

Proyecto Final DAW Tesseract Solutions

Iván García Carrasco



IES Arcipreste de HITA

13 de Mayo de 2024

Índice

[1. Introducción 3](#_Toc168563220)

[2. Definición del proyecto 3](#_Toc168563221)

[2.1. Descripción muy general del proyecto. 3](#_Toc168563222)

[2.2. Tipos de Empresas. Sectores Productivos 3](#_Toc168563223)

[3. Diseño y fase del proyecto 3](#_Toc168563224)

[3.1. Objetivos. Especificación de requisitos 3](#_Toc168563225)

[3.2. Identificación de las fases del proceso 4](#_Toc168563226)

[3.2.1. Análisis 4](#_Toc168563227)

[3.2.2. Diseño 4](#_Toc168563228)

[3.2.3. Codificación 15](#_Toc168563229)

[3.2.4. Pruebas 15](#_Toc168563230)

[3.2.5. Despliegue 15](#_Toc168563231)

[3.2.6. Mantenimiento 15](#_Toc168563232)

[3.3. Herramientas de diseño 15](#_Toc168563233)

[3.3.1. Herramientas en el lado cliente 15](#_Toc168563234)

[3.3.2. Herramientas en el lado servidor 16](#_Toc168563235)

[3.3.3. Herramientas de desarrollo 16](#_Toc168563236)

[3.4. Evaluación económica y financiación del proyecto 16](#_Toc168563237)

[3.4.1. Estudio de viabilidad 16](#_Toc168563238)

[3.4.2. Ayudas económicas 16](#_Toc168563239)

[4. Planificación del proyecto 16](#_Toc168563240)

[4..1 Diagrama de secuenciación del proyecto 16](#_Toc168563241)

[4.2. Secuenciación de actividades o tareas 17](#_Toc168563242)

[4.3 Asignación de recursos y tiempos en las actividades del proyecto 18](#_Toc168563243)

[4.4. Procedimientos en las actividades. 18](#_Toc168563244)

[4.5. Prevención de riesgos en el desarrollo del proyecto (Retrasos, bloqueos..etc.) 20](#_Toc168563245)

[5. Pruebas y validación del Proyecto 20](#_Toc168563246)

[5.1. Evaluación y seguimiento de las actividades. 20](#_Toc168563247)

[5.2. Indicadores de calidad. Elaborar una batería de pruebas para detectar errores. 21](#_Toc168563248)

[5.3. Informe de evaluación de incidencias. Solución de las mismas. 23](#_Toc168563249)

[6. Implantación del proyecto 24](#_Toc168563250)

[6.1. Plan de implantación. 24](#_Toc168563251)

[6.1.1. Implantación en máquina virtual 24](#_Toc168563252)

[6.1.2. Implantación en servidor propio 25](#_Toc168563253)

[6.2. Manual de instalación. 26](#_Toc168563254)

[6.2.1 Cambio de constantes del servidor 26](#_Toc168563255)

[6.2.2 Cambio de constantes de la aplicación 26](#_Toc168563256)

[6.2.3 Usuario administrador 26](#_Toc168563257)

[6.3. Manual de usuario. 26](#_Toc168563258)

[6.3.1. Introducción 26](#_Toc168563259)

[6.3.2. Sesión 27](#_Toc168563260)

[6.3.3 Jornada laboral 28](#_Toc168563261)

[6.3.4. Planificación 28](#_Toc168563262)

[6.3.5. Vehículos 29](#_Toc168563263)

[6.3.6. Mercancía 30](#_Toc168563264)

[6.3.7. Trabajadores 32](#_Toc168563265)

[6.3.8. Clientes 33](#_Toc168563266)

[6.3.9. Configuración 34](#_Toc168563267)

[7. Conclusión 34](#_Toc168563268)

[8. Bibliografía/Webgrafía 35](#_Toc168563269)

# 1. Introducción

En este documento se detallará toda la documentación técnica acerca de la aplicación web realizada como proyecto final del curso Desarrollo de Aplicaciones Web.

Este proyecto consiste en una web para la gestión de empresas logísticas.

# 2. Definición del proyecto

## 2.1. Descripción muy general del proyecto.

Este proyecto trata de una aplicación web para la gestión de todo tipo de empresas logística.

La logística es un sector muy amplio y es necesario llevar un registro muy riguroso de todo lo que pasa dentro de la empresa, es por ello que este proyecto abarcara todo lo que puede necesitar una empresa de este tipo para funcionar más rápido y sin errores.

Para los clientes es sumamente importante conocer que pasa en cada momento con la mercancía que han puesto en manos de una empresa externa, gracias a este proyecto se les dará a todos y cada uno la confianza necesaria y podrán saber el estado de aquello que hayan enviado.

Respecto a la parte no visible para los clientes, este proyecto llevará toda la gestión interna de la empresa, será la parte más potente de la aplicación, se podrá hacer un registro de todas las posibles partes de la empresa.

## 2.2. Tipos de Empresas. Sectores Productivos

La aplicación va enfocada completamente al sector **logístico**, a cualquier tipo de empresa que se dedique a la **recepción / envíos de mercancía**.

Está aplicación está desarrollada para agilizar el proceso logístico por el cual pasa una mercancía desde su recepción hasta su posterior entrega.

Muchas empresas aún utilizan sistemas comunes y antiguos además de inseguros pare llevar un control de este proceso. Muchas de estas empresas quieren mejorar la calidad de su trabajo, por lo que es una opción ideal el desarrollo de esta aplicación para su posterior venta.

# 3. Diseño y fase del proyecto

## 3.1. Objetivos. Especificación de requisitos

Son dos los objetivos de esta aplicación, la gestión interna y la consulta de los clientes.

Gestión interna:

* Control de mercancía
  + Añadir, eliminar y editar mercancía
  + Ver mercancía entregada
  + Asignar y desasignar a un vehículo
* Control de vehículos
  + Añadir, eliminar y editar vehículos
* Control de empleados
  + Añadir, eliminar y editar trabajadores
  + Crear y eliminar cuentas web
  + Registrar jornadas
  + Gestionar planificación
* Control de clientes
  + Añadir, eliminar y editar clientes
  + Asignar cliente a mercancía

Interfaz principal:

* Consulta de mercancía

## 3.2. Identificación de las fases del proceso

### 3.2.1. Análisis

Se evalúan las necesidades de las empresas y usuarios que hagan uso del software, con ello, se obtienen los requisitos de la aplicación, nombrados en el [punto 3.1](#_3.1._Objetivos._Especificación) y se planean varias rutas a seguir para la realización del proyecto, finalmente se escoge una de ellas.

### 3.2.2. Diseño

* **Tecnologías**

Se escogen las tecnologías que mejor encajan según los requisitos obtenidos.

En el [punto 3.3](#_3.3._Herramientas_de) están definidas las tecnologías y herramientas utilizadas.

* **Prototipos de las interfaces**

#### Principal

****

Imagen 1. Diseño principal

#### Localizar paquete



Imagen 2. Diseño localizar paquete

#### Login



Imagen 3. Diseño login

#### Gestión principal



Imagen 4. Diseño gestión principal

#### Vehículos



Imagen 5. Diseño vehículos

#### Mercancía



Imagen 6. Diseño mercancía

#### Mercancía entregada



Imagen 7. Diseño mercancía entregada

#### Trabajadores



Imagen 8. Diseño trabajadores

#### Cuentas usuario



Imagen 9. Diseño cuentas usuario

#### Clientes



Imagen 10. Diseño clientes

#### Configuración



Imagen 11. Diseño configuración

* **Logos**

Logos creados con el uso de distintas herramientas, la principal fue Photoshop.

Imagen 12. Logo página principal

Imagen 13. Logo página iniciar sesión Imagen 14. Logo superior navegador

* **Modelo entidad-relación**



Imagen 15. Modelo E-R

* **Casos de uso**



Imagen 16. Casos de uso

* **Modelo relacional y normalización**

Todas las tablas están normalizadas hasta al menos su ***2FN***. En **negrita** las **PK** y subrayadas las FK.

Empleados (**dni**, nombre, apellidos, teléfono, correo, fecha\_nacimiento, fecha\_inicio\_empresa, cargo)

Cuentas\_web (**dni\_empleado**, password)

Jornada\_empleados (**dni\_empleado**, **fecha\_jornada**, **hora\_entrada**, **hora\_salida**)

Mercancía (**localizador**, cliente, peso, tipo\_estado, tipo\_peso, descripción)

Usuarios (**dni**, nombre, apellidos, teléfono, correo)

Planificación\_empleados (**fecha**, **empleado**, **descripción**)

Vehículos (**matrícula**, carga\_útil, tipo\_peso, tipo\_estado)

Transporte\_mercancía (**localizador**, **matricula**)

Tipo\_cargo (**tipo**, nombre)

Tipo\_estado\_mercancía (**tipo**, nombre)

Tipo\_estado\_vehículo (**tipo**, nombre)

Tipo\_peso (**tipo**, nombre)

* **Base de datos**

Tabla: Empleados

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| Dni | Cadena | no | PK |
| Nombre | Cadena | no |  |
| Apellidos | Cadena | no |  |
| Teléfono | Numérico | no |  |
| Correo | Cadena | no |  |
| Fecha\_nacimiento | Numérico | no |  |
| Fecha\_inicio\_empresa | Numérico | no |  |
| Cargo | Numérico | no | FK (tipo\_cargo) |

Tabla 1. Empleados

Tabla: Cuentas\_web

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| Dni\_empleado | Cadena | no | PK, FK (empleados) |
| Password | Cadena | no | Encriptado |

Tabla 2. Cuentas\_web

Tabla: Jornada\_empleados

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| Dni\_empleado | Cadena | no | PK, FK (empleados) |
| fecha\_jornada | Cadena | no | PK |
| hora\_entrada | Cadena | no | PK |
| hora\_salida | Cadena | no | PK |

Tabla 3. Jornada\_empleados

Tabla: Mercancía

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| localizador | Numérico | no | PK, AI |
| Cliente | Cadena | no |  |
| PESO | Float | no |  |
| Tipo\_estado | Numérico | no | FK (tipo\_estado\_mercancía) |
| TIPO\_PESO | Numérico | no | FK (tipo\_peso) |
| Descripción | Cadena | no |  |

Tabla 4. Mercancía

Tabla: Usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| Dni | Cadena | no | PK |
| Nombre | Cadena | no |  |
| Apellidos | Cadena | no |  |
| Telefono | Numérico | no |  |
| Correo | Cadena | no |  |

Tabla 5. Usuarios

Tabla: Planificación\_empleados

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| Fecha | Numérico | no | PK |
| Empleado | Cadena | no | PK, FK (empleados) |
| Descripción | Cadena | no | PK |

Tabla 6. Planificación\_empleados

Tabla: Vehículos

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| matrícula | Cadena | no | PK |
| carga\_util | Float | no |  |
| Tipo\_peso | Numérico | no | FK (tipo\_peso) |
| Tipo\_estado | Numérico | no | FK (tipo\_estado\_vehículo) |

Tabla 7. Vehículos

Tabla: Transporte\_mercancía

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| Localizador | Numérico | no | PK, FK (mercancía) |
| Matrícula | Cadena | no | PK, FK (vehículos) |

Tabla 8. Transporte\_mercancía

Tabla: Tipo\_cargo

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| tipo | Numérico | no | PK |
| Nombre | Cadena | no |  |

Tabla 9. Tipo\_cargo

Tabla: Tipo\_estado\_mercancía

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| tipo | Numérico | no | PK |
| Nombre | Cadena | no |  |

Tabla 10. Tipo\_estado\_mercancía

Tabla: Tipo\_estado\_vehículo

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| tipo | Numérico | no | PK |
| Nombre | Cadena | no |  |

Tabla 11. Tipo\_estado\_vehículo

Tabla: Tipo\_peso

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Nulo | Descripción |
| tipo | Numérico | no | PK |
| Nombre | Cadena | no |  |

Tabla 12. Tipo\_peso

* **Diseño físico de la BD**

****

Imagen 17. Diseño físico de la BD

* **Diagrama de clases UML**



Imagen 18. Diagrama de clases UML

* **Estructura de carpetas**

La estructura será la siguiente:

- app // Principal  
 - ayax // Contiene los archivos ajax  
 - config // Archivos de configuración  
 - controllers // Controladores  
 - models // Modelos  
 - views // Vistas  
 - content // Contenido de la vista  
 - css // Archivos CSS  
 - img // Imágenes  
 - inc // Archivos que serán incluidos  
 - js // Archivos JavaScript

### 3.2.3. Codificación

Esta es la fase más larga, se codifica todo lo necesario para que el software creado sea correcto y contenga el menor número de errores posible.

La codificación seguirá el modelo vista controlador (MVC), un modelo que relaciona la vista de los usuarios con la base de datos (modelos) a través de los controladores. Es una implementación sencilla de realizar y más cercana a los entornos de trabajo reales.

### 3.2.4. Pruebas

Durante la misma codificación se detectan pruebas unitarias para encontrar errores ocultos. Al finalizar el desarrollo se le da la aplicación a un grupo de usuarios para que hagan uso del software durante varios días para detectar errores y corregirlos.

### 3.2.5. Despliegue

Durante el despliegue, se pone en marcha la aplicación web para su posterior explotación, a lo largo de esta fase se pueden detectar errores y se deben corregir.

### 3.2.6. Mantenimiento

Se mantiene el software a lo largo de su vida útil, corrigiendo errores que hubiesen quedado ocultos y añadiendo nuevas funcionalidades que sean necesarias.

## 3.3. Herramientas de diseño

### 3.3.1. Herramientas en el lado cliente

* HTML: lenguaje de marcas utilizado para la estructura visual de la aplicación web.
* CSS: lenguaje de diseño utilizado para dar formato visual.
* JavaScript: utilizado para dotar de funcionalidad la aplicación.
* Bootstrap: framework utilizado como complemento al lenguaje css para dar diseño y formato.
* SweetAlert2: librería de JavaScript utilizada para dotar de funcionalidad en forma de alertas.

### 3.3.2. Herramientas en el lado servidor

* PHP: lenguaje utilizado para realizar toda la lógica interna invisible para el usuario.
* phpMyAdmin: herramienta para la gestión de la base de datos
* MySQL: base de datos utilizada para almacenar la información de la aplicación.
* Apache2: servidor web utilizado para alojar la aplicación.

### 3.3.3. Herramientas de desarrollo

* Visual Studio Code y phpstorm: entornos de trabajo para la codificación.
* Oracle VM VirtualBox: software de virtualización para crear un servidor aislado en el que realizar el proyecto.
* Linux Mint 21.2: sistema operativo utilizado para el servidor.

## 3.4. Evaluación económica y financiación del proyecto

### 3.4.1. Estudio de viabilidad

Este proyecto es viable económicamente ya que los únicos requisitos que necesita es un servidor donde almacenar la aplicación y la base de datos, lo cual cuesta unos 50€ al mes, dependiendo como de grande sea la empresa que hará uso del software y cuantos datos maneje habitualmente.

Respecto al tiempo también es viable, ya que se disponen de 3 meses para realizarlo y solo vamos a abarcar la gestión básica de la empresa, no es un software a medida para una empresa en concreto, sino que más adelante, en caso de que una empresa comprase el software, se añadiría la funcionalidad necesaria para la parte especifica de cada una.

### 3.4.2. Ayudas económicas

No existen ayudas económicas para realizar este proyecto ya que es realizado solo por una persona gratuitamente en forma de proyecto final. Solo se necesitará una financiación para el mantenimiento, pero al ser una aplicación tan poco costosa solo será necesario pagar el alojamiento de la misma, valorado en unos 50€ al mes, es por ello que para un gasto tan insignificante no existen ayudas posibles.

# 4. Planificación del proyecto

## 4..1 Diagrama de secuenciación del proyecto

Para el desarrollo del proyecto he utilizado una metodología ágil en espiral, en las cuales se han identificado las siguientes fases:



Imagen 19. Diagrama metodología ágil

## 4.2. Secuenciación de actividades o tareas

Por cada fase del [punto anterior](#_4..1_Diagrama_de) identificamos las siguientes tareas:

1. Especificación de requisitos

* Requisitos funcionales (funcionalidad de la aplicación, que va a hacer…)
* Requisitos no funcionales (velocidad, magnitud de los datos…)

1. Análisis y planificación
   * Análisis de los requisitos
   * Planificación de las tareas que se llevarán a cabo
2. Diseño
   * Especificación de las tecnologías
   * Diseños de las interfaces
   * Logos
   * Modelo E-R
   * Casos de uso
   * Modelo relacional y normalización
   * Base de datos y diseño físico
   * Diagrama de clases
3. Codificación
   * Vistas de usuario
   * Archivo htaccess
   * Modelos
   * Controladores
4. Pruebas unitarias
   * Pruebas de la codificación realizada de forma unitaria
5. Pruebas funcionales e implementación
   * Pruebas del conjunto de la aplicación
   * Implementación en el entorno de producción

## 4.3 Asignación de recursos y tiempos en las actividades del proyecto



Imagen 20. Diagrama de Gantt, asignación de recursos y tiempos

## 4.4. Procedimientos en las actividades.

Los siguientes procedimientos se realizan en base a la metodología ágil elegida, en este caso, en espiral, es por ello que los procedimientos no tienen un principio y fin marcado, se desarrolla pasando varias veces por los mismos procedimientos, cada vuelta será más extensa que la anterior hasta el desarrollo final.

En cada tarea del [punto 4.2](#_4.2._Secuenciación_de) podemos distinguir los siguientes procedimientos:

* Requisitos funcionales

Se detalla cuál va a ser la funcionalidad de la aplicación, que queremos abarcar y que no.

* Requisitos no funcionales

Se detalla cuales son los requisitos que no son parte de la funcionalidad, cuántos datos se van a manejar, velocidades…

* Análisis de los requisitos

Analizamos los requisitos anteriores para tener una idea clara de las tareas que se deben realizar en cada fase del desarrollo.

* Planificación de las tareas que se llevarán a cabo

Se hace una planificación de las tareas analizadas, se utiliza un software de gestión de proyectos, en este caso Trello, para especificar las tareas y ver el estado de ellas en cada momento. En este punto también especificamos que tareas tienen prioridad sobre otras, tiempos de desarrollo…

* Especificación de las tecnologías

Según el análisis anterior, especificamos que tecnología es la más adecuada en cada caso para los requisitos descritos.

* Diseños de las interfaces

Hacemos un primer acercamiento a lo que será la interfaz final a partir de unos prototipos o diseños.

* Logos

Generamos los logos de nuestra aplicación.

* Modelo E-R

Se crea el modelo entidad relación en base a nuestro análisis, abarcando los máximos requisitos posibles.

* Casos de uso

Creamos los casos de uso de la aplicación en base a la funcionalidad.

* Modelo relacional y normalización

A partir de nuestro modelo E-R hacemos un primer modelo relacional, si no está normalizado, debemos normalizarlo como mínimo hasta la 2FN

* Base de datos y diseño físico

Se diseña la base de datos a partir de nuestro modelo relacional.

* Diagrama de clases

Creamos el diagrama de clases, contendrá las propiedades y los métodos de todas las clases.

* Vistas de usuario

Codificamos las vistas que verán los usuarios, según avance la codificación de otras partes seguiremos con estas.

* Archivo htaccess

Se crea el archivo .htaccess, es fundamental para el funcionamiento de nuestra aplicación ya que realiza una conversión de la url.

* Modelos

Codificación de los modelos principales que interactuaran con la base de datos.

* Controladores

Con los modelos ya creados, se codifican los controladores y se implementan junto las vistas y los modelos.

* Pruebas de la codificación realizada de forma unitaria

Se realizan las pruebas de cada componente codificado de forma unitaria.

* Pruebas del conjunto de la aplicación

Si se pasan las pruebas unitarias, se realizan pruebas funcionales del conjunto.

* Implementación en el entorno de producción

Se implementa en el entorno de producción las nuevas funcionalidades.

## 4.5. Prevención de riesgos en el desarrollo del proyecto (Retrasos, bloqueos..etc.)

Para la desarrollo del proyecto se han usado distintas técnicas de prevención de riesgos, a continuación, se detallan algunas de ellas:

* Realizar una planificación detallada, creando un plan con tareas claras y concisas para no “irse por las ramas” durante el desarrollo.
* Uso de Trello (software de gestión de proyectos) para evaluar en todo momento las tareas restantes y de esta forma organizar los posibles retrasos.
* Control de versiones con Git para revertir cambios no deseados o posibles confusiones.
* Desarrollo haciendo uso de la técnica TDD para prevenir futuros fallos de la aplicación.

# 5. Pruebas y validación del Proyecto

## 5.1. Evaluación y seguimiento de las actividades.

Las tareas han ido según lo planificado gracias a la metodología utilizada, ya que en caso de usar otra, muy probablemente tendríamos que haber pasado de nuevo por alguna tarea para finalizarla, de esta forma se ha ido poco a poco según lo necesitado en cada ocasión.

El calendario también fue según lo planeado excepto un apartado, en el apartado en el que se diseñaban las interfaces a lo largo del mes 1, tuve que volver a realizar alguna más que me di cuenta que íbamos a necesitar, por lo que entre el mes 2 y 3 tuve que volver a hacer un par de diseños de interfaces más.

## 5.2. Indicadores de calidad. Elaborar una batería de pruebas para detectar errores.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Localizar mercancía** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Localizar mercancía existente | Localizador: Existente | Existe | Existe | Ok |  |
| Localizador: No existente | No existe | No existe | Ok |  |
| No enviar localizador | Localizador: nulo | Redirección | Redirección | Ok | Se modifica el formulario para no enviarlo. |

Tabla 13. Pruebas localizar mercancía

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iniciar sesión** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Inicio de sesión | DNI: Existente  Contraseña: Correcta | Correcto | Correcto | Ok |  |
| DNI: No existente | No existe | No existe | Ok |  |
| DNI: Existente Contraseña: Incorrecta | Incorrecto | Incorrecto | Ok |  |

Tabla 14. Pruebas iniciar sesión

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Planificación** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Añadir planificación | Todo relleno | Añadida | Añadida | Ok |  |
| Algún campo incorrecto | No añadida | No añadida | Ok |  |

Tabla 15. Pruebas planificación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Jornada** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Iniciar jornada |  | Iniciada | Iniciada | Ok |  |
|  | Iniciada | Iniciada | Ok | Salir de la aplicación y al volver se queda guardada. |
| Finalizar jornada |  | Añadida | Añadida | Ok |  |
| Añadir jornadas |  | Añadidas | Añadidas | Ok | Sin recargar la página añadir y finalizar varias veces seguidas ya que hace uso de AJAX. |

Tabla 16. Pruebas jornada

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Vehículo** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Añadir vehículo | Matricula: ‘12’ | No válido | No válido | Ok |  |
| Todos correctos | Añadido | Añadido | Ok |  |
| Tipo peso: Incorrecto | No válido | No válido | Ok |  |
| Eliminar vehículo | Existe | Eliminado | Eliminado | Ok |  |
| No existe | No eliminado | No eliminado | Ok |  |
| Editar vehículo | Sin cambios | No editado | No editado | Ok |  |
| Cambios incorrectos | No editado | No editado | Ok |  |
| Cambios correctos | Editado | Editado | Ok |  |
| Asignar mercancía | No existe | No asignada | No asignada | Ok |  |
| Tipo de peso distinto | No asignada | No asignada | Ok |  |
| Supera la carga máxima | No asignada | No asignada | Ok |  |
| Todo correcto | Asignada | Asignada | Ok |  |
| Desasignar mercancía | No está en el vehículo | Incorrecto | Incorrecto | Ok |  |
| Está en el vehículo | Desasignada | Desasignada | Ok |  |

Tabla 17. Pruebas vehículo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mercancía** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Añadir mercancía | Todo relleno | Añadida | Añadida | Ok |  |
| Algún campo incorrecto | No añadida | No añadida | Ok |  |
| Eliminar mercancía | No existe | No eliminada | No eliminada | Ok |  |
| Editar mercancía | Sin cambios | No editada | No editada | Ok |  |
| Cambios incorrectos | No editada | No editada | Ok |  |
| Cambios correctos | Editada | Editada | Ok |  |

Tabla 18. Pruebas mercancía

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trabajadores** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Añadir trabajador | Todo relleno | Añadida | Añadido | Ok |  |
| Algún campo incorrecto | No añadida | No añadido | Ok |  |
| Eliminar trabajador | Existe | Eliminado | Eliminado | Ok | Si tiene cuenta se elimina también la cuenta |
| No existe | No eliminado | No eliminado | Ok |  |
| Crear cuenta web | Existe trabajador | Creada | Creada | Ok | Usuario: DNI Contraseña: DNI |
| No existe trabajador | No creada | No creada | Ok |  |
| Eliminar cuenta web | Existe cuenta | Eliminada | Eliminada | Ok | Si elimina la propia se cierra la sesión |
| No existe cuenta | No eliminada | No eliminada | Ok |  |

Tabla 19. Pruebas trabajadores

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Clientes** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Añadir cliente | Todo relleno | Añadido | Añadido | Ok |  |
| Algún campo incorrecto | No añadido | No añadido | Ok |  |
| Eliminar cliente | Existe | Eliminado | Eliminado | Ok |  |
| No existe | No eliminado | No eliminado | Ok |  |

Tabla 20. Pruebas clientes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Configuración** | | | | | |
| **Acción** | **Parámetros** | **Valor esperado** | **Valor obtenido** | **Resultado** | **Observaciones** |
| Propia | Sin cambios | No editada | No editada | Ok |  |
| Cambios incorrectos | No editada | No editada | Ok |  |
| Cambios correctos | Editada | Editada | Ok |  |
| De otro empleado | Empleado autorizado | Editada | Editada | Ok | Administradores y RR.HH |
| Empleado no autorizado | No editada | No editada | Ok | Solo puede actualizar su propio usuario |

Tabla 21. Pruebas configuración

## 5.3. Informe de evaluación de incidencias. Solución de las mismas.

Tuve una incidencia al implementar la asincronía con AJAX al añadir jornadas laborales de los empleados, el formato en el que se insertaban las horas era incorrecto al que necesitaba, la solución fue darle formato directamente en el servidor en vez de en el cliente.

También tuve problemas al usar la librería de SweetAlert2, me hubiese gustado mostrar las alertas directamente desde JavaScript, pero para esto hubiese necesitado crear todo el programa de forma asíncrona, esto no era buena idea ya que había ocasiones que necesitaba recargar la página para recargar los buffer del navegador y del servidor para no perder rendimiento y velocidad, la solución fue crear un controlador desde el servidor que mostrase la alerta que necesitábamos y realizar una composición desde el modelo principal, del cual heredan el resto de controladores.

Por último, he tenido que llevar mucho cuidado con que podía hacer cada usuario, al tener bastantes opciones en cada CRUD, he tenido varias incidencias en las que permitía acciones a usuarios que no debían tener permiso, la solución a esto ha sido el realizar muchas pruebas y probar mucho la aplicación para ver que podía hacer cada rol en todo momento.

# 6. Implantación del proyecto

## 6.1. Plan de implantación.

### 6.1.1. Implantación en máquina virtual

Duración: 1 hora. Para está opción debemos hacer uso de un software que permita la instalación de máquinas virtuales, la opción para la que se dará soporte de implementación será usando el software Oracle VM VirtualBox, el sistema operativo que utilizaremos será Linux Mint 21.2.

#### Descargar imagen VM

Debemos descargar la imagen de la máquina virtual en nuestro equipo, esta imagen se llama “Linux Mint 21.ova”.

#### Importación en VirtualBox 7.0 o superior

En el menú superior de la aplicación debemos pulsar sobre Archivo > Importar servicio virtualizado

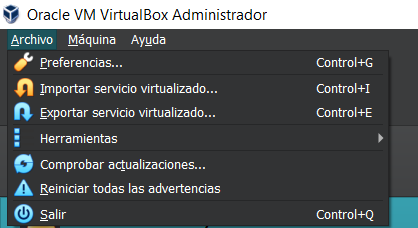


Imagen 21. Añadir máquina VirtualBox

Seleccionamos la imagen que nos hemos descargado en el [apartado anterior](#_Descargar_imagen_VM).



Imagen 22. Seleccionar imagen

Por último, podemos modificar su configuración y pulsamos en Terminar.



Imagen 23. Confirmar máquina virtual

Una vez termine la implantación ya nos aparecerá la máquina instalada, podremos cambiar su configuración o iniciarla.

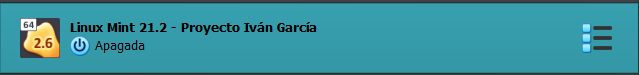


Imagen 24. Máquina virtual implantada

### 6.1.2. Implantación en servidor propio

Duración: 5-10 horas. Para está opción debemos tener en cuenta que **no se dará soporte en caso de fallos no relacionados con la aplicación web**, también debemos tener en cuenta que al hacer uso de un servidor propio es más probable que tengamos distintos fallos a lo largo de la implementación o que la aplicación web haga uso de componentes no compatibles con el servidor. Aunque el sistema operativo sea igual al utilizado para el desarrollo puede tener problemas de incompatibilidades que habrá que solucionar manualmente.

#### Prerrequisitos

El servidor debe contar como mínimo con los siguientes componentes:

* Apache (o servidores similares que permitan archivos .htaccess)
  + Módulos por defecto
  + Módulo rewrite
* PHP 8.1

#### Descarga de archivos

Debemos descargar los archivos contenidos en la carpeta “Archivos” o bien desde GitHub: <https://github.com/Ixvan22/ProyectoFinal>

#### Implementación en servidor web

En nuestro servidor web debemos indicar la ruta hacia el directorio que están contenidos nuestros archivos.

#### Habilitar archivos .htaccess

En nuestro servidor web debemos habilitar el uso de archivos .htaccess (Nota: si utilizamos un servidor Nginx no podremos hacer uso de este tipo de archivos, debemos buscar otra opción para implementar su contenido, es imprescindible para el funcionamiento de la aplicación)

#### Importar base de datos

Debemos hacer uso de un gestor de base de datos. Importaremos el archivo “db.sql” incluido en la carpeta “db”.

Una vez importado para mejorar la seguridad de nuestra aplicación eliminaremos la carpeta.

## 6.2. Manual de instalación.

### 6.2.1 Cambio de constantes del servidor

Modificaremos el archivo “app/config/server.php”.

DB\_SERVER: pondremos el host de nuestro servidor de base de datos

DB\_NAME: pondremos el nombre de la base de datos

DB\_USER: pondremos un usuario válido con acceso a la base de datos

DB\_PASS: podremos la contraseña del usuario indicado en DB\_USER

#### Cambios a tener en cuenta en seguridad

Si queremos mejorar significativamente la seguridad de nuestra aplicación deberíamos obtener estos datos a partir de otro lugar como puede ser variables de entorno dentro de nuestro sistema. En el archivo “app/models/mainmodel.php”, al inicio deberíamos obtener estas variables para no alterar el funcionamiento de nuestra aplicación.

### 6.2.2 Cambio de constantes de la aplicación

Modificaremos el archivo “app/config/app.php”.

APP\_URL: pondremos la url de nuestro servidor web

APP\_SESSION\_NAME: pondremos un nombre referente a la empresa (Ej. Tesseract)

En date\_default\_timezone\_set(…) podremos cambiar la zona horaria, en la siguiente web podemos ver los tipos de zonas horarias soportadas con el valor que hay que poner en cada caso: <http://www.php.net/manual/en/timezones.php>

### 6.2.3 Contraseñas

Para poder empezar a hacer uso de la aplicación se ha dejado creado un usuario administrador, en el login pondremos los siguientes datos:

***DNI: “00000000Z” - Contraseña: “00000000Z”***

Todas las contraseñas de la máquina virtual serán ***“alumno”.***

El usuario administrador de la base de datos por defecto es ***“root”*** y contraseña ***“alumno”***.

## 6.3. Manual de usuario.

### 6.3.1. Introducción

En este apartado se va a detallar toda la funcionalidad de la aplicación.

### 6.3.2. Sesión

#### Iniciar sesión

En url/login, introducimos el dni y la contraseña y pulsamos en iniciar sesión.

El DNI debe llevar un formato válido y debe existir en la aplicación, si el trabajador está registrado pero sin una cuenta web no podrá acceder.

Si es la primera vez que iniciamos sesión en la aplicación, la contraseña será el DNI, a no ser que nos la hayan cambiado antes algún administrador.



Imagen 25. Manual de usuario iniciar sesión

#### Cerrar sesión

Con la sesión iniciada, abrimos el navbar situado en el lateral izquierdo y pulsamos en “Salir”.



Imagen 26. Manual de usuario cerrar sesión

### 6.3.3 Jornada laboral

#### Fichar jornada

En url/gestionPrincipal pulsamos en “Iniciar”. Podemos salir de la aplicación una vez iniciada y se guardará al volver a entrar.



Imagen 27. Manual de usuario iniciar jornada

De la misma forma que se inicia la jornada, podemos finalizarla pulsando en “Finalizar”, al recargar la página podremos ver como consta en el primer registro de Jornadas recientes



Imagen 28. Manual de usuario finalizar jornada

#### Consultar jornadas

En url/gestionPrincipal, en el apartado inferior podremos consultar las jornadas más recientes. Tiene un límite, no mostramos todas las jornadas de la vida laboral del empleado, solo unas pocas



Imagen 29. Manual de usuario consultar jornadas

### 6.3.4. Planificación

#### Consultar planificación diaria

En url/gestionPrincipal, en el apartado superior podremos consultar nuestra planificación asociada en el día de hoy.



Imagen 30. Manual de usuario consultar planificación

#### Añadir planificación a un empleado

En url/gestionPrincipal, en el apartado superior pulsamos en “Añadir planificación”. Se abre un menú, rellenamos los datos y pulsamos en “Añadir”.

Todos los datos son obligatorios, si no rellenamos alguno, no podremos añadir la planificación.

La fecha debe ser el mismo día del sistema o mayor, pero no podremos introducir una fecha anterior al sistema.

En la descripción se debe poner lo que queremos asignar a un usuario en el día que tengamos marcado.

Para añadir la planificación, debemos pulsar en el botón de “Añadir”.



Imagen 31. Manual de usuario añadir planificación

### 6.3.5. Vehículos

#### Añadir vehículo

En url/vehículos, en el apartado superior pulsamos en “Añadir vehículo”. Se abre un menú, rellenamos los datos y pulsamos en “Añadir”.

Todos los campos son obligatorios, si no introducimos algún campo no se insertará el vehículo.

La matricula debe llevar un formato válido, los formatos son todas las matriculas españolas. Por ejemplo, 1234FFF y M1234FF serán matriculas válidas.

Se debe escoger el estado en el que queremos añadir el vehículo, debe ser uno válido del desplegable que se abre.

La carga útil debe ser mayor a 0, además debe ser de un tipo válido del desplegable que se abre.

Para añadir el vehículo, debemos pulsar en el botón de “Añadir”.



Imagen 32. Manual de usuario añadir vehículo

#### Eliminar vehículo

En url/vehículos, seleccionamos un vehículo y pulsamos en “Eliminar” en el apartado inferior.



Imagen 33. Manual de usuario eliminar vehículo

#### Asignar mercancía

En url/vehículos, seleccionamos un vehículo rellenamos el apartado “Añadir mercancía” y pulsamos en “Guardar cambios”.

La suma de las mercancías añadida no puede superar la carga útil del vehículo, tampoco podemos añadir una mercancía de un tipo que no sea el mismo que el del vehículo.

Para asignar la mercancía, debemos pulsar en el botón de “Guardar cambios”.



Imagen 34. Manual de usuario asignar mercancía

#### Desasignar mercancía

En url/vehículos, seleccionamos un vehículo, pulsamos en “Mercancía asignada”, se abre un menú y pulsamos en el icono de eliminar sobre la mercancía que queremos desasignar.

Cuando entregamos una mercancía, se desasigna del vehículo automáticamente.



Imagen 35. Manual de usuario desasignar mercancía

#### Modificar estado

En url/vehículos, seleccionamos un vehículo modificamos el apartado “Estado” y pulsamos en “Guardar cambios”.

El estado debe ser uno válido de los del desplegable que se abre.

Para editar el vehículo, debemos pulsar en el botón de “Guardar cambios”.



Imagen 36. Manual de usuario modificar estado

### 6.3.6. Mercancía

#### Añadir mercancía

En url/mercancia, en el apartado superior pulsamos en “Añadir mercancía”, se abre un menú, rellenamos los datos y pulsamos en “Añadir”.

El peso y el estado son obligatorios, si no rellenamos alguno la mercancía no se añadirá.

El peso debe ser mayor de 0, de lo contrario no se añadirá, también debe ser de un tipo válido de los del desplegable que se abre.

El estado debe ser uno válido de los del desplegable que se abre.

Podemos añadir descripción o no, no es obligatoria.

Podemos asignar la mercancía a un cliente ya existente, en caso de hacerlo así, debe ser un cliente válido de los del desplegable que se abre.

Podemos añadir un nuevo cliente desde este mismo menú, todos los campos serán obligatorios, si no rellenamos alguno, no se insertara ni la mercancía ni el cliente.

El DNI debe llevar un formato correcto, un ejemplo válido es 37147813F.

El teléfono debe llevar un formato correcto, un ejemplo válido es 645543765.

Para añadir el trabajador, debemos pulsar en el botón de “Añadir”.



Imagen 37. Manual de usuario añadir mercancía

#### Editar mercancía

En url/mercancia, seleccionamos una mercancía, se abre un menú, modificamos los datos y pulsamos en “Guardar cambios”.

Si no modificamos nada nuevo saltará un aviso de que no hemos seleccionado nada nuevo para editar.

Si editamos el cliente, debe ser uno válido de los del desplegable que se abre.

Si editamos el estado, debe ser uno válido de los del desplegable que se abre.

Para editar la mercancía, debemos pulsar en el botón de “Guardar cambios”.



Imagen 38. Manual de usuario editar mercancía

#### Consultar mercancía entregada

En url/mercanciaEntregada podemos consultar la mercancía ya entregada



Imagen 39. Manual de usuario consultar mercancía entregada

#### Eliminar mercancía entregada

En url/mercanciaEntregada podemos pulsar en el icono de eliminar para eliminar el registro de esa mercancía ya entregada.

Imagen 40. Manual de usuario consultar mercancía entregada

#### Localizar mercancía

En url introducimos el localizador y nos redirige a una página con sus datos.

****

Imagen 41. Manual de usuario localizar mercancía

Nos redirigirá a una nueva página en la que nos muestra los datos de la mercancía consultada.

Si la mercancía consultada no existe, nos mostrará que no existe ninguna mercancía con ese localizador.



Imagen 42. Manual de usuario localizar mercancía

### 6.3.7. Trabajadores

#### Añadir trabajador

En url/trabajadores, en el apartado superior pulsamos en “Añadir trabajador”, se abre un menú, rellenamos los datos y pulsamos en “Añadir”.

Todos los campos son obligatorios, si no rellenamos alguno no se añadirá el trabajador.

El DNI debe llevar un formato correcto, un ejemplo válido es 37147813F.

El teléfono debe llevar un formato correcto, un ejemplo válido es 645543765.

El cargo debe ser uno válido de los del desplegable que se abre.

Si pulsamos en Crear cuenta, se creará junto al empleado una cuenta web para poder iniciar sesión.

Para añadir el trabajador, debemos pulsar en el botón de “Añadir”.



Imagen 43. Manual de usuario añadir trabajador

#### Eliminar trabajador

En url/trabajadores, sobre el trabajador deseado pulsamos en “Eliminar trabajador”.

Si el trabajador tenía una cuenta web creada, se eliminará también.



Imagen 44. Manual de usuario eliminar trabajador

#### Añadir cuenta usuario

En url/trabajadores, al añadir un trabajador podemos seleccionar el apartado “Crear cuenta” o en un trabajador ya añadido podemos pulsar en “Crear cuenta”. El usuario y contraseña será su DNI.



Imagen 45. Manual de usuario añadir cuenta usuario

#### Eliminar cuenta usuario

En url/usuarios sobre la cuenta deseada pulsamos en “Eliminar”.



Imagen 46. Manual de usuario eliminar cuenta usuario

#### Editar configuración trabajador

En url/trabajadores, sobre el trabajador deseado pulsamos en “Editar”, solo podrán hacer uso de esto los administradores y RR.HH., nos lleva al apartado de Configuración explicado en el [punto 6.3.9](#_6.3.9._Configuración).

Si el usuario que pulsa en editar no tiene privilegios para editar la configuración de otros empleados, le redirigirá a editar su propia configuración.



Imagen 47. Manual de usuario editar configuración trabajador

### 6.3.8. Clientes

#### Añadir cliente

En url/clientes, en el apartado superior pulsamos en “Añadir cliente”, se abre un menú, rellenamos los datos y pulsamos en “Añadir”.

Todos los campos son obligatorios, si no rellenamos alguno no se añadirá el trabajador.

El DNI debe llevar un formato correcto, un ejemplo válido es 37147813F.

El teléfono debe llevar un formato correcto, un ejemplo válido es 645543765.

El cargo debe ser uno válido de los del desplegable que se abre.

Para añadir el cliente, debemos pulsar en el botón de “Añadir”.



Imagen 48. Manual de usuario añadir cliente

#### Eliminar cliente

En url/clientes, sobre el cliente deseado pulsamos en el icono de eliminar.



Imagen 49. Manual de usuario eliminar cliente

### 6.3.9. Configuración

#### Editar configuración

En url/configuración, modificamos los datos y pulsamos en “Guardar”. Solo los administradores y RR.HH. podrán ingresar a url/configuración/DNI, y modificar los datos de la persona asociada al DNI introducido.

Si no modificamos ningún cambio y pulsamos en guardar, saltará un aviso advirtiendo de que no hemos seleccionado nada nuevo.

El DNI nunca se podrá modificar.

El teléfono debe llevar un formato correcto, un ejemplo válido es 645543765.

El cargo debe ser uno válido de los del desplegable que se abre, si el usuario no tiene privilegios para editar el cargo, no tendrá disponible está opción.

El apartado de contraseña solo estará disponible para los empleados que dispongan de una cuenta web.

Para confirmar los cambios debemos pulsar en el botón de “Guardar”.



Imagen 50. Manual de usuario editar configuración

# 7. Conclusión

Echando la vista atrás, después de todos estos meses de trabajo, puedo decir que se han cubierto todos los requisitos especificados al inicio de esta documentación.

A lo largo del desarrollo, he podido aprender a usar nuevas herramientas que antes no conocía, lo que ha convertido este proyecto en una aventura en vez de un trabajo más.

Durante este periodo, he podido ver con mis propios ojos la importancia de probar cosas nuevas y no tenerle miedo al cambio, no todo lo que ya conocemos es lo único que existe, cuanto más he aprendido, más me he dado cuenta de que me queda mucho por aprender.

Dejo pendiente para mejorar bastantes cosas de este proyecto, pero para una primera versión he quedado más que satisfecho, me hubiese gustado implementar que en la base de datos no se eliminen los datos, sino que tuviesen una columna en la que nos decía si ese registro estaba activo o no, sobre todo de cara a la legalidad no perder esos datos que debemos guardar.

Por último, quiero dar las gracias a mis profesores por todo el apoyo que me han dado y por haberme enseñado todo lo necesario para enfrentarme al entorno laboral, sin ellos hoy no estaría escribiendo esto.

# 8. Bibliografía/Webgrafía

#### PHP

<https://www.php.net/manual/es/index.php>

<https://devdocs.io/php-apache/>

Gran Libro De PHP. Creación De Páginas Web Dinámicas - 2ª Edición - Amazon

Modern PHP: New Features and Good Practices - Amazon

#### JavaScript

<https://es.javascript.info/>

<https://www.w3schools.com/>

JavaScript elocuente. Una introducción moderna a la programación - Amazon

#### AJAX

<https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data>

<https://lenguajejs.com/javascript/peticiones-http/ajax/>

#### MVC

<https://dcreations.es/cursos/curso-patrones-de-disenio-en-php-gratis/patron-mvc-en-php>

Write PHP Like a Pro: Build a PHP MVC Framework From Scratch - Udemy