

**Ingeniería Web**

**Documentación proyecto web colaborativo**

**Grupo 6:**

**Jon Ander, Gaizka y Andrea**

**Vitoria-Gasteiz - 28/05/2024**

ÍNDICE

[INTRODUCCIÓN 1](#_Toc167792154)

[OBJETIVOS DEL PROYECTO 1](#_Toc167792155)

[Resumen para la dirección 1](#_Toc167792156)

[Tareas principales 1](#_Toc167792157)

[Planificación temporal 1](#_Toc167792158)

[ESPECIFICACIONES DE REQUISITOS DEL SISTEMA 2](#_Toc167792159)

[Descripción general 2](#_Toc167792160)

[Catálogo de requisitos 2](#_Toc167792161)

[Modelo lógico de datos 3](#_Toc167792162)

[Representación entidad relación 3](#_Toc167792163)

[Transformación a relacional 3](#_Toc167792164)

[Descripción de la interfaz del sistema 5](#_Toc167792165)

[Representación jerárquica de la interfaz 5](#_Toc167792166)

[ESPECIFICACIÓN DEL DISEÑO 5](#_Toc167792167)

[Principales funciones del software 5](#_Toc167792168)

[Descripción del entorno de desarrollo 6](#_Toc167792169)

[Representación gráfica 6](#_Toc167792170)

[MANUAL DE USUARIO 8](#_Toc167792171)

[INCIDENCIAS DEL PROYECTO Y CONCLUSIONES 12](#_Toc167792172)

[Bibliografía 13](#_Toc167792173)

ÍNDICE DE ILUSTRACIONES

[Ilustración 1 - Diagrama E/R 3](#_Toc167792139)

[Ilustración 2 – Flujo de navegación 5](#_Toc167792140)

[Ilustración 3 - Interacción entre clientes/servidor 6](#_Toc167792141)

[Ilustración 4 - Pantalla de inicio de sesión 8](#_Toc167792142)

[Ilustración 5 - Pantalla de registro 8](#_Toc167792143)

[Ilustración 6 - Página principal 9](#_Toc167792144)

[Ilustración 7 - Creación de producto 10](#_Toc167792145)

[Ilustración 8 - Pantalla de soporte 11](#_Toc167792146)

[Ilustración 9 - Validación de campos 11](#_Toc167792147)

ÍNDICE de tablas

[Tabla 1 - Producto 3](#_Toc167792148)

[Tabla 2 – Componente 3](#_Toc167792149)

[Tabla 3 – ProductoComponente 3](#_Toc167792150)

[Tabla 4 - Cliente 4](#_Toc167792151)

[Tabla 5 – Pedido 4](#_Toc167792152)

[Tabla 6 – ProductoPedido 4](#_Toc167792153)

# INTRODUCCIÓN

El proyecto desarrollado es una aplicación web diseñada para la gestión eficiente de productos, clientes, pedidos y componentes. Ha sido desarrollado utilizando Django, un framework de desarrollo web de alto nivel basado en Python.

# OBJETIVOS DEL PROYECTO

## Resumen para la dirección

La aplicación desarrollada para Deustronic S. L. es una plataforma web diseñada para optimizar la gestión de inventarios, clientes, pedidos y componentes dentro de una empresa. Utilizando el framework Django, se busca crear una herramienta intuitiva y eficiente que permita a los usuarios administrar estos elementos de manera centralizada y efectiva. La aplicación proporcionará funcionalidades completas para la creación, modificación, visualización y eliminación de productos, clientes y pedidos, además de facilitar la asignación de componentes a productos y productos a pedidos. Con esta solución, se espera mejorar significativamente la eficiencia operativa, reducir errores y aumentar la satisfacción del cliente.

## Tareas principales

A la hora de realizar el proyecto, las tareas principales han sido las siguientes:

1. Programación en Python
2. Renderización en HTML
3. Diseño y aplicación de estilos en css
4. Diseño de la BBDD

## Planificación temporal

(qué tareas, cómo nos hemos organizado, reparto)

* Andrea
* Gaizka
* Jon Ander

# ESPECIFICACIONES DE REQUISITOS DEL SISTEMA

## Descripción general

### Catálogo de requisitos

El sistema debe cumplir con las siguientes funciones:

1. Gestión de productos:
   1. Crear nuevos productos, ingresando detalles como nombre, descripción, precio y stock.
   2. Modificar los detalles de productos existentes.
   3. Eliminar productos del inventario.
   4. Visualizar una lista de todos los productos, con opciones de búsqueda y filtrado.
   5. Detalle de producto con información completa.
2. Gestión de clientes:
   1. Registrar nuevos clientes con información como nombre, dirección, correo electrónico y teléfono.
   2. Modificar la información de clientes existentes.
   3. Eliminar clientes del sistema.
   4. Visualizar una lista de todos los clientes, con opciones de búsqueda y filtrado.
   5. Detalle de cliente con información completa.
3. Gestión de pedidos:
   1. Modificar la información de clientes existentes.
   2. Eliminar clientes del sistema.
   3. Visualizar una lista de todos los clientes, con opciones de búsqueda y filtrado.
   4. Detalle de cliente con información completa.
4. Gestión de componentes:
   1. Crear nuevos componentes para productos.
   2. Asignar componentes a productos específicos.
   3. Modificar y eliminar componentes.
   4. Visualizar una lista de todos los componentes, con opciones de búsqueda y filtrado.
5. Interfaz de usuario:
   1. Proveer una interfaz intuitiva y fácil de usar para todas las funcionalidades mencionadas.
   2. Utilizar un diseño responsivo que funcione bien en dispositivos móviles y de escritorio.
   3. Incluir formularios claros y validaciones para la entrada de datos.
6. Seguridad y Soporte:
   1. Autenticación de usuarios para proteger el acceso a la aplicación.
   2. Gestión de mensajes de soporte recibidos a través de e-mail.

### Modelo lógico de datos

Diagrama

Descripción generada automáticamente con confianza mediaRepresentación entidad relación

Ilustración 1 - Diagrama E/R

Precio total es un atributo derivado ya que se obtiene de la multiplicación de cantidad de producto por el precio.

Transformación a relacional

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descripción |
| id (PK) | Integer | Identificador único del producto |
| nombre | String | Nombre del producto |
| categoria | String | Categoría del producto |
| precio | Decimal | Precio del producto |
| cantidad | Integer | Cantidad disponible en inventario |
| descripcion | Text | Descripción del producto |

Tabla 1 - Producto

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descripción |
| id (PK) | Integer | Identificador único del componente |
| nombre | String | Nombre del componente |
| marca | String | Marca del componente |

Tabla 2 – Componente

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descripción |
| id (PK) | Integer | Identificador único de la relación |
| producto\_id (FK) | Integer | Identificador del producto |
| componente\_id (FK) | Integer | Identificador del componente |
| cantidad | Integer | Cantidad del componente en el producto |

Tabla 3 – ProductoComponente

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descripción |
| id (PK) | Integer | Identificador único del cliente |
| cif | String | Identificador fiscal del cliente |
| nombre\_empresa | String | Nombre de la empresa del cliente |
| direccion | String | Dirección física del cliente |
| telefono | String | Número de teléfono del cliente |
| email | String | Correo electrónico del cliente |

Tabla 4 - Cliente

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descripción |
| id (PK) | Integer | Identificador único del pedido |
| cliente\_id (FK) | Integer | Identificador del cliente |
| fecha | Date | Fecha del pedido |
| precio\_total | Decimal | Precio total del pedido |

Tabla 5 – Pedido

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descripción |
| id (PK) | Integer | Identificador único de la relación |
| pedido\_id (FK) | Integer | Identificador del pedido |
| producto\_id (FK) | Integer | Identificador del producto |
| cantidad | Integer | Cantidad del producto en el pedido |

Tabla 6 – ProductoPedido

Siguiendo el esquema entidad relación y tratando de evitar el mayor número de redundancias, se han obtenido las anteriores tablas, especificadas y generadas a través del models.py de la aplicación.

## Descripción de la interfaz del sistema

Representación jerárquica de la interfaz

Diagrama

Descripción generada automáticamente

Ilustración 2 – Flujo de navegación

Gracias a este diagrama de flujo se puede observar el funcionamiento y navegación en transiciones en la aplicación. El estado de cerrar sesión es accesible desde todas las vistas exceptuando la vista de registro e inicio de sesión.

# ESPECIFICACIÓN DEL DISEÑO

## Principales funciones del software

1. Gestión de productos.
2. Gestión de pedidos.
3. Gestión de componentes
4. Gestión de clientes.
5. Interfaz de usuario.
6. Seguridad y soporte.

## Descripción del entorno de desarrollo

En cuanto al lenguaje de programación utilizado, ha sido Python, que es la base del framework Django, utilizado para el desarrollo web. Django proporciona una estructura robusta para el desarrollo de aplicaciones web complejas, manejando tareas comunes como la autenticación de usuarios, la gestión de bases de datos y la creación de interfaces de administración.

El IDE utilizado para el desarrollo ha sido PyCharm, una herramienta diseñada específicamente para el desarrollo en Python.

En cuanto al control de versiones, se ha utilizado la herramienta de git, alojando el repositorio en GitHub.

La base de datos es gestionada a través del sistema SQLite que Django maneja a través de su sistema ORM, lo que permite una primera definición de los modelos en Python para una posterior migración a un esquema de base de datos. Además, es posible realizar consultas a la base de datos a través de Python, sin la necesidad de escribir consultas SQL.

## Representación gráfica

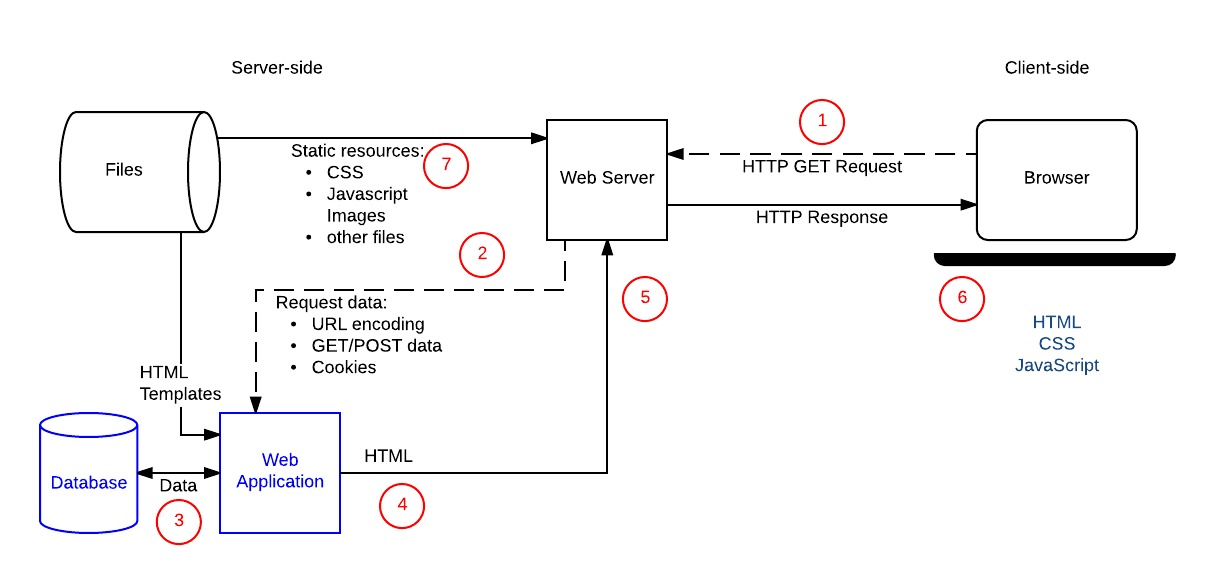


Ilustración 3 - Interacción entre clientes/servidor

La interacción entre el cliente y el servidor en una aplicación web sigue una serie de pasos, el proceso comienza cuando el cliente, utilizando un navegador web, realiza una petición HTTP GET al servidor web solicitando una página específica o un recurso.

El servidor web recibe la petición HTTP GET y procesa los datos de la solicitud, como la URL y los parámetros GET/POST. A continuación, el servidor web determina qué recursos se necesitan para responder a la solicitud. Si la aplicación web necesita datos almacenados en una base de datos, envía una consulta a la base de datos y espera la respuesta con los datos necesarios.

Una vez que la aplicación web tiene los datos necesarios, genera una respuesta HTML basada en estos datos y las plantillas HTML predefinidas. Esta respuesta HTML es la página web que será enviada al navegador del cliente. El servidor web entonces envía la respuesta HTTP, que incluye el contenido HTML, de vuelta al navegador del cliente.

El navegador web recibe la respuesta HTTP y procesa el contenido HTML, CSS y JavaScript, renderizando la página web para que el usuario pueda interactuar con ella. Además de las respuestas dinámicas generadas por la aplicación web, el servidor web también envía recursos estáticos como archivos CSS, JavaScript, imágenes u otros archivos directamente al navegador sin necesidad de procesamiento adicional.

Este flujo asegura que el cliente reciba tanto las páginas web dinámicas como los recursos estáticos necesarios para interactuar de manera efectiva con la aplicación web, proporcionando una experiencia de usuario fluida y eficiente.

# MANUAL DE USUARIO

Para poder acceder como usuario con rol de administrador en la aplicación es imprescindible iniciar sesión con el usuario ‘deusto’ con contraseña ‘deusto’.Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 4 - Pantalla de inicio de sesión

En la pantalla de registro es posible crear nuevos usuarios, aunque por motivos de seguridad, sin permisos suficientes para poder iniciar sesión.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Ilustración 5 - Pantalla de registro

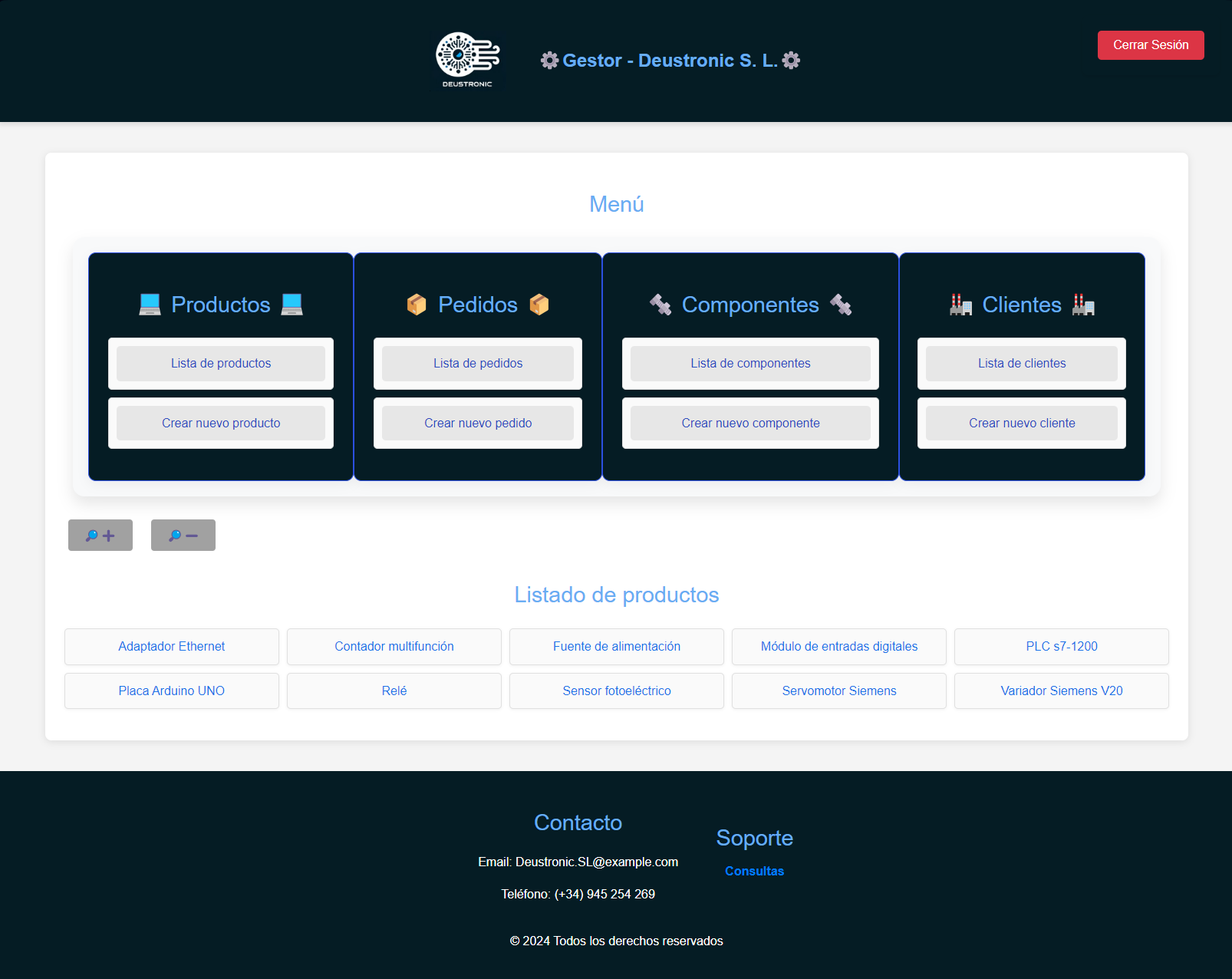


Ilustración 6 - Página principal

En la aplicación web se han implementado varios listados que siguen un formato y diseño similar, utilizando “cajas” para mostrar los diferentes elementos existentes en la base de datos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración 7 - Creación de producto

En cuanto a los formularios de la aplicación, todos mantienen un mismo estilo, requiriendo la escritura de todos los valores para su correcto envío.

En cuanto al diseño de los formularios:

1. **Encabezado:** Cada formulario de creación tiene un encabezado que indica claramente el propósito del formulario (por ejemplo, "Nuevo producto", "Nuevo pedido").
2. **Campos de Entrada:** Los formularios incluyen campos de entrada claramente etiquetados para capturar la información necesaria. Por ejemplo, en el formulario de creación de producto, los campos pueden incluir nombre del producto, precio, descripción y categoría.
3. **Botón de Envío:** Al final del formulario, se incluye un botón de envío que permite al usuario guardar el nuevo registro. El botón está claramente etiquetado (por ejemplo, "Nuevo producto").

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración 8 - Pantalla de soporte

En el apartado de soporte, se ha implementado una validación de campos para asegurar que los mensajes enviados no contengan palabras prohibidas. Esto ayuda a mantener un lenguaje adecuado y profesional en las comunicaciones. El formulario consta de un campo de texto y un botón de envío.

A continuación, se detalla cómo funciona esta validación utilizando el formulario de mensaje de soporte como ejemplo. La pantalla de soporte como se puede ver en la ilustración 6, la pantalla incluye un formulario simple donde el usuario puede escribir su mensaje.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 9 - Validación de campos

En la ilustración 9, se muestra un ejemplo en el que el usuario intenta enviar un mensaje que contiene palabras prohibidas ("palabra1" y "palabra2"). Cuando el usuario escribe estas palabras inapropiadas y trata de enviar el mensaje, el sistema no permite el envío y muestra un mensaje de advertencia en la parte inferior del formulario.

Esta funcionalidad se muestra visualmente al usuario en tiempo real, y además, se realiza una verificación adicional en el servidor para mantener la integridad del sistema. Este enfoque dual de validación ayuda a mantener una comunicación clara y profesional con el equipo de soporte.

# INCIDENCIAS DEL PROYECTO Y CONCLUSIONES

Tras finalizar el proyecto, se puede concluir que las mayores incidencias para desarrollar el proyecto ha sido la coordinación y gestión del repositorio en GitHub, que a pesar de haber generado ramas individuales para cada participante del grupo, ha sido complicado llevar a cabo correctamente la fusión de implementaciones.

Además, algunos compañeros han sufrido incidencias a la hora de clonar, hacer pull o commit & push, lo que ha dificultado y entorpecido el proceso, teniendo que realizar los commits otra persona en su lugar.

A pesar de estas incidencias, el desarrollo de la aplicación ha sido completo, implementando todo tipo de funcionalidades en Python y javascript. Es destacable que como en la última fase del proyecto se requiere investigar la implementación de las funcionalidades de forma autónoma, ha sido posible aprender a buscar y analizar código externo.

# Bibliografía

Login:

<https://youtu.be/6rMSouSx3II?si=XbTXD1qkjXikD4Eh>

Vista de detalle de pedido:

<https://stackoverflow.com/questions/53023775/simple-math-on-django-views-with-decimals>

<https://docs.djangoproject.com/en/5.0/ref/models/expressions/>

Validación de texto en formulario:

<https://www.freecodecamp.org/news/form-validation-in-javascript/>

<https://stackoverflow.com/questions/72188432/form-validation-must-contain-a-specific-word-from-a-list>

Fetch API:

<https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_forms_through_JavaScript>

<https://www.freecodecamp.org/news/javascript-fetch-api-for-beginners/>