

# **Sistema DGT**

## **1. Planificación**

### **1.1 Decisiones de diseño**

#### **1.1.1 Arquitectura**

Se ha decidido que el sistema va a tener una arquitectura monolítica. Dicha arquitectura incorporará el patrón de diseño MVC, es decir, el sistema se dividirá en una parte gráfica, con la cual interactúan los usuarios, una parte de dominio encargada de ofrecer las funcionalidades a la parte gráfica y por último otra parte dirigida a almacenar y obtener los datos de la base de datos.

#### **1.1.2 Metodología de desarrollo**

Para el desarrollo del sistema se utilizará Proceso Unificado de Desarrollo, es decir, el desarrollo del sistema estará guiado por casos de uso, será iterativo e incremental y estará guiado por la arquitectura. Las fases de las que se compondrá el desarrollo son: Inicio, elaboración, construcción y transición.

#### **1.1.3 Diseño del Software**

En la primera iteración de desarrollo se creará una interfaz genérica y una versión básica del sistema, que servirán como guía y dará soporte a todas las demás funcionalidades, que se implementarán en sucesivas iteraciones, ya que supondrá la arquitectura básica del sistema.

Se utilizará Java para el desarrollo del sistema ya que gracias a esto será multiplataforma.

Para almacenar los datos se utilizará una base de datos relacional MySQL.

## **1.2 Requisitos funcionales**

Requisito Funcional 1: El radar debe poder abrir un expediente sancionador.

Requisito Funcional 2: Sancionar a un conductor de un expediente determinado.

Requisito Funcional 3: Pagar una sanción.

Requisito Funcional 4: Cambiar el propietario de un vehículo.

## **1.3 Casos de uso**

Cada caso de uso deberá cumplir un requisito funcional ya que se ha decidido hacer un mapeo 1:1, es decir, un requisito funcional dará soporte a un requisito funcional.

Caso de Uso 1: Apertura de expediente. Cumple el Requisito Funcional 1.

Caso de Uso 2: Sanción de conductor. Cumple el Requisito Funcional 2.

Caso de Uso 3: Pago de sanción. Cumple el Requisito Funcional 3.

Caso de Uso 4: Cambiar propietario. Cumple el Requisito Funcional 4.

A su vez, cada caso de uso tiene una prioridad dentro del desarrollo del sistema, ya que puede que lo que se implementa con ese caso de uso sea necesario para realizar otros casos de uso. Las prioridades van de menor a mayor, siendo la mayor prioridad la 1.

Caso de Uso 1: Prioridad 1.

Caso de Uso 2: Prioridad 2.

Caso de Uso 3: Prioridad 3.

Caso de Uso 4: Prioridad 4.

## **1.4 Plan de iteraciones**

Se ha decidido realizar una iteración para cada caso de uso y a la hora de asignar el caso de uso a cada iteración se tendrá en cuenta la prioridad, ya que será necesario realizar los casos de uso en dicho orden. También se realizará una iteración inicialmente para realizar toda la planificación del proyecto y otra iteración al final para la etapa de transición. por lo tanto el proyecto se divide en las siguientes iteraciones:

- Iteración 0: Planificación del proyecto
- Iteración 1: Caso de Uso 1
- Iteración 2: Caso de Uso 2
- Iteración 3: Caso de Uso 3
- Iteración 4: Caso de Uso 4
- Iteración 5: Transición (integración, despliegue, documentación y entrega).

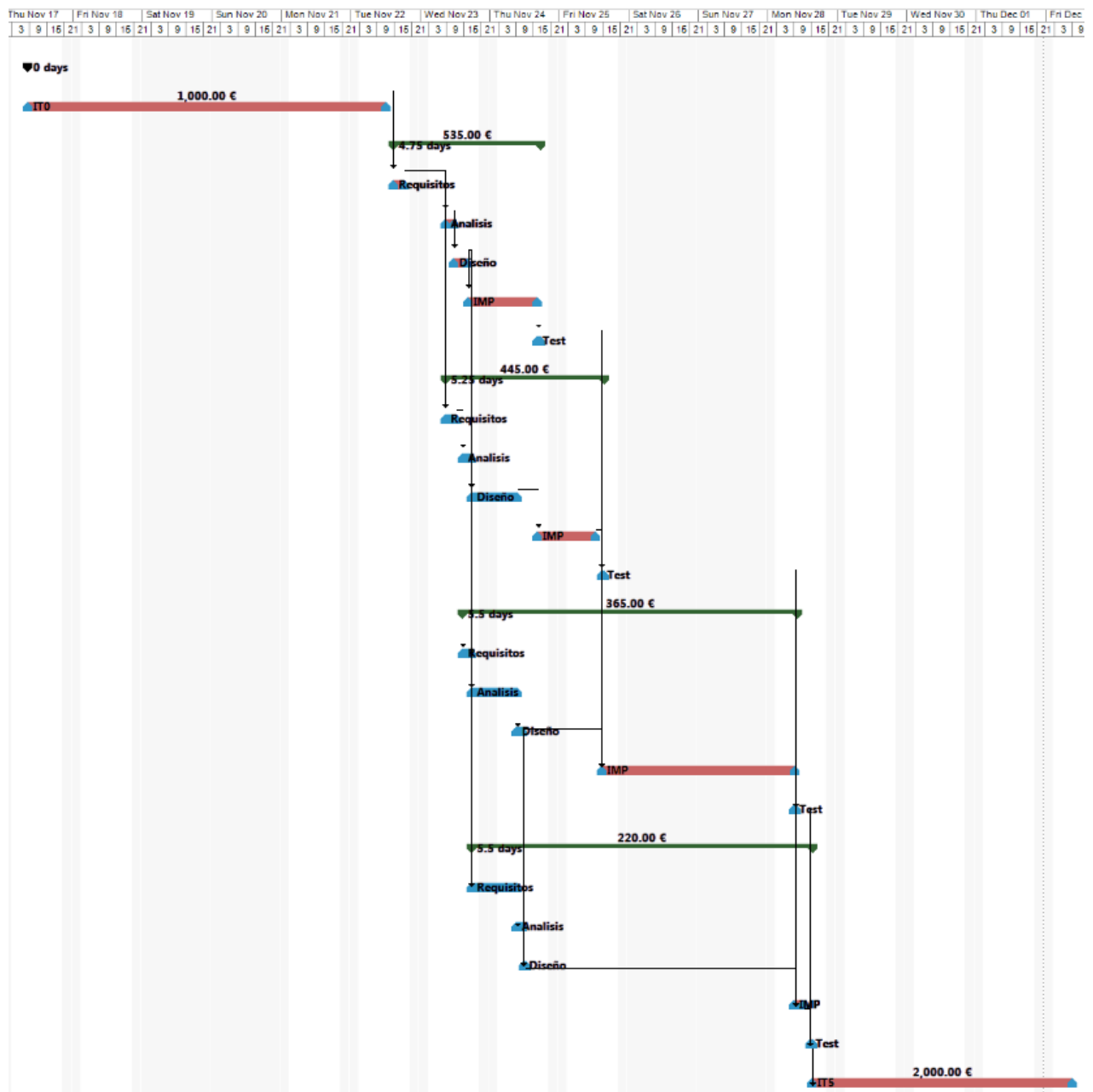
## **1.5 Estimación de agenda**

En las iteraciones 1 a 4 se tendrán las siguientes etapas: Requisitos, análisis, diseño, implementación y pruebas.

Inicio del proyecto: 17/11/2016

Fin del proyecto: 2/12/2016

Task Name	Duration	Start	Finish	Predecessors	Resource Names
☐ <b>Proyecto Swbuilder</b>	<b>22.25 days</b>	<b>Thu 11/17/16</b>	<b>Fri 12/2/16</b>		
☐ <b>Inicio</b>	<b>0 days</b>	<b>Thu 11/17/16</b>	<b>Thu 11/17/16</b>		<b>Aire acondicionado</b>
ITO	7 days	Thu 11/17/16	Tue 11/22/16		ITO[1]
☐ <b>IT1</b>	<b>4.75 days</b>	<b>Tue 11/22/16</b>	<b>Thu 11/24/16</b>		
Requisito	4 hours	Tue 11/22/16	Tue 11/22/16	3	Adri
Analisis	3 hours	Wed 11/23/16	Wed 11/23/16	5	Pablo
Diseño	3 hours	Wed 11/23/16	Wed 11/23/16	6	Oscar,Christian Di
IMP	8 hours	Wed 11/23/16	Thu 11/24/16	7	Adri Impl
Test	1 hour	Thu 11/24/16	Thu 11/24/16	8	Pablo test
☐ <b>IT2</b>	<b>5.25 days</b>	<b>Wed 11/23/16</b>	<b>Fri 11/25/16</b>		
Requisito	4 hours	Wed 11/23/16	Wed 11/23/16	5	Adri
Analisis	3 hours	Wed 11/23/16	Wed 11/23/16	11,6	Pablo
Diseño	2 hours	Wed 11/23/16	Thu 11/24/16	12,7	Oscar,Christian Di
IMP	6 hours	Thu 11/24/16	Fri 11/25/16	13,8	Pablo Impl
Test	1 hour	Fri 11/25/16	Fri 11/25/16	14,9	Oscar test
☐ <b>IT3</b>	<b>5.5 days</b>	<b>Wed 11/23/16</b>	<b>Mon 11/28/16</b>		
Requisito	3 hours	Wed 11/23/16	Wed 11/23/16	11	Adri
Analisis	2 hours	Wed 11/23/16	Thu 11/24/16	17,12	Pablo
Diseño	2 hours	Thu 11/24/16	Thu 11/24/16	13,18	Oscar,Christian Di
IMP	5 hours	Fri 11/25/16	Mon 11/28/16	19,14	Oscar Impl
Test	1 hour	Mon 11/28/16	Mon 11/28/16	20,15	Christian
☐ <b>IT4</b>	<b>5.5 days</b>	<b>Wed 11/23/16</b>	<b>Mon 11/28/16</b>		
Requisito	2 hours	Wed 11/23/16	Thu 11/24/16	17	Adri
Analisis	2 hours	Thu 11/24/16	Thu 11/24/16	23,18	Pablo
Diseño	1 hour	Thu 11/24/16	Thu 11/24/16	24,19	Oscar
IMP	3 hours	Mon 11/28/16	Mon 11/28/16	25,20	Christian Impl
Test	1 hour	Mon 11/28/16	Mon 11/28/16	26,21	Adri Test
IT5	7 days	Mon 11/28/16	Fri 12/2/16	27	IT5[1]



Nombre	Puesto	Horas	€/hora	Total
Pablo	Requisitos	10	30	300
Pablo	Testing	1	15	15
Pablo	implementación	6	20	120
Coste total				435

Nombre	Puesto	Horas	€/hora	Total
Oscar	Diseño	8	25	200
Oscar	implementación	5	20	100
Oscar	testing	1	15	15
Coste total				315

Nombre	Puesto	Horas	€/hora	Total
Christian	Testing	1	15	15
Christian	Diseño	7	25	175
Christian	implementación	3	20	60
Coste total				250

Por tanto el coste total de las 4 iteraciones es:

Coste final de plantilla	1565€
--------------------------	-------

## 1.6.2 Coste total del proyecto

Iteración 0	1000€
Iteraciones 1 a 4	1565€
Iteración 5	3000€
Total:	4565€

## 2. Análisis y Diseño

### 2.1 Descripción de los casos de uso

Caso de uso	1
Nombre	Apertura de expediente
Referencias	Requisito funcional 1
Responsabilidades	Permite abrir el expediente de un conductor
Actor	Usuario
Salida	Expediente creado
Pre-condiciones	El expediente que se desea crear tiene que ser sobre un conductor
Post-Condiciones	Se muestra el expediente recientemente creado
Escenario alternativo	Si intentamos crear un expediente para una persona que no es conductor, nos debe salir un mensaje de error indicándonos el error.

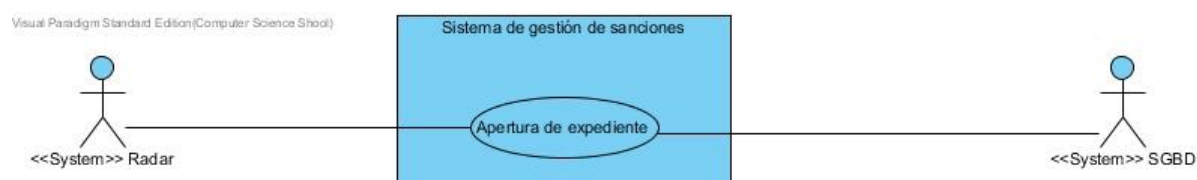
<b>Caso de uso</b>	2
<b>Nombre</b>	Sanción del conductor
<b>Referencias</b>	Requisito funcional 2
<b>Responsabilidades</b>	Permite sancionar a un conductor que ha superado la velocidad máxima permitida
<b>Actor</b>	Usuario
<b>Salida</b>	Sanción a un determinado conductor realizada
<b>Pre-condiciones</b>	Tiene que estar creado el expediente del conductor en cuestión
<b>Post-Condiciones</b>	La sanción se lleva a cabo, indicándolo mediante un mensaje
<b>Escenario alternativo</b>	Que se produzca algún error al llevar a cabo la sanción, por lo que se debe mostrar un mensaje indicándonos el error

<b>Caso de uso</b>	3
<b>Nombre</b>	Pago de sanción
<b>Referencias</b>	Requisito funcional 3
<b>Responsabilidades</b>	Permite a un conductor pagar una determinada sanción
<b>Actor</b>	Conductor
<b>Salida</b>	Sanción pagada
<b>Pre-condiciones</b>	Tiene que estar creado el expediente del conductor y tiene que haberse llevado a cabo la sanción a ese conductor
<b>Post-Condiciones</b>	Se realiza correctamente el pago de la sanción y se elimina esa sanción de ese conductor
<b>Escenario alternativo</b>	Que haya algún problema al realizar el pago, por lo que se debe indicar mediante un mensaje el error cometido

<b>Caso de uso</b>	4
<b>Nombre</b>	Cambio de propietario del vehículo
<b>Referencias</b>	Requisito funcional 4
<b>Responsabilidades</b>	Permite al usuario del sistema cambiar el propietario de un vehículo
<b>Actor</b>	Usuario
<b>Salida</b>	Cambio realizado correctamente
<b>Pre-condiciones</b>	Que el vehículo exista
<b>Post-Condiciones</b>	Mensaje en el sistema de que el cambio se realizado correctamente
<b>Escenario alternativo</b>	Los datos introducidos para el nuevo propietario no sean correctos

## 2.2 Diagramas de casos de uso

El siguiente diagrama se corresponde con el diagrama de casos de uso de la iteración 1 de nuestro proyecto:



El siguiente diagrama se corresponde con el diagrama de casos de uso de la iteración 2 de nuestro proyecto:



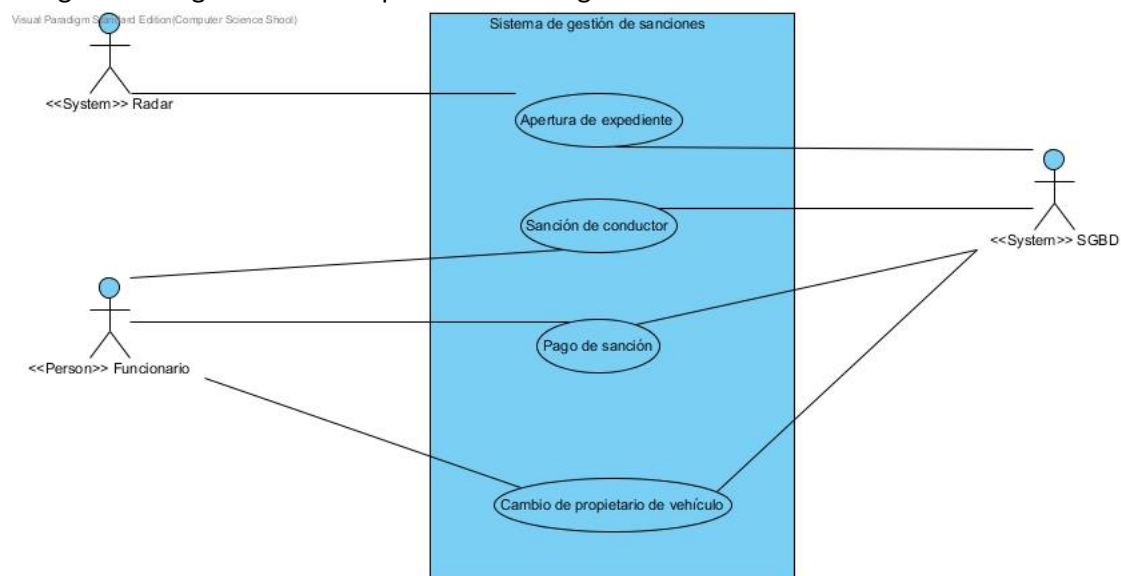
El siguiente diagrama se corresponde con el diagrama de casos de uso de la iteración 3 de nuestro proyecto:



El siguiente diagrama se corresponde con el diagrama de casos de uso de la iteración 4 de nuestro proyecto:

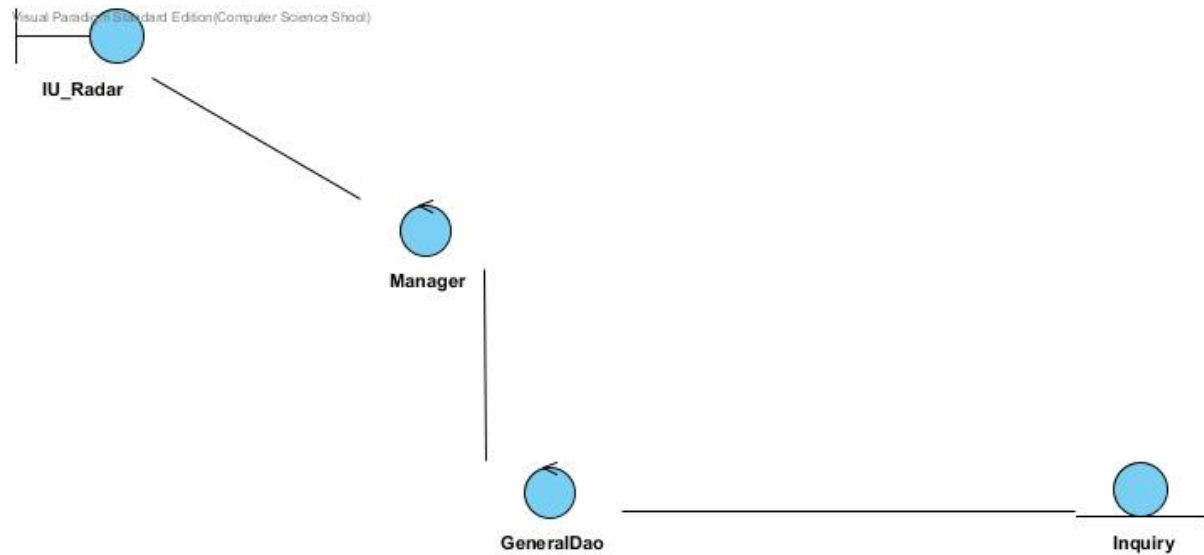


El siguiente diagrama se corresponde con el diagrama de casos de uso de todas las iteraciones:

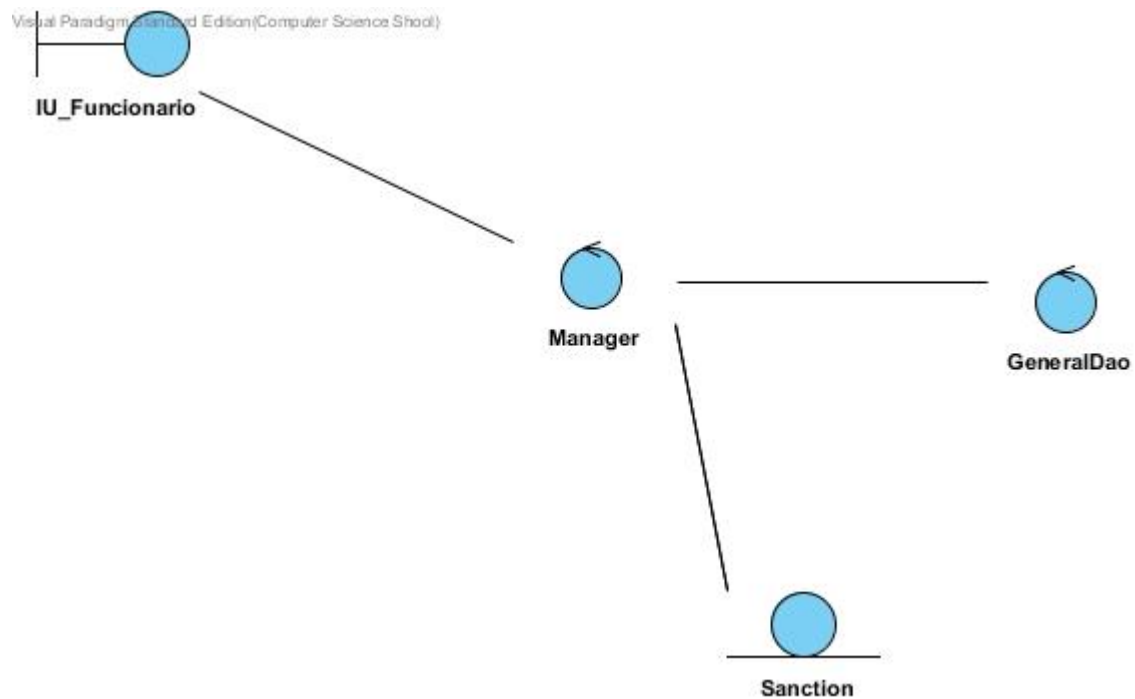


## 2.3 Diagramas de clases de análisis

El siguiente diagrama se corresponde con el diagrama de clases de análisis de la iteración 1 de nuestro proyecto:

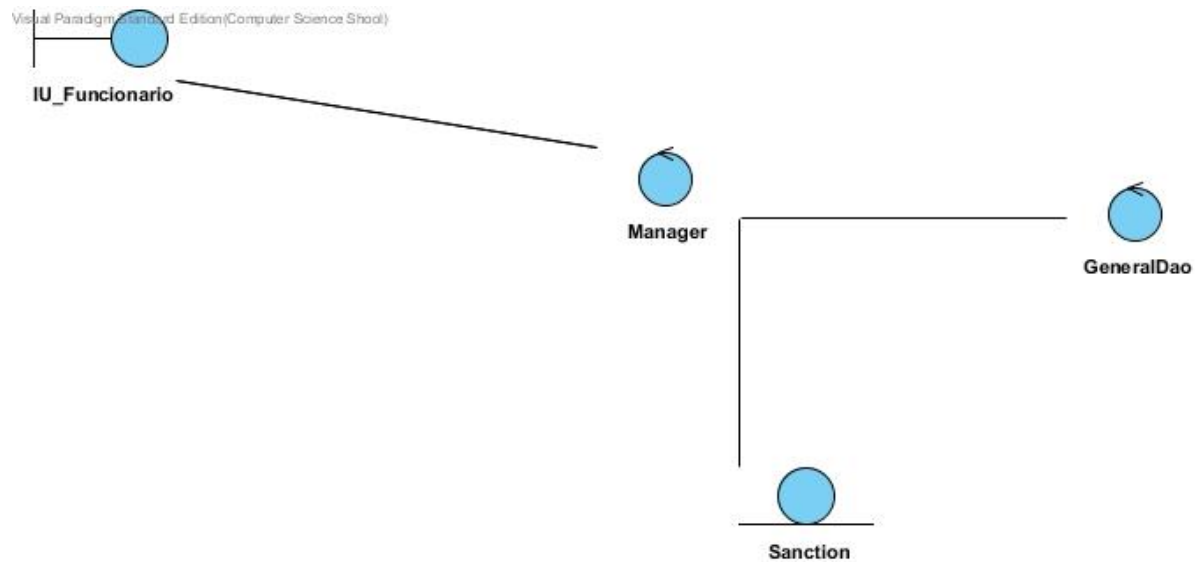


El siguiente diagrama se corresponde con el diagrama de clases de análisis de la iteración 2 de nuestro proyecto:

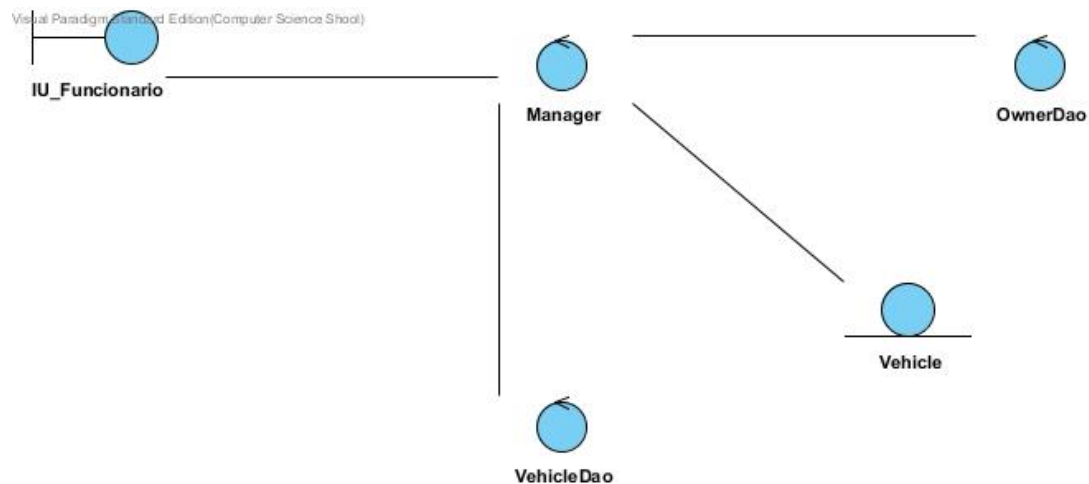




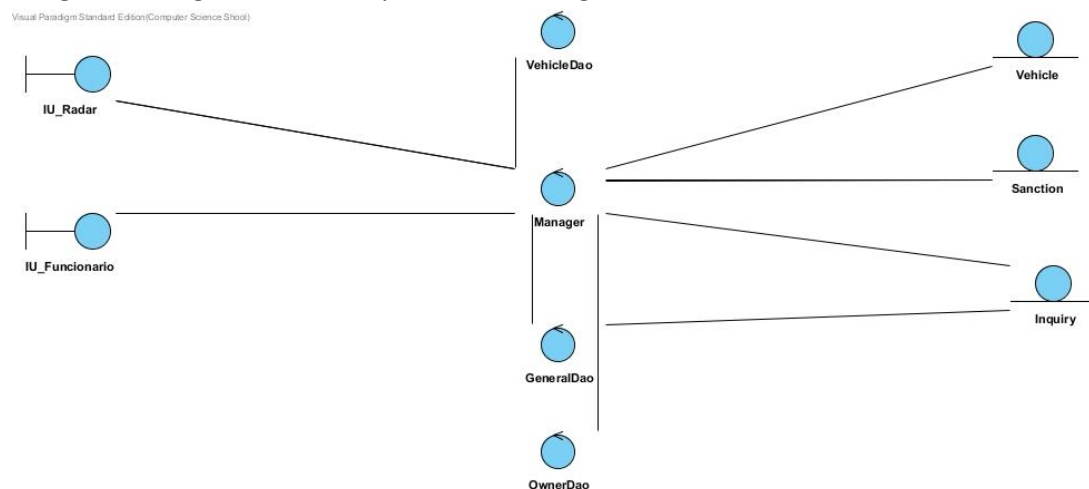
El siguiente diagrama se corresponde con el diagrama de clases de análisis de la iteración 3 de nuestro proyecto:



El siguiente diagrama se corresponde con el diagrama de clases de análisis de la iteración 4 de nuestro proyecto:

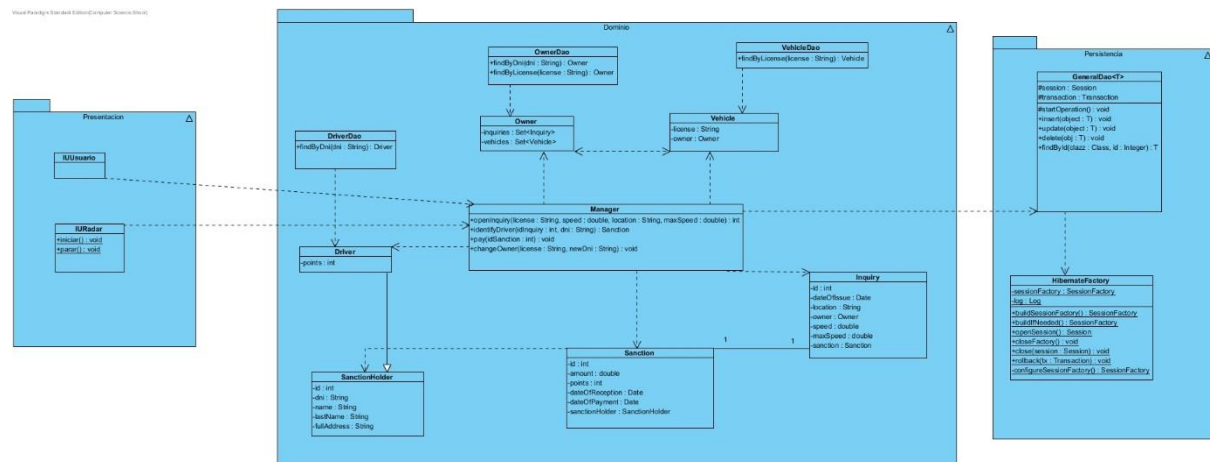


El siguiente diagrama se corresponde con el diagrama de clases de análisis de todas las iteraciones:



## 2.4 Diagramas de clases de diseño

El siguiente diagrama se corresponde con el diagrama de clases de diseño de nuestro proyecto:



## 3. Implementación

La implementación de este proyecto ha seguido el proceso unificado de desarrollo (proceso iterativo e incremental).

Este proyecto se ha dividido en módulos utilizando Maven, de forma que cada iteración pueda ser desarrollada por separado facilitando su gestión.

Cada módulo tendrá una estructura de paquetes que consistirá en presentación, dominio y persistencia, separando los elementos del sistema según la función que realicen.