Deusto
Universidad de Deusto
Deustuko Unibertsitatea

Ingeniería Web – Proyecto Web Colaborativo

Titulo: APLICACIÓN WEB PARA LA GESTIÓN DE PEDIDOS

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

Materia: Ingeniería Web

Estudiantes: Ane Sanchiz Díaz de Corcuera

Aritz Ibañez Esteban

Grupo: IW-14

Profesor: Jon Vadillo Romero

Facultad de Ingeniería
UNIVERSIDAD DE DEUSTO

Vitoria - Gasteiz, mayo de 2020

Resumen

Lo que se ha tratado de realizar con este proyecto es una aplicación web real que simula las funcionalidades de un sistema de gestión empresarial, siguiendo los requerimientos del Reto 2. En él se han puesto a prueba los conocimientos del equipo y la capacidad para resolver problemas que pueden darse en situaciones reales.

Se trata de una aplicación de gestión de pedidos en la que el usuario podrá gestionar los pedidos, productos, componentes y clientes de la empresa. Esta aplicación debería ser privada y restringida a los empleados de la empresa a pesar de que no hemos realizado la función de log in.

Para su desarrollo en la parte del cliente, se han utilizado conocimientos de CSS, HTML5 y JavaScript, mientras que en el servidor se ha utilizado Django y Python.

Descriptores

- CSS
- HTML5
- Django
- Python
- JavaScript
- API Fetch
- AJAX

Contenido

Capítulo 1: Introducción

Este proyecto ha sido realizado por los alumnos, Ane Sanchiz y Aritz Ibañez, para la asignatura de Ingeniería Web del grado en Industria Digital. El propósito es el de realizar una aplicación web donde la empresa Deustronics pueda gestionar los pedidos realizados.

Capítulo 2: Objetivos del proyecto

Tareas principales

Las tareas principales de este proyecto son las siguientes:

- Generar el HTML de la pagina web
- Dar estilo al HTML mediante CSS
- Desarrollar Python con la ayuda de Django
- Ampliar funcionalidades con JavaScript

Planificación temporal

La planificación temporal ha sido la siguiente:

- Estudio del proyecto
- Establecer el alcance del proyecto
- Organización del equipo
- Planificar fechas limite
- Reuniones semanales para ver el avance del proyecto y gestionar las siguientes semanas

Capitulo 3: Especificación de requisitos de sistema

Alcance del proyecto

Este proyecto se comprenderá de las funcionalidades iniciales establecidas en el catálogo de requisitos, y alguna funcionalidad extra para facilitar la gestión de la empresa. Se van a poder añadir, modificar o borrar los modelos de datos de la aplicación que son pedidos, productos, clientes y componentes .

Catálogo de requisitos:

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

Siendo esta la información principal que se maneja, las funcionalidades más importantes requeridas son:

- Gestión de productos: creación, visualización (listado y detalle) y baja.
- Gestión de pedidos: creación, visualización (listado y detalle) y actualización.
- Gestión de clientes: Alta, visualización (listado y detalle) y actualización
- Gestión de componentes que componen cada producto: creación, visualización (listado y detalle) y actualización.

Descripción de requisitos del nuevo sistema

Tras añadir al proyecto los conocimientos de Javascript, Fetch API y AJAX las nuevas funciones requieren:

- Cargar datos y modificar DOM mediante JavaScript: llamada a API utilizando Fetch para obtener datos y mostrar los valores modificando el DOM.

- Envío de datos de un formulario mediante AJAX para su almacenamiento en BBDD.

- Ampliación de funcionalidades en Python

- Implementar una funcionalidad extra mediante JavaScript

Modelo funcional

Las funcionalidades principales de la aplicación son las siguientes:

- Gestión de productos: creación, visualización (listado y detalle) y baja.
- Gestión de pedidos: creación, visualización (listado y detalle) y actualización.
- Gestión de clientes: Alta, visualización (listado y detalle) y actualización
- Gestión de componentes que componen cada producto: creación, visualización (listado y detalle) y actualización.

La gestión de la aplicación es sencilla e intuitiva para cualquier usuario.

Capítulo 4: Especificación del diseño

Introducción

Principales funciones del software

Como bien indican en los requerimientos impuestos por Deustronics Components S.L. las funcionalidades necerarias para la realización de esta aplicación son las siguientes:

- Gestión de productos:
    - Creación de nuevos productos e introducirlas en la BBDD
    - Visualización de productos mediante un listado y opción de ver detalles
    - Eliminar productos de la BBDD
    - Modificar los productos de la BBDD
- Gestión de pedidos:
    - Creación de nuevos pedidos e introducirlas en la BBDD
    - Visualización de pedidos mediante un listado y opción de ver detalles
    - Eliminar pedidos de la BBDD
    - Modificar los pedidos de la BBDD
- Gestión de clientes:
    - Dar de alta a nuevos clientes e introducirlas en la BBDD
    - Visualización de clientes mediante un listado y opción de ver detalles
    - Eliminar clientes de la BBDD
    - Modificar los clientes de la BBDD
- Gestión de componentes que componen cada producto:
    - Creación de nuevos componentes e introducirlas en la BBDD
    - Visualización de componentes mediante un listado y opción de ver detalles
    - Eliminar componentes de la BBDD
    - Modificar los componentes de la BBDD

Por otro lado, se ha añadido funcionalidades extra requeridas.

- Dinamización de imagenes con selector para mejorar la apariencia de las páginas principales de Productos y Componentes

- Modificación del formulario anterior que añade productos para realizarlo mediante AJAX.

- Añadimos un template nuevo llamado “facturas.html” que utiliza el Fetch API para recoger la fecha y el precio del pedido introducido por el usuario y modifica el DOM.

- Añadimos un file uploader que permite la subida de archivos .jpg y su visualización

- Mejoramos la visualización de las templates detalle del pedido y producto para que sea posible mostrar todos los elementos.

Descripción del entorno de desarrollo

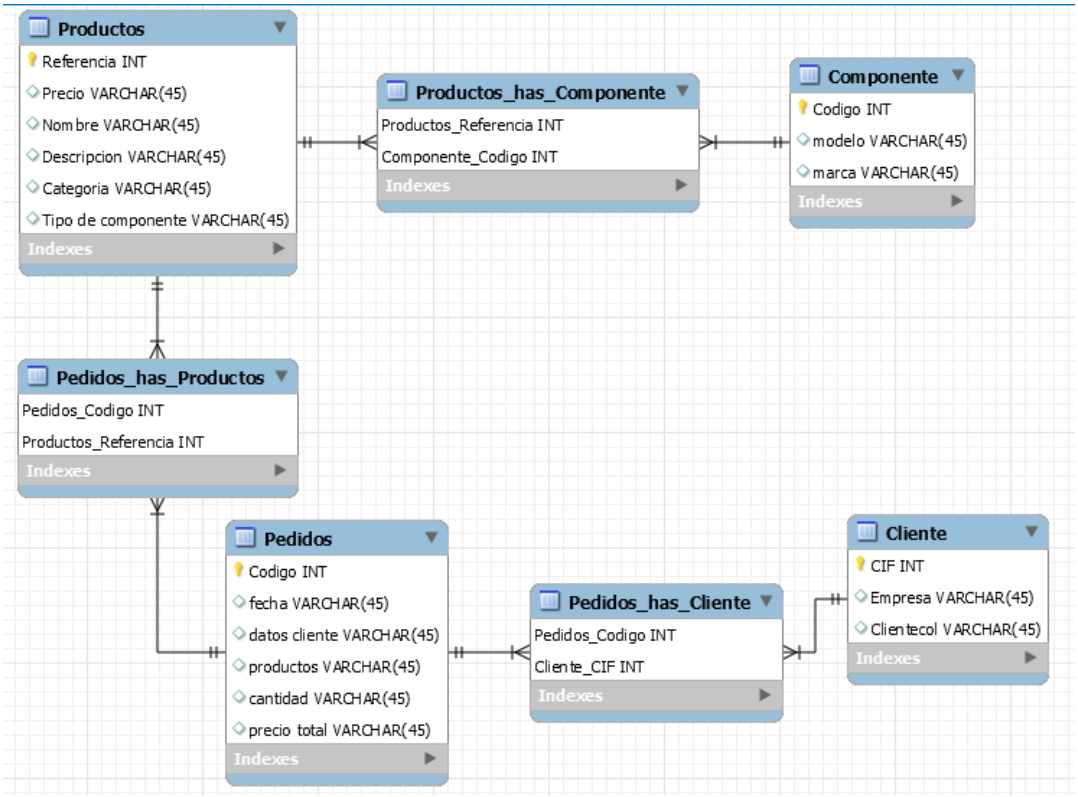
Para desarrollar el proyecto hemos utilizado pyCharm. El proyecto incluye dos directorios distintos para las plantillas html, templates, y otro para incluir archivos estáticos llamado “static” en el que se encuentran los archivos.js. Todo esto además de las propias carpetas y archivos generados por Django.

En todas las ventanas se comparte la función del Traductor de Google, el menú superior negro y la estructura de dos bloques de html denominados “intro” y “contenido”.

Como limitación tenemos que para mantener el formato el bloque “intro” no se expande a diferencia del otro bloque. Es por ello que decidimos situar las listas de objetos en la parte inferior de las páginas y reservar el apartado superior para otras funciones.

Descripción del diseño

Diseño de la estructura física de los datos



Definición de vistas

La aplicación generalmente se compone de cuatro tipos de vistas: Lista, CreateView, DetailView, DeleteView y UpdateView.

Lista

Esta vista se utiliza principalmente para generar una lista de los objetos creados en la aplicación, tanto de componentes, como productos, pedidos y clientes. En este caso mostramos el de clientes:

```
# Listado de clientes
def cliente(request):
    clientes = Cliente.objects.order_by('CIF')  //Guarda en clientes los objetos ordenados por el cif
    context = {                                 //Crea un Diccionario con la información necesaria
                                                  para la vista
                'titulo_form': 'Listado de clientes',
                'titulo_pagina':'Clientes',
                'lista_clientes': clientes}
    return render(request, 'clientes.html', context) // Devuelve un render con el html y el context
```

CreateView

Esta vista se utiliza principalmente para generar un formulario para crear objetos en la aplicación, tanto de componentes, como productos, pedidos y clientes. Esta es la CreateView de pedidos:

```
#Añadir
class PedidoCreateView(CreateView):
  model = Pedido                    //Modelo que va a utilizar
  form_class = PedidoForm           //Forma que va a utilizar
  template_name = 'añadir.html'//Template en la que va a generar el formulario

  def get_success_url(self):
    return reverse('pedidos')    // Template a la que va a volver
```

DetailView

Esta vista se utiliza principalmente para mostrar los datos de los objetos de la aplicación, en este caso le tendremos que pasar el id del objeto en cuestión. Esta vista está tanto de productos como pedidos. Mostramos como ejemplo la DetailView de componentes:

```
#Detalles
class ComponenteDetailView(DetailView):
  model = Componente                                    //Modelo que va a utilizar
  template_name = 'componente_detalle.html'          //Template en la que va a mostrar los detalles

  def get_context_data(self, **kwargs):
    context = super(ComponenteDetailView, self).get_context_data(**kwargs)
    context['titulo_pagina'] = 'Detalles del componente'
    return context
```

DeleteView

Esta vista se utiliza principalmente para eliminar un objeto de la aplicación, en este caso le tendremos que pasar el id del objeto en cuestión. Esta vista esta tanto de componentes, como productos, pedidos y clientes. En este caso veremos el de DeleteView de productos:

```
#Eliminar
class ProductoDelete(DeleteView):
  model = Productos                                     //Modelo que va a utilizar
  template_name = 'producto_eliminar.html'           //Template en la que va a mostrar los detalles
  success_url = reverse_lazy('indexprod')             //Template an la que va regresar

  def get_context_data(self, **kwargs):
    context = super(ProductoDelete, self).get_context_data(**kwargs)
    return context
```

UpdateView

Esta vista se utiliza principalmente para modificar un objeto de la aplicación, en este caso le tendremos que pasar el id del objeto en cuestión. Esta vista esta tanto de componentes, como productos, pedidos y clientes. Por último mostramos la UpdateView de productos:

```
#Actualizar
class ProductoUpdate(UpdateView):
   model = Productos                          //Modelo que se va a utilizar
   fields = ['referencia','precio','nombre','descripcion','categoria', 'tipo_componentes']
   template_name = 'añadir.html'              //Template en la que se va a mostrar
   success_url = reverse_lazy('indexprod')    //Template a la que va a regresar
```

Capítulo 5: Plan de pruebas globales

Tras algunos problemas iniciales al modificar el código sin comprobar cada cambio hemos cogido costumbre de actualizar la página a cada nueva comprobación. El console.log ha sido una herramienta de gran ayuda para entender el funcionamiento de javascript y el orden de ejecución sobre el html. Gracias a ello hemos conseguido corregir fallos de mala sintaxys que no sabíamos por qué daban error. Hemos utilizado una template llamada Usos.html como página de pruebas y errores así como un file.js que también servía de pruebas para evitar manchar y modificar el código bueno y no arriesgarnos a generar fallos nuevos.

Capítulo 6: Manual de usuario

Para poder comenzar a gestionar la aplicación lo primero que debemos hacer es arrancar el servidor. Para ello utilizaremos el comando “py manage.py runserver”. Una vez hecho esto entraremos al navegador e iremos a la siguiente url: http://127.0.0.1:8000/miApp

Este enlace abrirá la primera página de nuestra aplicación, el inicio, en el cual mostramos un menú muy intuitivo con las diferentes opciones. Por un lado, nos encontramos con el bloque “header” común para todas las templates, que contiene un menú que nos da opción de navegar por las diferentes pestañas de nuestra aplicación (Inicio, clientes, pedidos, productos y componentes)

Inicio

La página de inicio es la principal de la aplicación, desde la cuál se pueden observar las opciones de la aplicación. Dispone de un calendario, un gráfico de las empresas con las que se trabaja y un menú dinámico en la parte inferior algo más visual y que ahora incluye la opción de “FACTURAS”.

Clientes

Esta ventana es la que muestra información de nuestros clientes. En la parte superior se muestran comentarios de interés para los empleados de Deustronic en relación con los clientes de la empresa. En la parte inferior, se muestra un listado de todos los clientes de la BBDD con diferentes opciones de añadir, editar o borrar.

Pedidos

En esta ventana es donde se muestra información de los pedidos que están en curso. En la parte superior, se muestra información sobre los pedidos, los que están en proceso y los impagados. En la parte inferior, se muestra un listado de todos los pedidos de la BBDD con diferentes opciones de añadir, editar, ver detalles o borrar.

Productos

En esta ventana es donde se muestra información de los productos existentes en stock. En la parte superior, se muestran diferentes imágenes de los productos estrella. En la parte inferior, se

muestra un listado de todos los productos de la BBDD con diferentes opciones de añadir, editar, ver detalles o borrar.

Componentes

En esta ventana es donde se muestra información de los componentes de los productos existentes en stock. En la parte superior, se muestran imágenes de los diferentes tipos de componentes. En la parte inferior, se muestra un listado de todos los componentes de la BBDD con diferentes opciones de añadir, editar o borrar.

.

Añadir

En la opción de añadir en cliente, pedidos, productos o componentes, nos abrirá una ventana con un formulario para poder añadir el objeto deseado. Una vez rellenados los campos (dependiendo de qué objeto sea tendrá unos campos u otros), al registrarlo y será automáticamente añadido en la BBDD y la lista correspondiente. El registro de nuevos productos se realizará mediante AJAX, también al pulsar el botón Añadir.

Editar

Para poder acceder a esta ventana, habrá que pulsar sobre la imagen de un lápiz en el listado. Una vez hecho, se nos abrirá una ventana donde nos mostrará los detalles del objeto seleccionado a los cuales podremos cambiar los campos. Al darle a registrar, nos modificara el objeto con la nueva información y la actualizara en la BBDD.

Ver detalles

Sólo los productos y los pedidos disponen de esta funcionalidad puesto que es la manera más clara de mostrar toda la información. Para ello tendremos que pulsar sobre el “+” del listado. Una vez pulsado nos abrirá una ventana donde mediante una tabla se nos mostrará todos los detalles del objeto seleccionado, con una visualización mejorada respecto a la anterior versión de la aplicación.

Borrar

Para poder acceder a esta ventana, habrá que pulsar sobre la imagen de una basura en el listado. Una vez hecho, se nos abrirá una ventana de confirmación donde nos mostrará los detalles del objeto seleccionado. Dándole a confirmar el objeto se borrará de la BBDD.

Capítulo 7: Incidencias del proyecto y conclusiones

La mayor incidencia o dificultad del proyecto ha sido la falta de tiempo para perfeccionar muchos detalles que hubiéramos deseado mejorar. Los conceptos al principio eran algo confusos y hemos tenido que buscar información extra para lograr algunos de los objetivos del proyecto. Ha sido un trabajo bonito de realizar cuando ya se tenían claras las bases y el método de ejecución, pero el inicio fue complicado. En cuanto a la practicidad de la aplicación, como he comentado, nos hubiera gustado mejorar varios aspectos para hacerla más realista y funcional aun, pero por lo general cumple sus funciones y se podría dar uso de ella sin problemas.

Como ya he comentado en uno de los apartados anteriores una de las limitaciones de nuestra aplicación era el encajar los elementos correctamente en el bloque de inicio que no podía modificar u tamaño. Como funciones que han quedado pendientes de mejora resaltaría el botón de “ordenar”, en la ventana de productos que funciona mediante vistas y no utiliza el fetch API, de manera que hay que recargar la página al completo para mostrar los cambios en vez de modificarse sólo esa parte del DOM. Por otro lado hubieramos deseado mejorar el input file para que sólo permitiera agregar archivos de un tipo y visualizarlos, no únicamente en formato .jpg. La página de facturas tiene varias cosas que podrían mejorarse pero ha sido un punto clave para entender y trabajar mejor los conocimientos sobre el Fetch API. En la página principal también querríamos sustituir la imagen de gráfico por un widget de estadísticas o algo similar. Por último lado del gráfico se encuentra un calendario al que se le podrían haber añadido funciones de javascript

.