

# PRESENTACIÓN

## TRABAJO FIN DE GRADO



### *SISTEMA DE CONTROL DE DEMANDAS CIUDADANAS*

**2º CICLO DE INGENIERÍA EN INFORMÁTICA**  
**ÁREA: INGENIERÍA DEL SOFTWARE**  
**ALUMNO: JOAQUÍN LÓPEZ CARREÑO**  
**CONSULTOR: JUAN JOSÉ CUADRADO GALLEGO**



# ÍNDICE DE CONTENIDOS



- Objetivos del proyecto
- Planificación del proyecto
- Análisis de requisitos
- Diseño técnico
- Implementación
- Pruebas del sistema
- Conclusiones del proyecto

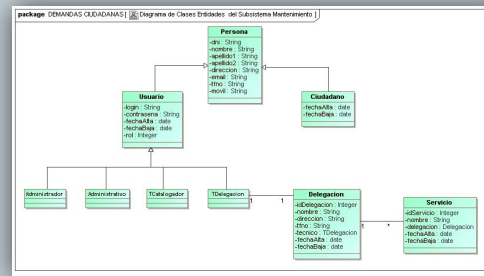
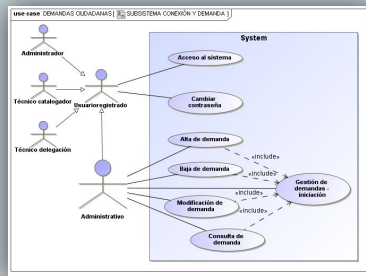
# Objetivos del proyecto

## Académicos

- Poner en práctica las competencias adquiridas durante el Grado. Especialmente en el itinerario de Ingeniería del Software.

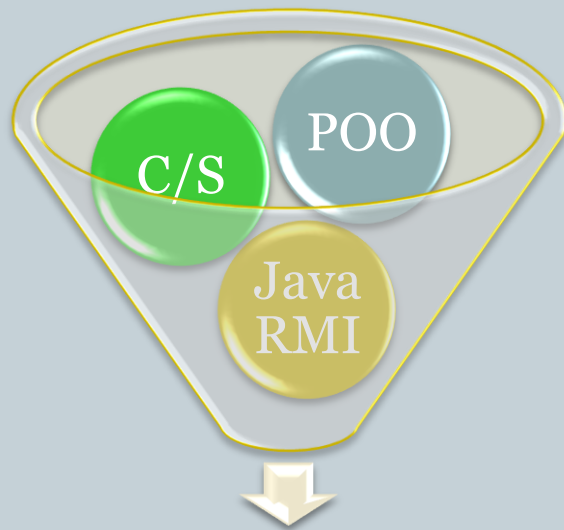
## Profesionales

- Desarrollar una aplicación para controlar y mejorar la gestión de demandas de servicios de los ciudadanos de un municipio.



Fecha	Usuario	Rol	Tipo Observación	Descripción
01/04/2013	administrador	Catalogador	De valoración	Añadir acorrido de CI Escasa si procede
01/04/2013	usuario	Catalogador	De Aceptación	OK, dos operarios se pasan en 3 días

# Planificación del proyecto (I)



Software a medida

Metodología  
Ciclo de Vida en Cascada



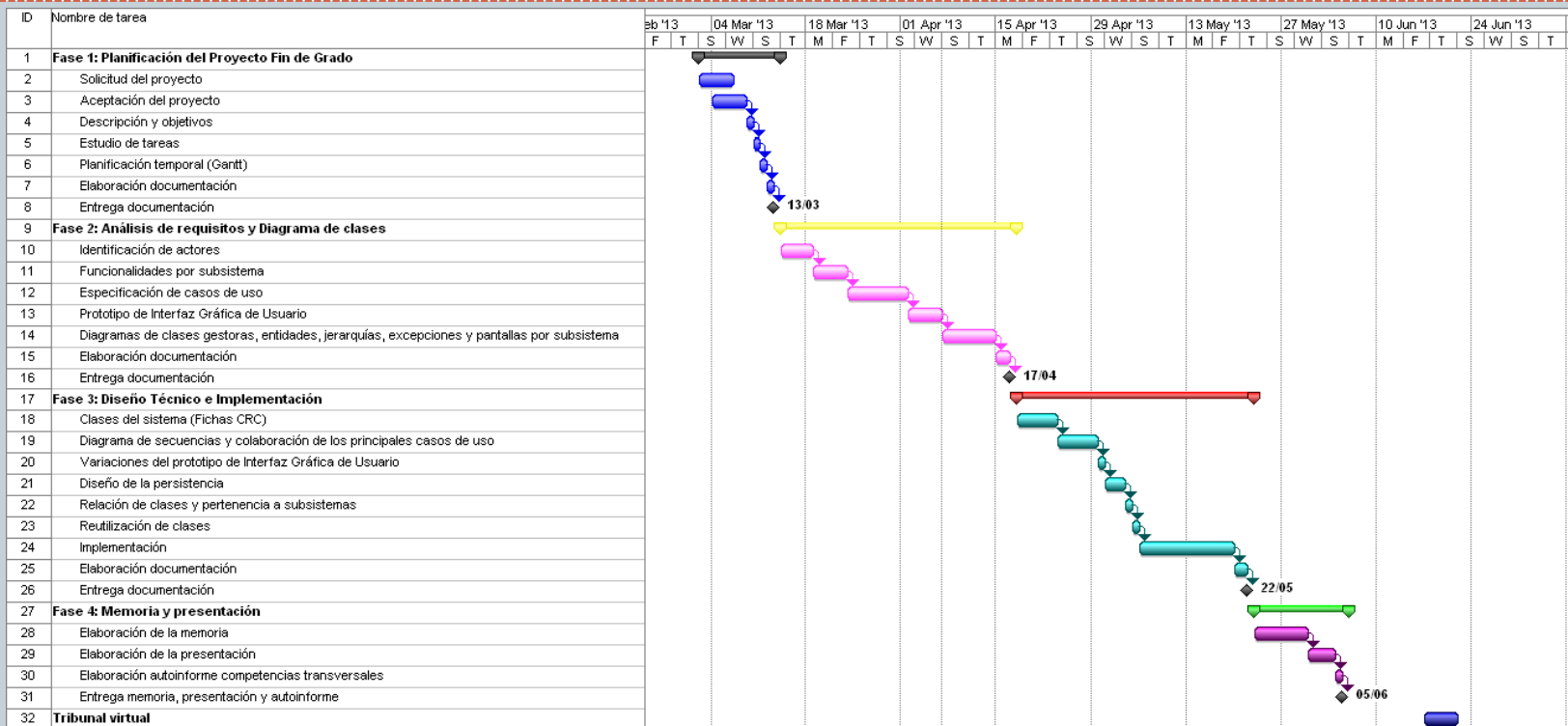
# Planificación del proyecto (II)



Descomposición del proyecto en actividades (WBS)

Diagrama de PERT

Diagrama de GANTT



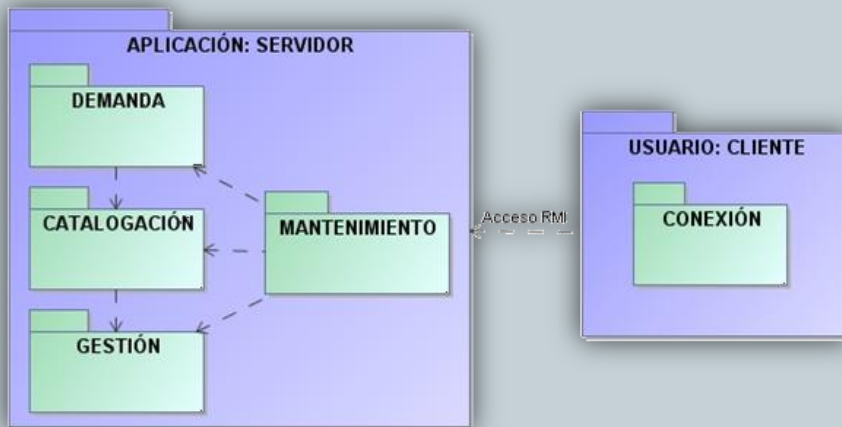
# Análisis de requisitos (I)

## 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.



Mantenimiento de los diferentes elementos que se utilizarán en el resto de subsistemas.



Dar de alta en el sistema las distintas demandas de los ciudadanos y consultar el estado de las mismas.



Valoración y catalogación inicial de las demandas ciudadanas. Control del estado de ejecución de las mismas.



Encargado de informar del estado de ejecución de cada una de las demandas ciudadanas.

# Análisis de requisitos (II)

## Funcionalidades

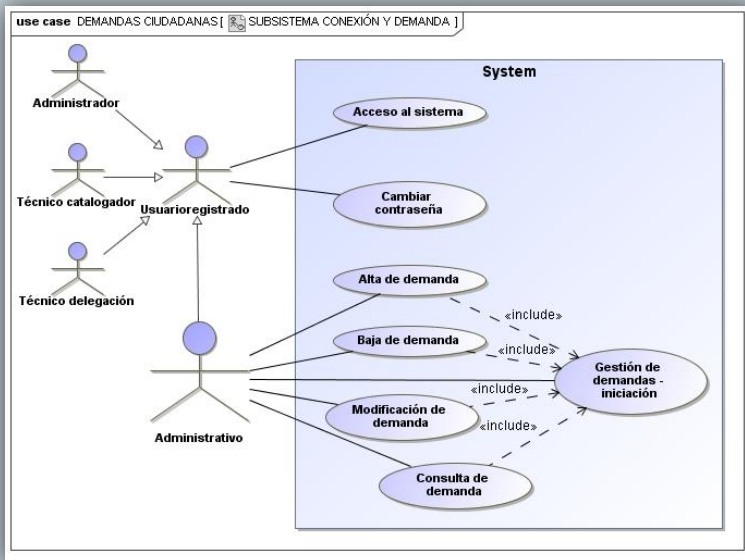
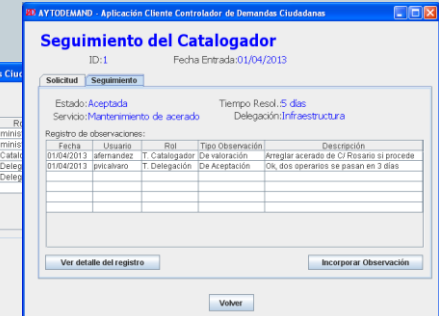
- Describir el proceso para todas las funciones requeridas para solucionar el problema.

## Casos de uso

- Describir cómo el software va a ser usado en una determinada situación.



## PROTOTIPO



### Caso de uso 17: Acceso al sistema

**Resumen de la funcionalidad:** Permite el acceso de un usuario a las funcionalidades de la aplicación según su rol.

**Casos de uso relacionados:** Administrador, administrativo, técnico catalogador, técnico delegación.

**Actores:** El usuario debe estar registrado en el sistema y conocer su login y password.

**Precondición:** El usuario accede a las funcionalidades del sistema según su perfil.

**Postcondición:** El usuario accede a las funcionalidades del sistema según su perfil.

**Escenario principal de éxito:**

1. El sistema solicita al actor el nombre de usuario y contraseña para acceder a la aplicación
2. El actor introduce el nombre de usuario y contraseña
3. El sistema valida los datos del actor
4. El sistema muestra las funcionalidades de la aplicación según el perfil del actor identificado

**Extensiones:**

- 2a. El actor elige la opción Cancelar
  - 2a1. Se finaliza el caso de uso
- 3a. Existe un error en el nombre de usuario o la contraseña
  - 3a1. El sistema informa al actor del error en la introducción de los datos
  - 3a2. El actor confirma el mensaje del sistema y se vuelve al punto 2

# Diseño técnico (I)

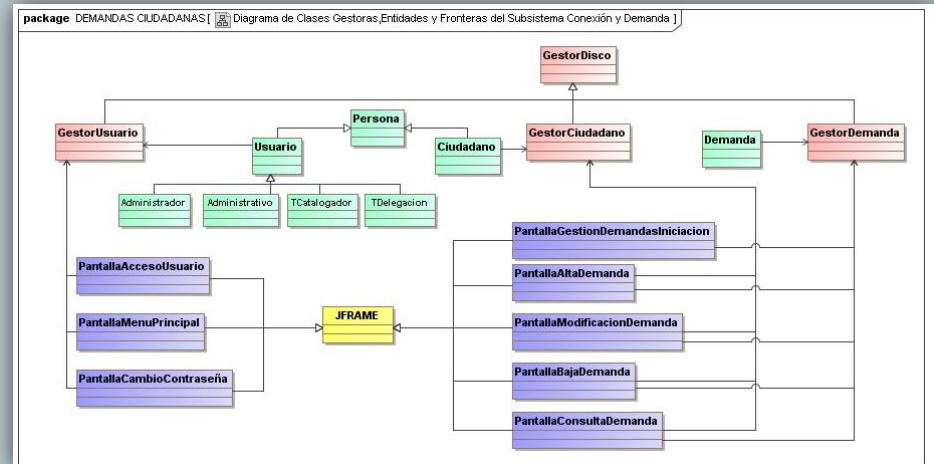
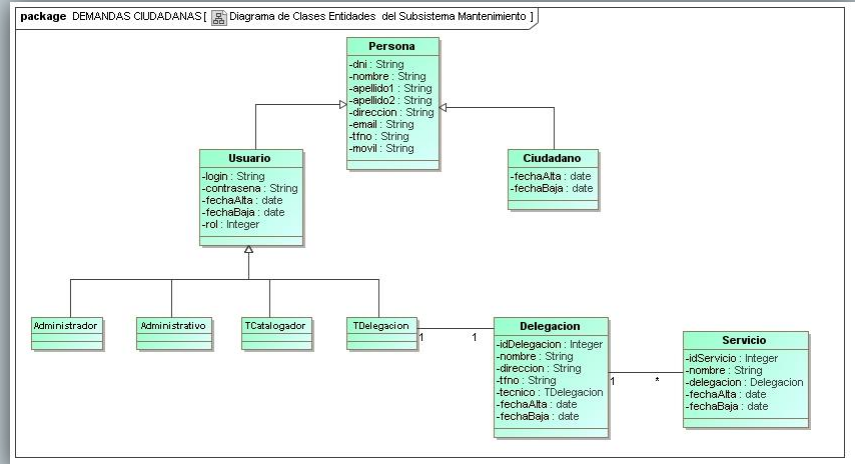
## Fichas CRC

- Identificar las clases candidatas del sistema .
- Identificar atributos y operaciones para cada objeto del sistema .

## Diagramas de clases y jerarquías

- Definir estructuras y jerarquías que organicen las clases.
- Visualizar, especificar, construir y documentar los aspectos estáticos de un sistema.

Servicio	
<b>Descripción:</b>	Representa a un servicio en el sistema.
<b>Tipo:</b>	Principal
<b>Características:</b>	Concreta, persistente
<b>Responsabilidades:</b>	Mantiene los datos de un servicio ofrecido por la administración local
<b>Colaboraciones:</b>	Delegacion, Demanda
<b>Constructores:</b>	
+Servicio(idServicio:Integer)	
+Servicio(idServicio:Integer, nombre:String, delegacion:Delegacion, fechaAlta:Date, fechaBaja:Date)	
<b>Atributos:</b>	
// Identificador del servicio (debe ser único)	
-idServicio:Integer	
// nombre descriptivo del servicio	
-nombre:String	
// Delegación responsable del servicio municipal	
delegacion:Delegacion	
// Fecha de alta del servicio	
-fechaAlta:Date	
// Fecha de baja del servicio. Si fechaBaja IS NOT NULL, el servicio es BAJA	
-fechaBaja:Date	
<b>Métodos:</b>	
// Métodos accesores a los atributos privados de la clase	
+getIdServicio():Integer	
+setIdServicio(idServicio:Integer)	
+getNombre():String	
+setNombre(nombre:String)	
+getDelegacion():Delegacion	
+setDelegacion(delegacion:Delegacion)	
+getFechaAlta():Date	
+setFechaAlta(fechaAlta:Date)	
+getFechaBaja():Date	
+setFechaBaja(fechaBaja:Date)	





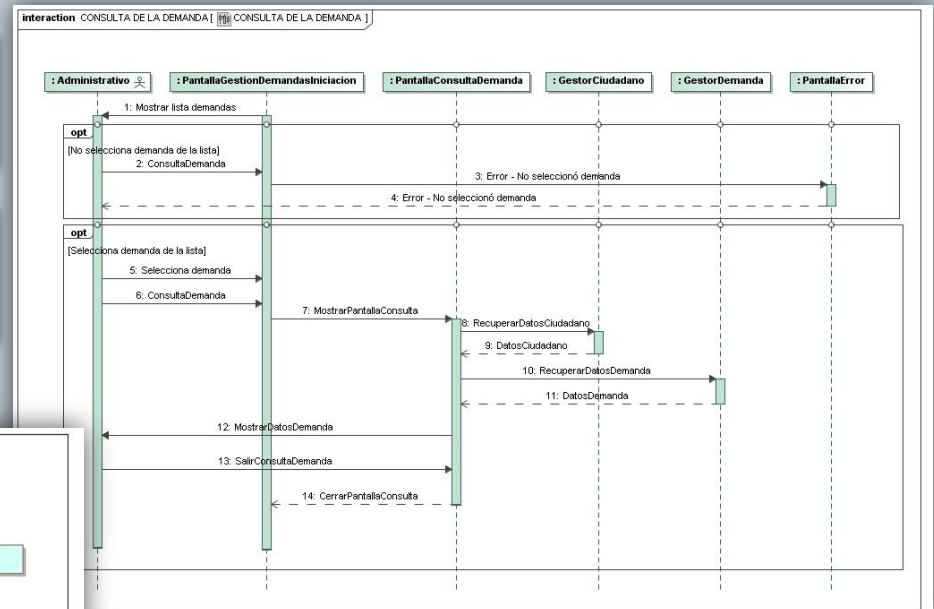
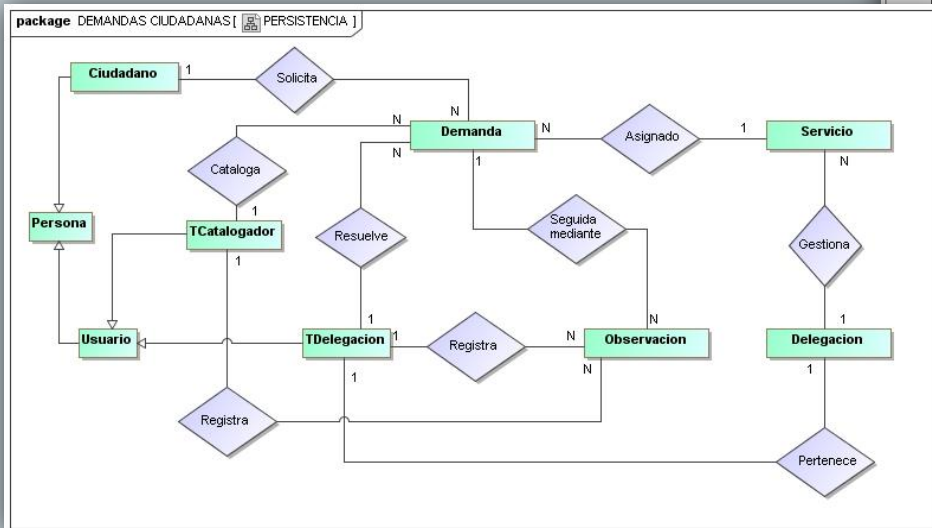
# Diseño técnico (II)

## Diagramas de interacción

- Vista del comportamiento del sistema, construyendo el modelo objeto-comportamiento.

## Diseño de la persistencia

- Modelo conceptual de los datos que serán precisos disponer más allá de la ejecución de la aplicación.



Relación de Clases por subsistema

Reutilización de Clases

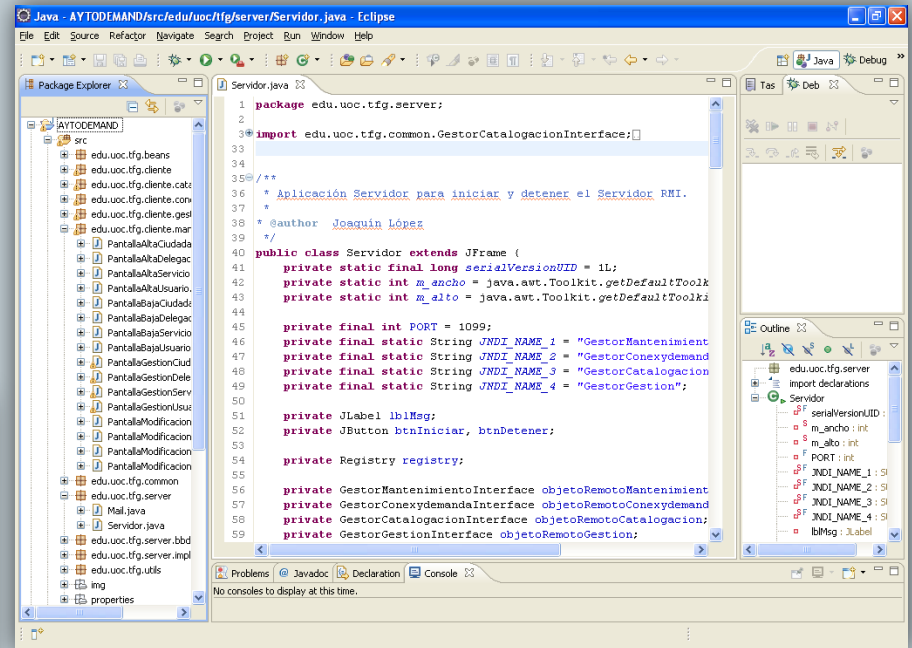
# Implementación (I)

## Programación

- Conversión de un diseño orientado a objetos en un código de programa, mediante el lenguaje de programación orientado a objetos Java.

## Elementos innovadores

- Automatizar el envío de información de interés a los distintos actores participantes en el sistema, sobre el estado de los expedientes de demandas ciudadanas.



```
1 package edu.uoc.tfg.server;
2
3 import edu.uoc.tfg.common.GestorCatalogacionInterface;
4
5 /**
6  * Aplicación Servidor para iniciar y detener el Servidor RMI.
7  */
8  @author Joaquín López
9  */
10 public class Servidor extends JFrame {
11     private static final long serialVersionUID = 1L;
12     private static int m_ancho = java.awt.Toolkit.getDefaultToolkit
13         .getToolkit().getScreenWidth();
14     private static int m_alto = java.awt.Toolkit.getDefaultToolkit
15         .getToolkit().getScreenHeight();
16
17     private final int PORT = 1099;
18     private final static String JNDI_NAME_1 = "GestorMantenimiento";
19     private final static String JNDI_NAME_2 = "GestorConexydemanda";
20     private final static String JNDI_NAME_3 = "GestorCatalogacion";
21     private final static String JNDI_NAME_4 = "GestorGestion";
22
23     private JLabel lblMsg;
24     private JButton btnIniciar, btnDetener;
25
26     private Registry registry;
27
28     private GestorMantenimientoInterface objetoRemotoMantenimiento;
29     private GestorConexydemandaInterface objetoRemotoConexydemanda;
30     private GestorCatalogacionInterface objetoRemotoCatalogacion;
31     private GestorGestionInterface objetoRemotoGestion;
32 }
```



# Implementación (II)

## Estructura de los paquetes

- Ayudan a organizar los elementos con el fin de comprenderlos más fácilmente y controlar las líneas de separación de la arquitectura del sistema.

## Documentación Javadoc

- Complementa la documentación generada desde las fases de análisis y diseño, ampliando la información que explique lo que realiza cada clase incorporada a la aplicación.

The screenshot shows a web browser displaying the Javadoc documentation for the 'edu.uoc.tfg.cliente' package. The browser address bar shows the file path: file:///C:/Temp/AYTODEMAMD/javadoc/index.html. The page has a navigation bar with 'Overview', 'Package', 'Class', 'Use', 'Tree', 'Deprecated', 'Index', and 'Help' tabs. Below the navigation bar, there is a table listing the packages within the 'edu.uoc.tfg.cliente' package:

Package
<a href="#">edu.uoc.tfg.beans</a>
<a href="#">edu.uoc.tfg.cliente</a>
<a href="#">edu.uoc.tfg.cliente.catalogacion</a>
<a href="#">edu.uoc.tfg.cliente.conexydemanda</a>
<a href="#">edu.uoc.tfg.cliente.gestion</a>
<a href="#">edu.uoc.tfg.cliente.mantenimiento</a>
<a href="#">edu.uoc.tfg.common</a>
<a href="#">edu.uoc.tfg.server</a>
<a href="#">edu.uoc.tfg.server.bbdd</a>
<a href="#">edu.uoc.tfg.server.impl</a>
<a href="#">edu.uoc.tfg.utils</a>

The screenshot shows two Package Explorer windows from an IDE. The left window shows the project structure for 'AYTODEMAMD', with a tree view including 'src' and sub-packages like 'edu.uoc.tfg.beans', 'edu.uoc.tfg.cliente', and 'edu.uoc.tfg.server'. The right window shows a detailed view of the 'edu.uoc.tfg.common' package, listing various Java classes such as 'ExcepcionPantalla.java', 'GestorCatalogacionInterface.java', and 'GestorConexionInterface.java'.

Manuales de la aplicación

Control de Excepciones

# Pruebas del sistema (I)

## Pruebas unitarias

- En SOO es la prueba de una clase, incluyendo en cada clase un método main() para comprobar el correcto funcionamiento de los métodos de la clase.

## Pruebas de integración

- En SOO son pruebas de agrupamiento de las clases colaboradoras, intentando revelar errores en las colaboraciones.

Nombre	Joaquín López Carreño		
Clase	GestorConexydemandaInterfacelmpl		
Método Main()	<pre>public static void main(String[] args) {     try{         test = new GestorConexydemandaInterfacelmpl();         testObtenerDemandas();         testCrearDemandaCiudadanoNoExiste();         testCrearDemandaCiudadano();         testObtenerObservacionesDemandaNoExiste();         testObtenerObservacionesDemanda();         testObtenerUsuarioInyeccionCodigo();         System.exit(0);     }catch (Exception e) {         System.out.println(e.toString());         e.printStackTrace();         System.exit(1);     } }</pre>		
Nº	Método	Acción Esperada	Verificación
1	public List<Demanda> getDemandas()	Devuelve una lista de las demandas con todos sus datos	✓
2	public void grabaDemanda(Demanda demanda)	No graba la demanda porque no existe el ciudadano en la BD.	✓
3	public void grabaDemanda(Demanda demanda)	Inserta la nueva demanda en la BD.	✓
4	public List<Observacion> getObservaciones(int iddemanda)	No se presenta datos de observaciones, y no se produce error aunque la demanda no exista	✓
5	public List<Observacion> getObservaciones(int iddemanda)	Devuelve una lista de las observaciones asociadas a la demanda con todos sus datos	✓
6	public Usuario getUsuario(String login, String pass)	No se obtienen los datos de usuario, ya que se evita la inyección de código	✓

Nombre	Joaquín López Carreño		
Subsistema	Mantenimiento		
Implicados	Mantenimiento – Conexión y demanda		
Propósito	Realizar el máximo número de pruebas cuyo diagrama de flujo involucre a los subsistemas mantenimiento y conexión y demanda (iniciando el proceso desde mantenimiento).		
Nº	Entrada	Acción Esperada	Verificación
1	Crear un usuario con el rol Administrativo	Acceder a la aplicación con el nuevo usuario y comprobar las opciones del perfil correctas para administrativo	✓
2	Modificar el password de un usuario	Acceder a la aplicación con el usuario y comprobar que la contraseña se ha actualizado	✓
3	Se da de baja un usuario	El usuario no se puede identificar en el sistema.	✓
4	Se crea un nuevo ciudadano en el sistema	Al dar de alta una demanda, al buscar el ciudadano aparece el nuevo ciudadano	✓
5	Se modifica un dato de contacto de un ciudadano	Si se accede a consultar una demanda asociada al ciudadano el dato de contacto aparece modificado	✓
6	Se da de baja un ciudadano	Al dar de alta una demanda, al buscar el ciudadano no aparece el ciudadano dado de baja	✓

### Observaciones

No existen observaciones a errores en las pruebas del subsistema

Prueba correctas	Errores	Valoración
6	0	100%

# Pruebas del sistema (II)

## Resultado auditoría

- Muy Bueno
- Número total de pruebas realizadas: 191
- Evaluación total del 100%

## Conclusiones

- La aplicación cumple con todas las funcionalidades solicitadas por el cliente.
- Primera versión preparada para pasar a la fase de producción.

## Mantenimiento

- Estar preparado ante la más que posible aparición de errores en la fase de producción.
- Posibles adaptaciones que haya que realizar según requerimientos del cliente.

Pruebas Unitarias		
Clases	Nº pruebas	Evaluación
GestorMantenimientoInterfacelImpl.java	9	100%
GestorConexydemandaInterfacelImpl.java	6	100%
GestorCatalogacionInterfacelImpl.java	4	100%
GestorGestionInterfacelImpl.java	3	100%
<b>Totales de las pruebas unitarias</b>	<b>22</b>	<b>100%</b>

Pruebas de Integración			
Subsistema		Nº pruebas	Evaluación
<b>Por subsistema</b>	Mantenimiento	51	100%
	Conexión y demanda	32	100%
	Catalogación y Seguimiento	23	100%
	Gestión de la demanda	27	100%
<b>Por parejas de Subsistemas</b>	Mantenimiento – Conexión y demanda	6	100%
	Mantenimiento – Catalogación y seguimiento	6	100%
	Mantenimiento – Gestión de la demanda	4	100%
	Conexión y demanda - Mantenimiento	2	100%
	Conexión y demanda – Catalogación y seguimiento	3	100%
	Conexión y demanda – Gestión de la demanda	1	100%
	Catalogación y seguimiento - Mantenimiento	1	100%
	Catalogación y seguimiento – Conexión y demanda	3	100%
	Catalogación y seguimiento – Gestión de la demanda	3	100%
	Gestión de la demanda - Mantenimiento	1	100%
	Gestión de la demanda – Conexión y demanda	3	100%
	Gestión de la demanda – Catalogación y seguimiento	3	100%
	<b>Totales de las pruebas de integración</b>	<b>169</b>	<b>100%</b>

# Conclusiones del proyecto



Proyecto actual

- Alto nivel de satisfacción por los objetivos conseguidos.
- Sienta las bases para poder enfrentarse con un mayor nivel de fiabilidad ante el desarrollo de proyectos de un mayor nivel de complejidad.

Proyectos futuros

- Mayor nivel de incertidumbre tanto desde el análisis de los requerimientos como de la mejor solución a aplicar.
- Aplicación de métodos iterativos e incrementales.

Ámbitos de mejora

- Conseguir una óptima reutilización de las clases.
- Manifiestamente mejorable en la reutilización que se hace de las clases sobre todo en la capa de interfaz de usuario.

Se han desarrollado los conocimientos adquiridos durante el Grado, constituyendo un buen ejemplo para afrontar el desarrollo de futuros proyectos en el ámbito profesional.

**MUCHAS GRACIAS  
POR SU ATENCIÓN**



***SISTEMA DE CONTROL  
DE DEMANDAS CIUDADANAS***

**2º CICLO DE INGENIERÍA EN INFORMÁTICA  
ÁREA: INGENIERÍA DEL SOFTWARE  
ALUMNO: JOAQUÍN LÓPEZ CARREÑO  
CONSULTOR: JUAN JOSÉ CUADRADO GALLEGO**

