



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

Ingeniería Web: Proyecto Web Colaborativo

Gestión de Mantenimiento de Equipos

Grupo: IW-09

Alumno: Iván Diéguez Larios

Michael Hindle Codón

Profesor: Jon Vadillo Romero

Iker Pastor López

1	Introducción	3
2	Objetivos del proyecto	3
2.1	Resumen de la dirección.....	3
2.2	Visión general del reto.....	3
2.3	Tareas principales	3
2.4	Supuestos clave del proyecto	3
2.5	Estimación de tiempo	3
2.6	Planificación temporal.....	4
3	Especificaciones de requisitos del sistema.....	4
3.1	Descripción general	4
3.1.1	Funciones a cumplir.	4
3.1.2	Inconvenientes actuales.....	¡Error! Marcador no definido.
3.1.3	Necesidades de la nueva aplicación	4
3.2	Descripción de requisitos del nuevo sistema.....	5
3.2.1	Modelo funcional	5
3.2.2	Modelo lógico de datos	6
3.2.3	Modelo de comportamiento	7
4	Especificación del diseño	9
4.1	Introducción	9
4.1.1	Principales funciones del software.....	9
4.1.2	Descripción del entorno de desarrollo	10
4.2	Arquitectura física y entorno tecnológico	10
4.2.1	Descripción general.....	10
4.3	Descripción del diseño	10
5	Manual de usuario.....	10
6	Incidencias del proyecto y conclusiones	11
7	Bibliografía	11

1 Introducción

Nuestro reto colaborativo era la creación de una página web del mantenimiento de equipos de la empresa Doustomach Innovation S.L. En esta página web se desea la creación, modificación y eliminación de equipos, tickets y empleados.

2 Objetivos del proyecto

2.1 Resumen de la dirección

Se nos manda la creación de una página web de una empresa llamada Doustomach Innovation S.L. que se necesita una gestión de mantenimiento de equipos, teniendo en cuenta que los equipos son creados por empleados y los empleados tienen un registro de las acciones realizadas sobre los equipos que son llamadas tickets, donde se anota todos los detalles sobre lo realizado.

2.2 Visión general del reto

Por lo que se presenta en el reto, la vista que tenemos sobre el reto es la creación de tres páginas internas, separadas de entre sí y en cada página crear ahí la introducción, modificación y borrado de los equipos, tickets y empleados.

2.3 Tareas principales

Se debe realizar páginas web en las cuales se pueda recoger información, para que se guarde en Python por medio de un formulario, las cuales estarán separadas dependiendo de cual es el dato que se requiere guardar (equipo, empleado o ticket). Del mismo modo en la misma página web de equipo, empleado y ticket, tener la opción de poder modificar datos que ya estuvieran guardados en Python, en esta parte se tendría que mostrar los datos guardados y poder modificarlos y volver ser guardados. Y por último la opción de borrar el registro de cualquiera de las tres ramas importantes.

2.4 Supuestos clave del proyecto

- Crear: Poder crear un registro y pueda ser guardado en Python.
- Modificar: Sacar información de Python, mostrar el contenido guardado, capacidad de modificar cualquier dato y ser guardado de nuevo en Python.
- Borrar: Seleccionar un registro de la tabla y capacidad de ser eliminado de Python teniendo en cuenta las relaciones que tenga con el resto de las tablas.

2.5 Estimación de tiempo

El tiempo que se nos asignó para la creación de la página web sería de un mes y medio con festivos incluidos. Se nos dividió el trabajo en dos partes, una la creación de HTML, CSS y Python, para que se pudiera añadir, modificar y eliminar unos registros. Después se nos puso una parte donde se tendría que mejorar la parte de Python y el uso de JavaScript para poder tener dinamismo en la páginas web.

2.6 Planificación temporal

06/04/2021 → Inicio de creación del grupo y verificación del reto a realizar.
09/04/2021 → Reunión del equipo para la repartición de trabajo a realizar.
17/04/2021 → Recopilación del trabajo individual y ponerlo en conjunto.
20/04/2021 → Reunión del equipo para verificar los trabajos individuales y poder debatir modificaciones del trabajo, así como poder añadir nuevas ideas para mejor uso de la página web.
21/04/2021 → Repartición de nuevas tareas o modificación de algunas anteriores.
30/04/2021 → Reunión de participantes para unir en conjunto los trabajos individuales.

01/05/2021 → Comprobación de creación, modificación y eliminación de la página web
02/05/2021 → Enviar al profesor la página web de como esta siendo trabajada.
07/05/2021 → Feedback del trabajo junto mejoras que se deben realizar.
12/05/2021 → Reunión para implementar las mejoras, así como repartición de nuevas funciones de la página web.
16/05/2021 → Recopilación de las mejoras de la página web.
20/05/2021 → Reunión del equipo para comprobar las reparticiones e informar de errores.
24/05/2021 → Inicialización del documento de desarrollo del trabajo.
25/05/2021 → Reunión de grupo para implementar las soluciones y comprobar todo.
27/05/2021 → Terminación del documento de desarrollo y enviar todas las carpetas al profesor para la inicialización de la evaluación.

Iván → Encargado de Python y la relación de HTML con Python.

Michael → Encargado de HTML, CSS y la documentación del desarrollo.

3 Especificaciones de requisitos del sistema

3.1 Descripción general

3.1.1 Funciones a cumplir.

- Un botón donde se pueda agregar un registro nuevo.
- Mostrar una tabla de los registros guardados.
- Tener en cada registro una opción de mostrar más detalles.
- Tener en cada registro una opción para modificar el registro.
- Tener en cada registro una opción de eliminar el registro.
- Un buscador de registros.
- Un zoom para aumentar o disminuir la página.
- Tener un registro de usuario.

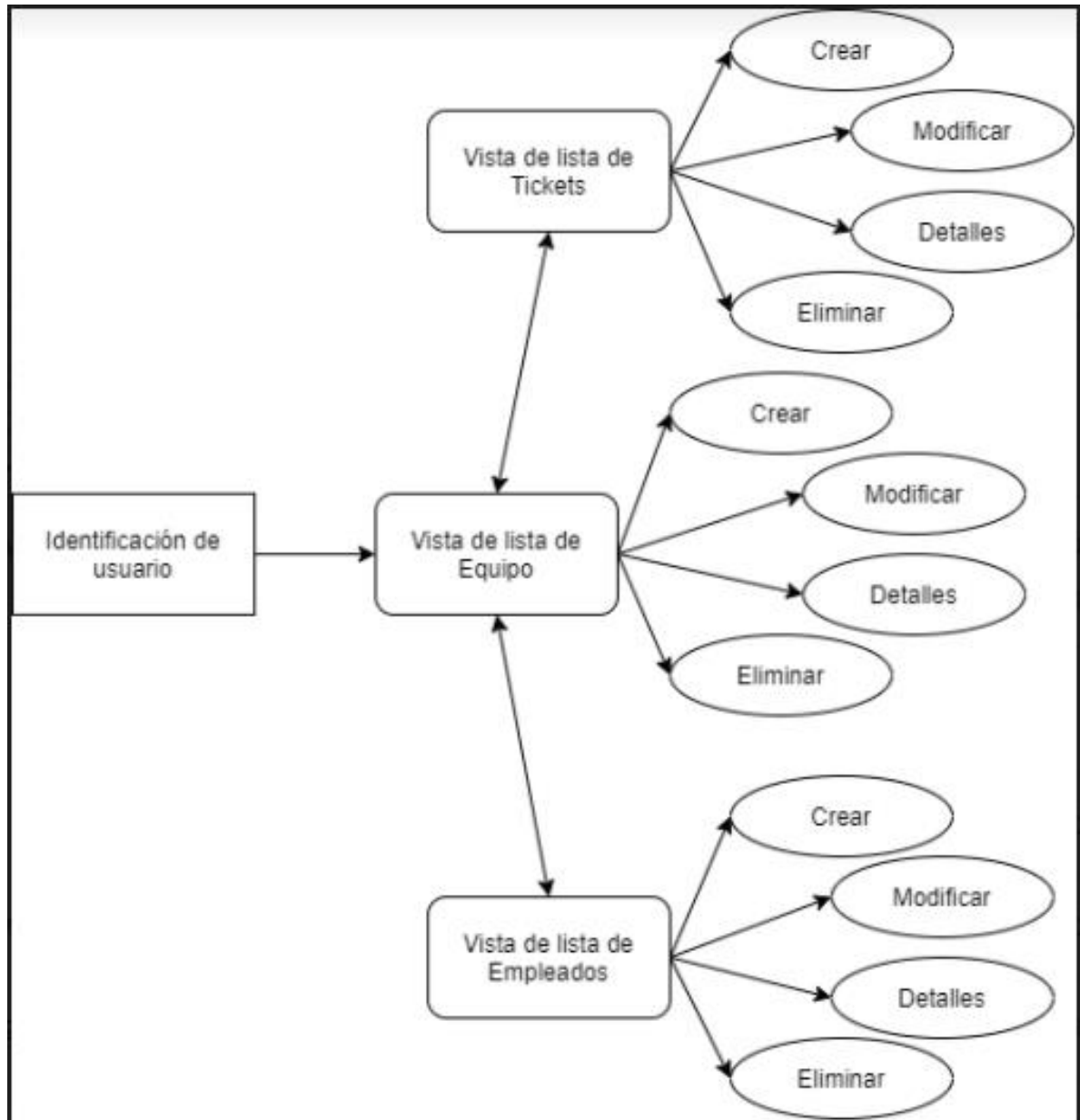
3.1.3 Necesidades de la nueva aplicación

- El funcionamiento optimo del acceso de usuario.
- Ordenar de forma correcta la función que ocultar los registros.

3.2 Descripción de requisitos del nuevo sistema

3.2.1 Modelo funcional

Una representación rápida del modelo funcional sería el que se ve en la imagen. Primero es necesario iniciar sesión para entrar en la página web. Una vez iniciada la sesión se mostrará la lista de equipos que esta guardada, donde en ella se puede acceder a la vista de las listas de Tickets o de Empleados. En cada una de las partes de las vistas de las listas se puede crear, modificar, mostrar mas detalles o eliminar un registro.



1) Modelo funcional

3.2.2 Modelo lógico de datos

Son cada uno de los registros con los campos necesarios que se necesita obtener en el proyecto, están metidas en el archivo models.py.

En esta imagen se crea los atributos de la clase empleado, en esta clase se requiere con un nombre, apellidos, email, DNI y un teléfono.

```
class Empleado(models.Model):
    nombre = models.CharField(max_length=25)
    apellidos = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    telefono = models.IntegerField(default=0)
    dni = models.CharField(max_length=9)
    def __str__(self):
        return f'Nombre: {self.nombre} {self.apellidos}, Correo: {self.email}, Telefono: {self.telefono}, DNI: {self.dni}'
```

2) Modelo de la clase Empleado

En esta modelo se encuentra la clase equipo, es una clase hija de ticket donde se guarda los atributos de numero de serie, modelo, marca, tipo de equipo, fecha de adquisición, fecha puesta en marcha, nombre del proveedor, teléfono del proveedor y planta de la empresa.

```
class Equipo(models.Model):
    numerodeserie = models.IntegerField(default=0)
    modelo = models.CharField(max_length=50, default='')
    marca = models.CharField(max_length=50, default='')
    tipoequipo = models.CharField(max_length=50, default='')
    fecha_adquisicion = models.DateField()
    fecha_puestaenmarcha = models.DateField()
    proveedor_nombre = models.CharField(max_length=50, default='')
    proveedor_tlf = models.IntegerField(default=0)
    planta = models.CharField(max_length=50, default='')
    def __str__(self):
        return f'Proveedor:{self.proveedor_nombre} - {self.proveedor_tlf}, Nserie: {self.numerodeserie}, Tipo:{self.tipoequipo}'
```

3) Modelo de la clase Equipo

Y este modelo padre, es el de la clase ticket donde se guarda unos atributos básicos pero se debe tener en cuenta que si no hay un registro previo de las clases Equipo y Empleado, no se podrá añadir correctamente el registro del ticket.

```
class Ticket(models.Model):
    numeroreferencia = models.IntegerField(default=0)
    titulo = models.CharField(max_length=50)
    descripcion = models.TextField(max_length=1000)
    fecha_apertura = models.DateField()
    fecha_resolucion = models.DateField()
    nivel_urgencia = models.CharField(max_length=50)
    tipo_ticket = models.CharField(max_length=50)
    estado_ticket = models.CharField(max_length=50)
    comentario = models.TextField(max_length=1000)
    FK_Equipo_ID = models.ForeignKey(Equipo, on_delete=models.CASCADE)
    FK_Empleado_ID = models.ForeignKey(Empleado, on_delete=models.CASCADE)
```

4) Modelo de la clase Ticket

3.2.3 Modelo de comportamiento

Para una mejor visualización del modelo de comportamiento, se mostrara diferentes imágenes con la intención de que sea más intuitivo.

Esta es una vista previa de la página web, se escogió los colores azul y negro para aparentar una zona ordenada y limpia.



5) Página web

En la parte de arriba (head) se encuentra con el nombre de la empresa, y también los diferentes apartados de los registros junto con el zoom.



6) Nombre de la empresa, zoom para aumentar o disminuir la vista de la página y diferentes apartados de la página.

En el medio, se encuentra el contenido de los registros guardados y un filtro para una búsqueda de todos los tipos de registros que hay, del mismo modo están las funciones de crear, borrar, mostrar más detalles y modificar

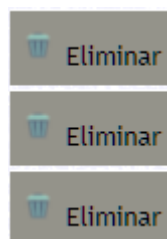


7) Botones para la creación de nuevos registros.

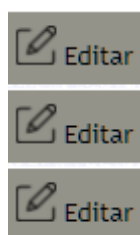
8) Filtro por la columna de tipo

Número de Serie	Tipo	Nombre del proveedor	Telefono del proveedor
763762763	Alpha	Juan	98989988
989898989898	patata	Andres e Hijos	98989898
7676	Bravo	Illo	987551145

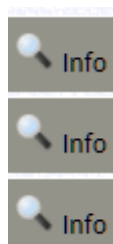
9) Tabla de los registros guardados.



10) Botones para borrar los registros dependiendo en que fila se encuentre

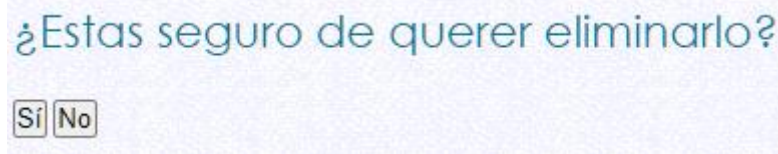


11) Botones para la opción de modificar los campos de un registro



12) Botones para mostrar más datos sobre el registro

Una vez que se selecciona la opción de borrar un registro, se tiene que confirmar antes de que el registro sea eliminado por completo, esta sería el aviso para la confirmación.



13) Confirmación para la eliminación de un registro

Y en su vez si te has metido dentro de una función que no era la deseada, siempre se puede volver a la página anterior con el siguiente botón.



14) Botón para volver a la página anterior.

Y por ultimo se encuentra un pie de la página, donde en la parte izquierda se informa un poco sobre a lo que la página consiste, en la columna central se encuentran enlaces a páginas de redes sociales y en la columna de la derecha una información básica de atención al cliente de la empresa.



15) Pie de la página web

4 Especificación del diseño

4.1 Introducción

4.1.1 Principales funciones del software

Esta planteado en diferentes archivos CSS para un mejor acceso y rapidez de mantener el código si es necesario modificar algo nuevo para así poder acceder a diferentes partes de la página web. Se dividen en los siguientes archivos: footer.css, contenido.css y style.css.

En cada uno de ellos tienen una función diferente, en el archivo footer.css se encuentra el diseño del pie de la página web, en el archivo contenido.css se enfoca

como esta estructurado la parte del contenido y en la parte de style.css se encuentra todo lo relacionado al resto de la página web.

4.1.2 Descripción del entorno de desarrollo

Visual Studio Code

En visual studio code fue el principal entorno de desarrollo de la página web, en el se uso la mayor parte del tiempo para la creación de la página web.

Github

Se uso mayormente para guardar el progreso de cada uno y así poder tener la versión del proyecto mas actualizada del momento. Nos comunicábamos previamente para que el progreso de uno que haya modificado el trabajo no se le borre cuando quiera actualizar los archivos.

CMD

Usamos la consola virtual para poder iniciar el servidor y así tener una vista de como se estaba generando los HTML y los CSS.

4.2 Arquitectura física y entorno tecnológico

4.2.1 Descripción general

Su estructura es simple, fácil de usar y completa para que cuando el usuario acceda a la página web, se encuentre cómodo y de fácil de entender, de así, el usuario pudiera manejar la página sin la necesidad de unos pasos a seguir.

4.3 Descripción del diseño

Una estructura simple, con un head, un body y un footer. En el head se encuentra el nombre de la empresa y un menú para acceder a una página diferente. En el body solo esta la tabla donde se muestra los diferentes registros de cada página. Y en el footer se encuentra una pequeña descripción de para qué sirve la página web, una salida a unas redes sociales y una descripción de la empresa.

5 Manual de usuario

- 1º → Registrar un usuario.
- 2º → Iniciar usuario.
- 3º → Visualización de la página web.
- 4º → Para poder añadir un nuevo registro, darle click la imagen de más.
- 5º → Para mostrar mas detalles, darle click a la imagen de la lupa.
- 6º → Para editar un registro, darle click a la imagen de la libreta.
- 7º → Para visualizar otra tabla, darle click en una de las tres opciones que se encuentra debajo del nombre de la empresa.

6 Incidencias del proyecto y conclusiones

Hemos tenido el proyecto las cosas justas con respecto al tiempo, el motivo fue que el día de que nos asignaron el grupo, se nos asignó una semana de retraso con respecto al resto de los grupos, eso conllevó que tuviéramos que pedir unos días extras para poder entregarlo a la fecha prevista del primer hito del trabajo. Otra incidencia del proyecto fue que uno de nuestro equipo no participaba en ninguna ocasión, eso conllevó que encima que éramos 3 en grupo (un participante menos que en los grupos restantes), éramos 2 participantes del proyecto los que tendrían que llevar el trabajo hacia adelante. Gracias a Iván que sabía más al respecto sobre Python que pudo avanzar rápidamente con el proyecto y gracias a Michael que sabía más sobre HTML y CSS que se pudo encargarse de dejar una buena presentación de la página web.

Una conclusión que destacaría sería que los problemas se pueden presentar en cualquier momento del proyecto, haciendo un apartado donde el tiempo de ser terminado sería de una hora, llegue a tardar más, hasta encontrar una solución al problema.

7 Bibliografía

-Plantillas

<https://www.html5webtemplates.co.uk/templates/>

-Información

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

<https://www.w3schools.com/js/default.asp>

<https://www.w3schools.com/python/default.asp>