

Ingeniería Web – Proyecto Web Colaborativo

Título: Reto 2: Gestión de pedidos

Curso: 2º Grado en Industria Digital (Semestre 2º)

Materia: Ingeniería Web

Estudiantes: Eneko Galdos Sánchez

Yeray Lamelas Cancho

Jon Zatón Toledo

Grupo: IW-13

Facultad de Ingeniería

UNIVERSIDAD DE DEUSTO

VITORIA - GASTEIZ, MAYO DE 2020

RESUMEN

La empresa llamada Deustronic Components S.L. ha decidido que ya ha llegado la hora de solucionar sus problemas derivados del uso de hojas Excel y comenzar a utilizar una aplicación para la gestión de pedidos. Tras un análisis de las alternativas disponibles, ha optado por encargar el desarrollo de una nueva aplicación al equipo de ingenieros IW-13 del Grado Industria Digital de la Universidad de Deusto.

La empresa se dedica a la fabricación de productos electrónicos (dispone de varias categorías y modelos) que comercializa a empresas de todo el planeta proporcionándoles los mejores productos del mercado y un servicio de alta calidad. Le gustaría poder gestionar su catálogo de productos, así como los pedidos realizados por los clientes.

Entonces, como equipo hemos planeado y realizado dicho proyecto sobre un entorno virtual en Django en el que nos hemos empleado funcionando como distintos departamentos de programación desempeñando plantillas, ventanas y sobre todo diversas funcionalidades utilizando diferentes tipos de código (HTML, CSS, Python y JavaScript) para llevar a cabo la propuesta anteriormente explicada en fecha prevista y con una alta calidad.

Descriptores

- Django
- Plantillas
- Funcionalidades
- Python
- JavaScript

Índice

| | |
|---|-----|
| Vitoria - Gasteiz, mayo de 2020 | iii |
| Resumen..... | v |
| 1. EXPLICACIÓN GENERAL DEL PROYECTO EN DJANGO | 1 |
| Descripción de los escenarios y funcionalidades platenadas..... | 1 |
| 1.1.1 Header y footer..... | 1 |
| 1.1.2 Página Principal..... | 2 |
| 1.1.3 1.1.3 Menú..... | 3 |
| 1.1.4 Vistas de listado..... | 4 |
| 1.1.5 1.1.5 Vistas de celda..... | 5 |
| 1.1.6 1.1.6 Crear..... | 6 |
| 1.1.7 1.1.7 Editar. | 7 |
| 1.1.8 1.1.8 Ver..... | 8 |
| 1.1.9 Borrar. | 8 |
| 1.2 Descripción del modelo de datos | 9 |
| 1.3 Descripción de las plantillas utilizadas | 10 |
| 1.3.1 Plantilla Base..... | 10 |
| 1.3.2 Plantilla Productos y Pedidos | 11 |
| 1.3.3 Plantilla Componentes y Clientes | 11 |
| 1.3.4 Plantilla de Producto, Componente, Cliente y Pedido | 12 |
| 1.4 Descripción de la estructura del proyecto web..... | 13 |
| 1.2 Descripción de funcionalidades añadidas implementadas | 15 |
| 2. Explicación general del proyecto en javascript..... | 17 |
| 2.1 Descripción de la estructura del proyecto web..... | 17 |

| | |
|---|----|
| 2.2 Descripción de las funcionalidades implementadas..... | 17 |
| 3. Conclusiones | 19 |
| 4. Bibliografía..... | 20 |

1. EXPLICACIÓN GENERAL DEL PROYECTO EN DJANGO

Nuestra aplicación consiste en la página de una empresa de componentes y productos informáticos.

Desde ella se podrán registrar nuevos clientes, así como componentes y productos. De la misma forma se podrán realizar pedidos de dichos productos, a nombre del cliente que deseemos.

Todos estos campos anteriormente mencionados se podrán ver en detalle, editar o borrar, navegando a través de las diferentes páginas y pestañas de forma muy intuitiva.

A continuación, se mostrará una vista de las diferentes pantallas explicando que se puede hacer desde cada una de ellas.

DESCRIPCIÓN DE LOS ESCENARIOS Y FUNCIONALIDADES PLATENADAS

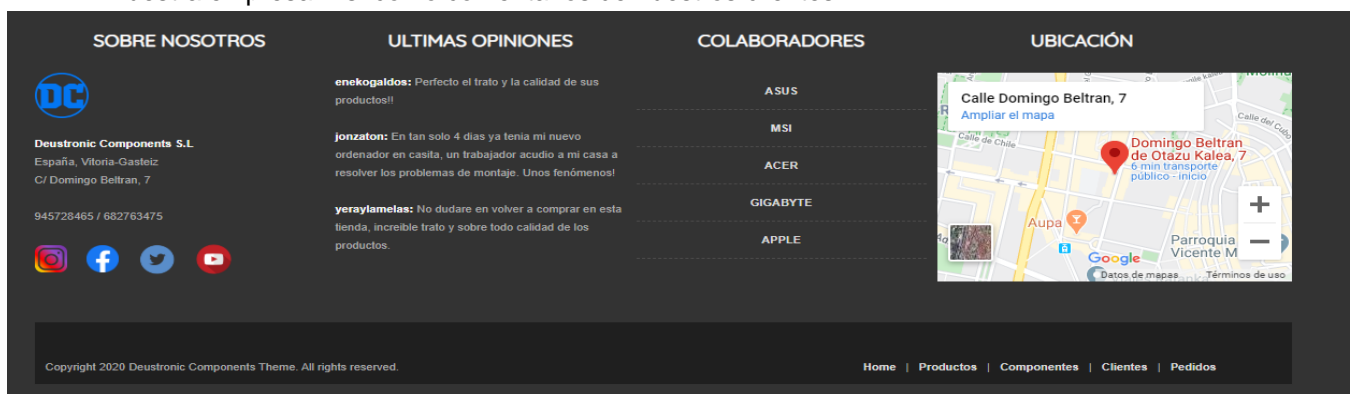
1.1.1 Header y footer

Todas nuestras pantallas cuentan con un header y un footer. En el header, tenemos menús desplegables que nos llevarán a las páginas de crear y visualizar los clientes, componentes, productos y pedidos.

Por otro lado, en el footer, tendremos enlaces a sitios de interés e información relevante sobre



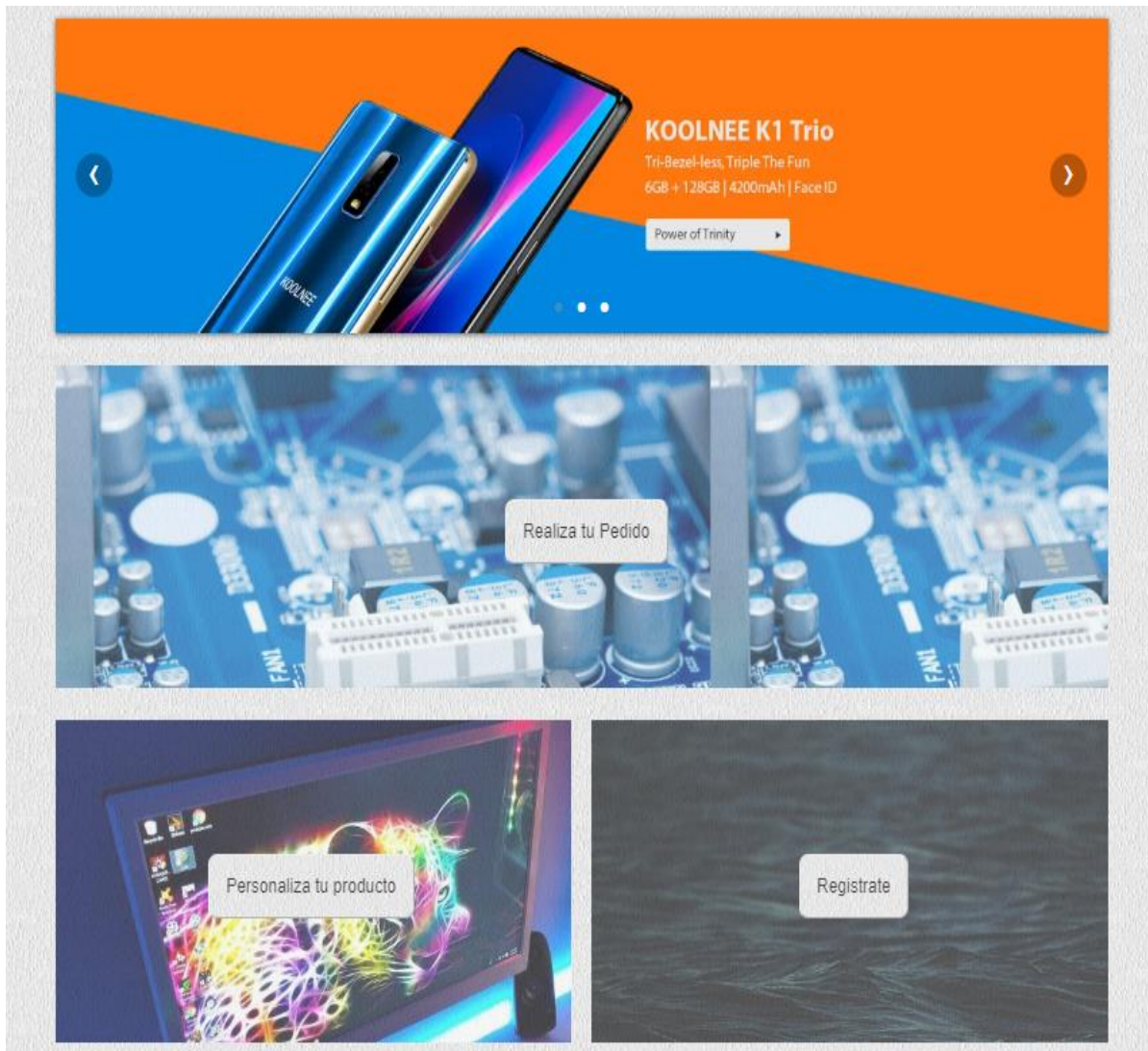
nuestra empresa. Así como comentarios de nuestros clientes.



1.1.2 Página Principal

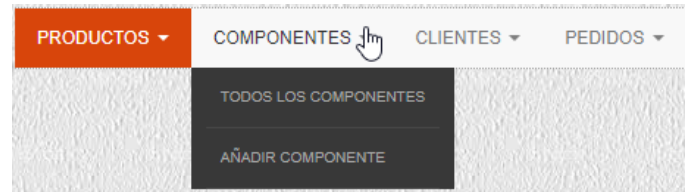
En esta pantalla, encontraremos un banner publicitario, con novedades y los últimos productos. Podremos cambiar de productos gracias a dos botones de siguiente y anterior a cada lado del banner.

Por otro lado, tendremos tres botones que nos llevarán a las pestañas más relevantes. La de personalizar producto, la de registrarse y la de hacer un nuevo pedido.



1.1.3 1.1.3 Menú

Se ha dispuesto de un menú horizontal con cuatro celdas que se encuentra en la esquina superior derecha en todas las pantallas a las que accedamos, este menú, nos da una mayor facilidad y rapidez a la hora de navegar hacia las ventanas en las que queremos interactuar. Cada celda contiene un submenú desplegable para acceder a distintas ventanas.



A continuación, podremos visualizar las distintas ventanas a las que se pueden acceder mediante este menú, en las que se podrán ejecutar diferentes acciones en cada una de ellas.

1.1.4 Vistas de listado

| CLIENTES | | | |
|----------|------------|-------------------|---|
| CLIENTE | CIF | CONTACTO | OPCIONES |
| Aernova | 1234567 | aernova@gmail.com | Editar Borrar Ver |
| dyfugij | lsndyfugih | yvuhbjn | Editar Borrar Ver |
| ikerlan | 435678 | iker@gmail.com | Editar Borrar Ver |
| yerysl | 4546456 | bhjbnjkn | Editar Borrar Ver |

| COMPONENTES | | | |
|-------------|-------------------|---------------------|---|
| REFERENCIA | MODELO | MARCA | OPCIONES |
| 4444444 | 444444444 | 4444444444444444444 | Editar Borrar Ver |
| 123 | PruebaComponente | dfgh | Editar Borrar Ver |
| 0 | modelo | marca | Editar Borrar Ver |
| 2 | prueba componente | deustronic | Editar Borrar Ver |
| 10 | tfghvjb | yghkj | Editar Borrar Ver |
| 234567767 | lrduykvhjbnk | tyfuhgjk | Editar Borrar Ver |

Desde las vistas de listado podremos acceder a todos los clientes o los productos registrados en la aplicación. En la lista podremos ver los datos más relevantes, y tendremos tres botones desde los cuales podremos EDITAR, VER, O BORRAR.

1.1.5 1.1.5 Vistas de celda.



Desde las vistas de celdas podremos todos los clientes o los productos registrados en la aplicación. En cada celda podremos ver los datos más relevantes, y tendremos tres botones desde los cuales podremos EDITAR, VER, O BORRAR.

1.1.6 1.1.6 Crear.

AÑADIR NUEVO PRODUCTO

Referencia:

Precio:

Nombre:

Descripción:

Categoría:

Tipo componente:

- PruebaComponente
- prueba componente
- trdyukvhjkb
- modelo

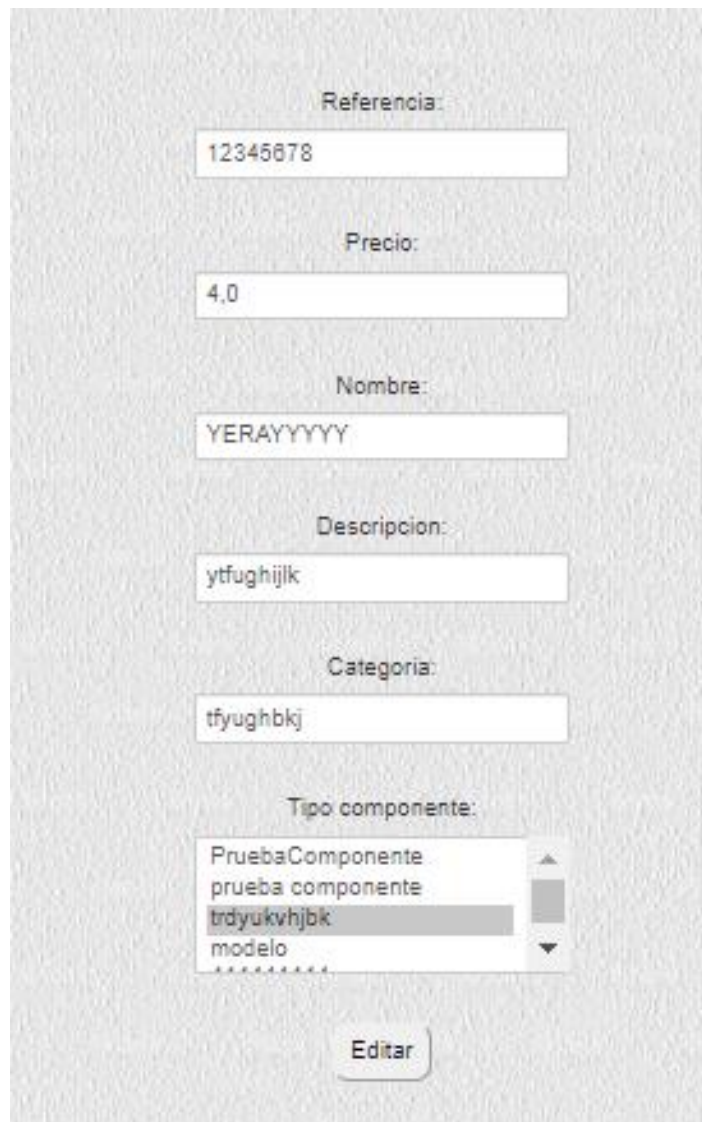
Añadir

En las pestañas de crear tendremos campos que tendremos que rellenar con los datos solicitados. Algunos datos los tendremos que introducir nosotros y otros nos los proporcionará la app a través de listados.

La imagen del ejemplo es de “añadir un nuevo producto”, para la cual necesitamos ver los componentes de los que puede disponer y que han sido previamente guardados desde “añadir componente”. Dichos componentes nos aparecen en una lista.

Una vez completados todos los campos le daremos al botón añadir y se nos guardará en nuestra base de datos.

1.1.7 1.1.7 Editar.



Referencia:

12345678

Precio:

4.0

Nombre:

YERAYYYY

Descripcion:

ytfughijkl

Categoria:

tfyugbkbj

Tipo componente:

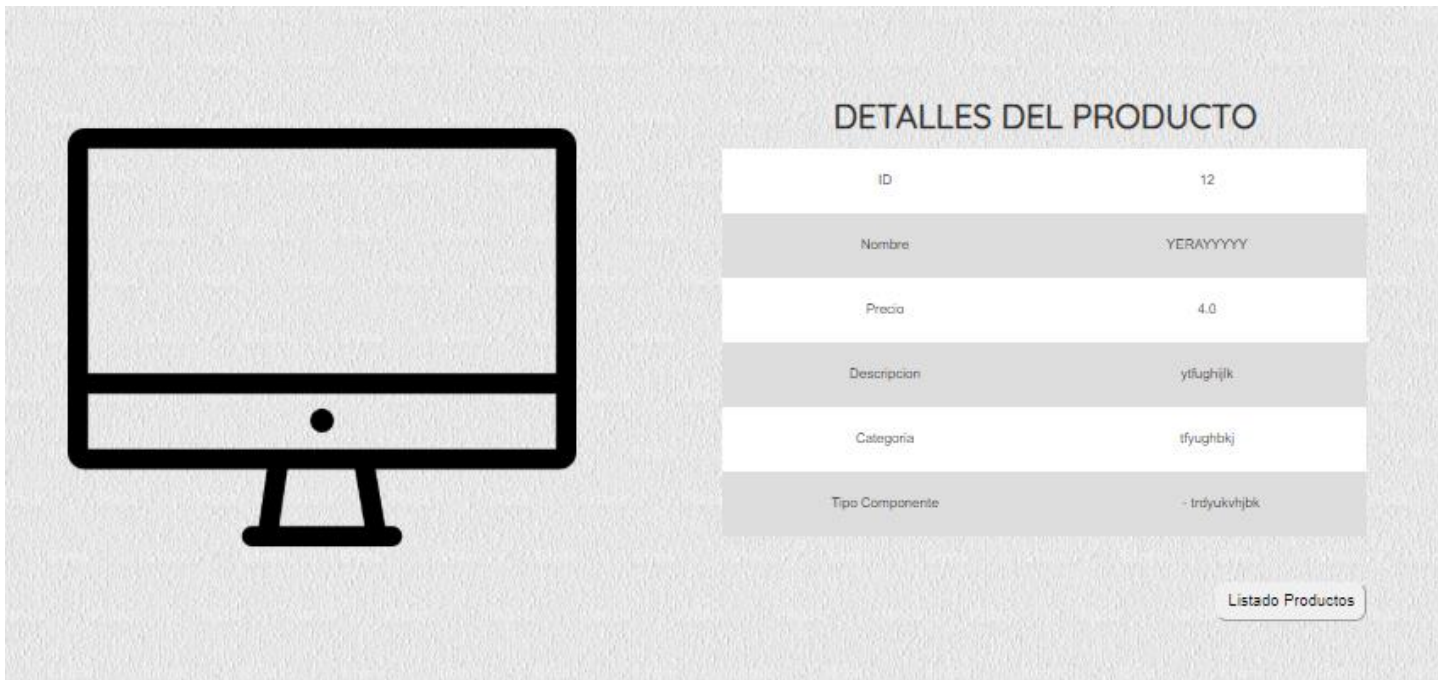
PruebaComponente
prueba componente
trdyukvhbjk
modelo
.....

Editar

Desde el botón editar accederemos a una página donde se nos mostrará la información que tiene el elemento seleccionado. Una vez cambiada le daremos a botón Editar y esa nueva información se guardará en nuestra base de datos.

En este apartado se ha definido que las relaciones “ManyToMany” no se puedan editar, cada pedido tendrá los productos que se han seleccionado en su creación, si se desea cambiar, el usuario deberá realizar un nuevo pedido con los productos nuevos. Ocurre lo mismo en productos, que no se pueden cambiar los componentes.

1.1.8 1.1.8 Ver.



Desde el botón ver accederemos a una vista detalle del elemento en cuestión, donde nos mostrará toda la información en detalle. Si deseamos volver a la lista tendremos un botón en la parte inferior derecha.

1.1.9 Borrar.

127.0.0.1:8000 dice

¿Seguro que quieres borrar los datos de: 1?

Aceptar

Cancelar

Si pulsamos el botón borrar, nos aparecerá una pantalla emergente donde tendremos la posibilidad de confirmar o de cancelar.

Si aceptamos dicho elemento se borrará de la base de datos y de nuestra aplicación.

1.2 DESCRIPCIÓN DEL MODELO DE DATOS

Hemos creado una serie de modelo de datos que serán gestionados mediante ciertas funcionalidades para conseguir lograr el objetivo de la aplicación web, los modelos de datos diseñados son los siguientes:

- **Productos**

- Referencia → tipo Char
- Precio → tipo Float
- Nombre → tipo Char
- Descripción → tipo Char
- Categoría → tipo Char
- Tipos de componentes que contiene el producto → asociado al modelo "Componente" en relación "ManyToMany"

Con este modelo podemos crear y guardar en la base de datos una serie de productos con estas características.

- **Componente**

- Código de referencia → tipo Int
- Nombre de modelo → tipo Char
- Marca → tipo Char

Con este modelo podemos crear y guardar en la base de datos una serie de componentes con estas características.

- **Cliente**

- Cif → tipo Char
- Nombre_empresa → tipo Char
- Datos_contacto → tipo Char
-

Con este modelo podemos crear y guardar en la base de datos una serie de clientes con estas características.

- **Pedido**

- Codigo_referencia → tipo Char
- Fecha → tipo Date
- Datos_cliente → tipo ForeignKey
- Producto_solicitado → asociado al modelo "Producto" en relación "ManyToMany"
- Cantidad_producto → tipo Int
- Precio_total → tipo Float

Con este modelo podemos crear y guardar en la base de datos una serie de pedidos con estas características.

1.3 DESCRIPCIÓN DE LAS PLANTILLAS UTILIZADAS

1.3.1 Plantilla Base

En esta plantilla se ha establecido un diseño fijado para las distintas ventanas de las que se compone la aplicación web mediante el uso de código HTML y CSS.

La plantilla se conforma por:

- Un fondo de pantalla
- Una cabecera en la que aparece el logo y nombre de la empresa (al clickar redirecciona a la página principal) y un menú para poder navegar a través de las distintas ventanas.
- Un footer el cual está formado de los contactos de las redes sociales de la empresa, últimas opiniones publicadas por los clientes, colaboradores de la empresa y la ubicación en Google Maps.

1.3.2 Plantilla Productos y Pedidos

Esta plantilla proporciona el formato diseñado a las ventanas “Todos los productos” y “Todos los pedidos” apoyado de la plantilla base, ventanas en las que se muestran mediante celdas, la cantidad de productos y pedidos que existen en la base de datos.

Las celdas están diseñados con la capacidad de mostrar la información de cada producto y pedido, y también poseen tres botones cada uno; “Editar”, “Borrar” y “Ver”.

Celda de producto

REFERENCIA: P1

NOMBRE: Producto1

PRECIO: 1.0\$

Editar
Borrar
Ver

Celda de pedido

CODIGO DE REFERENCIA: Ped2

FECHA: 2020-05-02

CLIENTE: undefined

Editar
Borrar
Ver

1.3.3 Plantilla Componentes y Clientes

Esta plantilla proporciona el formato diseñado a las ventanas “Todos los componentes” y “Todos los clientes” apoyado de la plantilla base, ventanas en las que se muestran mediante una tabla, la cantidad de componentes y clientes dispuestos en diferentes filas que existen en la base de datos.

Cada tabla posee diferentes columnas en las que se muestran los diferentes datos por los que están formados cada componente y cliente. Una de las columnas es genérica que proporciona tres botones en cada fila; “Editar”, “Borrar” y “Ver”.

Tabla de componentes

| REFERENCIA | MODELO | MARCA | OPCIONES |
|------------|--------------|------------------|--|
| 5 | Componente 5 | MarcaComponente5 | <div>Editar</div> <div>Borrar</div> <div>Ver</div> |

Tabla de clientes

| CLIENTE | CIF | CONTACTO | OPCIONES |
|----------|-----|--------------------|--|
| 222 | 222 | 222 | <div>Editar</div> <div>Borrar</div> <div>Ver</div> |
| Cliente2 | C2 | cliente2@gmail.com | <div>Editar</div> <div>Borrar</div> <div>Ver</div> |

1.3.4 Plantilla de Producto, Componente, Cliente y Pedido

Esta plantilla proporciona el formato diseñado a las ventanas “Añadir producto”, “Añadir componente”, “Regístrate” y “Añadir pedido” apoyado de la plantilla base, ventanas en las que se muestran un formulario en el que se piden registrar los datos requeridos para cada apartado. También existe un botón “Añadir” en cada una de las ventanas para registrar los datos del formulario excepto en la de “Regístrate” que se llama “Registrarse”.

AÑADIR NUEVO COMPONENTE

Codigo referencia:

Nombre modelo:

Marca:

Añadir

CREAR NUEVO CLIENTE

Cif:

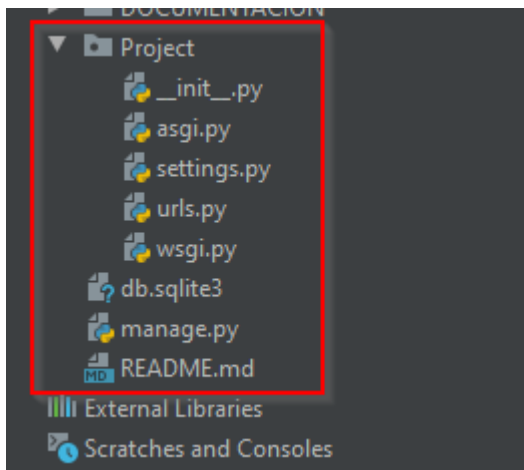
Nombre empresa:

Datos contacto:

Registrarse

1.4 DESCRIPCIÓN DE LA ESTRUCTURA DEL PROYECTO WEB

La estructura del proyecto ha sido creada mediante el comando “django-admin startproject Project”, aportando así los siguientes archivos:



“**manage.py**”: utilidad que permite interactuar con el proyecto.

“**__init__.py**”: archivo para que Python reconozca la carpeta como un Paquete y permita usar sus objetos en otras partes del proyecto.

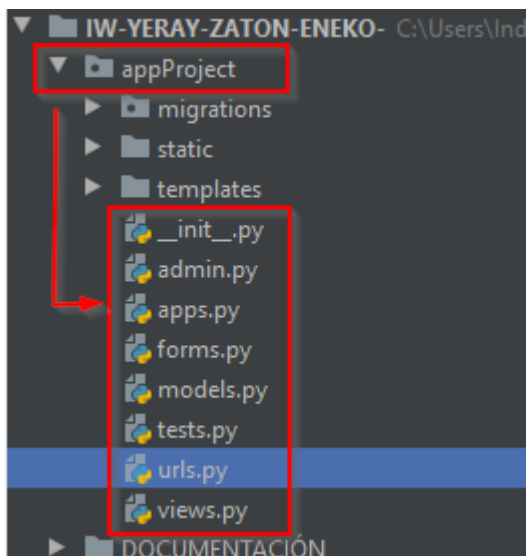
“**settings.py**”: archivo de configuración del proyecto.

“**urls.py**”: archivo donde definimos las URLs de la aplicación.

“**wsgi.py**”: archivo que ayuda a comunicarse la aplicación con el servidor.

“**asgi.py**”: archivo que ejecuta todo el código en un subproceso síncrono.

Por otro lado, la estructura de la aplicación ha sido creada mediante el comando “django-admin startapp appProject”, proporcionando así los siguientes archivos que se encargan de recibir las peticiones HTTP y hacer todas las operaciones necesarias desde esa recepción hasta el envío de una respuesta al cliente, pero también se han añadido varios archivos:



“**_init_.py**”: archivo que permite usar objetos en otras partes de la aplicación.

“**admin.py**”: archivo con el que podemos registrar los formularios de los distintos modelos de datos.

“**apps.py**”: archivo para ayudar al usuario a incluir cualquier configuración de aplicación para la aplicación.

“**forms.py**”: archivo que recoge los métodos para guardar el resultado de los formularios en los modelos.

“**models.py**”: archivo donde ubicaremos nuestros modelos de datos.

“**test.py**”: archivo proporciona una prueba Client para simular un usuario interactuando con el código al nivel de la vista.

“**urls.py**”: archivo donde definimos las URLs de la aplicación en conexión con **views.py**.

“**views.py**”: archivo donde definimos las distintas ventanas de la aplicación.

1.2 DESCRIPCIÓN DE FUNCIONALIDADES AÑADIDAS IMPLEMENTADAS

Una funcionalidad que se ha añadido al proyecto es un banner publicitario mostrando varias imágenes de una serie de componentes que vende la empresa mediante HTML.

Mediante dos botones, uno a cada lado de la imagen, nos permite cambiar ésta última a



derecha o a izquierda.

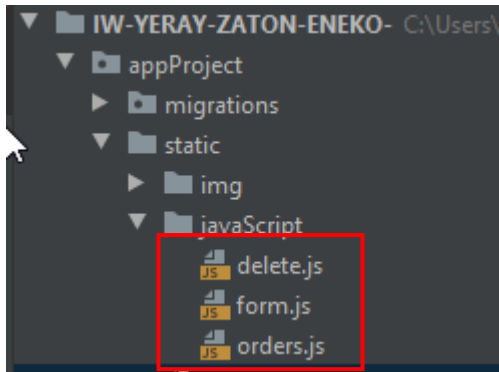


2. EXPLICACIÓN GENERAL DEL PROYECTO EN JAVASCRIPT

En este apartado se han implementado o modificado partes de código a JavaScript creando así una serie de funcionalidades recogidas en el documento de la entrega E3+E4.

2.1 DESCRIPCIÓN DE LA ESTRUCTURA DEL PROYECTO WEB

La estructura de JavaScript está compuesta en la carpeta “Static” ubicada en la aplicación “appProject” con los siguientes archivos:



“**delete.js**”: archivo con el que hemos implementado la funcionalidad de eliminar componentes.

“**forms.js**”: archivo que envía datos de un formulario mediante AJAX para su almacenamiento en BBDD.

“**orders.js**”: archivo con el que cargamos y modificamos datos de las DOM llamando a API y utilizando FETCH.

2.2 DESCRIPCIÓN DE LAS FUNCIONALIDADES IMPLEMENTADAS

Mediante el archivo **orders.js** implementamos la funcionalidad de poder cargar y modificar el DOM de pedidos, lo que nos permite mostrar los pedidos registrados en la BBDD en la ventana “Todos los pedidos”.

A través del archivo **forms.js** implementamos la funcionalidad de enviar los datos del formulario “cliente_create” de la aplicación mediante AJAX para así poder almacenar los datos introducidos por el usuario en la BBDD dando así de alta a un nuevo cliente.

La funcionalidad extra que hemos implementado en el archivo **delete.js** ha sido la de poder eliminar de la BBDD cualquier componente que aparezca en la tabla "COMPONENTES" de la ventana "Todos los componentes".

Esta funcionalidad se ejecuta al pulsar el botón "BORRAR" y esto hace que desaparezca la fila en la que se encontraba el componente en la tabla y actualiza el resto de los componentes desplazándolos hacia arriba ocupando la fila eliminada.

3. CONCLUSIONES

Tras realizar este proyecto, nos hemos dado cuenta de que tenemos la capacidad y conocimientos necesarios de poder realizar cualquier tipo de aplicación de cualquier tipo de ámbito.

Nos ha resultado muy interesante debido a que hemos podido profundizar tanto en HTML, CSS, Python como JavaScript.

4. BIBLIOGRAFÍA

Para este proyecto, nos hemos basado en diferentes fuentes para poder realizarlo:

- **Creación de archivos básicos del proyecto**

<https://github.com/jvadillo/curso-django-paso-a-paso>

- **Código funcional y conocimientos básicos**

<https://es.stackoverflow.com/>

<https://github.com/jvadillo/curso-django-paso-a-paso>

<https://www.youtube.com/user/padopadopado>

Pdfs de ALUD.

- **Plantilla (html, css)**

<https://plantillashtmlgratis.com/>