

Trabajo Fin de Grado

Ies federica montseny

Miguel Ángel Girona y David Pato | Desarrollo de Aplicaciones Multiplataforma

# INDICE

# 

1. Justificación
2. Objetivos
3. Metodología Desarrollo
4. Tecnologías
5. Planificación del proyecto
6. Desarrollo y ejecución del proyecto
7. Conclusiones
8. Líneas de investigación y desarrollo futuras
9. Bibliografía

# JUSTIFICACIÓN

Somos David Pato y Miguel Angel Girona, estudiantes de segundo de grado superior, Desarrollo de Aplicaciones Multiplataforma, también conocido como DAM. Dicho grado se estudia presencial en IES Federica Montseny, situado en Fuenlabrada.

Durante el transcurso de las prácticas, nos hemos lanzado a desarrollar, con los conocimientos adquiridos, una web de una tienda. Ambos decidimos hacer una página web parecida a PcComponentes, porque somos amantes de la tecnología y del hardware, por lo tanto nos propusimos crear una pagina lo mas parecida a la real.

# OBJETIVOS

El objetivo era crear una pagina en la que se pudiesen ver, comparar y comprar productos, con detalles.

Ya que en este sector hay muchas tiendas pero siempre tardan mucho, intentaríamos destacar por la sencillez a la hora de realizar pedidos y porque siempre tardaríamos como máximo 24 horas.

# METODOLOGÍA DESARROLLO

1. Análisis de requisitos

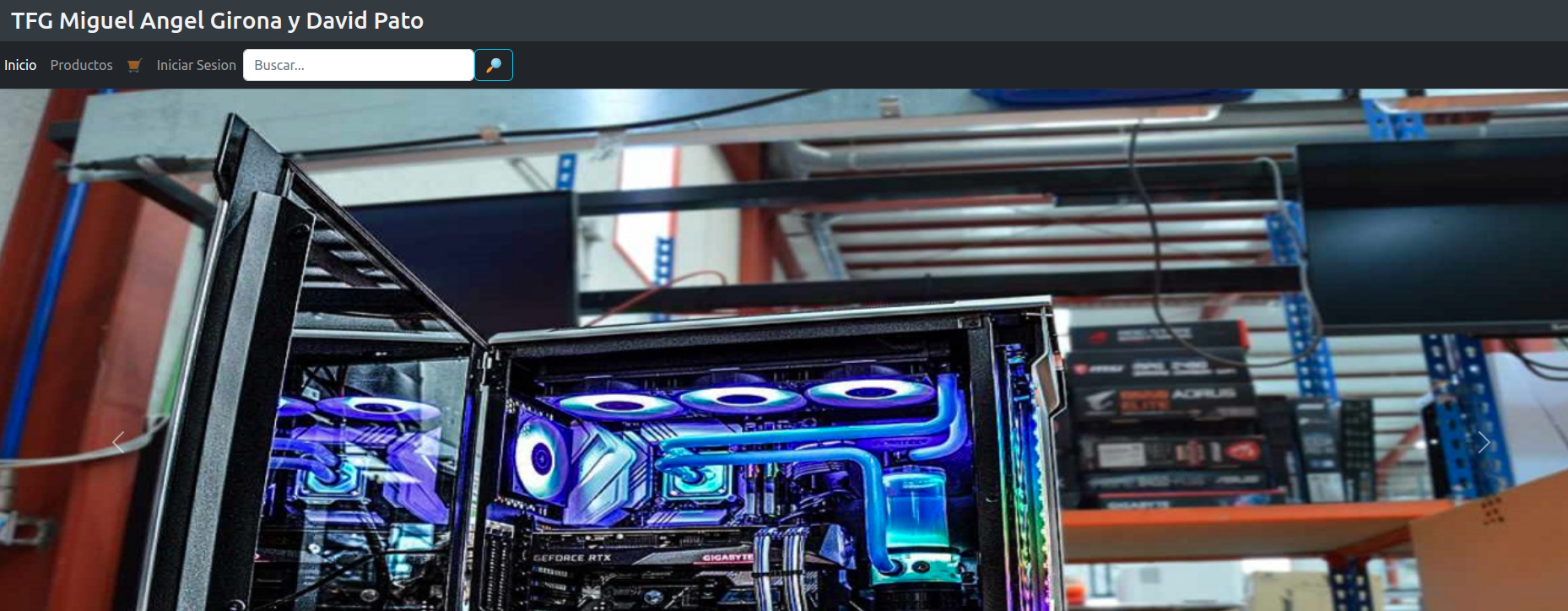
La web permitirá ver productos disponibles para su compra.

2. Diseño del sistema

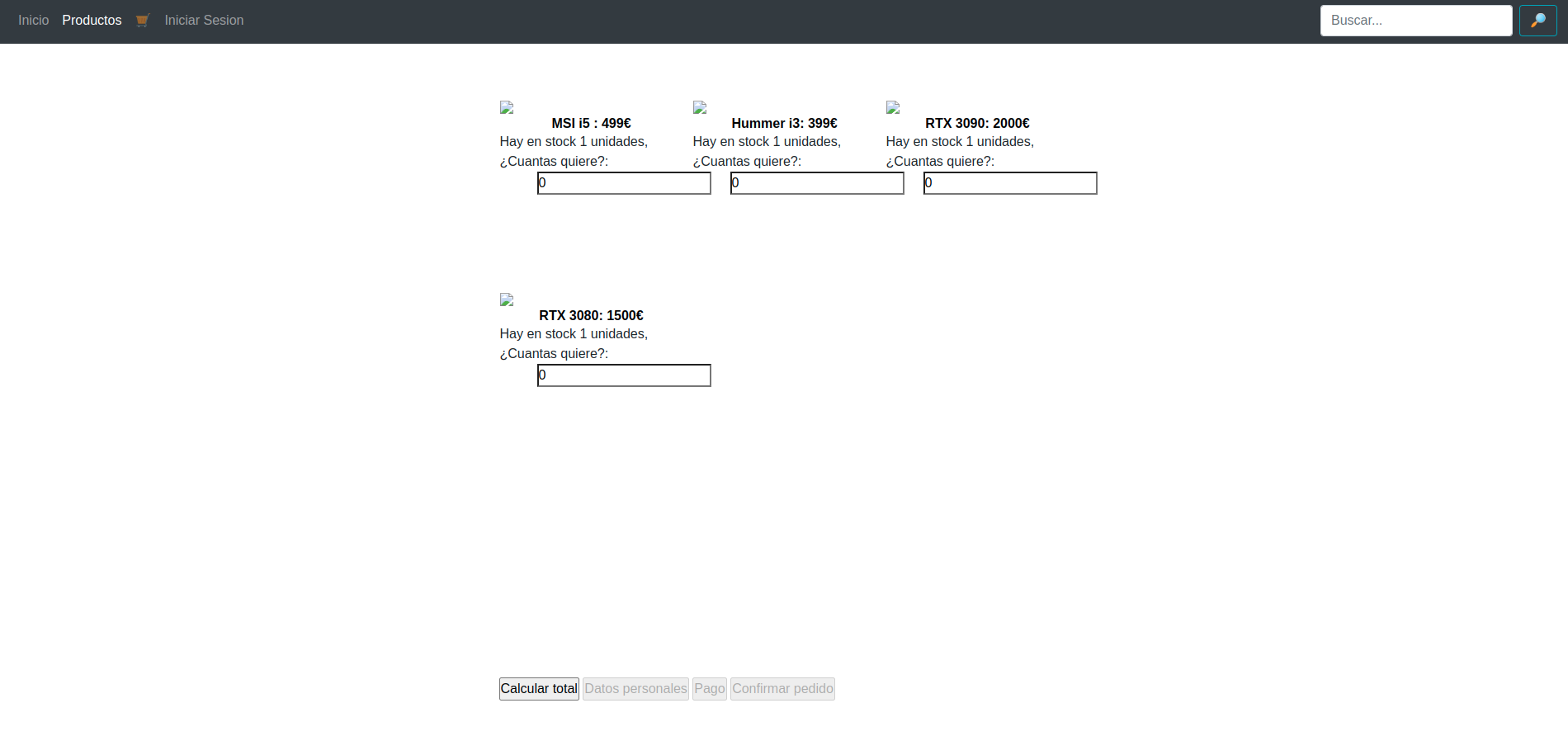
Se ha procedido a realizar una aplicación web, la cual recibirá los datos a través de diferentes peticiones a una API REST.

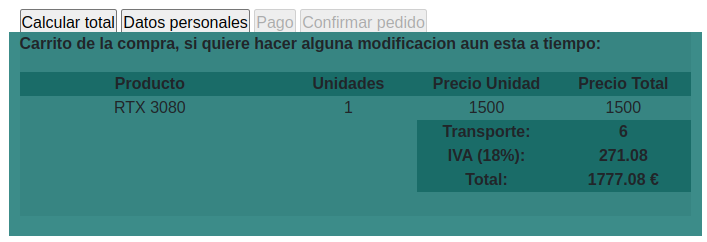
3. Diseño del programa

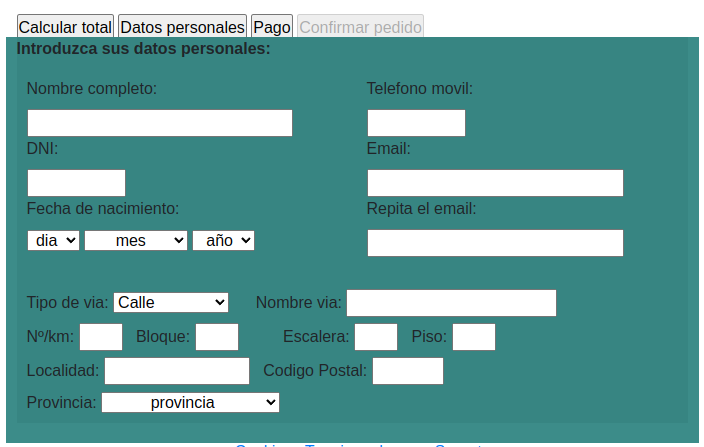
En la pagina de inicio tenemos un slider de 4 fotos en la cual nos habria gustado añadir una descripcion de la propia web con una api de google de maps para que muestre una localizacion

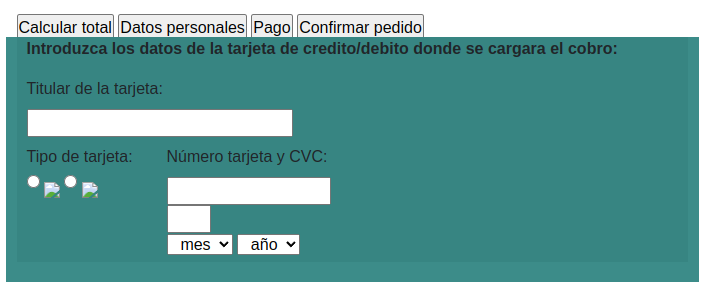
Tiene una topbar para poder navegar entre todas las secciones

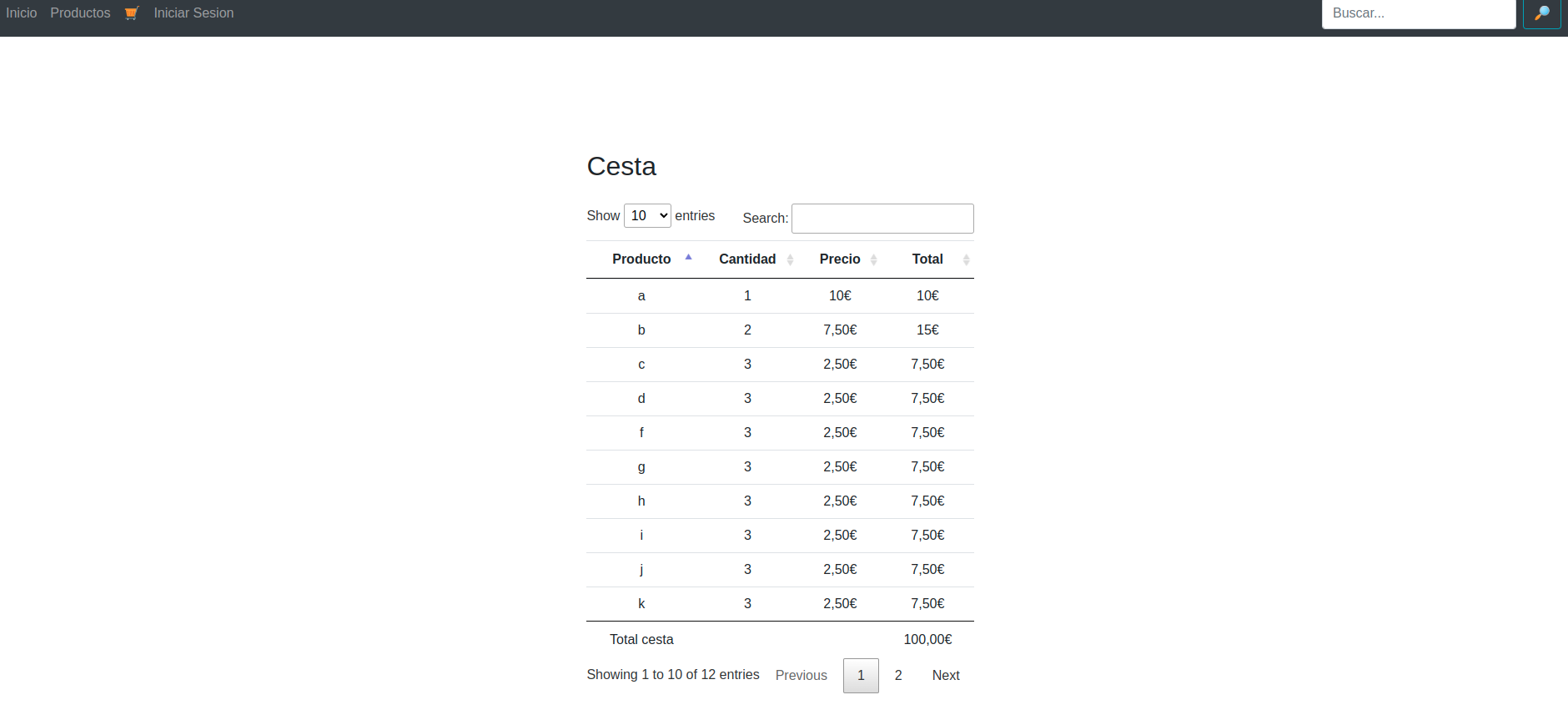
En la siguiente zona estan todos los productos que tenemos en la base de datos con los respectivos botones de compra

Abajo en los botones, se realiza una simulacion de compra al haber seleccionado alguno de los productos



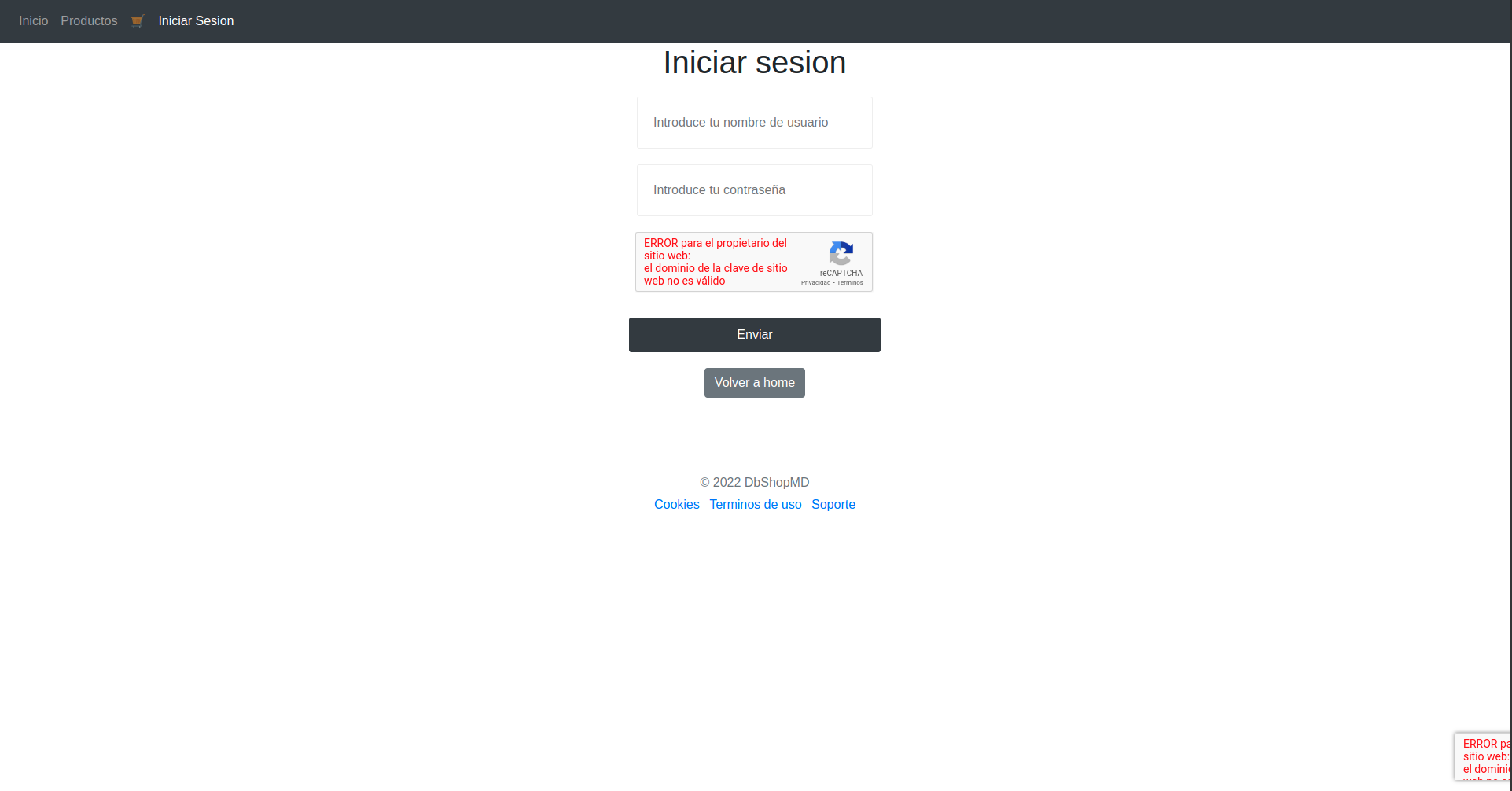




En la cesta tenemos una tabla con una simulacion de cesta de compra con mas de 10 productos la cual tiene un filtro de busqueda, paginacion y cantidad de productos a mostrar

En la siguiente pagina, esta la zona de iniciar sesion

El reCAPTCHA aparece en la imagen mal ya que se usa ejecutando el visual studio, para que funcione correctamente tendra que salir en localhost o en una propia url



4. Codificacion

El código perteneciente a todo el desarrollo de la web, su API Rest, se encuentra en los enlaces que se comentan a continuación:

<https://github.com/TFG-David-y-Giro/TFG.git>

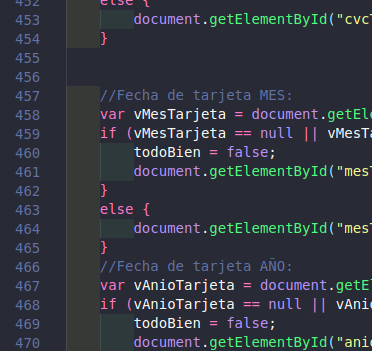
# TECNOLOGÍAS

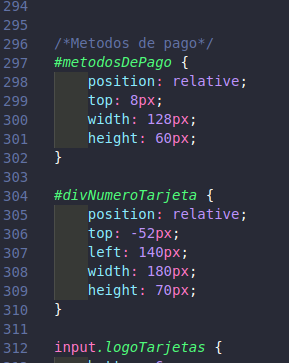
Para la parte de back hemos usado el ide IntelliJ, que es bastante sencillo ya que hemos aprendido a programar ahí.





Para la parte del front hemos usado visual studio code ya que permite editar y ver en tiempo real en el navegador los cambios realizados





# 

# 

# PLANIFICACIÓN DEL PROYECTO

El proyecto fue planificado para permitir lo máximo del desarrollo en el tiempo del que disponíamos, teniendo en cuenta que ambos miembros del equipo poseen un horario laboral y unas prácticas que realizar, situaciones que dificultan la disponibilidad de tiempo.

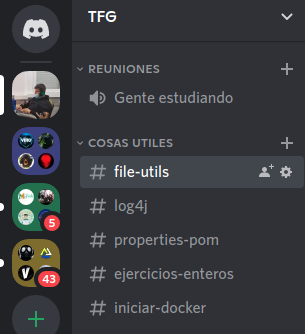
Fecha de inicio Duración Fecha final

Front 16/04/2022 18 13/06/2022

API Rest 16/04/2022 18 13/06/2022

Documentación 11/06/2018 2 13/06/2022

# DESARROLLO Y EJECUCIÓN DEL PROYECTO

 El desarrollo del proyecto, como hemos tenido tan poco tiempo, no lo íbamos a perder desplazándonos. Con lo cual tenemos un servidor destinado al tfg en discord (es gratis). En cual, hemos realizado las reuniones compartiendonos mutuamente pantalla para dar ideas y tomar decisiones a la hora de desarrollar todo.

Ha habido días que han sido a través de simples llamadas telefónicas por no poder coincidir y hemos intentado aportar en los trayectos a los respectivos trabajos

# CONCLUSIONES

# LÍNEAS DE INVESTIGACIÓN Y DESARROLLO FUTURAS

En un futuro nos habría gustado añadir:

* WEB
  + Pagina de administrador
    - Visor de la base de datos
    - Editor de usuarios con ayuda a recuperación de contraseña
    - Editor de proveedores
    - Editor de productos
    - Capacidad de poder crear mas administradores
    - Capacidad de modificar pequeñas partes de código de css para apariencia de la web mediante un editor de codigo implementado por una api.
  + Pagina de inicio de sesión
    - Implementar api para inicio de sesión con otras plataformas (apple id, google id o facebook id)
    - Terminar con un token el inicio de sesión el cual expira al salir o cerrar la ventana o navegador
  + Pagina de home
    - Implementar presentación tipo blog
  + Productos
    - Crear una pagina en especifico para cada producto
  + Pagina de carrito
    - Haber creado la pagina para realizar pagos
* Alojar la web en un contenedor con url para poder acceder desde internet, y acceder a otro contenedor de otra web al API para que funcione correctamente.
* Android
  + Crear aplicación android similar a la web con la condición de iniciar sesión en especifico para poder realizar operaciones

# BIBLIOGRAFÍA

# ANEXOS

<http://localhost:9090/item>

Devuelve un JSON de datos incluyendo un array de productos.

<http://localhost:9090/item/(id>)

PRE: Requiere un id de producto numérico.

Devuelve un JSON de datos incluyendo el producto si existe o un producto vacio si no existe.

<http://localhost:9090/supplier>

Devuelve un JSON de datos incluyendo un array de proveedores.

<http://localhost:9090/supplier/(id>)

PRE: Requiere un id de proveedor numérico.

Devuelve un JSON de datos incluyendo el proveedor si existe o un proveedor vacio si no existe.

<http://localhost:9090/user>

Devuelve un JSON de datos incluyendo un array de usuarios.

<http://localhost:9090/user/(id>)

PRE: Requiere un id de usuario numérico.

Devuelve un JSON de datos incluyendo el usuario si existe o un usuario vacio si no existe.

<http://localhost:9090/cart>

Devuelve un JSON de datos incluyendo un array de cestas.

<http://localhost:9090/cart/id>

PRE: Requiere un id de cesta numérico.

Devuelve un JSON de datos incluyendo la cesta si existe o una cesta vacia si no existe.

<http://localhost:9090/category>

Devuelve un JSON de datos incluyendo un array de categoria.

<http://localhost:9090/category/id>

PRE: Requiere un id de categoria numérico.

Devuelve un JSON de datos incluyendo el usuario si categoria o una categoria vacia si no existe.