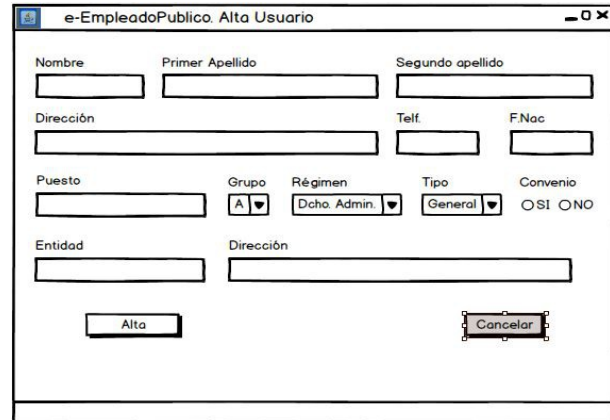
En el caso de las búsquedas o consultas bastará cubrir todos o alguno de los campos editables en cualquiera de las pantallas gestoras para que, seleccionando la ocurrencia deseada el sistema nos ofrezca todos los datos referentes a él. A partir de aquí, podremos modificar alguno y guardarlo o iniciar una nueva búsqueda, con lo cual volveríamos a la pantalla gestora precedente.



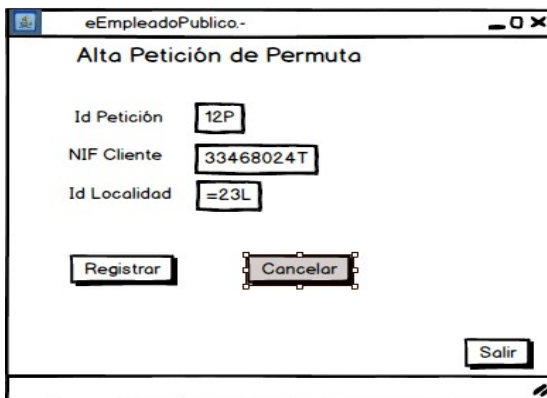
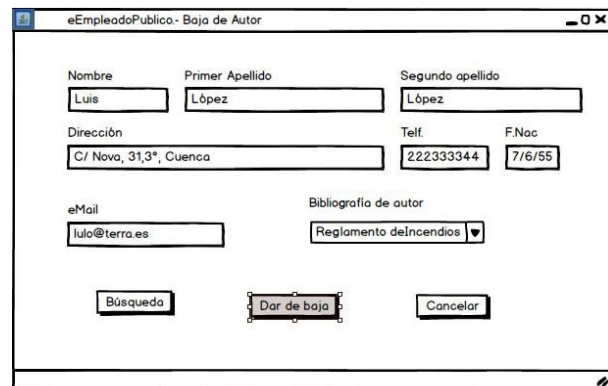
N° Factura	Fecha	Factura	Id Cliente	Concepto	Detalle	Importe
1	01/01/01	1AFC		recurso	a link	300
2	02/01/01	1BFB		permuta	a link	600



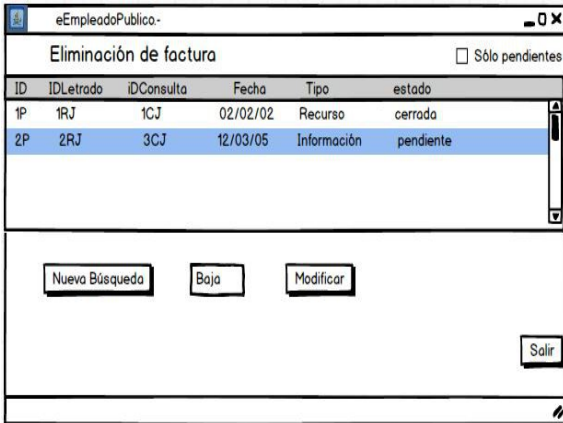
figuras 27 y 28. Altas de factura y Asesor

figuras 29 y 30. Altas de Autor y Cliente

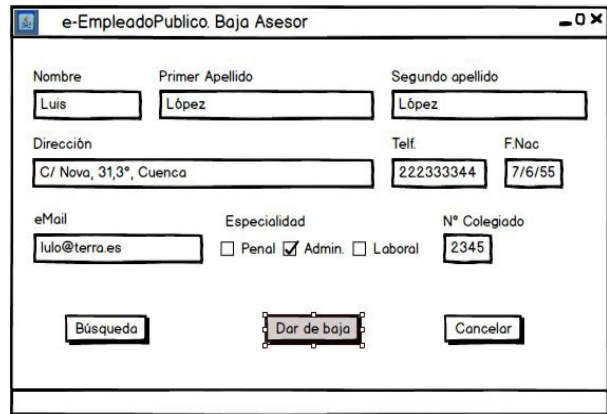



figuras 31 y 32. Alta de Permuta y baja de Autor



ID	IDLetrado	IDConsulta	Fecha	Tipo	estado
1P	1RJ	1CJ	02/02/02	Recurso	cerrada
2P	2RJ	3CJ	12/03/05	Información	pendiente

Buttons: Nueva Búsqueda, Baja, Modificar, Salir



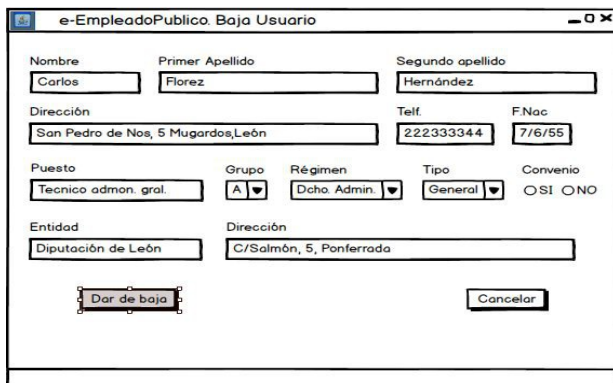
Nombre: Luis, Primer Apellido: López, Segundo apellido: López

Dirección: C/ Nova, 31,3°, Cuenca, Telf: 222333344, FNac: 7/6/55

eMail: lulo@terra.es, Especialidad: Penal Admin. Laboral, N° Colegiado: 2345

Buttons: Búsqueda, Dar de baja, Cancelar

figuras 33 y 34. Bajas de Factura y Asesor



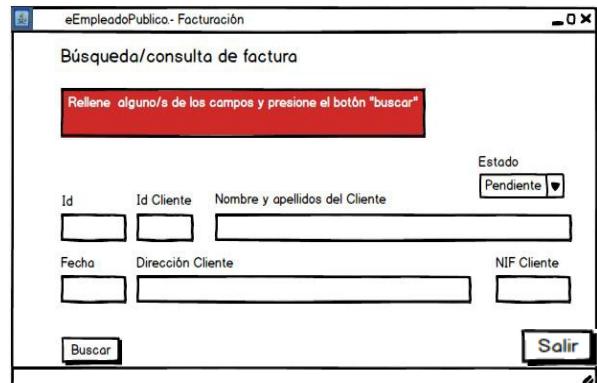
Nombre: Carlos, Primer Apellido: Florez, Segundo apellido: Hernández

Dirección: San Pedro de Nos, 5 Mugarodos, León, Telf: 222333344, FNac: 7/6/55

Puesto: Tecnico admon. gral., Grupo: A, Régimen: Dicho. Admin., Tipo: General, Convenio: SI NO

Entidad: Diputación de León, Dirección: C/Salmón, 5, Ponferrada

Buttons: Dar de baja, Cancelar



Búsqueda/consulta de factura

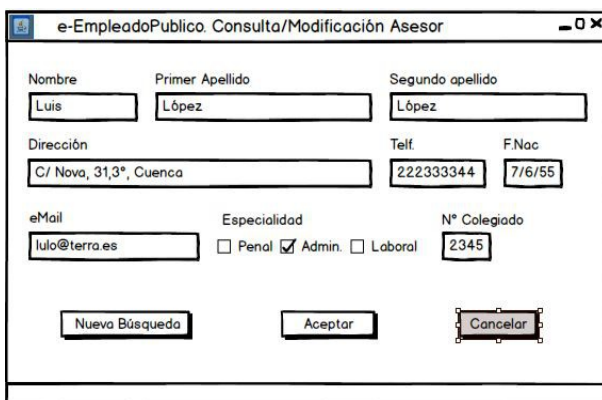
Rellene alguno/s de los campos y presione el botón "buscar"

Id, Id Cliente, Nombre y apellidos del Cliente, Estado: Pendiente

Fecha, Dirección Cliente, NIF Cliente

Buttons: Buscar, Salir

figuras 35 y 36 Bajas de Usuario y Consulta/modif. de Factura

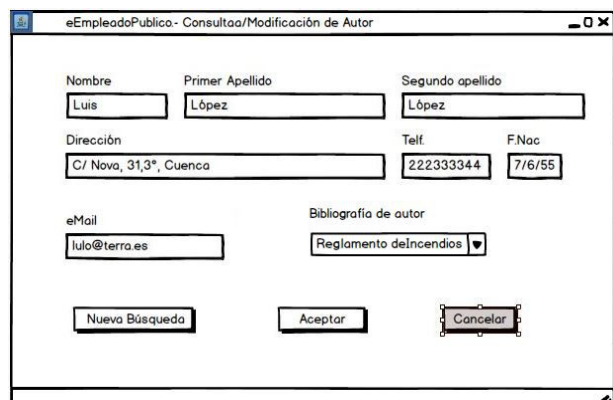


Nombre: Luis, Primer Apellido: López, Segundo apellido: López

Dirección: C/ Nova, 31,3°, Cuenca, Telf: 222333344, FNac: 7/6/55

eMail: lulo@terra.es, Especialidad: Penal Admin. Laboral, N° Colegiado: 2345

Buttons: Nueva Búsqueda, Aceptar, Cancelar



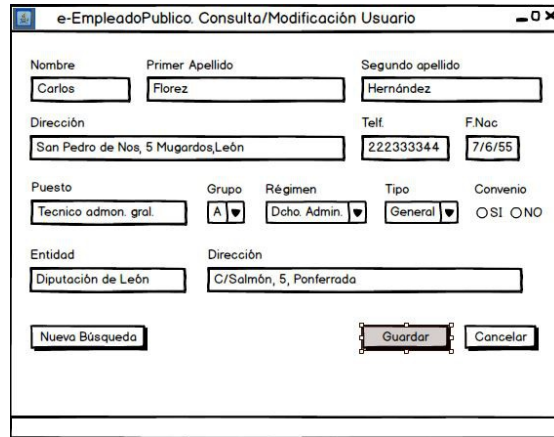
Nombre: Luis, Primer Apellido: López, Segundo apellido: López

Dirección: C/ Nova, 31,3°, Cuenca, Telf: 222333344, FNac: 7/6/55

eMail: lulo@terra.es, Bibliografía de autor: Reglamento de Incendios

Buttons: Nueva Búsqueda, Aceptar, Cancelar

figuras 37 y 38. Consulta/modif. de Asesor y de Autor

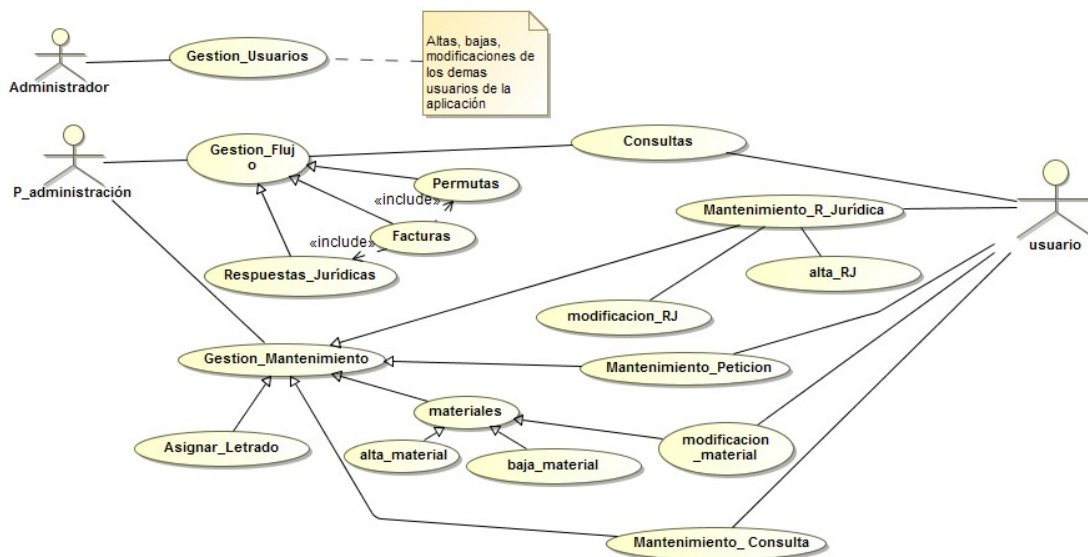


figuras 39. Consulta/modif. de Cliente

2.6.-Especificación textual. Casos de Uso

A continuación se ofrece un diagrama general de casos de uso con implicación de autores y una relación amplia y representativa de ellos

2.6.1.- Diagrama general



figuras 40. Diagrama general de casos de uso

2.6.2.- Descripción textual. Casos de uso

En este apartado se muestra una relación exhaustiva (salvo error u omisión) de los casos de uso del sistema con la intención de que se ponga de manifiesto el funcionamiento de este. Esta es su terminología:

Función. Describe brevemente la función del caso de uso.

Actores . Indica los actores que pueden realizar el caso de uso.

Relaciones. Enumera los casos de uso relacionados.

Pre. Indica la situación antes de que se realice la acción.

Secuencia. Explica los pasos que debe realizar el actor.

Pos condición. Indica la situación una vez realizada la acción.

Excepciones. Indica, si las hay, las excepciones que se pueden producir.

Anexo. Incluye, alguna explicación o aclaración adicional.

Caso de uso_1		Añadir autor	
Función	Incorpora un nuevo autor al sistema		
Relaciones	Búsqueda de Autor		
Actor/es	Administrador		
Pre	El autor no figura en la BB.DD del sistema		
Secuencia	<ul style="list-style-type: none"> • El propietario introduce los datos que le pide el sistema del nuevo autor. • Introducirá los datos correspondientes a los atributos relacionados en la ficha CRC correspondiente a <i>Autor</i>. • El sistema generará un código de usuario una vez introducidos todos los datos. • Por último, se guardará toda la información introducida en el sistema. 		
Post	El autor queda incorporado a la BB.DD		
Excepciones	Si el autor ya existía en la BB.DD, el sistema evitará insertarlo nuevamente y lanzará un aviso.		
Anexo			

Caso de uso_2		Añadir letrado
Función	Incorpora un nuevo letrado al sistema	
Relaciones	Búsqueda de letrado	
Actor/es	Administrador	
Pre	El letrado no figura en la BB.DD del sistema	
Secuencia	<ul style="list-style-type: none"> • El propietario introduce los datos que le pide el sistema del nuevo letrado. • Introducirá los datos correspondientes a los atributos relacionados en la ficha CRC correspondiente a letrado. • El sistema generará un código de usuario una vez introducidos todos los datos. • Por último, se guardará toda la información introducida en el sistema. 	
Post	El letrado queda incorporado a la BB.DD	
Excepciones	Si el letrado ya existía en la BB.DD, el sistema evitará insertarlo nuevamente y lanzará un aviso.	
Anexo		

Caso de uso_3		Añadir cliente
Función	Incorpora un nuevo cliente al sistema	
Relaciones	Búsqueda de cliente	
Actor/es	Administrador	
Pre	El cliente no figura en la BB.DD del sistema	
Secuencia	<ul style="list-style-type: none"> • El propietario introduce los datos que le pide el sistema del nuevo cliente. • Introducirá los datos correspondientes a los atributos relacionados en la ficha CRC correspondiente a <i>cliente</i>. • El sistema generará un código de usuario una vez introducidos todos los datos. • Por último, se guardará toda la información introducida en el sistema. 	
Post	El cliente queda incorporado a la BB.DD	
Excepciones	Si el cliente ya existía en la BB.DD, el sistema evitará insertarlo nuevamente y lanzará un aviso.	
Anexo		

Caso de uso_4 Buscar Autor	
Función	Busca un autor dentro de la BB.DD
Relaciones	Alta, baja, modificación/consulta de autor
Actor/es	Administrador, p_administrativo
Pre	El autor figura en la BB.DD
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato del autor que desea localizar.
Post	El sistema muestra los datos por pantalla
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido.
Anexo	

Caso de uso_5 Buscar letrado	
Función	Busca un letrado dentro de la BB.DD
Relaciones	Alta, baja, modificación/consulta de letrado
Actor/es	Administrador, p_administrativo
Pre	El letrado figura en la BB.DD
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato del letrado que desea localizar.
Post	El sistema muestra dos datos por pantalla
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido.
Anexo	

Caso de uso_6 Buscar cliente	
Función	Busca un cliente dentro de la BB.DD
Relaciones	Alta, baja, modificación/consulta de cliente
Actor/es	Administrador, p_administrativo

Pre	El cliente figura en la BB.DD
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato del cliente que desea localizar.
Post	El sistema muestra datos or pantalla
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido.
Anexo	

Caso de uso_7 Modificar Autor	
Función	Permite modificar los datos de un autor
Relaciones	Alta, baja, modificación/consulta de autor
Actor/es	Administrador
Pre	El autor existe en la BB.DD
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato del autor que desea localizar. El sistema mostrará los datos existentes El actor modificará los datos deseados El sistema guardará los cambios
Post	Los datos del autor están modificados en la base de datos.
Excepciones	Si el sistema no encuentra al autor o los datos introducidos no tienen el formato adecuado, lanzará un aviso
Anexo	

Caso de uso_8 Modificar letrado	
Función	Permite modificar los datos de un letrado
Relaciones	Alta, baja, modificación/consulta de letrado
Actor/es	Administrador
Pre	El letrado existe en la BB.DD
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato del letrado que desea localizar. El sistema mostrará los datos existentes

	<ul style="list-style-type: none"> • El actor modificará los datos deseados • El sistema guardará los cambios
Post	Los datos del letrado están modificados en la base de datos.
Excepciones	Si el sistema no encuentra al letrado o los datos introducidos no tienen el formato adecuado, lanzará un aviso
Anexo	

Caso de uso_9 Modificar cliente	
Función	Permite modificar los datos de un cliente
Relaciones	Alta, baja, modificación/consulta de cliente
Actor/es	Administrador
Pre	El cliente existe en la BB.DD
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato del cliente que desea localizar. • El sistema mostrará los datos existentes • El actor modificará los datos deseados • El sistema guardará los cambios
Post	Los datos del cliente están modificados en la base de datos.
Excepciones	Si el sistema no encuentra al cliente o los datos introducidos no tienen el formato adecuado, lanzará un aviso
Anexo	

Caso de uso_10 Eliminar Autor	
Función	Elimina del sistema a un autor.
Relaciones	Búsqueda autor

Actor/es	Administrador
Pre	El autor ya existe en la base de datos.
Secuencia	<ul style="list-style-type: none"> • El propietario introduce en el sistema algún dato del autor que desea localizar. • El sistema le mostrará todos los datos existentes en la base de datos referente al autor buscado. • El propietario ordenará al sistema la eliminación del autor de la base de datos. • Por último, el sistema hará un borrado lógico de toda la información referida a dicho autor de la base de datos.
Post	El autor está eliminado en la base de datos.
Excepciones	Si el autor NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_11 Eliminar letrado	
Función	Elimina del sistema a un letrado.
Relaciones	Búsqueda letrado
Actor/es	Administrador
Pre	El letrado ya existe en la base de datos.
Secuencia	<ul style="list-style-type: none"> • El propietario introduce en el sistema algún dato del letrado que desea localizar. • El sistema le mostrará todos los datos existentes en la base de datos referente al letrado buscado. • El propietario ordenará al sistema la eliminación del letrado de la base de datos. • Por último, el sistema hará un borrado lógico de toda la información referida a dicho letrado de la base de datos.
Post	El letrado está eliminado en la base de datos.
Excepciones	Si el letrado NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_12 Eliminar cliente	
Función	Elimina del sistema a un cliente.
Relaciones	Búsqueda cliente
Actor/es	Administrador
Pre	El cliente ya existe en la base de datos.
Secuencia	<ul style="list-style-type: none"> • El propietario introduce en el sistema algún dato del cliente que desea localizar. • El sistema le mostrará todos los datos existentes en la base de datos referente al cliente buscado. • El propietario ordenará al sistema la eliminación del cliente de la base de datos. • Por último, el sistema hará un borrado lógico de toda la información referida a dicho cliente de la base de datos.
Post	El cliente está eliminado en la base de datos.
Excepciones	Si el cliente NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_13 Alta de Factura	
Función	Incorpora una nueva factura al sistema
Relaciones	Permutas, Respuesta_Jurídica
Actor/es	P_administración
Pre	Existe un nueva permuta o un nuevo recurso o documento
Secuencia	<ul style="list-style-type: none"> • El personal de administración hace una búsqueda de nuevas permutas o actos jurídicos de letrados. • El sistema devolverá aquellas permutas o actos jurídicos que no estén vinculados a factura. • El operario introducirá los datos que le pide el sistema. Este generará automáticamente una nueva factura con un código. • Una vez aceptada, la factura quedará guardada en sistema

Post	La factura queda incorporada
Excepciones	No se cumplimentan alguno/os campos obligatorios
Anexo	

Caso de uso_14 Buscar Factura	
Función	Busca una factura dentro de la BB.DD
Relaciones	Alta, baja, modificación/consulta de factura
Actor/es	p_administrativo.
Pre	La factura existe en el sistema
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato de la factura que desea localizar.
Post	El sistema muestra por pantalla los datos relativos a la factura
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido.
Anexo	

Caso de uso_15 Modificar Factura	
Función	Modifica algún/o dato/s de una factura. .
Relaciones	Búsqueda factura
Actor/es	Personal Administración
Pre	La factura existe en la BB.DD.
Secuencia	<ul style="list-style-type: none"> El operario introduce en el sistema algún dato de la factura que desea localizar. El sistema le mostrará todos los datos existentes en la base de datos referente a la factura buscada. El operario modificará los campos activos que desea. Tras confirmar, los cambios quedan registrados en el sistema
Post	Los datos modificados están en la BB.DD.
Excepciones	Si la factura NO existe en la base de datos, o los datos introducidos no tienen el formato correcto el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una

	nueva búsqueda.
Anexo	Una factura solo se puede modificar cuando no está cerrada

Caso de uso_16 Eliminar Factura	
Función	Elimina del sistema a una factura.
Relaciones	Búsqueda factura
Actor/es	Personal Administración
Pre	La factura existe en la BB.DD.
Secuencia	<ul style="list-style-type: none"> • El operario introduce en el sistema algún dato de la factura que desea localizar. • El sistema le mostrará todos los datos existentes en la base de datos referente a la factura buscada. • El operario ordenará al sistema la eliminación de la factura de la base de datos. • Por último, el sistema hará un borrado lógico de toda la información referida a dicha factura de la base de datos.
Post	La factura será eliminada en la base de datos.
Excepciones	Si la factura NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	Una factura solo se puede eliminar cuando no está cerrada

Caso de uso_17 Alta de Petición	
Función	Incorpora una nueva Petición al sistema
Relaciones	Respuesta
Actor/es	Usuario Cliente, Respuesta
Pre	El usuario está incorporado al sistema
Secuencia	<ul style="list-style-type: none"> • El cliente introduce los campos requeridos • Introducirá los datos correspondientes a los atributos relacionados en la ficha CRC

	<p>correspondiente a <i>Petición</i>.</p> <ul style="list-style-type: none"> • El sistema generará un código de Petición una vez introducidos todos los datos. • Por último, se guardará toda la información introducida en el sistema.
Post	Existe una nueva petición en el sistema
Excepciones	No se cumplimentan alguno/os campos obligatorios
Anexo	

Caso de uso_18 Buscar Petición	
Función	Busca una petición dentro de la BB.DD
Relaciones	Alta, baja, modificación/consulta de petición
Actor/es	p_administrativo., cliente
Pre	La petición existe en el sistema
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato de la petición que desea localizar.
Post	El sistema muestra por pantalla los datos relativos a la petición
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido o no existe la petición.
Anexo	

Caso de uso_19 Modificar Petición	
Función	Modifica algún/o dato/s de una petición. .
Relaciones	Búsqueda petición
Actor/es	Cliente
Pre	La petición existe en la BB.DD. Está activa
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato de la petición que desea localizar. • El actor le mostrará todos los datos existentes en la base de datos referente a la petición buscada. • El actor modificará los campos activos que desea. • Tras confirmar, los cambios quedan registrados en el sistema

Post	Los datos modificados están en la BB.DD.
Excepciones	Si la petición NO existe en la base de datos, o los datos introducidos no tienen el formato correcto el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_20 Eliminar Petición	
Función	Elimina del sistema una petición.
Relaciones	Búsqueda Pétición
Actor/es	Personal Administración, cliente
Pre	La petición existe en la BB.DD. Y está activa
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato de la petición que desea localizar. • El sistema le mostrará todos los datos existentes en la base de datos referente a la petición buscada. • El actor ordenará al sistema la eliminación de la petición de la base de datos. • Por último, el sistema hará un borrado lógico de toda la información referida a dicha petición de la base de datos.
Post	La petición será eliminada en la base de datos.
Excepciones	Si la petición NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_21 Alta de Respuesta	
Función	Incorpora una nueva Petición al sistema
Relaciones	Cliente, Respuesta, Petición
Actor/es	Cliente
Pre	Existe una Petición, el cliente está registrado

Secuencia	<ul style="list-style-type: none"> • El cliente hace una búsqueda de peticiones. • Selecciona una petición activa y presiona el botón “responder” • Introduce los campos requeridos y pulsa aceptar. • El sistema genera un código para la nueva respuesta. • Por último, se guardará toda la información introducida en el sistema.
Post	Existe una nueva respuesta.
Excepciones	No se cumplimentan alguno/os campos obligatorios
Anexo	Una petición es activa si no ha concluido en permuta o no ha sido cancelada

Caso de uso_22 Modificar Respuesta	
Función	Modifica algún/o dato/s de una Respuesta
Relaciones	Búsqueda respuesta
Actor/es	letrado
Pre	La respuesta existe en la BB.DD.
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato de la respuesta que desea localizar. • El actor le mostrará todos los datos existentes en la base de datos referente a la respuesta buscada. • El actor modificará los campos activos que desea. • Tras confirmar, los cambios quedan registrados en el sistema
Post	Los datos modificados están en la BB.DD.
Excepciones	Si la respuesta NO existe en la base de datos, o los datos introducidos no tienen el formato correcto el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_23 Buscar Respuesta	
Función	Busca una respuesta dentro de la BB.DD

Relaciones	Alta, baja, modificación/consulta de respuesta
Actor/es	p_administrativo. cliente
Pre	La respuesta existe en el sistema
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato de la respuesta que desea localizar.
Post	El sistema muestra por pantalla los datos relativos a la respuesta
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido o no existe la respuesta.
Anexo	

Caso de uso_24 Eliminar Respuesta	
Función	Elimina del sistema una respuesta a una petición de permuta.
Relaciones	Búsqueda Respuesta.
Actor/es	cliente
Pre	Existe una respuesta por parte del cliente a una petición activa de permuta.
Secuencia	<ul style="list-style-type: none"> El actor introduce en el sistema algún dato de la respuesta que desea localizar. El sistema le mostrará todos los datos existentes en la base de datos referente a la petición buscada. El actor ordenará al sistema la eliminación de la respuesta de la base de datos. Por último, el sistema hará un borrado lógico de toda la información referida a dicha respuesta de la base de datos.
Post	La respuesta será eliminada en la base de datos.
Excepciones	Si la respuesta NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_25		Alta de Respuesta Jurídica
Función	Incorpora una nueva Respuesta Jurídica al sistema	
Relaciones	Cliente, Consulta, Permuta	
Actor/es	Letrado	
Pre	Existe una Consulta o una petición respondida y consentida por los clientes, el letrado está registrado y asignado.	
Secuencia	<ul style="list-style-type: none"> • El letrado cumplimenta los campos requeridos y genera el contenido documental correspondiente • Pulsa aceptar. • El sistema genera un código para la nueva respuesta. • Por último, se guardará toda la información introducida en el sistema. 	
Post	Existe una nueva respuesta jurídica.	
Excepciones	No se cumplimentan alguno/os campos obligatorios	
Anexo	La respuesta puede ser un informe, un recurso, o consistir en un documento formal(solicitando la anuencia de las administraciones en caso de permutas consensuadas por los clientes)	

Caso de uso_26		Buscar Respuesta Jurídica
Función	Busca una respuesta jurídica dentro de la BB.DD	
Relaciones	Alta, baja, modificación/consulta de respuesta jurídica	
Actor/es	p_administrativo., letrado	
Pre	La respuesta jurídica existe en el sistema	
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato de la respuesta jurídica que desea localizar. 	
Post	El sistema muestra por pantalla los datos relativos a la respuesta jurídica	
Excepciones	El sistema lanza un aviso si el filtro no tiene el formato requerido o no existe la respuesta buscada.	
Anexo		

Caso de uso_27		Modificar Respuesta Jurídica
Función	Modifica algún/o dato/s de una Respuesta Jurídica	
Relaciones	Búsqueda respuesta jurídica	

Actor/es	letrado
Pre	La respuesta jurídica existe en la BB.DD.
Secuencia	<ul style="list-style-type: none"> • El actor introduce en el sistema algún dato de la respuesta jurídica que desea localizar. • El actor le mostrará todos los datos existentes en la base de datos referente a la respuesta jurídica buscada. • El actor modificará los campos activos que desea. • Tras confirmar, los cambios quedan registrados en el sistema
Post	Los datos modificados están en la BB.DD.
Excepciones	Si la respuesta jurídica NO existe en la base de datos, o los datos introducidos no tienen el formato correcto el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	

Caso de uso_28 Eliminar Respuesta Jurídica	
Función	Elimina del sistema a una repuesta jurídica.
Relaciones	Búsqueda repuesta jurídica , factura
Actor/es	Personal Administración
Pre	La repuesta jurídica existe en la BB.DD. Y está vinculada a una factura.
Secuencia	<ul style="list-style-type: none"> • El operario introduce en el sistema algún dato de la repuesta jurídica que desea localizar. • El sistema le mostrará todos los datos existentes en la base de datos referente a la repuesta jurídica buscada. • El operario ordenará al sistema la eliminación de la repuesta jurídica de la base de datos. • Por último, el sistema hará un borrado lógico de toda la información referida a dicha repuesta jurídica de la base de datos.
Post	La repuesta jurídica será eliminada en la base de datos.
Excepciones	Si la repuesta jurídica NO existe en la base de datos, el sistema mostrará un mensaje de error y volverá a la ventana inicial para poder realizar una nueva búsqueda.
Anexo	El sistema eliminará la factura vinculada.

3.-Diseño Técnico.

En este capítulo se recoge toda la documentación relativa a la fase de diseño del sistema y que habrá de ser tenido en cuenta a la hora de realizar la implementación. En esta fase no nos centramos tanto en que queremos realizar, sino en como lo vamos a realizar, buscando la solución que satisfaga todos los requerimientos del sistema.

Partiendo de la especificación de los requisitos del capítulo anterior, para el diseño vamos a realizar las siguientes tareas:

- Seleccionar una muestra representativa de clases y objetos usando los requisitos básicos como guías (fichas CRC)
- Identificar atributos y operaciones para cada objeto del sistema (fichas CRC)
- Definir estructuras y jerarquías que organicen las clases (diagramas de clases)
- Construir un modelo objeto-comportamiento (diagramas de actividad y diagramas de colaboración y/o secuencia)
- Construir el modelo conceptual de los datos que serán persistentes (diagramas de persistencia)

3.1.-Fichas CRC

<i>PERSONA</i>	
Descripción:	Atributos y métodos de la clase persona
Tipo:	Principal
Características:	Abstracta, compuesta
Responsabilidades:	Modelar los datos de una persona
Colaboraciones:	DNI
Constructores:	+persona()

PERSONA	
	+persona(DNI: DNI) +persona(DNI: DNI, nombre: String, apellidos: String, dirección: String, email: String, telefono: String, fechaNac: Date, sexo: char)
Atributos:	// DNI de la persona -DNI: DNI // Nombre de la persona -nombre: String // Apellidos de la persona -apellidos: String // Dirección de la persona -dirección: String // email de la persona -email: String // Telefono de la persona -telefono: String // Fecha de nacimiento de la persona -fechaNac: Date // Sexo de la persona -sexo: char
Métodos	+getDNI(): DNI +setDNI: (DNI: DNI) +getNombre(): String +setNombre: (nombre: String) +getApellidos(): String +setApellidos: (apellidos: String) +getDirección(): String +setDirección: (dirección: String) +getEmail(): String +setEmail: (email: String) +getTelefono(): String +setTelefono: (telefono: String) +getFechaNac(): Date +setFechaNac: (fechaNac: Date) +getSexo(): char + setSexo: (sexo: char)

DNI	
Descripción:	Identificación exclusiva de la persona
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades :	Validar la individualidad existencial de la persona
Colaboraciones:	Persona
Constructores:	+DNI() +DNI(dni: String)

DNI	
Atributos:	// DNI -DNI: String
Métodos	+ getDNI(): String + setDNI: (DNI: String) + toString(): String + validar: (boolean)

Usuario	
Descripción:	Representa y engloba a personas con acceso a la aplicación
Tipo:	Herederas de Persona
Características:	Abstracta
Responsabilidades :	Interviene en la creación alta, baja, modificación y consulta de usuarios
Colaboraciones:	excepción, pantalla
Constructores:	+usuario(dni: DNI, usuario: String, contraseña: String) +usuario(DNI: DNI, usuario: String, contraseña: String, nombre: String, apellidos: String)
Atributos:	//Identificador único del usuario -IDUsuario: String //Perfil del usuario -perfil:(Perfil:perfil) // Contraseña exclusiva de acceso para Usuario -psswd: String
Métodos	+ getIDUsuario(): String +setIDUsuario: (Usuario: String) + getPerfil(): Perfil + setPerfil: Centro + getPasswd(): String + setPasswd: (String)

Administrador	
Descripción:	Modeliza un usuario con acceso ilimitado a la aplicación
Tipo:	Herederas de Usuario
Características:	Concreta, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación y consulta del resto de usuarios
Colaboraciones:	excepción, pantalla
Constructores:	+usuario(dni: DNI, usuario: String, contraseña: String)

Administrador	
	+usuario(DNI: DNI, usuario: String, contraseña: String, nombre: String, apellidos: String)

PersonalSecretaría	
Descripción:	Modeliza un usuario encargado de gestionar las actividades de tipo administrativo
Tipo:	Herederas de Usuario
Características:	Concreta, persistente
Responsabilidades:	Interviene en la gestión de actividades
Colaboraciones:	excepción, pantalla, actividad
Constructores:	+usuario(dni: DNI, usuario: String, contraseña: String) +usuario(DNI: DNI, usuario: String, contraseña: String, nombre: String, apellidos: String)

RespuestaJuridica	
Descripción:	Generaliza los servicios que los letrados pueden prestar a los clientes
Tipo:	principal
Características:	abstracta, compuesta
Responsabilidades:	Modelar los datos de una respuesta jurídica
Colaboraciones:	Consulta
Constructores:	+Universidad(dni: DNI, usuario: String, contraseña: String) +Universidad(DNI: DNI, usuario: String, contraseña: String, nombre: String, apellidos: String)
Atributos:	-idRespuestaJ:String -idConsulta -fecha:Date -descripción:String -contenido:String
Métodos	+ getIdRespuestaJ(): String + setIdRespuestaJ: (contenido: String) + getIdConsulta(): String + setIdConsulta: (id: String) + getFecha(): Date + setFecha: (date: date) + getDescripción(): String

RespuestaJuridica	
	+ setDescription: (descripción: String) + getContenido(): String + setContenido: (contenido:String)

Autor	
Descripción:	Representa y modeliza los autores de contenidos
Tipo:	Herederas de usuario
Características:	Concreta, persistente
Responsabilidades:	Alta, baja, modificación y consulta de un autor
Colaboraciones:	Material
Constructores:	+Autor() +Autor(fechaAlta:Date, fechaBaja: Date, contenidos:ArrayList)
Atributos:	//Listado de materiales +ListaMateriales<Material>:ArrayList // fecha de alta del autor en el sistema -fechaAlta: Date // fecha de baja del autor en el sistema -fechaBaja: Date
Métodos	+ getFechaAlta(): Date + getFechaBaja():Date + setFechaAlta(date:Date) + setFechaBaja: (date:Date)

Letrado	
Descripción:	Representa y modeliza los letrados asesores
Tipo:	Herederas de usuario
Características:	Concreta, persistente
Responsabilidades:	Alta, baja, modificación y consulta de un letrado
Colaboraciones:	Consulta
Constructores:	+Letrado() +Letrado(fechaAlta:Date, fechaBaja: Date, especialidad:Especialidad, Consultas:ArrayList<Consulta>)
Atributos:	//Listado de consultas por letrado //ListaConsultas<ArrayList>:Consulta // especialidad del letrado -especialidad:Especialidad //fecha alta del letrado -fechaAlta(): Date //fecha de baja del Letrado -fechaBaja():Date

Letrado	
Métodos	+ getFechaAlta(): Date + getFechaBaja():Date + setFechaAlta(date:Date) + setFechaBaja: (date:Date) +listaDeConsultas

Cliente	
Descripción:	Representa y modeliza los usuarios clientes
Tipo:	Hereda de usuario
Características:	Concreta, persistente
Responsabilidades:	Alta, baja, modificacion y consulta de clientes
Colaboraciones:	Petición consulta,cliente(reflexiva)
Constructores:	+Cliente(administración: String, convenio: Boolean, localidadTrabajo: String, tipoEnte:Ente, puesto:String, grupo:Grupo, cEspecifico:Int, cProductividad:Int, horario:String, gratificacion:Int, cSociales:List<cSociales>)
Atributos:	<pre>//ListaPersonal<ArrayList>:Usuario // nombre de la administración del cliente administración: String // existe convenio específico? -convenio:boolean // localidad del puesto de trabajo -localidadTrabajo: int //tipo de administración tipoEnte: Ente //denominación del puesto puesto:String //grupo de pertenencia grupo:Grupo //montante del Complemento Específico cEspecifico:Int //montante del Complemento de productividad cProductividad:Int //horario del cliente -horario: String //importe de gratificación si lo hay -gratificación:int //complementos ayudasSociales:List<cSociales></pre>
Métodos	+ getdmon(): String + setAdmon: (NIF: NIF) + getConvenio(): boolean + setConvenio: (nombre: String) + getLocalidaT(): String + setLocalidadT: (email: String) + getEnte(): String

Cliente	
	+ setEnte: (ente: Ente) + getPuesto(): String + setPuesto: (direccion: String) + getGrupo(): String + setGrupo: (grupo:Grupo) + getEspecifico(): String + setEspecifico: (codPos: String) + getProductividad(): String + setProductividad: (Usuario: String) + getHorario(): String + setHorario: (Centro) + getGratificacion(): String + setGratificacion: (String)

ConsultaJ	
Descripción:	Modeliza una consulta de un cliente a los letrados
Tipo:	Asociativa
Características:	Concreta, persistente
Responsabilidades:	Interviene en la búsqueda alta, baja, modificación de las consultas de los clientes a los asesores
Colaboraciones:	excepción, pantalla
Constructores:	+consultaJ(idConsulta:String, idCliente: String, idLetrado: String, fecha:Date, descripción:String,idRespuestaJ)
Atributos:	//identificador del cliente -idCliente:String //id de el letrado encargado -idLetrado:String //identificador único de la consulta -idConsulta:(String)

Permuta	
Descripción:	Describe el intercambio de puestod de trabajo entre 2 empleados públicos
Tipo:	Principal
Características:	Asociativa
Responsabilidades:	Interviene en la modelización de una permuta entre dos clientes

Permuta	
Colaboraciones:	excepción, pantalla, validación, petición, respuesta
Constructores:	+permuta(idClienteR:String, idClienteR, okClienteP:boolean, okClienteR: boolean fecha:Date, idPetición:String, idRespuesta:String)
Atributos	//identificador del cliente peticionario -idClienteR //identificador del cliente interesado en la petición -idClienteR //conformidad del cliente peticionario -okClienteP:Boolean //conformidad del 2º cliente interesado -okClienteR:Boolean //fecha -:fecha:Date //id petición --IdPetición:string //id respuesta -id respuesta.
Métodos	“getters y setters de los atributos anteriores

Informe	
Descripción:	Subclase de RespuestaJ que modeliza documentos generados por los letrados en respuesta a diversas cuestiones o dudas planteadas por los clientes
Tipo:	Herederas de RespuestaJ
Características:	Concreta, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación y consulta de informes letrados a petición de los clientes
Colaboraciones:	excepción, pantalla
Constructores:	+usuario(dni: DNI, usuario: String, contraseña: String) +usuario(DNI: DNI, usuario: String, contraseña: String, nombre: String, apellidos: String)
Atributos:	//Identificador único del usuario -IDUsuario: String //Perfil del usuario -perfil:(Perfil:perfil) // Contraseña exclusiva de acceso para Usuario -psswd: String
Métodos	+ getIDUsuario(): String + setIDUsuario: (Usuario: String) + getPerfil(): Perfil + setPerfil: Centro

Informe	
	+ getPasswd(): String + setPasswd: (String)

DocumentoFormal	
Descripción:	Modeliza las Respuestas Jurídicas generadas por los asesores plasmadas en documentos dirigidos a organismos oficiales en forma de quejas, solicitudes o puestas en conocimiento de aquellos
Tipo:	Hereda de Respuesta Jurídica
Características:	Concreta, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación y consulta de documentos dirigidos a organismos oficiales
Colaboraciones:	Permuta, factura
Constructores:	+documentoF() +documentoF(IdDocumentoF:String,tipo:TipoDocumentoF,Contenido:String, coste:String)
Atributos:	//Identificador del documento -IDDoc: String //tipo de documento -tipo:(tipo:TipoDocumentoF) // Contenido -contenido: String //coste -coste:Float
Métodos	+ getID(): String +setID: (id: String) + getTipo():TipoDocumentoF + setTipo: tipo:TipoDocumentoF) + getContenido(): String +setContenido: (contenido:String) +getCoste():Float +setCoste:(coste:Float)

Recurso	
Descripción:	Modeliza documentos generados por los letrados, contenedores de recursos formales ante la Admón de justicia o las Administraciones Públicas y que

Recurso	
	forman parte de un determinado procedimiento reglado
Tipo:	Herederas de Respuesta Jurídica
Características:	Concreta, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación y consulta de los referidos documentos descritos
Colaboraciones:	excepción, pantalla, factura.
Constructores:	+usuario(dni: DNI, usuario: String, contraseña: String) +usuario(DNI: DNI, usuario: String, contraseña: String, nombre: String, apellidos: String)
Atributos:	//Identificador único del usuario -IDUsuario: String //Perfil del usuario -perfil:(Perfil:perfil) // Contraseña exclusiva de acceso para Usuario -passwd: String
Métodos	+ getIDUsuario(): String +setIDUsuario: (Usuario: String) + getPerfil(): Perfil + setPerfil: Centro + getPasswd(): String + setPasswd: (String)

Factura	
Descripción:	Modela la entidad factura correspondiente a un servicio no gratuito prestado a un cliente
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación y consulta de facturas
Colaboraciones:	excepción, pantalla, permuta, documento, personal_administrativo
Constructores:	+factura(fecha:Date, idFactura: String, idRespuestaJ:String , importe:Float, abonada:boolean, idEmisor:String)
Atributos:	//Identificador único de la factura -idFactura: String // identificador del documento motivador -IDRespuesta: String //identificador del administrativo creador idEmisor:String //montante total

Factura	
	idImporte:Float //fecha de emisión fecha:Date
Métodos	+ getIDFactura(): String +setIDFactura:(idFactura: String) + getIdRespuesta: String + setIdRespuesta:(idRespuesta:String + getIdEmisor(): String +setIdEmisor(idEmisor:String) +getImporte():String +setImporte:(importe:Float)

Petición	
Descripción:	Plasma el interés de un cliente por hacer una permuta
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación y consulta de peticiones de permutas
Colaboraciones:	excepción, pantalla, cliente, respuesta
Constructores:	+petición() +petición(idPetición:String, idCliente:String; respuestas:<respuesta>:list, fecha:Date)
Atributos:	//Identificador único del cliente -IDCliente: String //id de la petición idPetición:String // listado de respuestas listaRespuestas:(ArrayList<respuesta>) //fecha de la petición -feccha:Date
Métodos	+ getIDCliente(): String +setIDUsuario: (Usuario: String) + getFecha: Date + setPerfil: Centro + getIdCliente(): String

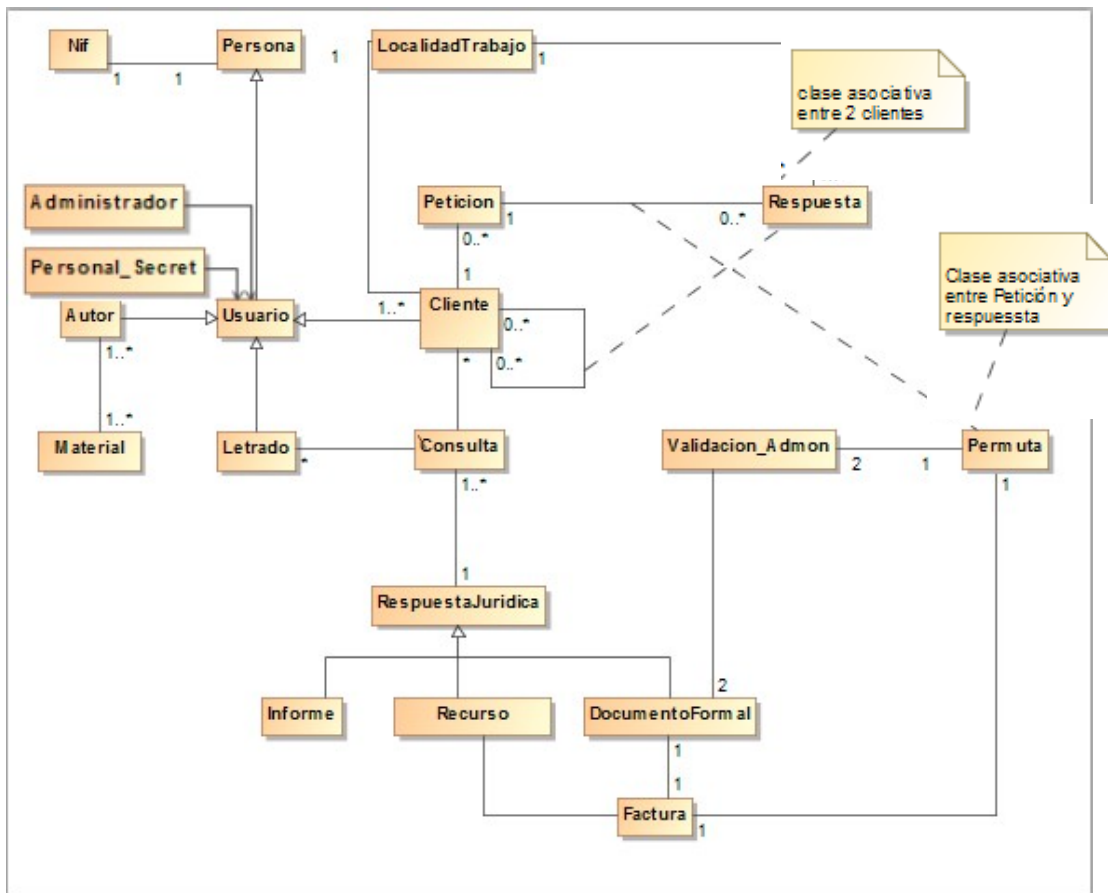
Respuesta	
Descripción:	Modeliza la respuesta de un cliente interesado a la petición de permuta de otro cliente
Tipo:	Asociativa
Características:	principal, persistente
Responsabilidades:	Interviene en la creación alta, baja, modificación de respuestas a peticiones de permuta
Colaboraciones:	excepción, pantalla
Constructores:	+respuesta(idRespuesta:String,idPeticion:String,fecha:Date,) +respuesta(idRespuesta:String,idPeticion:String,fecha:Date, localidad:LocalidadTrabajo)
Atributos:	//Identificador único del usuario -IDUsuario: String //Perfil del usuario -perfil:(Perfil:perfil) // Contraseña exclusiva de acceso para Usuario -psswd: String
Métodos	+ getIDUsuario(): String +setIDUsuario: (Usuario: String) + getPerfil(): Perfil + setPerfil: Centro + getPasswd(): String + setPasswd: (String)

Localidad_Trabajo	
Descripción:	Modela la entidad Localidad como atributo de un cliente
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Interviene en la asignación de una localidad de trabajo actual a un cliente.
Colaboraciones:	excepción, pantalla
Constructores:	+localidadT(id:String, localidad:String, provincia:String ,dirección:String, municipio:String, CP:integer)
Atributos:	//Identificador único del usuario -IdLocalidad: String //localidad del cliente -localidad:String //provincia del cliente -provincia:String // Código postal -CP:Int // municipio del trabajo del cliente -municipio:String

Localidad_Trabajo	
Métodos	+ getIDLocalidad(): String + setIDLocalidad: (idlocalidad: String) + getlocalidad(): String + setLocalidad: (localidad:String) + getProvincia(): String + setProvincia: (provincia:String)

3.2.-Diagramas de clases

En el siguiente diagrama podemos observar las clases principales que integran la aplicación , sus relaciones y la cardinalidad de éstas:



figuras 41. Diagrama general de clases

Este mismo diagrama se trocea en 2 partes para mostrar las clases con sus atributos y las clases “enumeradas” relacionadas:

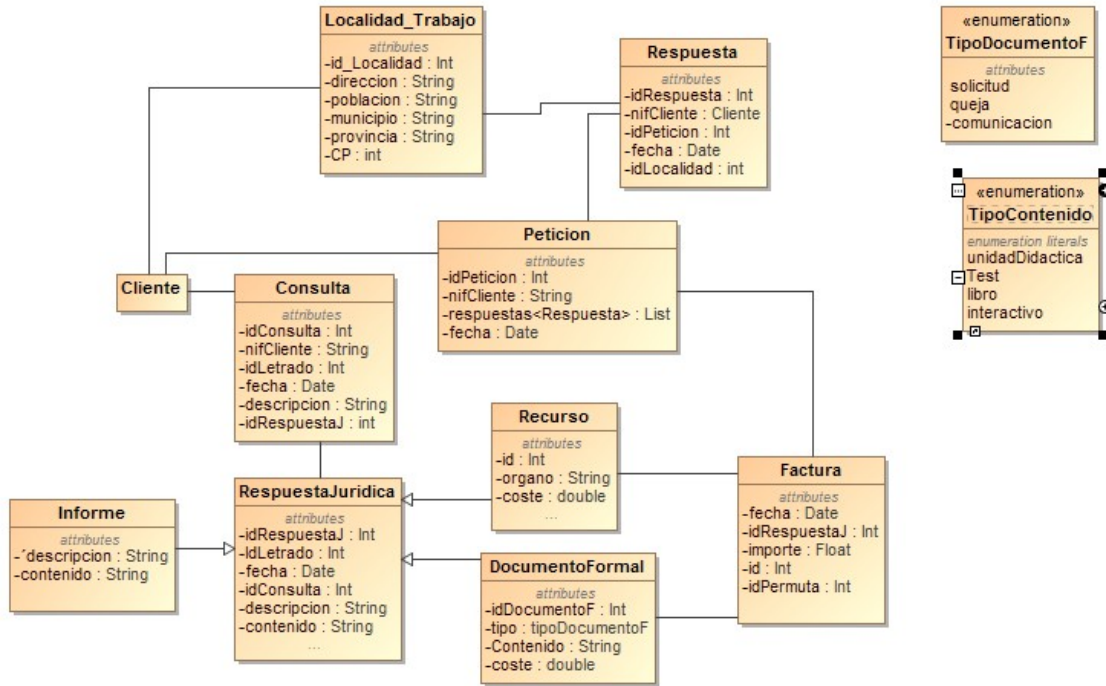


figura 42.-a Diagrama detallado de clases

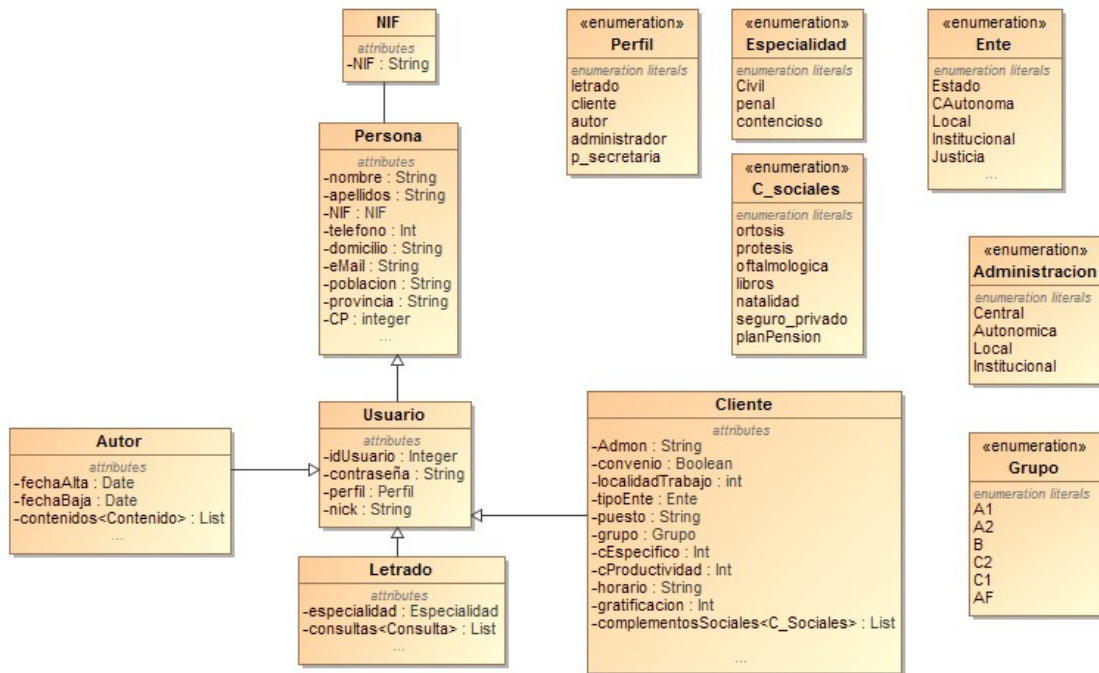
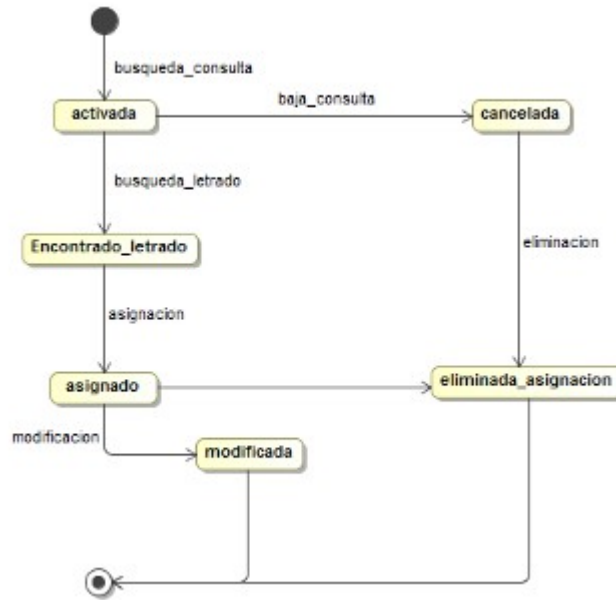


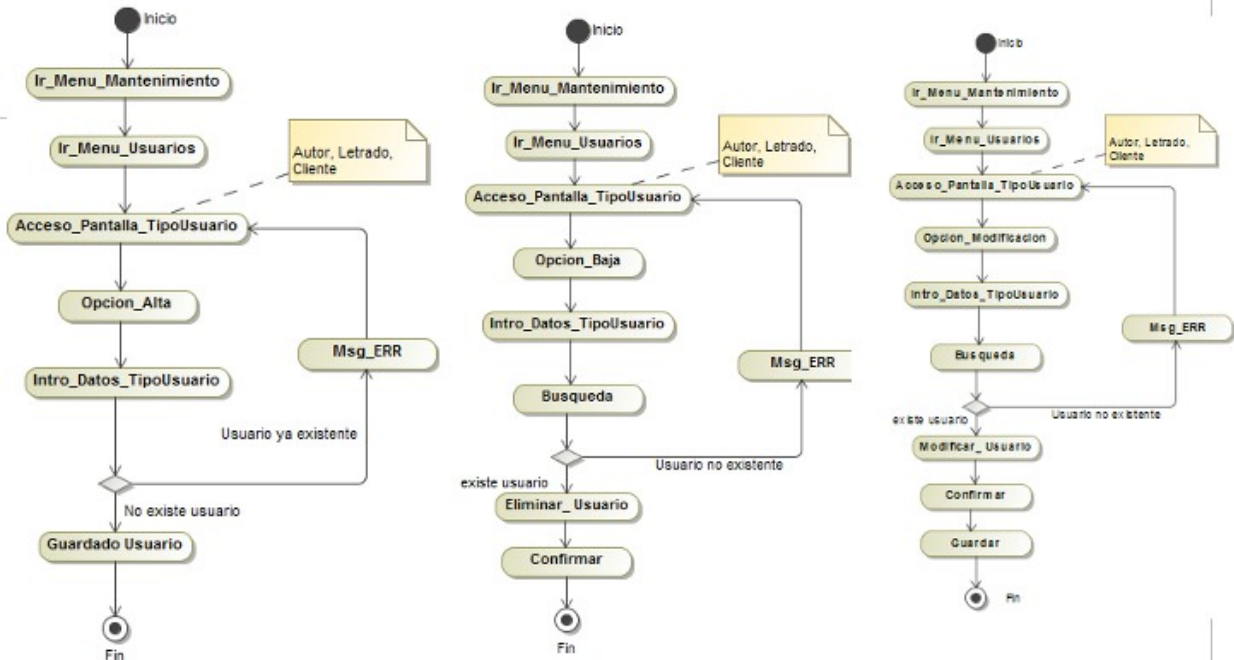
figura 42.-b Diagrama detallado de clases

3.3.-modelo de comportamiento

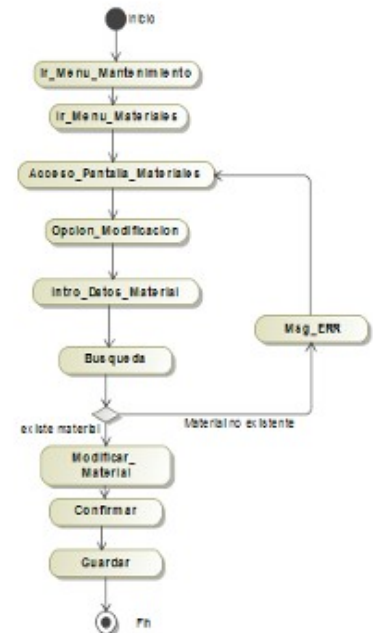
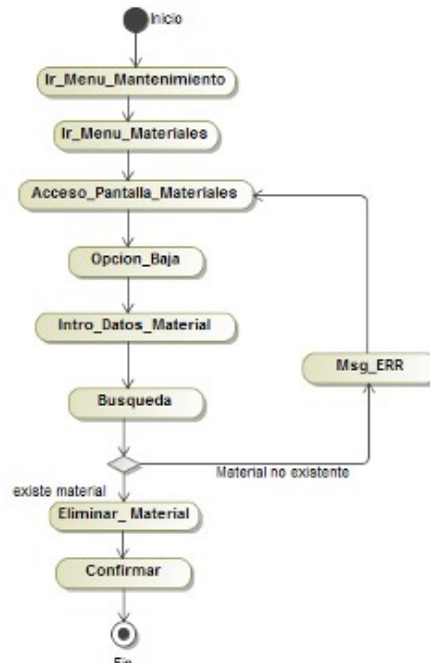
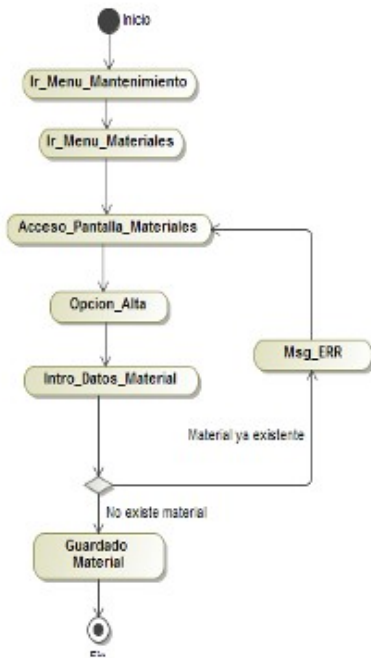
3.3.1.- Diagramas de actividad



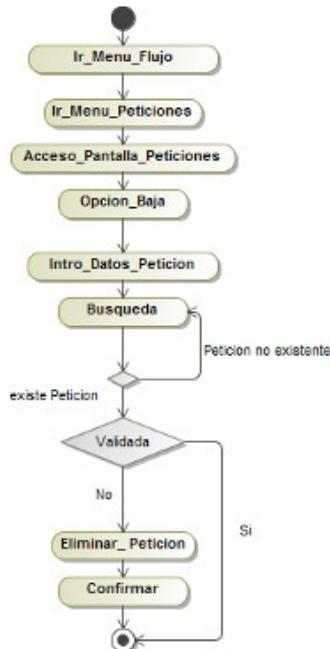
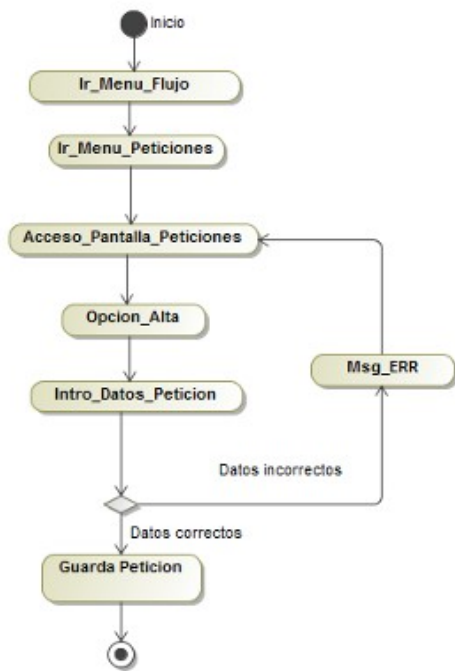
Asignación de letrado (fig. 43)



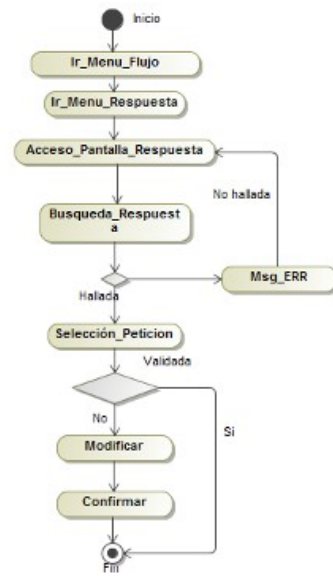
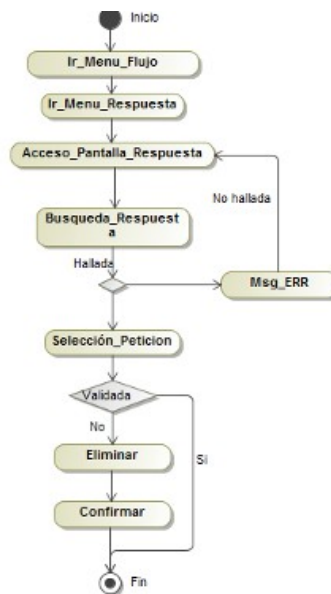
Alta, baja y modificación de Usuario (figs 44,45,46)



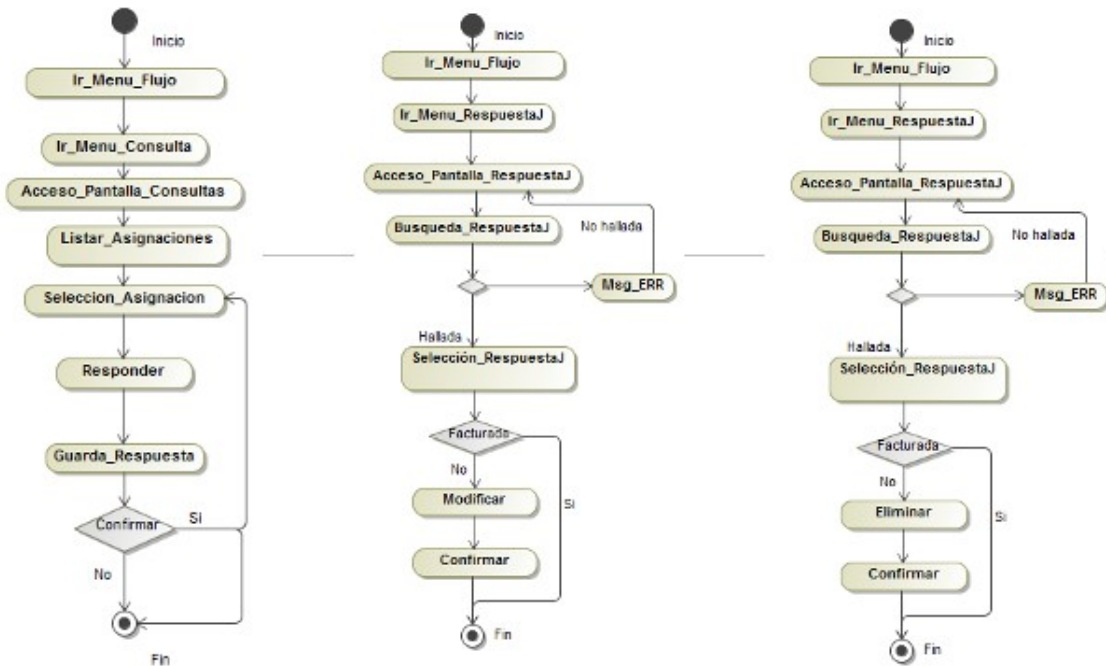
Alta baja y modificación de material (figs 47,48,49)



Alta baja y modificación de petición.(figs 50,51,52)



Alta baja y modificación de Respuesta(figs 53,54,55)



Alta baja y modificación de Respuesta Jurídica.(figs 56,57,58)

3.3.2.-Diagramas de secuencia y colaboración.

A continuación se muestran 3 casos que se han considerado representativos y frecuentes de la actividad de los usuarios interactuando con el sistema, referidos a un alta de usuario, una modificación de petición y, por supuesto, la conexión al sistema de cualquier usuario.

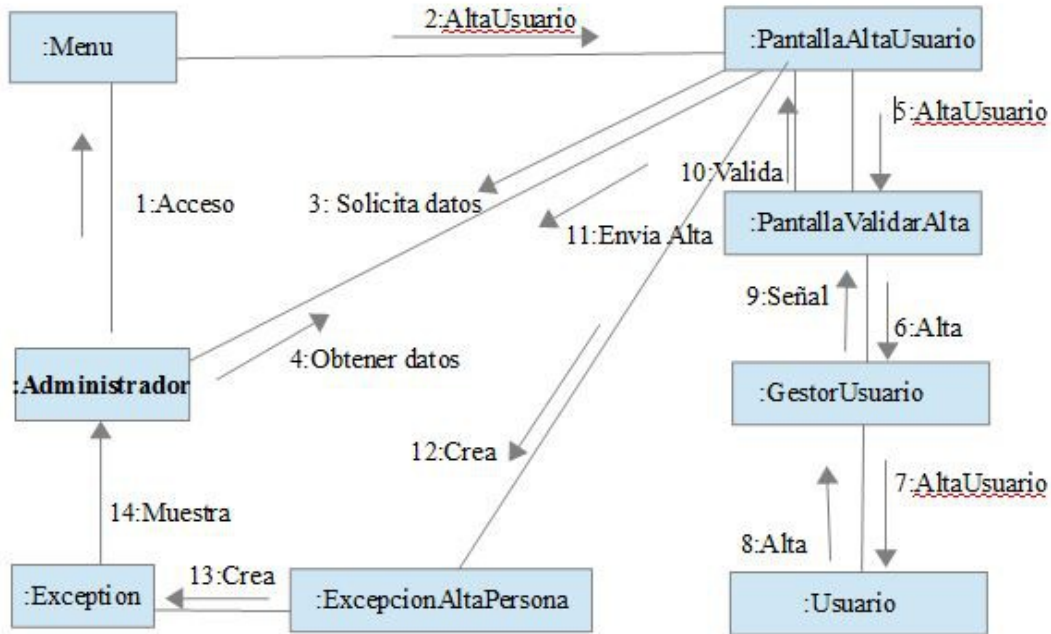


fig 59.- Alta de Usuario

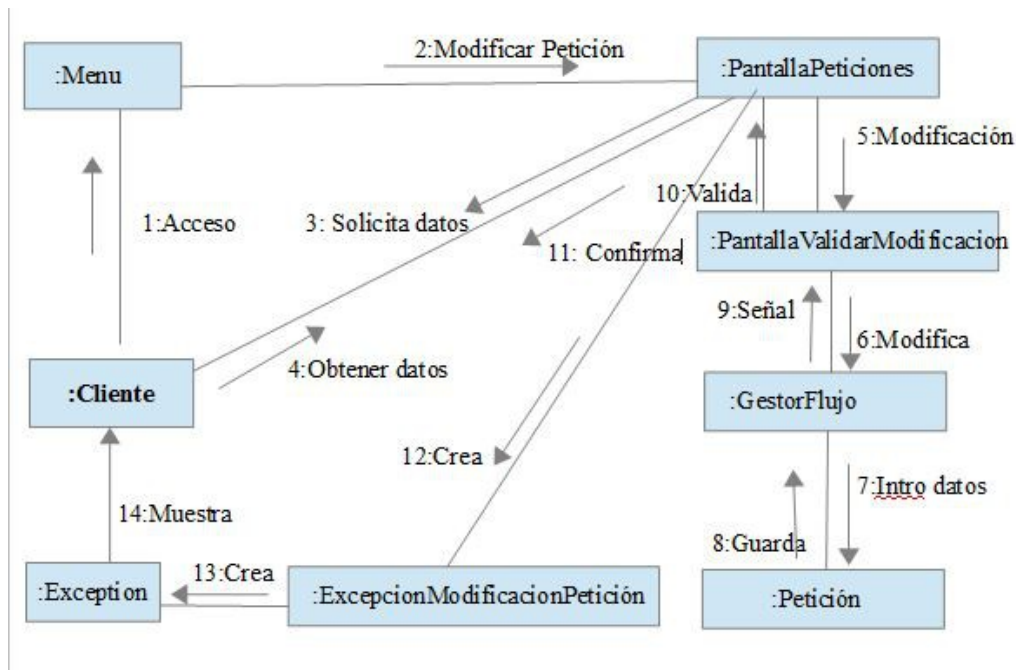


fig. 60 Modificación de petición

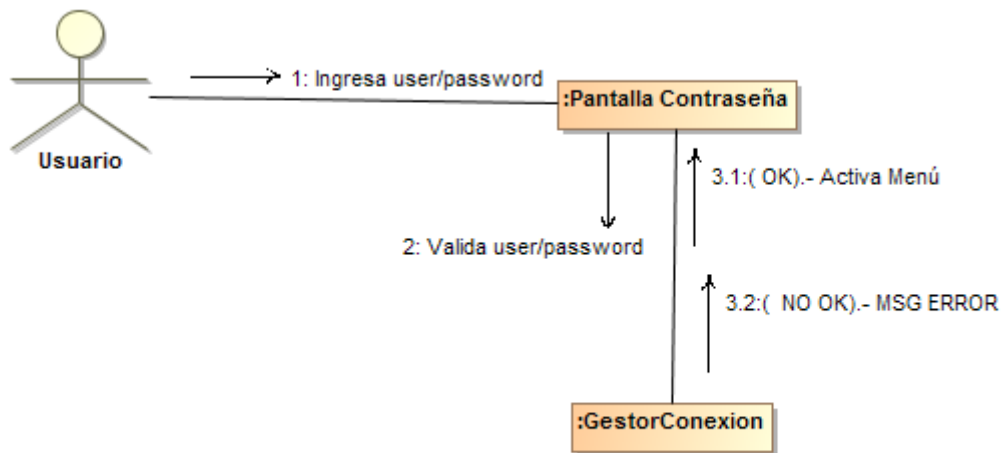


fig 61.- Conexión de un usuario

3.4.-Persistencia

3.4.1.-Modelo E-R

En el siguiente gráfico se muestra el diagrama E-R

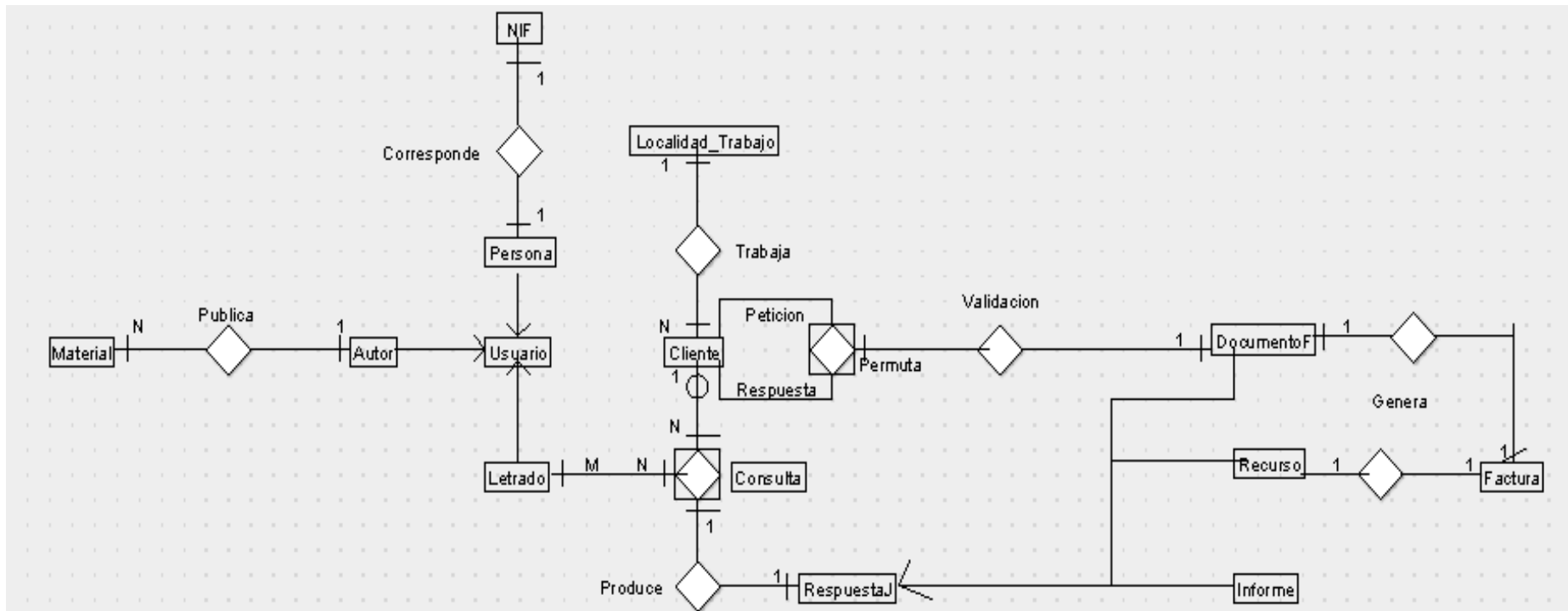


fig 62. Diagrama E-R

En el gráfico apreciamos las diferentes entidades y las asociaciones que se establecen entre ellas. En el diagrama hay tres entidades llamadas “PERSONA”, “USUARIO” y “RESPUESTA JURIDICA”. La clase “Usuario” que se especializa en “Autor”, “Cliente” y “Letrado” es, a su vez una especialización de la clase “Persona”. Por su parte, las clases “Informe”, “Recurso” y “DocumentoF” son especializaciones de la clase “RespuestaJ”. Las referidas especializaciones son Disjuntas y Totales, es decir, no puede haber ocurrencias de una superclase que pertenezcan a dos subclases a la vez y, también, que cualquier ocurrencia de una superclase tiene que pertenecer obligatoriamente a una de sus subclases.

Existe dos entidades asociativas “Consulta ” y “Permuta” que controlan las interacciones entre letrados y clientes y el proceso de concreción de peticiones de permuta y respuestas.

A partir de aquí se relacionan las entidades existentes en el diagrama anterior, con expresión de atributos, subrayando aquellos que son clave primaria (PK):

NIF

nif.

Persona

nif, nombre, apellidos, domicilio, mail, teléfono, población, provincia, CP.

Usuario

nif, idUsuario, contraseña, nick, perfil.

Cliente

nif, idCliente, admón, convenio, localidadT, tipoEnte, puesto, grupo, cEspecífico, cProductividad, horario, gratificación.

Letrado

nif, idLetrado, especialidad.

Autor

nif, idAutor, fechaAlta, fechaBaja.

Petición

idPetición, nifCliente, fecha.

Respuesta

idRespuesta, nifCliente, idPetición, fecha, idLocalidadT.

Localidad_Trabajo

idLocalidad, dirección, población, provincia, CP

Consulta

idConsulta, nifCliente, idLetrado, fecha, descripción, idRespuestaJ.

RespuestaJ

idRespuestaJ, idLetrado, idConsulta,

Informe

idRespuestaJ, fecha, descripción, contenido.

Recurso

idRespuestaJ, órgano, coste, descripción, contenido, fecha.

DocumentoF

idRespuestaJ, coste, descripción, contenido, fecha, tipo.

Factura

idFactura, idPermuta, idRespuestaJ, importe, fecha.

Material

idMaterial, idAutor, título, contenido, descripción.

Permuta

idPetición, idRespuesta, okClienteP, okClienteR, fecha

3.4.2.- Diseño lógico

NIF

nif.

Persona

nif, nombre, apellidos, domicilio, mail, teléfono, población, provincia, CP.

[**nif**] referencia a **NIF**

Usuario

nif, idUsuario, contraseña, nick, perfil.

[**nif**] referencia a **Persona**.

Cliente

nif, idCliente, admón, convenio, localidadT, tipoEnte, puesto, grupo, cEspecífico, cProductividad, horario, gratificación.

[**nif**] referencia a Usuario.

[**localidadT**] referencia a **localidadTrabajo**.

Letrado

nif, idLetrado, especialidad.
[**nif**] referencia a Usuario.

Autor

nif, idAutor, fechaAlta, fechaBaja.
[**nif**] referencia a Usuario

Material

idMaterial, idAutor, título, contenido, descripción.
[**idAutor**] referencia a Autor.

Consulta

idConsulta, nifCliente, idLetrado, idRespuestaJ_fecha, descripción,
[**nifCliente**] referencia a Cliente.
[**idLetrado**] referencia a Letrado
[**idRespuestaJ**] referencia a RespuestaJ.

RespuestaJ

idRespuestaJ, idConsulta,
[**idConsulta**] referencia a Consulta.

Informe

idRespuestaJ, fecha, descripción, contenido.
[**idRespuestaJ**] referencia a RespuestaJ.

Recurso

idRespuestaJ, órgano, coste, descripción, contenido, fecha, idFactura.
[**idRespuestaJ**] referencia a RespuestaJ.
[**idFactura**] referencia a Factura.

DocumentoF

idRespuestaJ, coste, descripción, contenido, fecha, tipo, idFactura.
[**idRespuestaJ**] referencia a RespuestaJ.
[**idFactura**] referencia a Factura.

Permuta

idPetición, idRespuesta, okClienteP, okClienteR, fecha, idFactura
[**idPetición**] referencia a Petición.
[**idRespuesta**] referencia a Respuesta.
[**idFactura**] referencia a Factura.

4.-Valoración Económica.

Al tratarse de un proyecto perteneciente al ámbito académico y, por tanto, hipotético realizado por una sola persona. Es difícil y aventurado estimar una cantidad monetaria del proyecto realizado. Únicamente cabría hacer una valoración más o menos aproximada del tiempo invertido en ello, que redondeando podría cifrarse en 450 o más horas. También hay que hacer constar que esta estimación es lineal y no se pondera la tipología del horario invertido y por tanto, se desprecia el hecho de que muchas horas hayan sido nocturnas, festivas o intempestivas.

5.-Conclusión.

A pesar de ser consciente de que el proyecto es incompleto, no en vano faltan por incluir la implementación y el testeo de calidad, es cierto que la aplicación de un caso práctico del nivel del presente proyecto, sienta las bases para poder enfrentarse con un mayor nivel de fiabilidad ante el desarrollo de proyectos de un mayor nivel de complejidad.

Conforme se obtenga una mayor experiencia en el desarrollo de proyectos orientados a objetos, se podrá avanzar en la consecución de estos objetivos. Por ejemplo, y una vez finalizado el proyecto se puede concluir que es manifiestamente mejorable tanto en la parte correspondiente a las especificaciones como a la de diseño que podían estar más “pulidas”. Imponderables sobrevenidos han hecho posible un mayor refinamiento

Como conclusión, indicar que a nivel personal ha sido una experiencia muy satisfactoria y que, sin duda, constituye en un buen ejemplo para afrontar el desarrollo de futuros proyectos en el ámbito profesional.

6.-Glosario de términos.

- **Usuario**
- Genérico que se refiere a cualquier persona que, con los permisos de acceso requeridos hace uso y accede a las funcionalidades que la aplicación le ofrece
- **Aplicación**
- Ingenio informático ideado para llevar a cabo una o varias tareas. En este caso llevará la gestión de ofrecer servicio de permutas, asesoría jurídica y formación a los empleados públicos
- **Contenido.**
- Cualquier material con contenidos didácticos.
- **Informe**

- Opinión que un asesor jurídico da a un usuario sobre cualquier asunto con trascendencia jurídica.
- **Permuta**
- Intercambio consentido del puesto de trabajo entre dos empleados públicos.
- **Recurso**
- Proceso compuesto por diferentes actos que se dirime en los tribunales.
- **Factura**
- Justificante para el usuario destinatario de algún servicio de pago.
- **Petición**
- Exhibición por parte de un usuario, de su interés por permutar.
- **Personal de Secretaria**

Es la persona encargada de realizar las tareas administrativas de la aplicación. En concreto la comunicación con los clientes y los otros usuarios y la actividad tocante a la gestión de facturas.

7.-Bibliografía.

- YermiSistac, J., Camps, R., Costal, D., Martín, C., Rodríguez, E. (sept. 2005). *Bases de Datos I* (2ª ed.). UOC. Barcelona: Eureka Media, SL.
- Sistac, J., Camps, R., Costa, P., Costal, D., Marco, J.M., Rodríguez, E., Segret, R., Urpí, T. (febrero 2004). *Bases de Datos II* (2ª ed.). UOC. Barcelona: Eureka Media, SL.
- Campderrich Falgueras, Benet (2004). *Ingeniería del Software*. UOC. Recerca Informàtica, SL.
- Xhata, Fatos, Caballe, Santi (2008). *Aplicaciones distribuidas en Java, con tecnología RMI*. Delta. Barcelona
- F. Javier Ceballos (2005) *Java 2*, 3ª edición. RA-MA. Madrid
- Pressman, R.S. (1998). *Ingeniería del software. Un enfoque práctico*. Madrid: McGraw-Hill.
- Beneito Montagut, Roser. *Presentación y elaboración de presentaciones*. UOC. Barcelona.
- Rumbaugh, J., Jacobson, I., Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Addison-Wesley.
- Schmuller, J. *Aprendiendo UML en 24 horas*. Prentice Hall.