



CAMPUS  
DE EXCELENCIA  
INTERNACIONAL



**POLITÉCNICA**

"Ingeniamos el futuro"

## **Graduado en Ingeniería Informática**

Universidad Politécnica de Madrid

Escuela Técnica Superior de  
Ingenieros Informáticos

### **TRABAJO FIN DE GRADO**

# **DESARROLLO DE UNA APLICACIÓN WEB DE TRAZABILIDAD PARA EL SECTOR DE LA CONSTRUCCIÓN**

Autor: Álvaro Sagarruy de la Rosa

Director: Marina Álvarez Alonso

MADRID, Junio 2018

## Índice

<b>Lista de figuras</b> .....	<b>2</b>
<b>Lista de tablas</b> .....	<b>2</b>
<b>Resumen</b> .....	<b>3</b>
<b>Capítulo 1 - Introducción</b> .....	<b>4</b>
<b>1.1 Descripción de antecedentes</b> .....	<b>4</b>
<b>1.2 Objetivos</b> .....	<b>7</b>
<b>1.3 Justificación</b> .....	<b>7</b>
<b>1.4 Alcance</b> .....	<b>8</b>
<b>Capítulo 2 - Estado del Arte</b> .....	<b>9</b>
<b>2.1 Trazabilidad</b> .....	<b>9</b>
2.1.1 Características de la trazabilidad .....	9
2.1.2 Importancia y ventajas .....	9
2.1.3 Tipos de Trazabilidad.....	10
<b>2.2 Herramientas Web</b> .....	<b>11</b>
2.2.1 Servicio Web.....	11
2.2.2 Bases de Datos .....	14
2.2.3 Visualización Web.....	15
2.2.4 Mapas.....	16
2.2.5 Elección para el proyecto .....	17
<b>Capítulo 3 - Metodología del desarrollo</b> .....	<b>18</b>
<b>3.1 Diseño del modelo</b> .....	<b>19</b>
<b>3.2 Desarrollo</b> .....	<b>19</b>
<b>3.3 Pruebas</b> .....	<b>20</b>
<b>Capítulo 4 – Diseño</b> .....	<b>21</b>
<b>4.1 Requisitos</b> .....	<b>21</b>
4.1.1 Requisitos No funcionales .....	21
4.1.2 Requisitos Funcionales .....	22
4.1.3 Requisitos del Usuario.....	23
4.1.4 Requisitos del Sistema.....	24
<b>4.2 Elementos de la aplicación</b> .....	<b>24</b>
4.2.1 Roles en la aplicación y sus acciones .....	24
4.2.2 Otros elementos .....	26
4.2.3 Diseño estructurado de alto nivel.....	28
<b>4.3 Planificación del trabajo</b> .....	<b>29</b>
<b>4.4 Base de Datos</b> .....	<b>30</b>
<b>Capítulo 5 - Desarrollo</b> .....	<b>32</b>
<b>5.1 Aplicación</b> .....	<b>32</b>
5.1.1 Sesión y Autenticación de usuario .....	32
5.1.2 Gestión de los mapas .....	32

5.1.3 Estructura de las páginas .....	34
<b>5.2 Redacción de Pruebas.....</b>	<b>35</b>
<b>Capítulo 6 - Conclusiones y líneas futuras .....</b>	<b>39</b>
6.1 Conclusiones .....	39
6.2 Líneas futuras.....	39
<b>Referencias web .....</b>	<b>40</b>
<b>Bibliografía .....</b>	<b>40</b>

## Lista de figuras

Figura 1 – Documentación necesaria para la gestión de residuos .....	04
Figura 2 – Ejemplo de albarán de servicio .....	05
Figura 3 – Ejemplo de albarán de admisión .....	06
Figura 4 – Fase de ejecución en la gestión de RCD .....	06
Figura 5 – Diagrama de flujo de trabajo .....	18
Figura 6 – Diseño estructurado de alto nivel .....	29
Figura 7 – Modelo Entidad Relación .....	31

## Lista de tablas

Tabla 1 – Comparativa de tecnologías para desarrollo web .....	12
Tabla 2 – Comparativa de servidores web .....	14
Tabla 3 – Comparativa de gestores de bases de datos .....	15
Tabla 4 – Comparativa de servidores web .....	16
Tabla 5 – Requisitos No Funcionales .....	21
Tabla 6 – Requisitos Funcionales .....	22
Tabla 7 – Requisitos del Usuario .....	23
Tabla 8 – Requisitos del Sistema .....	24
Tabla 9 – Pruebas Generales .....	35
Tabla 10 – Pruebas unitarias del Constructor .....	36
Tabla 11 – Pruebas Unitarias del Responsable .....	37
Tabla 12 – Pruebas Unitarias del Gestor .....	38

## Resumen

El siguiente proyecto pretende desarrollar una aplicación web que permita gestionar la trazabilidad de los residuos en el sector de la construcción. Controlando además la comunicación entre los distintos actores de este proceso.

## Capítulo 1 - Introducción

### 1.1 Descripción de antecedentes

Desde el año 2008 se han ido incorporado varias regulaciones en materia de gestión de Residuos de Construcción y Demolición (RCD) en España, el RD 105/2008 que actualmente regula la producción y gestión de los RCD y diversos Planes Nacionales. En la actualidad está vigente el Plan Estatal Marco de Gestión De Residuos (PEMAR) 2016-2022.

Estos documentos establecen varias medidas para fomentar el reciclaje de RCD en las obras de construcción, entre las que destacan: la obligación de desarrollar los documentos Estudio y Plan de gestión de RCD y el pago de una fianza económica que será devuelta al finalizar la obra una vez que el promotor acredite documentalmente que los RCD han sido gestionados conforme a la ley.

La figura 1 muestra un flujograma con las distintas fases y documentación necesaria para la gestión de los residuos en el sector de la construcción conforme a la legislación vigente.

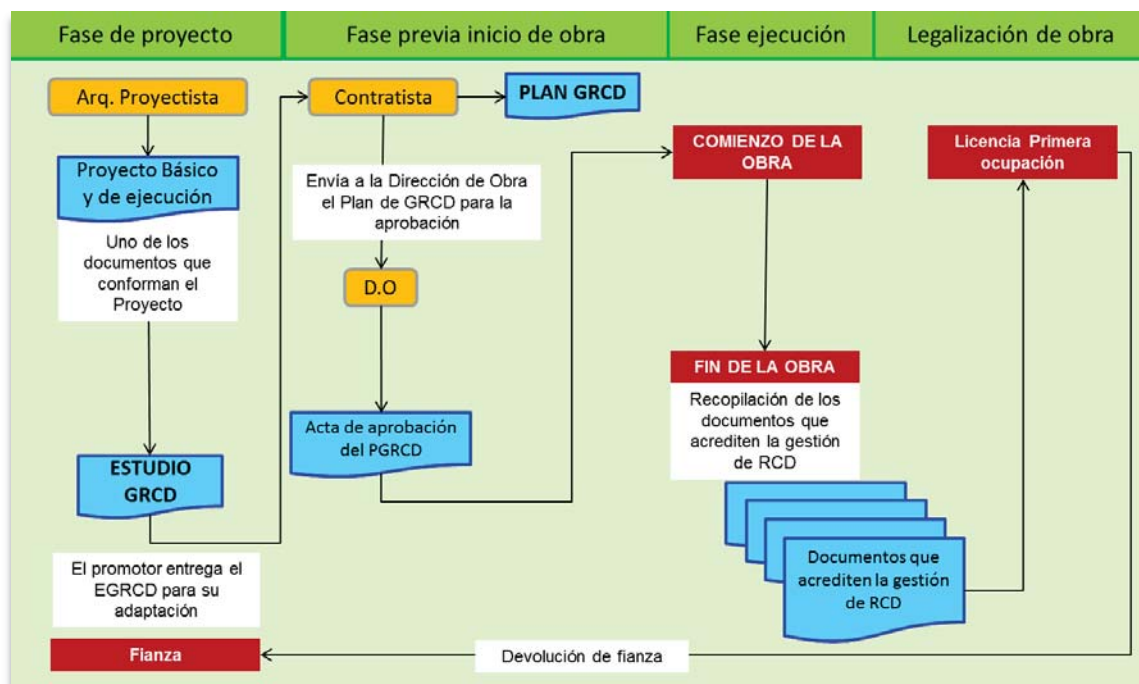


Figura 1 – Documentación necesaria para la gestión de residuos.

La figura 4 resume la gestión de RCD durante la fase de ejecución de la obra y engloba las siguientes fases:

1. **Retirada de contenedor y entrega de albarán de servicio:** cuando un contenedor sale de la obra lleno y traen otro vacío, el gestor autorizado entrega un “albarán de servicio”.

Este documento contiene información relativa a la localización de la obra, el número de contenedor que retira y el que trae vacío, la fecha, y su volumen. La figura 2 muestra un ejemplo de albarán actual.

MACOTRAN, S.L.U.  
CONTENEDORES, SACOS Y GRÚAS  
Nicaragua, 15 Local - 28016 MADRID  
Teléfono: 91 359 47 39 - 91 345 31 80 - 91 350 82 72 - Fax: 91 345 57 15  
E-mail: contenedores@macotra.com

ISO 9001  
ISO 14001

Nº albarán de servicio: **467061**

COCHE N.º: \_\_\_\_\_ ALBARAN N.º **467061**

CLIENTE: \_\_\_\_\_

DIRECCIÓN: **JUAN PABLO II (PUNO)** ← Datos de la obra

FECHA: **16.3.11** HORA: **13:15** ← Fecha y hora de recogida del RCD

GESTOR DE RESIDUOS NO PELIGROSOS AUTORIZADO POR LA COMUNIDAD DE MADRID  
N.º INSCRIPCIÓN RGN/MD/07256  
CENTRO-PLANTA: Ctra. Villaverde o Vallecas, n.º 277  
28031 MADRID

MACOTRAN, S. L. U.  
Empresa integrante de UTE Planta de Navalcarnero, le garantiza que sus residuos serán tratados y reciclados conforme a la Ley vigente.

Nº contenedor que entra en obra: **1236**

Nº contenedor lleno que se retira de obra: **873**

CONTENEDOR	LLEVA	RETIRA
CONTENEDOR 3 M/3	-----	RETIRA
CONTENEDOR 6 M/3	-----	RETIRA
CONTENEDOR 9 M/3	-----	RETIRA
CONTENEDOR 16 M/3	-----	RETIRA
CONTENEDOR 22 M/3	-----	RETIRA
CONTENEDOR 30 M/3	-----	RETIRA
SACO ESCOMBRO 1 M/3	-----	RETIRA

HORAS CAMIÓN GRÚA: \_\_\_\_\_

Capacidad del contenedor

**CONDICIONES GENERALES DE ALQUILER DEL CONTENEDOR**

1) Se debe presentar el contenedor de conformidad con el establecido en el art. 103 de Ley 22/2002 de residuos sólidos urbanos y en el art. 103 de Ley 22/2002 de residuos sólidos urbanos y en el art. 103 de Ley 22/2002 de residuos sólidos urbanos.

2) El albarán de servicio se emite en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

3) El albarán de servicio se emite en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

4) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

5) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

6) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

7) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

8) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

9) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

10) Cuando el contenedor sea retirado de la obra, el cliente deberá presentar el albarán de servicio en el momento de la entrega del contenedor vacío en la obra, a cargo del cliente, en la obra. La carga se podrá almacenar en cualquier lugar de la obra.

Figura 2 – Ejemplo de albarán de servicio.

2. **Transporte del contenedor y entrega a planta de tratamiento (albarán de admisión):** Cuando el transportista entrega el contenedor en la planta de reciclaje, esta lo pesa y emite un “albarán de admisión”. Este albarán contiene información relativa a la localización de la obra, peso, volumen y tipo de residuo (mediante un Código LER [1]) y el número del albarán de servicio.
3. **Archivo de documentación y entrega posterior para devolución de fianza:** Todos los albaranes se guardarán y se entregarán al finalizar la obra para poder obtener

la licencia de primera ocupación, así como la devolución de la fianza económica. En la figura 3 se muestra un ejemplo de albarán de admisión.

MACOTRAN, S.L. GESTOR DE RESIDUOS NO PELIGROSOS AUTORIZADO POR LA COMUNIDAD DE MADRID		
Nº albarán de servicio	ALBARAN N°: 020446 ALBARAN SERVICIO: 436336	Datos del gestor
	FECHA: 21/07/2010 H. ENTRADA: 08:08:37	Datos de la obra
Fecha factura	CODIGO TRANS.: 000001	Datos del RCD: Peso, volumen y cód. LER
	TRANSPORTISTA: MACOTRAN, S.L.U. MATRICULA:	PRODUCTO: 000001 RCD MEZCLADO LER 17 09 04 BRUTO: 13.520 TARA: 10.080 VOLUMEN: 6 NETO: 3.440
MACOTRAN, S.L. GESTOR DE RESIDUOS NO PELIGROSOS AUTORIZADO POR LA COMUNIDAD DE MADRID Nº INSCRIPCIÓN RES.MED.131 CENTRO PLANTA: Cda. Wladimir A. G. Nº 277 28014/M/295		FIRMA DEL TRANSPORTISTA:
(ALBARAN DE ADMISION)		

Figura 3 – Ejemplo de albarán de admisión.



Figura 4 – Fase de ejecución en la gestión de RCD.

A pesar de los avances legislativos en materia de gestión de RCD, existen numerosas dificultades para su aplicación en las obras, entre las que destacan: la dificultad en la segregación de los RCD en origen, la dificultad en la estimación a priori de la generación de RCD, y la dificultad para gestionar y controlar los residuos.

Además, son varios los agentes que intervienen en el proceso de gestión del RCD, con poca comunicación entre ellos, lo que dificulta tener una trazabilidad clara de todo el proceso.

Por este motivo, es fundamental que el sector de la construcción, así como los propios Ayuntamientos, establezcan medidas que permitan garantizar una correcta trazabilidad de los residuos que genera, y que permita conocer el trayecto realizado desde el lugar en el que el residuo se genera hasta su destino final.

Los sistemas de trazabilidad tienen implicaciones muy importantes en términos de calidad, seguridad y prevención. La trazabilidad se define como “la habilidad para trazar la historia, aplicación o localización de lo que se esté considerando” (ISO 9001:2000).

En definitiva, los sistemas de trazabilidad permiten el seguimiento y la localización del producto a lo largo de la cadena comercial y se utilizan principalmente en el sector alimenticio. La importancia de la trazabilidad como mecanismo para el seguimiento y conocimiento de la historia de un alimento ha sido ampliamente reconocida por distintas organizaciones internacionales (FAO y la OMS) y ha dado lugar a la aparición del Reglamento CE nº 178/2002, el cual obliga desde el año 2005 a establecer un sistema de trazabilidad de alimentos.

La trazabilidad aplicada a la gestión de los residuos permite tener un seguimiento de la vida de un residuo, aportando información suficiente para saber todas las vías que se han seguido hasta su depósito final.

## 1.2 Objetivos

Por este motivo, el trabajo que se va a desarrollar es un software que permita la trazabilidad de los residuos de construcción y demolición, similar al utilizado en el sector alimenticio, y que garantice la correcta gestión y control de los RCD generados en una obra de edificación.

Además, permitirá conectar los distintos agentes intervinientes en la gestión de RCD, y recogerá los documentos, e información adicional que acrediten el proceso que siguen los residuos desde la obra hasta su depósito final.

## 1.3 Justificación

Con este proyecto se pretende modernizar los sistemas de gestión de residuos, convirtiéndolos en un proceso más cómodo, rápido, seguro y eficiente. Al ser un sistema que se va completando en cada acción que realiza cada actor del proceso, es posible llevar una monitorización en tiempo real de la traza que siguen los RCD.

Como consecuencia, este sistema ayudará no solo a los promotores cuando presenten los justificantes necesarios para la devolución de la fianza, sino también ayudará a las empresas constructoras a conocer el residuo que genera su actividad, y a las administraciones locales y funcionarios encargados del control de la gestión de RCD, determinando con más precisión la responsabilidad de los vertidos ilegales o incluso los incidentes ocasionados con los residuos tóxicos.



## 1.4 Alcance

El proyecto consistirá en una aplicación que dé soporte vía web, y muestre, en función del usuario, las posibilidades para interactuar con el sistema, ya sea incluyendo información nueva o consultándola. El sistema permitirá la comunicación entre los distintos roles del sistema, mostrando la información en tiempo real.

Esta información se centrará en el seguimiento de los residuos, y estos estarán asociados a una obra en concreto. El resultado final será, acreditar que todos los residuos de una obra se gestionan correctamente mediante un gestor autorizado, y para ello el sistema será capaz de comparar los RCD estimados en una obra determinada con los gestionados al finalizar la misma, a fin de asegurar que todos ellos han sido gestionados de acuerdo a las leyes vigentes.

Además de llevar la gestión en tiempo real del proceso de gestión, el sistema almacenará un registro detallado de fechas de cada actividad, que permanecerán una vez finalizada la obra.

## Capítulo 2 - Estado del Arte

En este capítulo se detalla el estudio de las tecnologías actuales que resultan de interés para el correcto desarrollo del proyecto.

### 2.1 Trazabilidad

Se entiende trazabilidad como el conjunto de aquellos procedimientos preestablecidos y autosuficientes que permiten conocer el histórico, la ubicación y la trayectoria de un producto o lote de productos a lo largo de la cadena de suministros en un momento dado, a través de unas herramientas determinadas [2].

La trazabilidad permite conocer el origen de los materiales y los componentes de un producto. Sin embargo, este concepto se puede aplicar a otros entornos, en general refleja el estado de un elemento en un proceso. Así mismo la trazabilidad registra la historia en el proceso y la ubicación actual del elemento en cualquier punto del mismo.

#### 2.1.1 Características de la trazabilidad

La trazabilidad requiere procedimientos documentados orientados a la identificación de todo producto que se encuentre bajo la responsabilidad de cada operador. De esta manera recopila y compara información del proceso, de sus elementos y las relaciones entre ellos.

Gracias a la información recopilada, llevar un buen sistema de trazabilidad permite la adopción de medidas eficaces y el control de las distintas actividades, contribuyendo a mantener la transparencia necesaria.

Aunque en la mayoría de las ocasiones la trazabilidad es una respuesta voluntaria para la mejora y la transparencia de los procesos, en otros casos se trata de una actividad obligada, para cumplir las especificaciones legales que se imponen a la actividad que monitoriza.

#### 2.1.2 Importancia y ventajas

El seguimiento de la vida de elemento en un proceso puede aportar información suficiente para conocer todos los estados por los que ha pasado, incluyendo su ubicación y relación con otros elementos, hasta el final de su vida.

Como consecuencia, ayuda a determinar con más precisión la responsabilidad de defectos o de problemas de seguridad en el proceso. Del mismo modo, si se produjese un accidente se podrían localizar de forma fácil y precisa no sólo los elementos afectados, sino todos aquellos elementos que puedan haber estado en el origen del problema.

### 2.1.3 Tipos de Trazabilidad

Existen tres tipos de trazabilidad:

#### Hacia atrás

Se refiere a la recepción de elementos. En este tipo de trazabilidad se busca conocer las etapas y los estados por los que ha pasado el elemento hasta llegar al estado en el que se encuentra cuando se solicita esta información.

Los registros mínimos que se deben guardar para conocer esta información son:

- ✓ Etapa anterior, o de quien se recibe el producto.
- ✓ Qué se ha recibido exactamente. Recopilar toda la información referente al producto recibido que se quiera consultar.
- ✓ Cuando se ha recibido el producto.
- ✓ Que se hizo con el producto.

#### De proceso

Es la trazabilidad interna a una organización o empresa. Se refiere a los procesos internos de la organización, y cómo gestionan los productos. Podría entenderse como un proceso de trazabilidad separado de los demás. De esta manera este proceso separado es, en su totalidad, una etapa de un proceso de trazabilidad mas grande, que controla los elementos que le llegan, y los que salen.

#### Hacia adelante

Se refiere al envío de elementos. En este tipo de trazabilidad se busca conocer las etapas y los estados por los que va a pasar el elemento desde un punto determinado.

Los registros mínimos que se deben guardar para conocer esta información son:

- ✓ Etapa siguiente, o a quien se entrega el producto.
- ✓ Que se entrega exactamente. Recopilar toda la información referente al producto enviado que se quiera consultar.
- ✓ Cuando se envía el producto.
- ✓ Medio de transporte que se utiliza para el envío.

Para el caso particular de este proyecto, se trabaja con trazabilidad hacia atrás y hacia adelante, según los actores que participen. Cuando un contenedor sale de una obra, se genera y se sigue una trazabilidad hacia adelante, hasta que los RCD son tratados en la planta. Una vez terminada la obra y cuando se pretende conseguir las licencias, se sigue una trazabilidad hacia atrás para conocer los estados por los que los RCD han pasado.

## 2.2 Herramientas Web

Desarrollo web es un término que define la creación de sitios web para Internet o una intranet, haciendo uso de tecnologías de software del lado del servidor y del cliente, que involucran una combinación de procesos de base de datos con el uso de un navegador web, a fin de realizar determinadas tareas o mostrar información.[3]

Para el desarrollo web existen múltiples opciones. Sin embargo, todas ellas se basan en los mismos componentes:

- ✓ Servidor de base de datos
- ✓ Servidor Web

Estos dos sistemas que funcionan en conjunto permiten ofrecer una aplicación web que trabaje sobre unos datos que se almacenan en una base de datos diseñada a medida, para el caso particular.

### 2.2.1 Servicio Web

En la gestión del desarrollo web se emplean las siguientes tecnologías:

- ✓ Servidor Web que genere páginas siguiendo una lógica establecida.
- ✓ HTML 5 para definir los contenidos en la página.
- ✓ JavaScript para definir el comportamiento de uso de la página.
- ✓ CSS 3 para definir los estilos.

Un servidor web que gestione la aplicación y proporcione acceso es la base del servicio web. En este servidor se almacena documentación en html que conforma la estructura de las páginas, y css que aporta los estilos a los elementos introducidos con html.

Sin embargo, la mayoría de los sitios web proporcionan interfaces que se construyen de forma dinámica en función de otra información proporcionada, bien por el usuario o tomada de una base de datos que va cambiando en el tiempo. Esto hace que las páginas almacenadas en html se vuelvan obsoletas en poco tiempo, y gestionar los cambios se vuelve insostenible.

Como solución a este problema se necesita utilizar otro tipo de tecnología que permita insertar lógica ejecutable dentro del código en html. Esto permite la generación dinámica en tiempo real de una página personalizada y ajustada al momento de su solicitud sin necesidad de incluir los cambios en el código html.

Con esta posibilidad, el servidor es capaz de imprimir información obtenida en tiempo real de la base de datos, sobre el código html. Esto permite reutilizar código y generarlo dinámicamente de forma sencilla, en el propio servidor web, para posteriormente enviarlo al software cliente (el navegador).

En la tabla 1 se puede ver una breve descripción de las características de las tecnologías de desarrollo web más conocidas, mostrando sus ventajas e inconvenientes.

Lenguaje	Características	Fortalezas	Debilidades
<b>PHP</b>	<ul style="list-style-type: none"> <li>• Utilizado para generar páginas web dinámicas</li> <li>• Se ejecuta en el servidor</li> <li>• Los usuarios no pueden ver el código PHP únicamente reciben en sus navegadores código HTML</li> <li>• Las páginas que genera son visibles para prácticamente cualquier navegador y computadora o dispositivos móviles que pueda interpretar el HTML.</li> <li>• No se necesita la instalación de PHP en el lado del cliente.</li> <li>• Versiones reciente permiten la POO</li> <li>• Lenguaje de alto nivel</li> </ul>	<ul style="list-style-type: none"> <li>• Su sintaxis es muy similar a otros lenguajes</li> <li>• Fácil</li> <li>• Es un lenguaje muy popular tiene una comunidad muy grande</li> <li>• Rápido</li> <li>• Multiplataforma</li> <li>• Maneja base de datos</li> <li>• Bastante documentado</li> <li>• Libre y gratuito.</li> <li>• Varias funciones</li> <li>• No requiere definición de variables</li> <li>• Puede ser combinado junto a HTML</li> <li>• Tiene muchos frameworks que facilitan el desarrollo en este lenguaje.</li> <li>• Muchos servicios de alojamiento web lo tienen.</li> </ul>	<ul style="list-style-type: none"> <li>• Necesita un servidor para funcionar.</li> <li>• La POO es deficiente para aplicaciones grandes.</li> <li>• Todo el trabajo se realiza en el servidor y mucha información o solicitudes pueden ser ineficiente.</li> </ul>
<b>RUBY</b>	<ul style="list-style-type: none"> <li>• Orientado a objetos</li> <li>• Lenguaje de alto nivel</li> <li>• Sintaxis similar a Python y Perl</li> <li>• Opensource</li> <li>• Lenguaje para la creación de aplicaciones de escritorio y aplicaciones web.</li> </ul>	<ul style="list-style-type: none"> <li>• Diferencia entre mayúsculas y minúsculas</li> <li>• Maneja excepciones</li> <li>• Puede cargar librerías si el sistema operativo lo permite</li> <li>• Multiplataforma</li> <li>• Portátil</li> <li>• Desarrollo de bajo costo</li> <li>• Software libre</li> <li>• Multiplataforma</li> </ul>	<ul style="list-style-type: none"> <li>• Es relativamente nuevo y no cuenta con mucha documentación en comparación con otros lenguajes de programación o está muy difundido en relación a otros lenguajes.</li> </ul>
<b>JSP (Java Server Pages)</b>	<ul style="list-style-type: none"> <li>• Lenguaje para creación de sitios dinámicos</li> <li>• Necesita un servidor Tomcat</li> <li>• Motor basado en servlets de java</li> <li>• Multiplataforma</li> </ul>	<ul style="list-style-type: none"> <li>• ejecución rápida de servlets.</li> <li>• Código bien estructurado</li> <li>• Integridad con módulos java</li> <li>• La parte dinámica está escrita en java</li> </ul>	<ul style="list-style-type: none"> <li>• Complejidad de aprendizaje</li> </ul>
<b>ASP (Active Server Pages)</b>	<ul style="list-style-type: none"> <li>• Desarrollado por Microsoft</li> <li>• Tecnología del lado del servidor</li> <li>• Requiere de Internet Información Server (IIS)</li> </ul>	<ul style="list-style-type: none"> <li>• Usa visual Basic script</li> <li>• Comunicación optima con SQL server</li> <li>• Soporta JScript</li> </ul>	<ul style="list-style-type: none"> <li>• De paga</li> <li>• El hospedaje de sitios web es costos</li> <li>• Necesita de mucho código para funciones sencillas</li> </ul>
<b>ASP.ET</b>	<ul style="list-style-type: none"> <li>• Sucesor de ASP</li> <li>• Creada por Microsoft</li> <li>• De paga</li> <li>• Orientado a objetos</li> </ul>	<ul style="list-style-type: none"> <li>• Controles de usuarios y personalizados</li> <li>• Fácil mantenimiento</li> <li>• Incremento en velocidad</li> <li>• Mayor seguridad</li> </ul>	<ul style="list-style-type: none"> <li>• Mayor consumo de recursos</li> </ul>

Lenguaje	Características	Fortalezas	Debilidades
<b>Python</b>	<ul style="list-style-type: none"> <li>• Permite la creación de todo tipo de programas incluso sitios web</li> <li>• No requiere de compilación es un código interpretado</li> </ul>	<ul style="list-style-type: none"> <li>• Libre y código fuente abierto</li> <li>• Lenguaje de propósito general</li> <li>• Multiplataforma</li> <li>• Orientado a objetos</li> <li>• Portable</li> </ul>	<ul style="list-style-type: none"> <li>• Los lenguajes interpretados suelen ser relativamente lentos</li> </ul>
<b>JavaScript</b>	<ul style="list-style-type: none"> <li>• Es un lenguaje interpretado</li> <li>• Es similar a java</li> <li>• Es orientado a objetos</li> </ul>	<ul style="list-style-type: none"> <li>• Los scripts tienen capacidad limitada por razones de seguridad</li> <li>• Se ejecuta del lado del cliente</li> <li>• Lenguaje de scripting seguro y fiable.</li> </ul>	<ul style="list-style-type: none"> <li>• No soporta herencias</li> <li>• Código visible por cualquier usuario</li> <li>• El código debe ser descargado completamente</li> <li>• Puede poner en riesgo la seguridad del sitio con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).</li> </ul>
<b>C++</b>	<ul style="list-style-type: none"> <li>• Orientado a objetos</li> <li>• Rápido</li> </ul>	<ul style="list-style-type: none"> <li>• Ideal para sistemas robustos.</li> <li>• IDEs de desarrollo son DEV C++, BORLAND C, TURBO C</li> <li>• Es multiplataforma</li> </ul>	<ul style="list-style-type: none"> <li>• No soporta creación de aplicaciones web</li> <li>• Complejo visualmente</li> </ul>
<b>C</b>	<ul style="list-style-type: none"> <li>• Popular para la creación de software de sistema</li> </ul>	<ul style="list-style-type: none"> <li>• Rápido</li> <li>• Eficiente</li> <li>• Es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix</li> </ul>	<ul style="list-style-type: none"> <li>• No es popular para la creación de aplicación</li> <li>• Sintaxis compleja</li> </ul>
<b>C#</b>	<ul style="list-style-type: none"> <li>• Está orientado a objetos</li> <li>• Esta estandarizado por Microsoft como parte de su plataforma net.</li> </ul>	<ul style="list-style-type: none"> <li>• Se desempeña de forma plena en los sistemas operativos Windows.</li> <li>• Sintaxis más en comparación con C y C++</li> <li>• Posibilidad de realizar aplicaciones web, de escritorio y móviles.</li> </ul>	<ul style="list-style-type: none"> <li>• Requiere un mínimo de 4 gb para su instalación.</li> </ul>
<b>Java</b>	<ul style="list-style-type: none"> <li>• Es orientado a objetos</li> <li>• Multiplataforma</li> </ul>	<ul style="list-style-type: none"> <li>• Al ser orientado a objetos permite su modularización.</li> <li>• Permite la creación de aplicaciones de escritorio Tiene soporte a desarrollo de aplicaciones móviles y web.</li> </ul>	<ul style="list-style-type: none"> <li>• Es un lenguaje interpretado así que es relativamente lento en comparación con otros lenguajes.</li> </ul>

Tabla 1 – Comparativa de tecnologías para desarrollo web. [4]

En la tabla 2 se muestra una comparativa entre dos de los servidores web más utilizados para alojar sitios web.

Servidor	Características	Fortalezas	Debilidades
<b>Servidor http Apache</b>	Apache presenta entre otras características altamente configurables bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.	<ul style="list-style-type: none"> <li>• Altamente configurable</li> <li>• Estabilidad</li> <li>• Independencia de la plataforma</li> <li>• Código abierto</li> </ul>	<ul style="list-style-type: none"> <li>• Complejidad de formatos de configuración no estándar</li> <li>• Falta de integración Administración</li> </ul>
<b>Internet Information Services (IIS)</b>	Es una aplicación de servidor web y un conjunto de módulos de extensión de funciones creadas por Microsoft para su uso con Microsoft Windows. IIS 7.5 soporta HTTP, HTTPS, FTP, FTPS, SMTP y NNTP	<ul style="list-style-type: none"> <li>• Ofrece el mejor rendimiento porque la autenticación anónima no impone una sobrecarga apreciable.</li> <li>• No requiere administración de cuentas de usuario concretas.</li> </ul>	<ul style="list-style-type: none"> <li>• No autentica clientes individualmente.</li> <li>• Si IIS no controla la contraseña, la cuenta debe tener la capacidad de un inicio de sesión local.</li> </ul>

Tabla 2 – Comparativa de servidores web. [5]

### 2.2.2 Bases de Datos

Actualmente existen múltiples posibilidades y tecnologías para implementar base de datos en aplicaciones web. Las bases de datos que permiten consultas dinámicas a través de un conector, que permita a un software para desarrollo web realizar una conexión a la base de datos y realizar consultas, emplean lenguaje SQL.

Esta tecnología permite la gestión y acceso a bases de datos relacionales, que son extensamente utilizadas en muchos ámbitos, como el desarrollo web. Para poder realizar la implementación de la base de datos es necesario contar con un gestor de bases de datos relacionales, que reciba las conexiones y consultas y las ejecute en la base de datos, para después poder ofrecer los resultados.

En la tabla 3 se muestra una comparativa de dos de los gestores más utilizados en los modelos de bases de datos relacionales.

Gestor	Características	Ventajas	Desventajas
<b>MariaDB</b>	Nace como una alternativa libre a MySQL. Mantener un base de datos libre, garantizando la licencia GPL.	<ul style="list-style-type: none"> <li>• Nuevos motores de almacenamiento más eficientes.</li> <li>• Estadísticas para índices y tablas.</li> <li>• Columnas dinámicas</li> <li>• Mejor funcionamiento en general.</li> </ul>	Migración de sistemas ya existentes
<b>MySQL</b>	MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual. Es considerado el gestor mas popular del mundo.	<ul style="list-style-type: none"> <li>• Velocidad al realizar las operaciones.</li> <li>• Posee gran Fiabilidad</li> <li>• Facilidad de uso</li> <li>• Licencia GPL</li> <li>• Tiene mejor conectividad y seguridad</li> </ul>	<ul style="list-style-type: none"> <li>• No es intuitivo</li> <li>• Existe poca documentación</li> </ul>

Tabla 3 – Comparativa de gestores de bases de datos. [6][7]

### 2.2.3 Visualización Web

Para el caso contrario, se emplea JavaScript. Es un lenguaje de script que se ejecuta en el propio navegador, y que permite una gran cantidad de posibilidades. Una de ellas y la más básica, es recoger información que el usuario introduce en los formularios, y reenviarla de vuelta al servidor, ejecutando de nuevo otro código en PHP que trate la información, por ejemplo, introduciéndola en la base de datos.

Esta parte de la ejecución se realiza en la máquina cliente, y el software encargado de ejecutarlo se conoce como navegador web. Dentro del navegador se ejecuta tanto el código html, como el css y javascript. Esto permite la correcta visualización y funcionamiento de la página en la pantalla del usuario.

Los navegadores que se utilizan variados, pero en general se distinguen en seis grandes navegadores, dependiendo de las preferencias del usuario.

- ✓ Microsoft Edge
- ✓ Internet Explorer
- ✓ Opera
- ✓ Google Chrome
- ✓ Mozilla Firefox
- ✓ Safari

Aunque cada navegador tiene características propias, todos trabajan de la misma manera, y son compatibles en la mayoría de los casos, ya que trabajan sobre los mismos lenguajes. Dado que cada usuario usa navegadores diferentes, y las similitudes son muy



abundantes, se hace necesario desarrollar la aplicación compatible con todos los navegadores.

#### 2.2.4 Mapas

Por otro lado, en muchos sitios web, se incluyen mapas para mostrar localizaciones. Esto se implementa mediante frames incrustados en el código html. Existen librerías de mapas gratuitos para poder mostrar localizaciones. Para la aplicación a la trazabilidad, este tipo de herramientas es muy interesante, ya que permite mostrar la ubicación de los escenarios donde tiene lugar el tratamiento de los elementos involucrados en la cadena.

Para este proyecto resulta interesante para mostrar ubicaciones, tanto de obras como de plantas de gestión de RCD, o incluso para poder observar de un vistazo que otros usuarios se encuentran cerca de una localización concreta.

Actualmente existen diversas librerías que contienen mapas para su inserción.

- ✓ OpenLayer
- ✓ Leaflet
- ✓ Google Maps

En la tabla 4 se muestra una comparativa de las diferentes tecnologías para mostrar mapas incrustados en código html.

Tecnología	Características	Ventajas	Desventajas
<b>Google Maps</b>	Permite crear aplicaciones completas y visualizaciones de los datos, aprovechando la estructura de Google Maps.	<ul style="list-style-type: none"> <li>• Gratis</li> <li>• Introducción de direcciones a través de la Api</li> <li>• Integración en html mediante un marco.</li> <li>• Posibilidad de compartir las direcciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Uso de condiciones impuestas por Google.</li> <li>• Se muestra el logo de Google</li> </ul>
<b>Leaflet</b>	Es una biblioteca de código abierto de JavaScript para mapas interactivos optimizados para dispositivos móviles.	<ul style="list-style-type: none"> <li>• Gratis</li> <li>• Elección del proveedor de mapas</li> <li>• Actualizable mediante plugins</li> <li>• Comunicación Fuerte</li> </ul>	Uso sobre Javascript
<b>Openlayers</b>	Es una biblioteca javascript opensource para cargar, visualizar y renderizar mapas de múltiples fuentes en páginas web.	<ul style="list-style-type: none"> <li>• Elección del proveedor de mapas.</li> <li>• Flexible</li> <li>• Buen soporte</li> <li>• Extensible</li> </ul>	Uso sobre Javascript

Tabla 4 – Comparativa de servidores web. [9]

### 2.2.5 Elección para el proyecto

Para este proyecto se ha decidido utilizar la tecnología de desarrollo de páginas dinámicas con PHP, por su facilidad de uso, licencia libre, y gran soporte. Para la gestión de la base de datos se empleará MySQL, por la gran compatibilidad que ofrece con PHP. En concreto se va a utilizar la herramienta de Bitnami (Lamp para servicios Linux), que incorpora ambas tecnologías y ofrece una instalación rápida y sencilla. [10]

Para la implementación de los mapas se ha optado por usar la tecnología de Google, por su facilidad de uso sobre el código html, y la gran estabilidad y compatibilidad que proporciona.

## Capítulo 3 - Metodología del desarrollo

Para el desarrollo de este proyecto se ha elegido seguir una metodología en cascada. Esta metodología consiste en el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. [11]

Esta metodología es la mas tradicional, y se ajusta perfectamente a proyectos pequeños, con las metas y objetivos claramente definidos. Por esta razón se ha considerado que es la mas adecuada para este proyecto, ya que permite una organización clara en cada fase, planificando cada tarea a realizar en cada momento.

A la hora de ejecutar el proceso, se han establecido reuniones del alumno con ambas tutoras cada dos semanas para revisar el progreso del proyecto, y discutir los diferentes aspectos de cada fase, y los problemas encontrados.

La figura 5 muestra un diagrama de flujo de todas las actividades que se van a llevar a cabo a lo largo de la ejecución del proyecto. En él se pueden distinguir las diferentes fases del proyecto, y las respectivas partes en las que se basan.

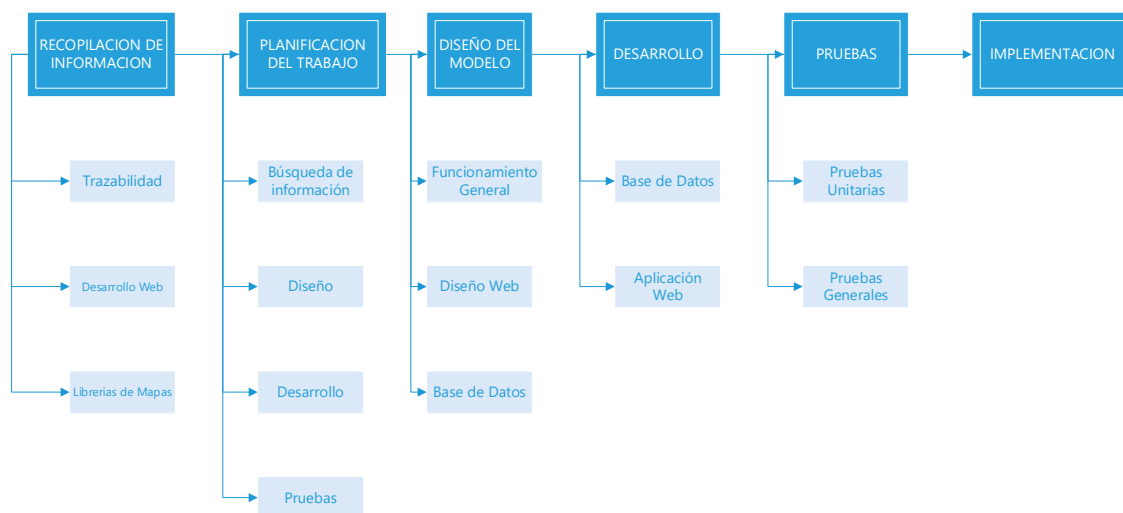


Figura 5 – Diagrama de flujo de trabajo.

Una vez definido cual es el objetivo a seguir y se ha hecho un estudio y recopilación de la información necesaria, la siguiente fase consiste en planificar la ejecución del proceso, y definir las fases que se van a seguir y las tareas a realizar en cada fase.

El proyecto se dividirá en las siguientes fases:

- ✓ Diseño
- ✓ Desarrollo
- ✓ Pruebas

Previamente a estas fases se ha definido el objetivo, y la información que buscar. Las fases de búsqueda de información y diseño junto con la planificación son imprescindibles para el correcto desarrollo del proyecto. Una vez en la fase de desarrollo, realizar cambios en estas fases tiene un alto coste, ya que supondría volver a empezar desde el punto que se modifique.

### 3.1 Diseño del modelo

En esta fase se realiza el modelo de requisitos que detalle las necesidades a cubrir por la aplicación. Una vez definidos se estudia el diseño de la aplicación. Esto comprende las partes en las que se va a componer la aplicación, y como se van a implementar.

En esta fase también se va a detallar como va a ser el funcionamiento de la aplicación, cuales van a ser sus opciones y como distribuirlas en las paginas que van a componer la aplicación.

### 3.2 Desarrollo

La fase de desarrollo comprende la realización del proyecto y creación de la aplicación para que cumpla con los requisitos establecidos en las fases anteriores. Es en esta fase se deben implementar las tecnologías estudiadas para los propósitos que se han elegido.

En primer lugar, se implementa la base de datos que contendrá la información, posteriormente se instalará el servidor que soportará la aplicación, y la conexión a la base de datos. Después se comienza con el desarrollo de la aplicación.

En esta fase se van realizando, según se vayan completando los módulos que la conforman, las pruebas unitarias más básicas para dar por finalizados los módulos según se vayan implementando.

### 3.3 Pruebas

En esta fase, toda la aplicación debe estar finalizada, y se pasa a la fase de comprobación del funcionamiento de la aplicación. Si en esta fase se detectan errores en el funcionamiento será necesario solucionar el problema o volver a implementarlo, dependiendo de la naturaleza del error. La fase de pruebas se divide en dos partes esenciales.

En primer lugar, se deben comprobar las pruebas unitarias, verificar que los requisitos más específicos se cumplen correctamente. En esta fase la mayoría de las pruebas unitarias ya se han comprobado previamente en la fase de desarrollo, pero es necesario volver a realizar las pruebas para verificar que en módulos posteriores no se han realizado cambios que afecten a otros módulos anteriores.

Una vez realizadas las pruebas unitarias, se pasan a hacer las pruebas generales. Estas pruebas son aquellas que verifican el comportamiento de la aplicación con todos los módulos interconectados. El objetivo es asegurar las dependencias de unos módulos con otros.

Después de la fase de pruebas, cuando se certifica que el sistema funciona como se espera y los requisitos se cumplen, se pasaría a la fase final de implementación, donde se instala el sistema en un escenario real, y se pasa al estado de producción, donde personas reales explotan la aplicación. Esta fase no se va a contemplar en este proyecto.

## Capítulo 4 – Diseño

En este capítulo se va a tratar el diseño del sistema a desarrollar. En primer lugar, se detallan los requisitos que debe cumplir la aplicación, seguido de los elementos en los que se va a componer y que van a dar cuerpo a la aplicación. Después se identifican los módulos en los que se va a componer y se planifica su orden de desarrollo. Por último, se detalla el diseño de la base de datos.

### 4.1 Requisitos

En la fase de diseño se desarrollan los aspectos que van a definir la aplicación y su comportamiento. Los requisitos explican y detallan el funcionamiento del sistema. Consta de una serie de puntos de que detallan las necesidades que debe cubrir el sistema, y el funcionamiento del mismo. Sin embargo, en ningún momento deben especificar como implementar ninguna funcionalidad del sistema, sino simplemente exponerla detalladamente.

Para comprender mejor estas necesidades, los requisitos se dividen en grupos dependiendo de su naturaleza. En este trabajo se van a considerar requisitos no funcionales, requisitos funcionales, requisitos del usuario y requisitos del sistema. [12]

#### 4.1.1 Requisitos No funcionales

Los requisitos no funcionales son aquellos que no se refieren directamente a las funcionalidades del sistema, pero que sí debe cumplir. Se suelen representar como restricciones en el comportamiento del sistema, como aspectos en la seguridad, tiempo de ejecución, interfaces gráficas etc.

<b>Requisito 8</b>	<b>Requisito No Funcional</b>
<b>Descripción</b>	Las constructoras solo podrán ver la información de las obras de sus responsables.
<b>Razón</b>	Por motivos de seguridad.
<b>Dependencias</b>	Requisito 1 <span style="float: right;">Prioridad: 3</span>

<b>Requisito 9</b>	<b>Requisito No Funcional</b>
<b>Descripción</b>	Los responsables solo podrán ver la información de sus obras.
<b>Razón</b>	Por motivos de seguridad.
<b>Dependencias</b>	Requisito 1 <span style="float: right;">Prioridad: 3</span>

<b>Requisito 10</b>	<b>Requisito No Funcional</b>
<b>Descripción</b>	Las gestoras y los fabricantes solo podrán ver la información de los contenedores recogidos.
<b>Razón</b>	Por motivos de seguridad
<b>Dependencias</b>	Requisito 1 <span style="float: right;">Prioridad: 3</span>

Tabla 5 – Requisitos No Funcionales.

#### 4.1.2 Requisitos Funcionales

Los requisitos funcionales son aquellos que detallan lo que el sistema debe hacer. Son requisitos que expresan la funcionalidad y objetivos del sistema.

<b>Requisito 1</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Deben existir diferentes roles: Constructora, responsable de obra, Fabricante, y gestor de residuos.
<b>Razón</b>	Existen diferentes actores que usan el sistema.
<b>Dependencias</b>	- Prioridad: 3
<b>Requisito 2</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Los responsables deben estar asociados a una constructora, y esta debe darlos de alta.
<b>Razón</b>	Una constructora tiene responsables de obra, y estos son contratados por la constructora.
<b>Dependencias</b>	- Prioridad: 3
<b>Requisito 3</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Constructoras, gestoras y fabricantes deben poder registrarse directamente.
<b>Razón</b>	Para facilitar el registro ya que no existe un rol de administrador.
<b>Dependencias</b>	- Prioridad: 2
<b>Requisito 4</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Se deben almacenar obras existentes.
<b>Razón</b>	Se debe identificar las obras.
<b>Dependencias</b>	- Prioridad: 3
<b>Requisito 5</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Se deben almacenar gestoras existentes.
<b>Razón</b>	Se debe identificar las gestoras.
<b>Dependencias</b>	- Prioridad: 3
<b>Requisito 6</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Se deben almacenar los contenedores de cada obra.
<b>Razón</b>	Se debe identificar los contenedores.
<b>Dependencias</b>	- Prioridad: 3
<b>Requisito 7</b>	<b>Requisito Funcional</b>
<b>Descripción</b>	Se deben almacenar las partidas de cada obra.
<b>Razón</b>	Se debe identificar las partidas.
<b>Dependencias</b>	- Prioridad: 3

Tabla 6 – Requisitos Funcionales.

### 4.1.3 Requisitos del Usuario

Los requisitos de usuario son aquellos que reflejan el comportamiento externo del sistema, desde el punto de vista del usuario. Se trata de requisitos sin expresiones técnicas y claros, de manera que cualquier usuario pueda comprenderlos sin necesidad de conocimientos avanzados.

<b>Requisito 13</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Las constructoras deben dar de alta las obras en el sistema.
<b>Razón</b>	Son las constructoras las que contratan las obras.
<b>Dependencias</b>	- Prioridad: 3
<b>Requisito 14</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los responsables deben registrar las partidas.
<b>Razón</b>	Son los responsables los que gestionan los materiales, y se debe almacenar los materiales que entran para calcular los residuos.
<b>Dependencias</b>	Requisito 13 Prioridad: 3
<b>Requisito 15</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los responsables deben generar los contenedores, y asociarlos a un gestor.
<b>Razón</b>	Son los responsables los que gestionan los contenedores que generan, y van a una planta de reciclaje.
<b>Dependencias</b>	Requisito 13 Prioridad: 3
<b>Requisito 16</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los responsables deben introducir las fechas de sus obras.
<b>Razón</b>	Son los responsables los que gestionan la obra.
<b>Dependencias</b>	Requisito 13 Prioridad: 1
<b>Requisito 17</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los responsables deben cerrar las obras cuando terminen.
<b>Razón</b>	Son los responsables los que gestionan la obra, y recogen los certificados de tratamiento de residuos.
<b>Dependencias</b>	- Prioridad: 1
<b>Requisito 18</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los gestores deben registrar sus plantas de gestión de residuos.
<b>Razón</b>	Los gestores suelen tener varias plantas de tratamiento.
<b>Dependencias</b>	- Prioridad: 2
<b>Requisito 19</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los gestores deben poder recoger los contenedores, marcando los contenedores como recogidos.
<b>Razón</b>	Son los gestores los que certifican que los residuos se tratan adecuadamente.
<b>Dependencias</b>	Requisito 15 Prioridad: 3
<b>Requisito 20</b>	<b>Requisito de Usuario</b>
<b>Descripción</b>	Los fabricantes deben poder aceptar contenedores con residuos.
<b>Razón</b>	En ocasiones los fabricantes aceptan residuos.
<b>Dependencias</b>	- Prioridad: 1

Tabla 7 – Requisitos del Usuario.





Por cada obra recopilará la siguiente información:

- ✓ Cantidad total de residuos generados en m<sup>3</sup> y kg.
- ✓ Cantidad total de residuos por metros construidos.
- ✓ Cantidad total de residuos por tipo de obra.
- ✓ Cantidad total de residuos por tipo de proyecto.
- ✓ Cantidad total de residuos por tipo de estructura.
- ✓ Cantidad total de residuos por tipo de fachada.
- ✓ Cantidad total de residuos por tipo de partición interior.
- ✓ Cantidad total de residuos por fase de construcción.

Por todas las obras se calculará la media de todas las obras de la siguiente información:

- ✓ Cantidad total de residuos por tipo de obra.
- ✓ Cantidad total de residuos por tipo de proyecto.
- ✓ Cantidad total de residuos por tipo de estructura.
- ✓ Cantidad total de residuos por tipo de fachada.
- ✓ Cantidad total de residuos por tipo de partición interior.

#### *4.2.1.2 Responsable de obra*

Los responsables son los empleados de la constructora que se encargan de la gestión de la obra. Debe ser el constructor quien, de alta a sus responsables, ya l asignarles una obra estos se convierten en administradores de la obra.

El responsable debe gestionar la obra, asignando las fechas de cada fase, e insertando las primas de material que lleguen a la obra en el sistema, para que quede constancia. También deben registrar los contenedores que se contratan y una vez estén llenos asignarlos a una planta gestora de residuos para que sea tratado correctamente. Una vez finalizada la obra deberán cerrar la obra en el sistema.

Además de esta labor pueden consultar información de todas sus obras de la misma manera que los constructores.

#### *4.2.1.3 Gestor de Residuos*

El gestor de residuos es una empresa autorizada para tratar los residuos generados en las obras de construcción, y para emitir certificados a los responsables de obra que verifiquen que los residuos han sido tratados correctamente.

Cuando un responsable asocia un contenedor a una planta de reciclaje, esta envía un transportista a recoger el contenedor, y entrega otro vacío. El contenedor lleno se lleva de vuelta a la planta donde se pesa y se termina de rellenar la información correspondiente. Una vez aceptado, el gestor registra que ha sido tratado correctamente.

El gestor podrá además visualizar información acerca de las obras con las que tenga contrato. Por cada obra podrá visualizar la siguiente información:

- ✓ Quien es el responsable de la obra.
- ✓ Ubicación de la obra
- ✓ Cuales son los contenedores que ha generado, y cuales ha aceptado.

Además, podrá gestionar sus plantas de gestión de residuos, visualizando su ubicación en un mapa. De esta manera los responsables podrán visualizar las plantas mas cercanas a sus obras y ahorrar costes en transporte.

#### 4.2.1.4 Fabricante

El rol de fabricante es muy similar al de gestor de residuos. En ocasiones los fabricantes deciden recibir contenedores de residuos que contienen materiales que les pueden interesar, bien sea material sobrante o material reciclable para su producción. Por este motivo se ha decidido incluir la opción a los fabricantes de recibir contenedores con residuos directamente de las obras.

En este caso los fabricantes podrán desempeñar el papel de una gestora de residuos, y su perfil en el sistema tendrá las mismas operaciones.

#### 4.2.2 Otros elementos

Además de los roles con usuario en la aplicación, es necesario crear otra serie de elementos en el sistema que soporten el correcto funcionamiento del mismo. Estos elementos son parte del proceso, sin embargo, carecen de usuario activo, ya que son tratados por los elementos que si representan al usuario del sistema.

##### 4.2.2.1 Obras

Las obras representan, los proyectos que llevan a cabo las constructoras. Tienen asociado un responsable que es quien las gestiona y es responsable de las obras. Cada obra contiene la siguiente información.

- ✓ Responsable a cargo de la obra.
- ✓ Título descriptivo de la obra. Sirve para identificar las obras.
- ✓ Dirección de la obra.
- ✓ Coordenadas, latitud y longitud para ubicar la obra en el mapa.
- ✓ El tipo de edificación, que distingue entre Residencial Unifamiliar, Residencial en altura, Sector terciario y otros.
- ✓ El tipo de obra, que distingue entre Rehabilitación, Demolición, Nueva y Civil.
- ✓ El tipo de cimentación que distingue entre Pilotes, Zapatas, Losas y otros.
- ✓ El tipo de fachada que distingue entre Prefabricada, Panelada, Ladrillo y otros.

- ✓ El tipo de estructura que distingue entre Hormigón, Metálica y Madera.
- ✓ El tipo de partición interior que distingue entre Ladrillo, Gran Formato y Yeso.
- ✓ Número de viviendas de la obra.
- ✓ Número de plantas sobre suelo.
- ✓ Número de plantas bajo ceros.
- ✓ Superficie construida en metros cuadrados.
- ✓ Porcentaje de zonas comunes construidas.
- ✓ Fecha de inicio de la obra
- ✓ Fecha de Fin de la obra.
- ✓ Estimación de residuos generados.
- ✓ Fecha de inicio de la fase de estructura.
- ✓ Fecha de fin de la fase de estructura.
- ✓ Fecha de inicio de la fase de albañilería.
- ✓ Fecha de fin de la fase de albañilería.
- ✓ Fecha de inicio de la fase de acabados 1.
- ✓ Fecha de fin de la fase de acabados 1.
- ✓ Fecha de inicio de la fase de acabados 2.
- ✓ Fecha de fin de la fase de acabados 2.

Al crear la obra, el constructor, añade al responsable, el título, dirección, y todas las características de la obra, y el responsable se encarga de introducir las fechas de cada fase.

#### 4.2.2.2 Partidas

Además de rellenar la información de cada obra, los responsables deben registrar las partidas que se contraten. Las partidas son entregas de material para la obra, y se emplean para calcular la estimación de residuos que se van a generar.

Cada partida viene con una estimación de residuos calculada por el fabricante. Esta medida se emplea para calcular cuántos residuos se van a generar en la obra. Por cada partida se almacena la siguiente información:

- ✓ Nombre del fabricante del material.
- ✓ Obra de destino.
- ✓ Descripción del producto suministrado.
- ✓ Fecha de fabricación del material.
- ✓ Cantidad en peso y volumen del material.
- ✓ Medidas del paquete.
- ✓ Estimación de residuos del fabricante.

#### 4.2.2.3 Contenedores

De la misma forma que se deben registrar las partidas con material que entran en la obra es necesario registrar los residuos que salen. Estos residuos salen en contenedores que proporciona las plantas gestoras. Inicialmente el responsable solicita uno o varios contenedores vacíos, que llegan a la obra. Cuando estos contenedores están llenos, se solicita su recogida, pudiendo solicitar un contenedor nuevo vacío.

Los contenedores, por tanto, representan los residuos generados por las obras, los cuales van a viajar hasta la planta de reciclaje. Se trata del elemento principal de la aplicación y sobre el cual se pretende hacer la trazabilidad.

Por cada contenedor se registra la siguiente información:

- ✓ Obra de la que procede.
- ✓ Planta de gestión encargada de la recogida y tratamiento del contenedor.
- ✓ Número de contenedor.
- ✓ Fecha de recogida del contenedor de la obra.
- ✓ Fecha de entrega del contenedor en la planta de reciclaje.
- ✓ Tipo de residuo que contiene el contenedor.
- ✓ Cantidad en peso y volumen.
- ✓ Coste de recogida para la obra.

Una vez entregados en la planta de reciclaje, los contenedores se marcan como tratados. Al finalizar la obra se puede comprobar cuantos contenedores han sido recogidos por plantas autorizadas, o fabricantes en caso de ser enviados a un fabricante para su reciclaje.

Esta información se puede contrastar con la cantidad de material empleado en la obra procedente de las partidas contratadas, la estimación de la obra y la proporcionada por el fabricante para cada partida. De esta manera se puede demostrar el correcto tratamiento de los residuos sin necesidad de albaranes físicos en papel, y de forma completamente automatizada.

#### 4.2.3 Diseño estructurado de alto nivel

Como resumen del proceso de funcionamiento del sistema, la figura 6 muestra el diseño estructurado de alto nivel, en el que se pueden ver las interacciones de todos los participantes en el sistema.

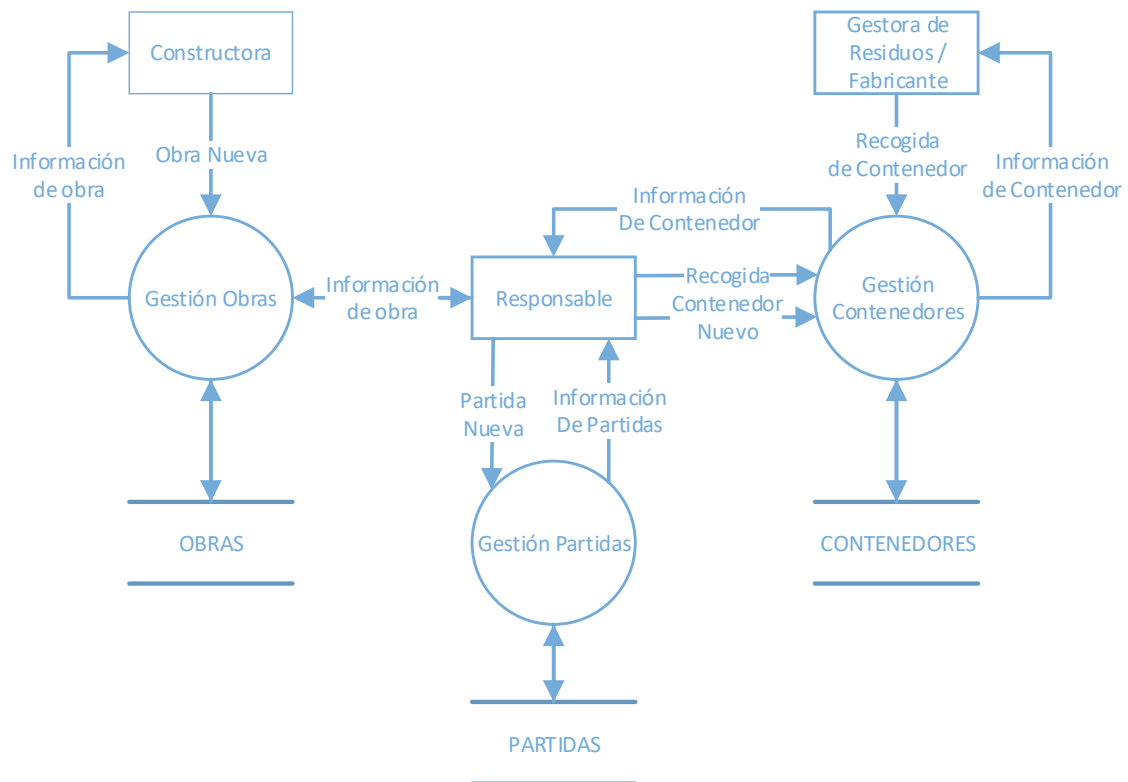


Figura 6 – Diseño estructurado de alto nivel.

### 4.3 Planificación del trabajo

Una vez finalizado el diseño, se destacan bloques principales sobre los que trabajar. Por un lado, los constructores y los responsables, que gestionan las obras y los contenedores que estas generan, y por otro lado los gestores y los fabricantes que recogen los contenedores para tratar los residuos.

En el primer bloque se distinguen dos módulos, constructores y responsables, y en el segundo gestores y fabricantes. Cada módulo se implementa en el sistema como una página web, a la que se accede con las credenciales correspondientes.

Además de estos cuatro módulos se implementa la base de datos que contenga los datos de cada módulo y las relaciones entre ellos.

Como funcionalidad adicional se incluye un módulo que proporciona la implementación de los mapas, con los diferentes puntos obtenidos de la base de datos, y que permite insertar un mapa en cualquier módulo proporcionando las coordenadas de los puntos deseados.

Para la correcta implementación es necesario en primer lugar crear la base de datos, y posteriormente insertar datos básicos para las pruebas y el correcto funcionamiento de los demás módulos.

En segundo lugar, se implementará la funcionalidad de los mapas, para poder integrarla en el resto de módulos que sea requerida. Para ello será necesario introducir obras manualmente en la base de datos.

En tercer lugar, se comenzará con los constructores, y sus operaciones, permitiendo insertar responsables en el sistema, y posteriormente obras que serán asignadas a los responsables existentes.

En cuarto lugar, se implementarán los responsables, que completarán y gestionarán las obras, generando partidas y contenedores.

En último lugar, se completará el sistema con los módulos de los gestores y fabricantes, que gestionarán los residuos, finalizando el proceso de trazabilidad. Los fabricantes y los gestores son dos módulos diferentes, pero que realizan las mismas operaciones, y por tanto se unifican a nivel de software, aunque se muestren por separado al usuario.

#### 4.4 Base de Datos

Toda la información que se genera en el sistema se almacena en la base de datos. Esta base de datos da soporte a la aplicación y en ella se reflejan las relaciones entre los distintos elementos que la componen. La base de datos está compuesta por una tabla por cada elemento de la aplicación.

Para albergar a los usuarios hay dos tablas. La tabla usuario que almacena la información común a todos los usuarios, como el nombre, el email, contraseña, teléfono etc, y la tabla de cada rol. Es en la tabla de cada rol donde se almacena la información específica de cada tipo de usuario. Las tablas de cada rol se encuentran relacionadas con la tabla usuario.

Para almacenar a los fabricantes se utiliza internamente la misma tabla que para los gestores de residuos. Esto es así porque, aunque en la aplicación son dos roles diferentes, las acciones y el comportamiento en el sistema es el mismo. Los fabricantes que recojan residuos estarán actuando como un gestor, ya que lo que se haga después de la entrega del contenedor con el material, sale fuera de los límites de la aplicación.

No obstante, para evitar confusiones, ya que en la vida real un fabricante no es un gestor de residuos, se ha optado por mantener los dos roles de cara al usuario.

En la figura 7 se puede ver un diagrama completo con las tablas y los campos por cada tabla que contiene la base de datos.

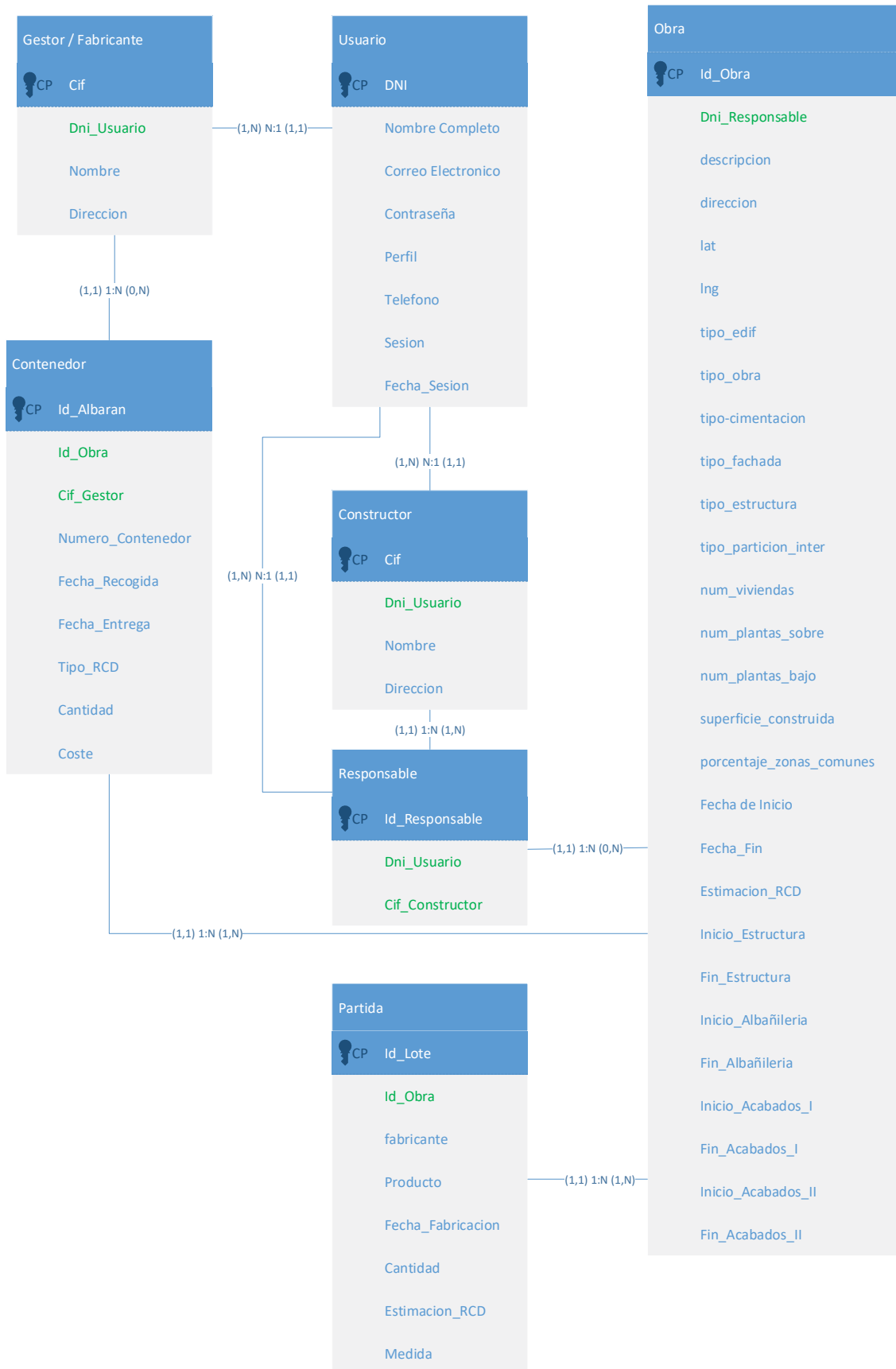


Figura 7 – Modelo entidad relación.



## Capítulo 5 - Desarrollo

En este capítulo se detalla, una vez finalizado el proceso de diseño, el desarrollo de la aplicación, y la batería de pruebas llevadas a cabo durante el desarrollo del mismo y al final. Por este motivo se detallan ambos apartados en el mismo capítulo.

### 5.1 Aplicación

#### 5.1.1 Sesión y Autenticación de usuario

Al acceder por primera vez a la aplicación se muestra una página con información sobre los diferentes roles que existen y las posibilidades que estos ofrecen. Para poder acceder al sistema es necesario darse de alta, y crear una cuenta para poder entrar. Para ello se habilita una opción en la esquina superior derecha, a la que se accede al formulario de alta para cada rol, excepto el de responsable, que será dado de alta por el constructor correspondiente.

Otra opción que aparece al lado del registro es Login, que permite entrar en el sistema si ya se dispone de cuenta. Para ello introducir el email registrado y la contraseña.

Este modelo se ha implementado, almacenando los perfiles en la base de datos, en una tabla llamada usuario. Esta tabla contiene todos los usuarios independientemente del rol que ocupen.

Cada página de la aplicación comprobará en la carga si el usuario está autenticado. Esto se realiza mediante las sesiones que proporciona PHP. Al iniciar una solicitud de una página al servidor, este genera una sesión, la cual se almacena en la base de datos asociada al usuario que inició sesión. El servidor almacena internamente la sesión de cada conexión, y cuando se solicita una página se comprueba la sesión activa del servidor con la sesión almacenada en la base de datos para el usuario activo.

Una vez se encuentre en la página correspondiente al rol del usuario, en la parte superior derecha se muestran dos opciones: logout, y perfil. Para cerrar la sesión se borra de la base de datos, haciendo que la sesión activa del servidor no corresponda con la almacenada en la base de datos, y siendo necesario autenticarse de nuevo. La opción perfil permite cambiar la información de usuario almacenada en la base de datos.

#### 5.1.2 Gestión de los mapas

Para poder implementar los mapas, se ha decidido realizar un módulo que obtenga el mapa y pueda insertarlo en la página realizando una llamada al servidor. Este módulo se divide en dos partes: cálculo de direcciones y obtención del mapa.

La necesidad en el sistema consiste en proporcionar un mapa personalizado que marque distintas ubicaciones dinámicamente. Estas ubicaciones se encuentran almacenadas en la base de datos y cambian según los usuarios vayan introduciendo las ubicaciones. Estas ubicaciones son de dos tipos: obras, y plantas de gestión de residuos. Además, no siempre se van a visualizar todas las obras, o todas las plantas de gestión. Por tanto, las ubicaciones dependen de distintas consultas que se realicen a la base de datos, que devolverán diferentes resultados.

Para implementar esto, haciendo uso de los mapas de Google, es necesario utilizar la API de Google Maps. Esta API permite proporcionar un fichero xml con la información de cada ubicación, y obtener el mapa ya construido. [13]

Dentro del xml se proporciona toda la información que se desea mostrar en las etiquetas de cada ubicación. Pero la información más importante a enviar es la ubicación de cada punto en el mapa. Estos puntos se deben representar como coordenadas de latitud y longitud.

Estas coordenadas deben almacenarse en la base de datos, sin embargo, no se solicitan al usuario. En los formularios para la creación, tanto de obras como de plantas de gestión de residuos, el usuario únicamente introduce la dirección, nombre de la calle y número, de la misma manera que se introduce una dirección en un gps, o un mapa de Google.

El módulo encargado de obtener el mapa, envía la dirección escrita por el usuario, a un servidor de Google, que devuelve un fichero en formato JSON con la información detallada de la ubicación enviada. Dentro de esta información se encuentran las coordenadas en latitud y longitud necesarias para ubicar la dirección en el mapa.

Una vez obtenidas las coordenadas se almacenan en la base de datos. Esta operación se realiza al insertar las obras y plantas de gestión. De esta manera, posteriormente al solicitar el mapa al servidor, las coordenadas ya están calculadas.

Para realizar esta consulta y obtener el fichero JSON con la información de la dirección se hace uso de la librería axios para javascript. [14]

Así, cuando se quiera obtener e insertar un mapa únicamente hay que realizar la consulta a la base de datos y convertir la información devuelta a formato xml antes de enviarla al servidor de Google que devolverá el mapa. Esta conversión se realiza mediante PHP, y se ha implementado creando un fichero en PHP que codifica las diferentes consultas a realizar a la base de datos y convierte los resultados en xml.

Cabe destacar que para poder hacer uso del servicio de Google Maps, es necesario crearse una cuenta, y habilitar el servicio desde la configuración de la cuenta de Google. Este servicio es gratuito, pero es limitado, y aunque los límites se establecen en 2500

solicitudes al día y 50 solicitudes por minuto, si se lleva a un ámbito real sería necesario realizar una cuenta premium. Sin embargo, para este proyecto esto no se va a contemplar.

Un problema al realizar las consultas a un servidor externo es la latencia que se produce. Esta latencia hace que la respuesta llegue demasiado tarde y el código de la aplicación ya haya devuelto un error. A la hora de insertar el mapa en el fichero HTML resultante esto no supone un problema, ya que el espacio para el mapa está definido con anterioridad, y la latencia es apenas perceptible por el usuario.

Sin embargo, a la hora de realizar el cálculo de las coordenadas, esta latencia supone un problema, ya que se almacenan en la base de datos antes de recibir la respuesta del servidor. Esto se debe a la asincronía de las llamadas al servidor, y el desconocimiento del tiempo de respuesta.

Para solucionar el problema de la asincronía, se van a controlar las llamadas mediante javascript. Existen muchas soluciones para tratar la asincronía, y se ha elegido la opción de las promesas en javascript, porque son muy sencillas de implementar y en las versiones más recientes se encuentra implementada de forma nativa. [15]

### 5.1.3 Estructura de las páginas

Se ha implementado una página para cada rol, aunque tienen el mismo diseño y estructura. Las páginas se encuentran divididas en secciones. Cada sección se encarga de la gestión de un proceso. Las secciones por rol son las siguientes:

Para los constructores hay de tres secciones:

- ✓ **Obras**, donde se crean, modifican y consultan las obras de la constructora.
- ✓ **Responsables**, donde se dan de alta, y de baja los responsables de la constructora.
- ✓ **Estadísticas**, donde se consulta el estado de la constructora, y se obtienen estadísticas generales y por cada obra.

Para el responsable hay tres secciones:

- ✓ **Obras**, donde gestionan las partidas y visualizan la información de la obra.
- ✓ **Plantas de Gestión**, donde gestionan los contenedores, y visualizan información de los contenedores generados en la obra
- ✓ **Estadísticas**, donde se consulta el estado de todas las obras del responsable.

Para los gestores de residuos y los fabricantes únicamente hay una sección. En esta sección se gestionan las obras con las que el gestor o fabricante tiene contrato, y se

aceptan los contenedores que los responsables de cada obra solicitan. Además, se pueden ver las estadísticas de los contenedores aceptados por obra.

## 5.2 Redacción de Pruebas

Para comprobar que la aplicación funciona correctamente se realiza una batería de pruebas basada en los requisitos especificados en el capítulo 4.

En primer lugar, se especifican las pruebas generales y básicas del sistema. Estas pruebas son necesarias para poder continuar con las demás.

<b>Prueba 1</b>	<b>Creación de Usuarios</b>
<b>Descripción</b>	Dar de alta a todos los usuarios de todos los roles.
<b>Requisito</b>	1 y 3
<b>Resultado esperado</b>	Todos los usuarios se dan de alta sin problemas
<b>Superada</b>	Si
<b>Prueba 2</b>	<b>Inicio de sesión</b>
<b>Descripción</b>	Iniciar sesión con todos los roles, y comprobar que la página cargada es la correcta
<b>Requisito</b>	1 y 3
<b>Resultado esperado</b>	Todos los usuarios inician sesión correctamente.
<b>Superada</b>	Si
<b>Prueba 3</b>	<b>Cerrar Sesión</b>
<b>Descripción</b>	Cerrar sesión una vez abierta
<b>Requisito</b>	1 y 3
<b>Resultado esperado</b>	La sesión se cierra y no se puede acceder de nuevo a no ser que se vuelva a iniciar.
<b>Superada</b>	Si
<b>Prueba 4</b>	<b>Visualización del mapa</b>
<b>Descripción</b>	Comprobar que el mapa se visualiza correctamente
<b>Requisito</b>	11 y 12
<b>Resultado esperado</b>	Se debe visualizar el mapa con los puntos especificados en la consulta.
<b>Superada</b>	Si
<b>Prueba 5</b>	<b>Seguridad de los datos</b>
<b>Descripción</b>	Comprobar la información mostrada a cada usuario.
<b>Requisito</b>	1, 8, 9 y 10
<b>Resultado esperado</b>	Cada usuario solo puede ver la información que le corresponde.
<b>Superada</b>	Si

Tabla 9 – Pruebas Generales

En segundo lugar, se detallan las pruebas unitarias, que son aquellas más específicas y que se corresponden a cada rol.

Para el Constructor:

<b>Prueba 6</b>	<b>Gestión de responsables</b>
<b>Descripción</b>	Dar de alta y baja a responsables.
<b>Requisito</b>	2
<b>Resultado esperado</b>	Los responsables de dan de alta y de baja sin errores.
<b>Superada</b>	Si
<b>Prueba 7</b>	<b>Gestión de obras</b>
<b>Descripción</b>	Crear y modificar obras.
<b>Requisito</b>	4 y 13
<b>Resultado esperado</b>	Se pueden crear obras, y después modificarlas.
<b>Superada</b>	Si
<b>Prueba 8</b>	<b>Visualización de estadísticas</b>
<b>Descripción</b>	Comprobar la información estadística mostrada.
<b>Requisito</b>	8
<b>Resultado esperado</b>	Se muestran las estadísticas, tanto de cada obra como del conjunto de todas ellas.
<b>Superada</b>	Si
<b>Prueba 9</b>	<b>Visualización del mapa</b>
<b>Descripción</b>	Comprobar el mapa de las obras.
<b>Requisito</b>	11
<b>Resultado esperado</b>	El mapa muestra el conjunto de obras que tiene la constructora almacenadas en el sistema.
<b>Superada</b>	Si

Tabla 10 – Pruebas unitarias del Constructor.

Para el responsable:

<b>Prueba 10</b>	<b>Visualización de obras</b>
<b>Descripción</b>	Comprobar el mapa de las obras.
<b>Requisito</b>	9 y 11
<b>Resultado esperado</b>	El mapa muestra el conjunto de obras que tiene bajo su responsabilidad.
<b>Superada</b>	Si
<b>Prueba 11</b>	<b>Visualización de plantas de gestión</b>
<b>Descripción</b>	Comprobar el mapa de plantas de gestión.
<b>Requisito</b>	12
<b>Resultado esperado</b>	El mapa muestra el conjunto de plantas de gestión a las que puede enviar los residuos.
<b>Superada</b>	Si
<b>Prueba 12</b>	<b>Alta de partida</b>
<b>Descripción</b>	Dar de alta una partida
<b>Requisito</b>	14
<b>Resultado esperado</b>	La partida queda registrada en la base de datos.
<b>Superada</b>	Si
<b>Prueba 13</b>	<b>Alta de contenedor</b>
<b>Descripción</b>	Generar un contenedor.
<b>Requisito</b>	15
<b>Resultado esperado</b>	El contenedor queda registrado en el sistema
<b>Superada</b>	Si
<b>Prueba 14</b>	<b>Visualización de estadísticas</b>
<b>Descripción</b>	Comprobar la información estadística mostrada.
<b>Requisito</b>	9
<b>Resultado esperado</b>	Se muestran las estadísticas de las partidas y contenedores por obra, y por el total de obras propias.
<b>Superada</b>	Si

Tabla 11 – Pruebas Unitarias del Responsable.

Para el gestor:

<b>Prueba 15</b>	<b>Visualización de obras</b>
<b>Descripción</b>	Comprobar obras con las que se tiene contrato.
<b>Requisito</b>	10
<b>Resultado esperado</b>	Se muestran las estadísticas de las partidas y contenedores por obra, y por el total de obras propias.
<b>Superada</b>	Si
<b>Prueba 16</b>	<b>Gestión de contenedores</b>
<b>Descripción</b>	Aceptar la entrada de un contenedor.
<b>Requisito</b>	19 y 20
<b>Resultado esperado</b>	El contenedor se queda marcado como aceptado.
<b>Superada</b>	Si
<b>Prueba 17</b>	<b>Visualización de estadísticas</b>
<b>Descripción</b>	Comprobar información estadística mostrada.
<b>Requisito</b>	10
<b>Resultado esperado</b>	Se muestran las estadísticas de los contenedores por obra, y por el total de obras con contrato.
<b>Superada</b>	Si

Tabla 12 – Pruebas Unitarias del Gestor.

## Capítulo 6 - Conclusiones y líneas futuras

### 6.1 Conclusiones

La trazabilidad nos permite seguir la trayectoria de un producto a lo largo de un proceso. Junto con las tecnologías actuales, es posible automatizar este proceso y convertirlo en una actividad más cómoda y eficiente, ahorrando dinero y tiempo.

Una aplicación que gestione la trazabilidad proporciona una mejor comunicación entre todos los elementos del proceso, poniendo en contacto a los distintos agentes involucrados, y permite una mejor escalabilidad del sistema.

En definitiva, esta aplicación permite un mayor control sobre los residuos generados en un área determinada, cuantificando los residuos generados. Esta característica es esencial para organismos que controlen la actividad de gestión de residuos, como puede ser por ejemplo un ayuntamiento.

### 6.2 Líneas futuras

Un sistema que gestiona la trazabilidad de los residuos resulta interesante desde el punto de vista de la eficiencia. Sin embargo, para hacer el sistema más completo, la gestión del mismo por parte del ayuntamiento ayudaría al proceso de obtención de licencias de obras. De esta manera un rol de ayuntamiento ayudaría en el control del proceso.

Por otro lado, este sistema gestiona la trazabilidad únicamente de los residuos generados, pero como posible ampliación se contempla la gestión de los materiales desde su origen como materias primas hasta el reciclado de los mismos. Además, se podría ampliar a la gestión del residuo una vez llega a la planta de tratamiento y su reciclaje como materia prima alternativa.

Por último, para agilizar la recopilación de datos, se podría incorporar al sistema la capacidad de gestión de los datos con códigos QR en los contenedores, de manera que se pueda incorporar la información de forma automática.




## Referencias web

- [1] - Lista Europea de Residuos - [http://www.gestoresresiduos.es/codigos\\_ler.html](http://www.gestoresresiduos.es/codigos_ler.html)
- [2] - Trazabilidad, definición wikiPedia - <https://es.wikipedia.org/wiki/Trazabilidad>
- [3] - Desarrollo Web – Definición wikiPedia - [https://es.wikipedia.org/wiki/Desarrollo\\_web](https://es.wikipedia.org/wiki/Desarrollo_web)
- [4] - Tecnologías de desarrollo web - <http://desarrollowebbydesarrolloweb.blogspot.com.es/2015/02/tabla-comparativa-de-los-lenguajes-de.html>
- [5] - Comparativa de servidores Web - <https://es.scribd.com/document/83122571/Cuadro-Comparativo-Apache-IIS>
- [6] - Ventajas de Maria DB- <http://www.tuprogramacion.com/bases-de-datos/mariadb-vs-mysql/>
- [7] - Ventajas de MySql - <https://mysqldaniel.wordpress.com/ventajas-y-desventajas/>
- [8] - OpenLayers vs Leaflet - <https://mappinggis.com/2016/11/openlayers-vs-leaflet-mejor/>
- [9] - Comparativa mapas en web - <https://stackshare.io/stackups/google-maps-vs-leaflet-vs-openlayers>
- [10] – Herramientas de desarrollo Web – Bitnami - <https://bitnami.com/stack/lamp>
- [11] – Desarrollo en Cascada - Wikipedia - [https://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](https://es.wikipedia.org/wiki/Desarrollo_en_cascada)
- [12] – Juan Camilo Zuluaga Henao – Análisis y diseño II – <https://es.slideshare.net/juanchenao/tipos-de-requisitos>
- [13] – Geocoding API – Google - <https://developers.google.com/maps/documentation/geocoding/start?hl=es#client-library>
- [14] – Librería AXIOS en GitHub - <https://github.com/axios/axios>
- [15] – Controlar la ejecución Asíncrona – Charla de Pablo Almunia – Todo JS - <https://www.todojs.com/controlar-la-ejecucion-asincrona/>

## Bibliografía

- [a] Limpia-Way Ltd - <https://sen.world/project/clean-way-software/>
- [b] Natalia Llano Ríos. Construcción del plan de trazabilidad en las diferentes líneas (recepción) manejadas por AVINCO S.A. Corporación Universitaria LaSallista - Facultad de Ingeniería de alimentos. CALDAS. 2010
- [c] Larry Ullman. Visual Quickstart Guide PHP for the Web. Cuarta Edición. Berkeley, CA: Peachpit Press;2011.
- [d] Steve Suehring and Janet Valade. PHP, MySQL®, JavaScript® & HTML5 All-in-One For Dummies®. Primera edición. Hoboken, NJ: John Wiley & Sons, Inc; 2013

Este documento esta firmado por



<b>Firmante</b>	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
<b>Fecha/Hora</b>	Wed Jun 27 17:57:21 CEST 2018
<b>Emisor del Certificado</b>	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
<b>Numero de Serie</b>	630
<b>Metodo</b>	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)