11-7-2025

Trabajo Práctico Integrador

Dolce Freddo

INTEGRANTES:

DÍAZ KATHERINE – GOMEZ TOBÍAS – CABALLARO JUAN – CACERES ISAIAS

índice

Consigna

Descripción del Sistema

El sistema permitirá:

- Gestión de usuarios y permisos (Login con roles).
- CRUD de productos (Altas, Bajas, Modificaciones y Visualización).
- Alertas de confirmación y advertencia de stock bajo.
- Interfaz responsiva para múltiples dispositivos.
- Panel de administración para control de inventario y usuarios.

Objetivos

- Diseñar y construir un sistema modular, escalable y funcional.
- Desarrollar habilidades de trabajo colaborativo entre áreas.
- Comprender e implementar el patrón arquitectónico MVC.
- Simular un entorno real de desarrollo de software.

Requisitos Funcionales

Alertas

- Confirmación visual al eliminar o editar un producto.
- Notificación de “stock bajo” si las unidades < 5.

CRUD de Productos

- Altas: Carga de nombre, descripción, stock, precio e imagen.
- Bajas: Eliminar productos.

- Modificaciones: Editar datos de productos.
- Visualización: Mostrar catálogo ordenado y buscable.

Login y Permisos

- Registro e inicio de sesión de usuarios.
- Usuarios estándar: solo pueden visualizar productos.
- Administradores: pueden gestionar productos y usuarios.

Requisitos No Funcionales

- Seguridad: Control de acceso según el rol.
- Interfaz de Usuario: Adaptada a escritorio y móvil.
- Escalabilidad: Preparado para expansión de base de datos y nuevas funciones. Software
- Lenguaje backend: Python o JavaScript (Node.js)
- Framework sugerido: Django / Express.js
- Frontend: HTML, CSS, JavaScript (con Bootstrap, React opcional) • Base de datos: MySQL o SQLite
- Navegadores compatibles: Chrome, Firefox, Edge
- Herramientas: Git, VS Code, XAMPP o servidores locales

Arquitectura del Sistema

Diagrama General

- Frontend: Interfaz para usuarios y admins.
- Backend: Lógica del sistema, autenticación, gestión de datos. • Base de Datos: Almacena productos, usuarios.

Patrón MVC

- Modelo: Manejo de datos (productos, usuarios, etc.)
- Vista: Interfaz web.
- Controlador: Lógica de interacción entre modelo y vista.

Guía de Instalación y Configuración

1. Instalar dependencias del backend (pip install o npm install)
2. Configurar base de datos local.
3. Cargar datos iniciales (productos de prueba).
4. Ejecutar el servidor local.
5. Acceder desde navegador (localhost:8000 o puerto configurado).

Diagrama de Gantt

Se debe entregar un cronograma visual del proyecto, con las etapas:

- Diseño UI
- Desarrollo Frontend
- Backend y lógica de negocio
- Integración con base de datos
- Pruebas
- Documentación
- Presentación final

Tipos de Usuarios

ROL	ACCESO
Usuario	Visualizar productos, editar perfil
Administrador	CRUD productos, gestionar usuarios

Material Gráfico

- Logo personalizado por equipo para identificar su sistema.
- Banners promocionales (mínimo 2) dentro del sistema.
- Capturas de pantalla del sistema en uso para el informe final.

Informe Final

1. Introducción y objetivo
2. Alcance del sistema
3. Requisitos de hardware/software
4. Arquitectura del sistema
5. Diagrama de Gantt
6. Caso de Uso
7. Guía de instalación
8. Evaluación del trabajo
9. Propuestas de mejora
10. Reflexión de equipo

11. Anexos: capturas, logos, banners, código destacado.

El equipo está formado por:

Roles	Funciones Principales
Programación (Backend) **Caballaro Juan Pablo**	Lógica del servidor, conexión a base de datos, autenticación, validaciones.
Programación (Frontend) **Caceres Isaias**	Diseño de la interfaz, vistas, interactividad (HTML, CSS, JS).
Documentación Gomez Tobias	Informe completo, guías, diagramas, presentación final.
Marketing / UI Design Diaz Katherine	Logo, banners, experiencia de usuario, nombre del sistema, estética general.

Introducción

En este trabajo práctico desarrollamos un sistema web de gestión pensado para Dolce Freddo. La idea principal fue simular cómo funcionaría un sistema real dentro de un negocio con al menos 20 puestos de trabajo, como si se tratara de una empresa que necesita organizarse mejor y digitalizar su forma de trabajar.

El sistema permite llevar un control eficiente de los productos que se venden —como sabores de helado, postres o combos—, gestionar el stock de manera automática con alertas cuando quedan pocas unidades disponibles, y diferenciar los accesos de los usuarios según sus roles. Por ejemplo, un administrador tiene acceso completo al sistema para editar productos y gestionar usuarios, mientras que un usuario común solo puede visualizar el catálogo.

Elegimos una heladería como temática porque nos pareció un ejemplo cercano, cotidiano y fácil de imaginar. Esto nos ayudó a visualizar mejor qué funciones debía tener el sistema, cómo debería organizarse la información, y qué tipo de usuarios lo usarían en el día a día.

A lo largo del desarrollo aplicamos conocimientos técnicos como el patrón arquitectónico MVC (Modelo-Vista-Controlador), y trabajamos con herramientas actuales como Django, Bootstrap para el diseño responsivo, y bases de datos como SQLite o MySQL. Además, nos organizamos en roles bien definidos lo cual nos permitió simular cómo funciona un verdadero equipo de desarrollo en una empresa.

Alcance del Sistema

El sistema que desarrollamos está pensado para cubrir algunas de las principales necesidades que puede tener Dolce Freddo al momento de gestionar sus productos, mantener un control organizado del stock y permitir que tanto el personal como los usuarios puedan interactuar con el catálogo de forma simple y ordenada.

Desde el inicio del proyecto nos enfocamos en construir una solución web accesible, que pueda ser utilizada desde distintos dispositivos y que se adapte a un entorno real de trabajo, donde muchas veces hay poco tiempo y es fundamental que todo funcione de manera rápida, clara y sin complicaciones.

Objetivos del Sistema

El desarrollo del sistema de gestión para Dolce Freddo no surge solamente como un trabajo técnico, sino también como una experiencia integral de aprendizaje. Desde el inicio nos propusimos construir una herramienta que fuera útil, clara y lo más cercana posible a una solución real para un negocio que trabaja día a día con productos de alta rotación, como es el caso de una heladería.

En este marco, el sistema busca responder a dos tipos de objetivos: por un lado, aquellos relacionados con la funcionalidad directa que necesita el comercio para operar de manera eficiente; y por otro, los objetivos pedagógicos y formativos que nos propusimos alcanzar como grupo, aplicando los conocimientos adquiridos en programación, diseño, documentación y trabajo en equipo.

Desarrollar un sistema web completo, funcional y adaptable, orientado a la gestión de productos, stock e interacción de usuarios, que le permita a Dolce Freddo mejorar sus procesos internos, mantener su catálogo digital actualizado y brindar una experiencia ordenada y accesible tanto al personal administrativo como a los usuarios comunes que consultan la oferta del local.

Este objetivo general fue el punto de partida para definir los distintos módulos, herramientas y enfoques que íbamos a implementar, buscando siempre mantener un equilibrio entre la viabilidad técnica y la utilidad práctica.

Objetivos Específicos

1. **Construir una Estructura de Sistema Clara y Modular.**

Desde el inicio, uno de nuestros objetivos fue trabajar con una arquitectura organizada. Esto implicó estructurar el proyecto bajo el patrón MVC (Modelo - Vista - Controlador), que separa claramente los componentes de lógica (backend), interfaz visual (frontend) y manipulación de datos (base de datos).

Esto no solo mejora la legibilidad del código y la posibilidad de futuras mejoras, sino que también permite que diferentes miembros del equipo puedan trabajar en paralelo en distintas partes del sistema sin pisarse ni generar conflictos.

2. Implementar un Sistema de Gestión de Productos Completo (CRUD)

Una de las necesidades más importantes en un comercio como una heladería es poder tener control sobre su catálogo. Por eso, el sistema permite al administrador realizar operaciones de:

- Creación de productos: ingresar nuevos helados y postres al catálogo, asignándoles nombre, precio, imagen, stock y descripción.
- Edición: modificar cualquiera de esos campos cuando cambie algún detalle o haya errores.
- Eliminación: quitar productos que ya no se ofrezcan.
- Visualización: tanto en el panel de administración como en la vista pública.

Este CRUD se aplica específicamente a helados y postres, ya que son los productos que más cambian en este tipo de negocio. Otros elementos como combos o sabores disponibles se muestran en la interfaz como parte del contenido visual y de navegación, pero no están pensados para editarse en esta primera versión.

3. Agregar Alertas de Stock Bajo para Mejorar la Organización

El sistema incluye una funcionalidad clave: alertas visuales automáticas que avisan al administrador cuando un producto tiene menos de 5 unidades en stock. Este control, si bien es simple, resulta muy útil para anticiparse a posibles faltantes, evitar demoras y garantizar que siempre haya disponibilidad de los productos más vendidos.

Esta lógica está integrada al backend y se muestra de forma clara en el panel de control del administrador.

4. Gestionar Roles y Accesos de Usuarios de Forma Segura

Otro de los objetivos fue dividir claramente los accesos al sistema. Existen dos tipos de usuarios:

- Usuarios comunes: pueden registrarse por su cuenta, consultar el catálogo, navegar los productos y acceder a su perfil personal.
- Administradores: tienen acceso a un panel exclusivo donde pueden gestionar todo el sistema, incluyendo productos y usuarios.

Cada rol tiene sus permisos definidos, y se implementaron validaciones para que no puedan acceder a funciones que no les corresponden. Esto nos ayudó a mantener un sistema más organizado, seguro y acorde a la realidad de muchas plataformas de gestión actuales.

5. Crear una Interfaz Moderna, Intuitiva y Responsiva

Nos propusimos diseñar una interfaz visual que sea clara, fácil de entender, y sobre todo que funcione bien en cualquier dispositivo. Hoy en día, muchas tareas se realizan desde el celular o una tablet, y por eso el sistema fue construido con un diseño responsivo que se adapta automáticamente a distintos tamaños de pantalla.

También se cuidaron los detalles de navegación: botones accesibles, textos legibles, alertas visibles y una estructura lógica que permita encontrar fácilmente cada función.

6. Incorporar Herramientas para Facilitar la Búsqueda de Productos

Sabemos que, a medida que el catálogo crece, encontrar un producto específico puede volverse complicado. Por eso se agregó un buscador por nombre, que permite filtrar los productos fácilmente. Además, se contempla para una futura actualización la posibilidad de agregar filtros por tipo de producto (helado, postre, combo) para mejorar aún más la navegación.

7. Aplicar Validaciones de Seguridad Básicas en el Inicio de Sesión

La seguridad es un aspecto que no puede dejarse de lado, por más simple que sea el sistema. En este caso se implementaron:

- Validaciones de campos vacíos.
- Restricciones en contraseñas (como uso de caracteres especiales).
- Rutas protegidas para evitar que usuarios sin permisos accedan al panel de administración.

8. Fomentar el Trabajo en Equipo y la Colaboración entre Áreas

Además de programar, uno de los objetivos más importantes del proyecto fue trabajar en equipo, dividiendo roles y tareas de forma equilibrada. Cada miembro se enfocó en un área (backend, frontend, diseño, documentación), lo que nos obligó a comunicarnos constantemente, coordinar avances, resolver errores juntos y tomar decisiones en grupo.

Esto nos permitió experimentar cómo se trabaja en proyectos reales de software, donde cada integrante aporta desde su especialidad, pero el resultado final depende del compromiso de todos.

9. Simular un Entorno Real de Desarrollo de Software

Durante todo el proceso tratamos de seguir una lógica similar a la que se utiliza en proyectos profesionales: planificación por etapas, cronograma con fechas, control de versiones, pruebas y correcciones, documentación técnica y presentación del trabajo terminado. Este enfoque nos ayudó a entender cómo se vive el desarrollo de un sistema desde adentro, y nos dio herramientas que seguramente aplicaremos en el futuro.

Requisitos de Hardware/Software

Hardware

Componentes	Requisitos Mínimos	Requisitos Recomendados
Procesador	Dual Core 1.6 GHz o superior	Intel Core i3 / AMD Ryzen 3 o superior
Memoria RAM	2 GB	4 GB o más
Almacenamiento	Navegador + caché (100 MB aprox.)	500 MB libres para navegación fluida
Pantalla	Resolución mínima: 1024 x 768px	Full HD (1920 x 1080 px)
Conectividad	Conexión a internet (Wi-Fi o cableada)	Internet estable (>10 Mbps)

Software

Elemento	Detalle
Sistema Operativo	Windows 7 o superior, macOS, Linux, Android, iOS
Navegador	Google Chrome, Mozilla Firefox, Microsoft Edge, Safari
Versión del Navegador	Preferentemente versiones actualizadas (últimos 2 años)
JavaScript	Debe estar habilitado en el navegador
Compatibilidad	100% funcional en dispositivos móviles y escritorio

Arquitectura del Sistema

El sistema web desarrollado está basado en una arquitectura moderna, clara y modular que sigue el patrón MVC (Modelo - Vista - Controlador), utilizado en el desarrollo de aplicaciones web debido a su eficacia para organizar la lógica del sistema, mejorar la escalabilidad y facilitar el mantenimiento del código. La arquitectura fue pensada para ofrecer una separación ordenada entre la lógica de negocio, la presentación visual y la gestión de los datos, permitiendo que distintas partes del equipo pudieran trabajar en simultáneo sin interferencias.

Modelo

Es la capa encargada de representar y gestionar la información que utiliza el sistema. En esto se definen las estructuras de datos que reflejan los elementos centrales del proyecto, como los productos (helados y postres), los usuarios y su stock. Además, el modelo se comunica directamente con la base de datos, lo que permite crear, leer, actualizar y eliminar registros de forma controlada. Esta interacción se da mediante ORM (Object-Relational Mapping).

Algunos de los atributos definidos en nuestro sistema son:

- Nombre del Producto.
- Precio.
- Descripción.
- Imagen.

Vista

La vista es la parte visible de la aplicación, la interfaz con la que interactúan tanto el usuario común como el administrador. Acá es donde se le brinda la experiencia al usuario, el diseño visual y la organización del contenido.

La vista fue desarrollada con:

- HTML Para la estructura.
- CSS y Bootstrap Para el diseño responsive.
- JavaScript Para dar interactividad.

Y se muestran páginas como:

- Inicio.
- Carrito.
- Acerca de Nosotros.
- Login.

Controlador

El controlador actúa como intermediario entre el modelo y la vista. Su función principal es gestionar las acciones del usuario, interpretar las solicitudes (por ejemplo, "ver productos", "editar producto", "iniciar sesión") y devolver la respuesta correspondiente.

Cada vez que un usuario interactúa con el sistema, por ejemplo "Eliminar producto" o busca un nombre en el buscador, el controlador se encarga de procesar esa acción, aplicar la lógica necesaria, comunicarse con el modelo para obtener los datos, y finalmente mostrar la vista correspondiente con la información actualizada.

COMPONENTES ADICIONALES

Frontend

Es la parte que se ejecuta en el navegador del usuario. En esto se muestra todo el contenido visual (productos, menús, botones, formularios). Su principal función es mostrar información y permitir la interacción del usuario.

- **Herramientas:** HTML, CSS, Bootstrap, JavaScript.

Backend

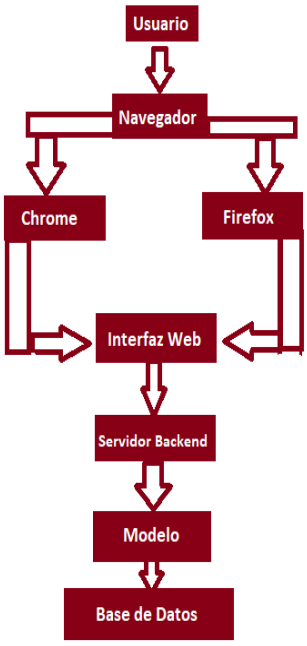
Es el núcleo del sistema. Aquí se procesa toda la lógica del negocio, como el login, el control de stock, las restricciones de acceso según el rol del usuario, y la validación de formularios. El backend se comunica constantemente con la base de datos y con el frontend.

- **Herramientas:** Python con Django y JavaScript.

Base de Datos

La base de datos funciona como almacenamiento permanente de toda información necesaria para el funcionamiento del sistema.

- Usuarios registrados.
- Productos cargados.
- Cantidad de stock.
- Roles de cada cuenta.



Gestión de Productos

Uno de los principales pilares del sistema es la gestión de productos, ya que se trata de una heladería con una oferta variada y dinámica. Para esta primera versión se decidió trabajar con cuatro categorías principales:

- Helados.
- Postres.
- Combos Promocionales.
- Sabores Disponibles.

De todas estas categorías, los helados y postres son los productos que se pueden gestionar completamente desde el panel de administración. Esto significa que el administrador puede cargar nuevos productos, editar la información (como nombre, descripción, precio, imagen y stock) y eliminarlos si ya no forman parte del menú.

Por otro lado, los combos y los sabores funcionan como acompañamiento visual dentro del catálogo, y si bien por ahora no se editan desde el sistema, cumplen una función importante a nivel comercial: ayudan a mostrar promociones o a que el cliente conozca qué sabores hay disponibles al momento de hacer un pedido o consulta.

Control de Stock con Alertas

Para evitar que los productos se agoten sin que nadie lo note, el sistema incluye una alerta automática de stock bajo. Esta alerta se activa cada vez que un producto tiene menos de cinco unidades disponibles. De esta forma, el administrador puede actuar con tiempo y reponer lo necesario antes de que se genere un problema.

Este detalle, aunque puede parecer menor, es clave en negocios como una heladería, donde la demanda puede variar mucho de un día a otro, y quedarse sin un sabor o un postre muy pedido puede impactar en la experiencia del cliente.

Gestión de Usuarios y Permisos

En cuanto al acceso al sistema, se implementó un mecanismo de registro simple y amigable. Cualquier persona puede registrarse desde la plataforma, y al hacerlo se le asigna automáticamente el rol de usuario común.

- Este tipo de usuario puede:
- Navegar por el catálogo de productos.
- Consultar imágenes, descripciones y precios.
- Acceder a su perfil personal.

Por otra parte, existen los administradores, que tienen acceso a una sección exclusiva del sistema desde donde pueden:

- Gestionar completamente el inventario de productos (crear, editar, eliminar).
- Ver las alertas de stock.
- Administrar los usuarios registrados.

Esta división de roles permite mantener el orden y la seguridad dentro del sistema, asegurando que solo las personas autorizadas puedan hacer modificaciones importantes.

Interfaz Moderna y Adaptable

El sistema fue diseñado con la intención de que pueda ser utilizado en cualquier momento y desde cualquier lugar. Por eso, se trabajó especialmente en que sea responsive, es decir, que se vea bien y funcione correctamente tanto en computadoras de escritorio como en celulares o tablets.

Esto es importante porque permite que el sistema se pueda usar en un entorno real de trabajo, donde tal vez el encargado necesite hacer una consulta rápida desde su celular, o mostrarle a un cliente el catálogo desde una tablet, sin necesidad de depender de una computadora fija.

A nivel estético, se buscó que la interfaz sea clara, sin elementos innecesarios, para que cualquier persona. con o sin conocimientos técnicos, pueda usarla sin dificultad.

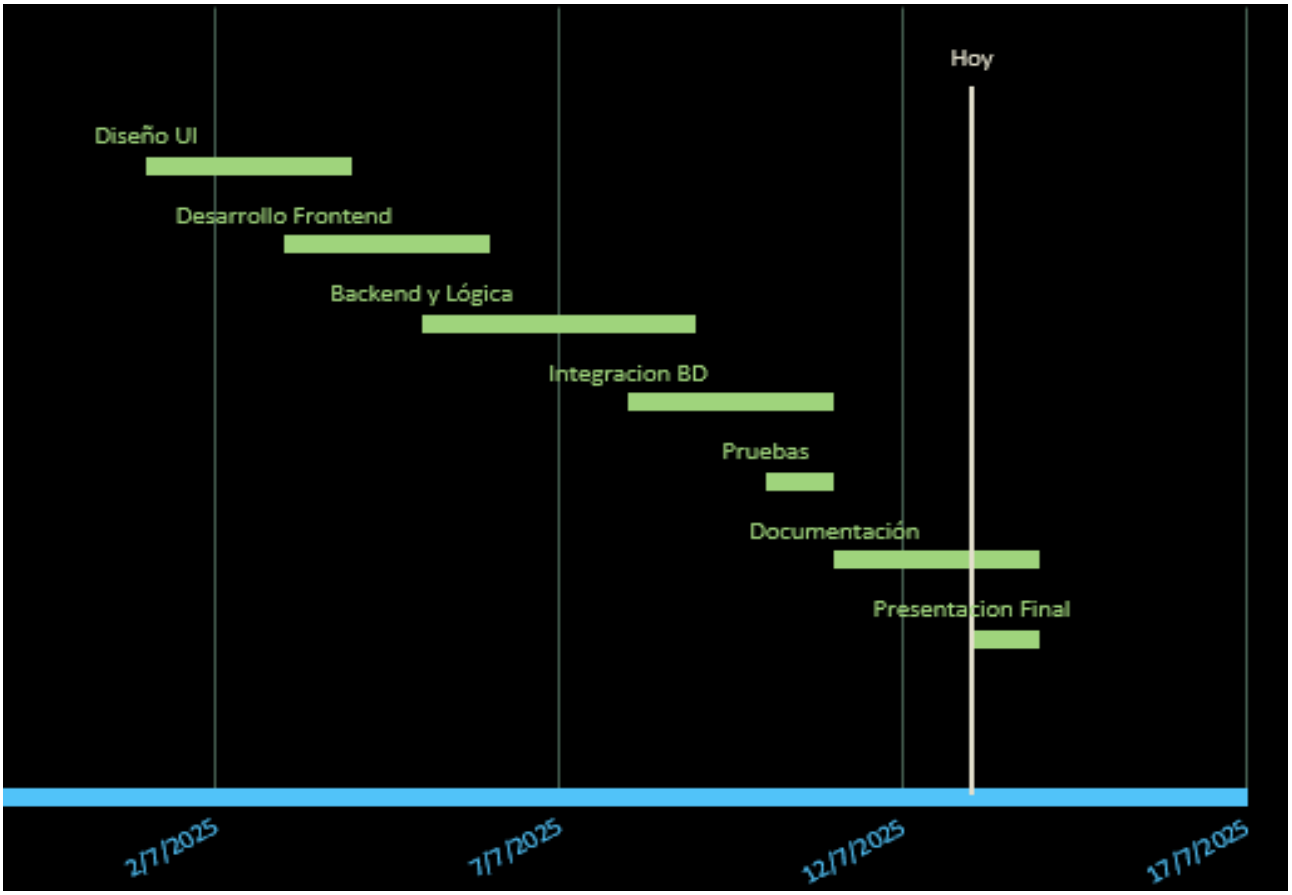
Funciones Adaptables

Además de las funciones principales, se decidió agregar algunas herramientas que ayudan a mejorar la navegación dentro del sistema:

- Buscador de productos: permite encontrar rápidamente un producto por su nombre o palabra clave.
- Filtro por tipo de producto (en evaluación): se está considerando agregar un filtro que permita al usuario seleccionar si quiere ver solo helados, solo postres, combos, etc., lo cual puede ser útil si el catálogo se expande en el futuro.

Estas funciones no son esenciales, pero aportan a una mejor experiencia para quienes usan el sistema, y lo acercan a las herramientas reales que hoy en día se usan en el comercio digital.

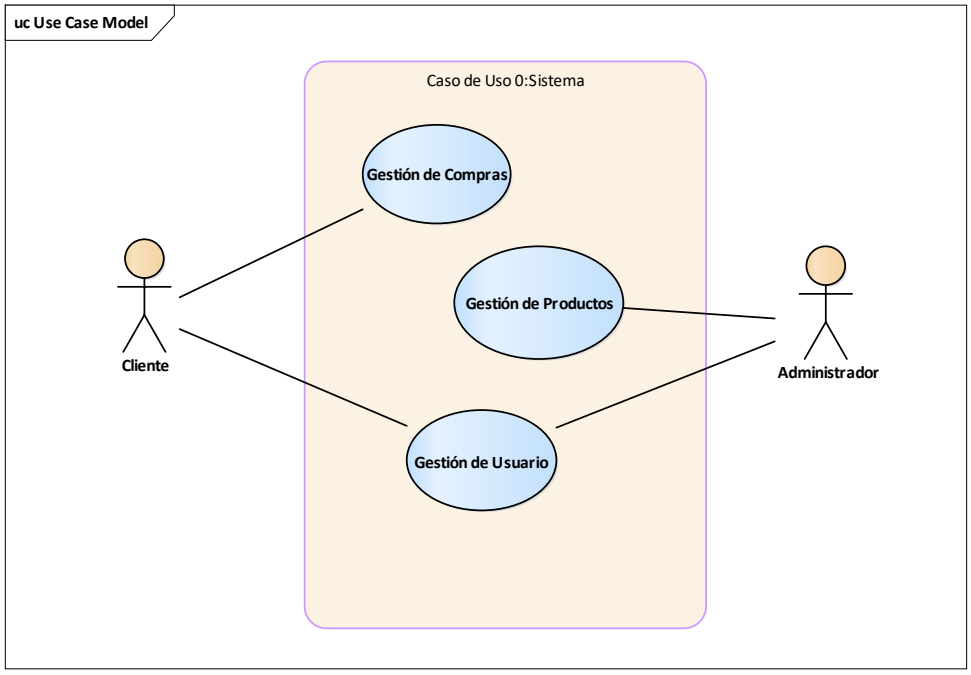
Diagrama de Gantt

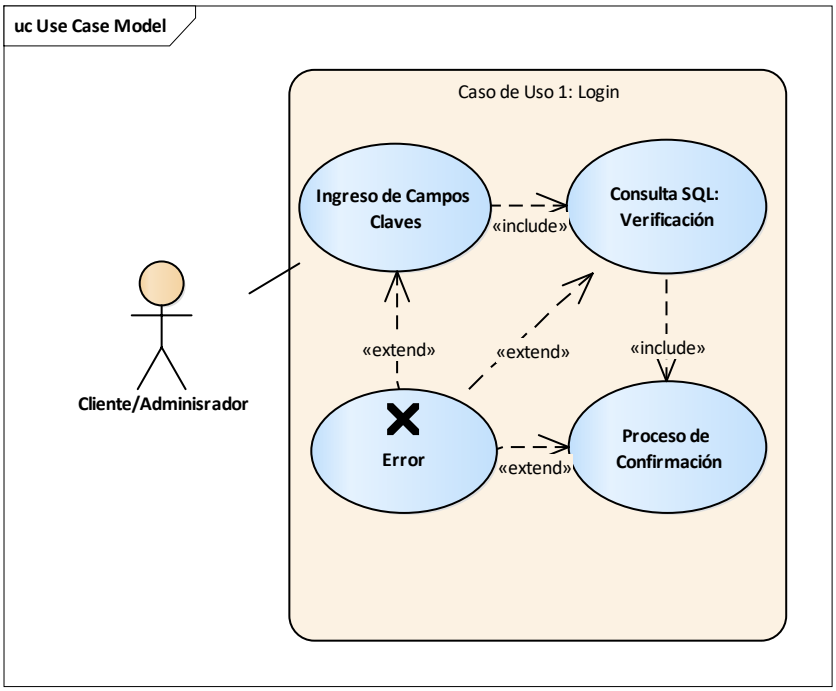


N.º	Fecha de inicio	Fecha de finalización	Tarea
1	1/7/2025	3/7/2025	Diseño UI
2	3/7/2025	5/7/2025	Desarrollo Frontend
3	5/7/2025	8/7/2025	Backend y Lógica
4	8/7/2025	10/7/2025	Integracion BD
5	10/7/2025	10/7/2025	Pruebas
6	11/7/2025	13/7/2025	Documentación
7	13/7/2025	13/7/2025	Presentacion Final

Caso de Uso

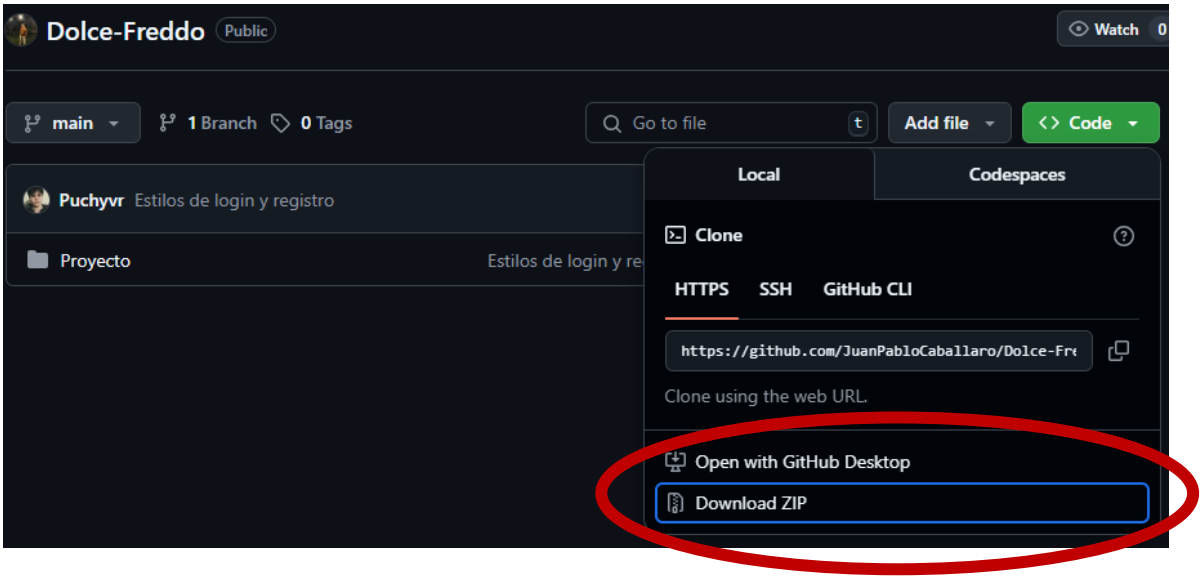
Este punto no llegamos a completarlo debido a la escasez de entendimiento por parte del equipo de documentación. Se adjunta lo entendido hasta el momento.



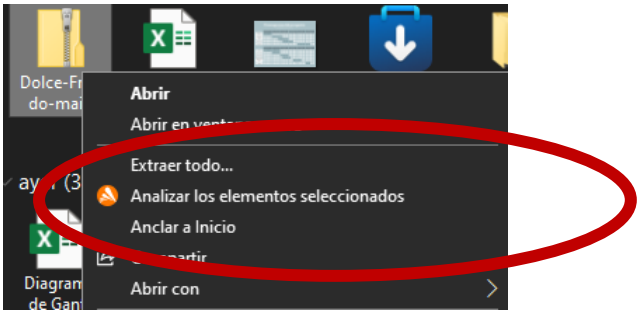


Guía de Instalación

Como primer paso descargamos el ZIP del repositorio.

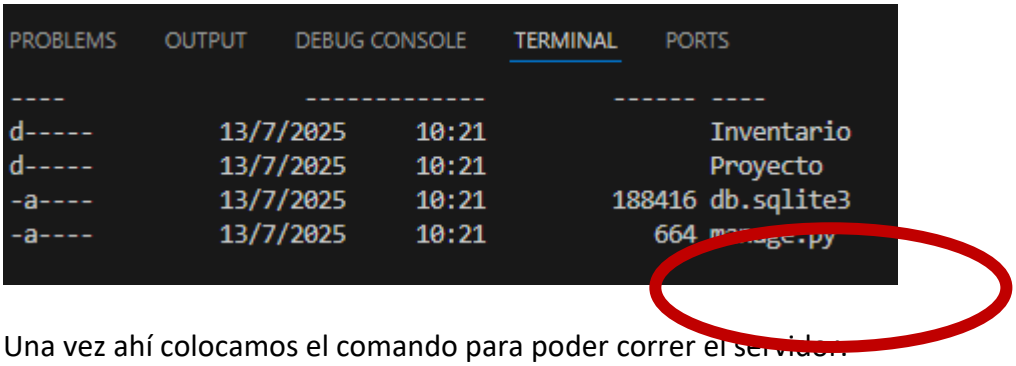


Una vez que el ZIP esta descargado, se extrae en el escritorio.

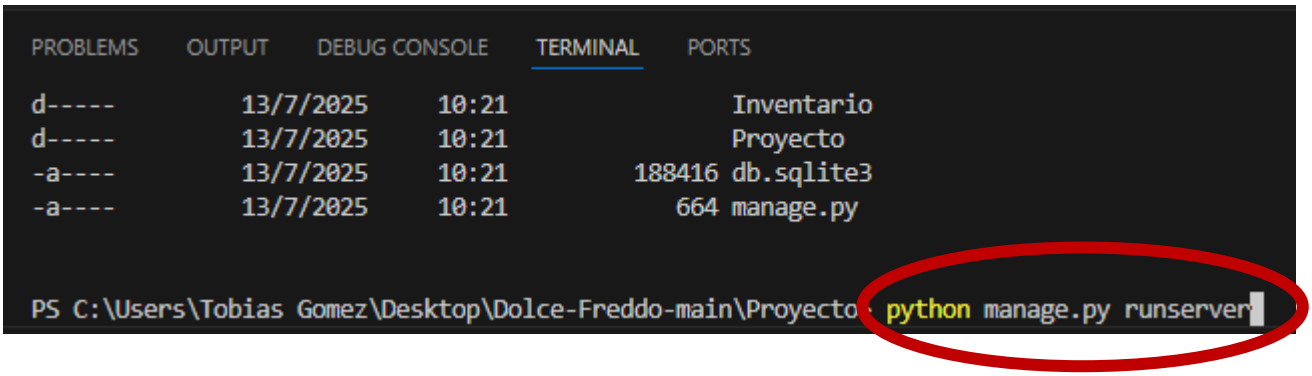


Luego se abre la carpeta desplazandolo hacia el VisualStudio.

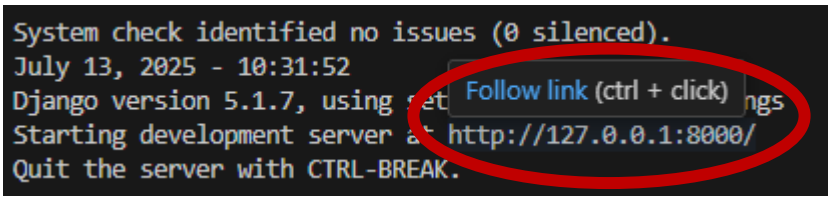
Al entrar hay que verificar que estemos ubicados en Manage.py asi podemos correr la pagina.



Una vez ahí colocamos el comando para poder correr el servidor.

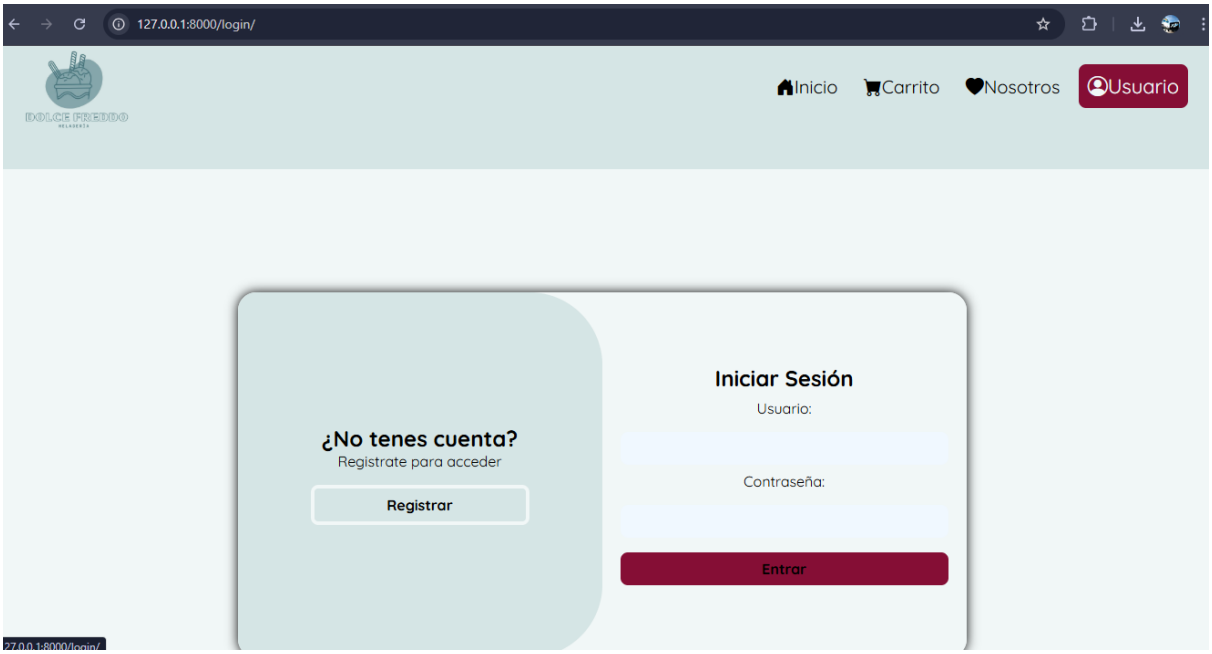


Ingresamos al link.



Una vez ingresados ya podemos navegar por lo que es la pagina.

Una de las funciones de la página es poder loguearse para poder hacer compras.



Para crear una cuenta hay que ir al apartado de “¿No tenes cuenta?” y crearse una poniendo un usuario y una contraseña.

Una vez creado se coloca en la parte de Inicio de sesión y ya se puede navegar estando logueado.

Evaluación del Trabajo

El desarrollo de este sistema web fue una experiencia muy valiosa para todos los integrantes del equipo. Más allá de los conocimientos técnicos aplicados, nos permitió organizarnos, distribuir tareas y trabajar como si estuviéramos dentro de un equipo real de desarrollo.

Desde el principio, tuvimos claro lo que queríamos lograr debido a la consigna, y en base a eso, pudimos hacer algo a nuestro gusto.

Lo que hicimos bien:

Supimos organizarnos y repartir las tareas de forma equilibrada, sabiendo quien era mejor para cada cosa y así liberar toda nuestra capacidad. Cada uno cumplió su rol y se apoyó en los demás cuando hizo falta.

Logramos que el sistema fuera totalmente funcional, cumpliendo con los requisitos.

Pudimos mejorar nuestro conocimiento con herramientas como Django, Bootstrap, Git, y nos sentimos más cómodos trabajando con ellas.

Lo que nos costó:

Conectar el backend con el frontend no fue tan fácil como esperábamos. Tuvimos que entender bien cómo pasaban los datos entre ambos.

Algunos errores con la base de datos o el servidor nos hicieron perder tiempo, pero aprendimos a solucionarlos leyendo documentación o buscando ejemplos.

El diseño responsivo también nos dio trabajo, sobre todo para que la interfaz se viera bien tanto en computadoras como en celulares.

Propuesta de Mejora

Una vez finalizado el proyecto y viendo el sistema en funcionamiento, empezamos a pensar en qué cosas podríamos mejorar o agregar si tuviéramos más tiempo o si se tratara de una versión pensada para uso real en una heladería.

Durante el desarrollo, nos enfocamos en cumplir con todos los requisitos principales: login con roles, CRUD de productos, alertas de stock y una interfaz clara. Sin embargo, también nos dimos cuenta de que el sistema puede crecer mucho más y adaptarse a otras necesidades.

Algunas ideas que surgieron como posibles mejoras fueron:

Incorporar estadísticas

Nos gustaría sumar un panel con gráficos o datos que muestren, por ejemplo, qué productos se quedan sin stock más seguido o cuáles se usan más. Sería útil para tomar decisiones en la heladería.

Organizar los productos por categorías

Si el sistema tuviera muchas variedades de helado o productos extra como toppings o postres, sería más cómodo agruparlos por tipo para encontrarlos más rápido.

Adaptarlo mejor a celulares

Aunque hicimos un diseño responsivo, creemos que podríamos mejorar la experiencia en celulares. Incluso pensamos que podría ser útil convertir el sistema en una aplicación web que se pueda usar como una app.

Notificaciones automáticas

Otra idea sería que el sistema pueda enviar alertas por correo o mostrar notificaciones cuando el stock de un producto esté por debajo de cierto número, para no depender solo de entrar a revisar.

Vincularlo con un sistema de ventas

Si este proyecto siguiera creciendo, podríamos conectarlo con un módulo de ventas, donde además de controlar el stock, se pueda registrar qué se vendió, cuánto y cuándo.

Reflexión del Equipo

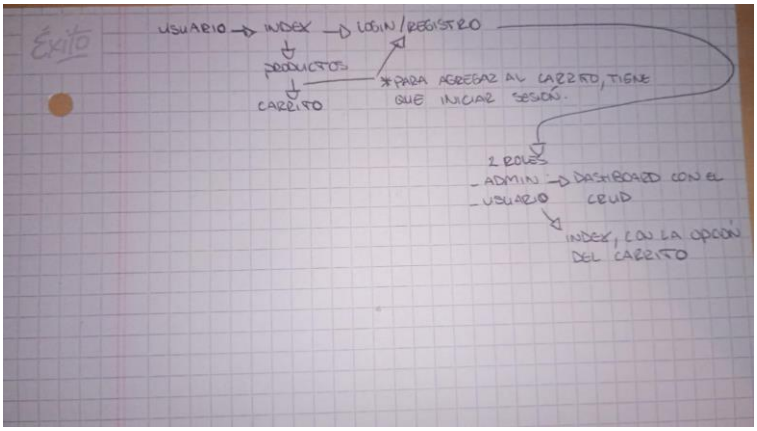
Este proyecto fue un desafío, pero al mismo tiempo una buena experiencia. Pudimos aplicar lo que veníamos aprendiendo y llevarlo a algo real, viendo el sistema funcionar de manera correcta.

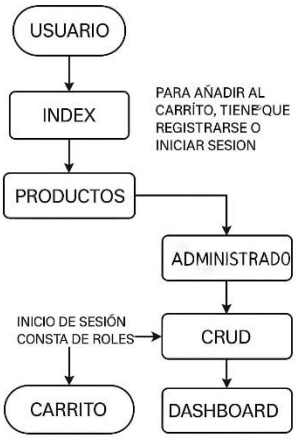
Hubo errores, problemas y momentos de frustración, pero también mucha alegría cuando todo empezaba a resolverse. Lo mejor fue el trabajo en equipo: cada uno aportó lo suyo desde sus conocimientos y aprendimos a organizarnos.

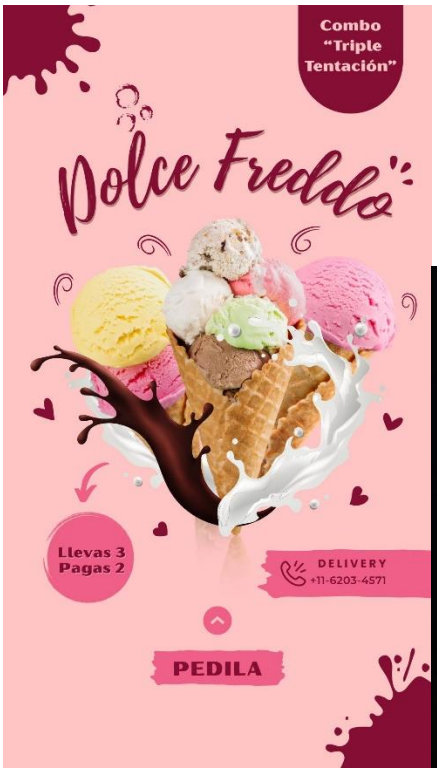
Además de lo técnico, nos llevamos la satisfacción de poder crear algo desde cero, con nuestro diseño, nuestras ideas y mucho esfuerzo. Estamos muy contentos con el resultado (aunque algunas cosas quedaron sin resolver), nos motiva a seguir aprendiendo para una próxima entrega de proyecto con más calidad y más contenido, es decir, este proyecto nos motiva a mejorar para futuros proyectos.

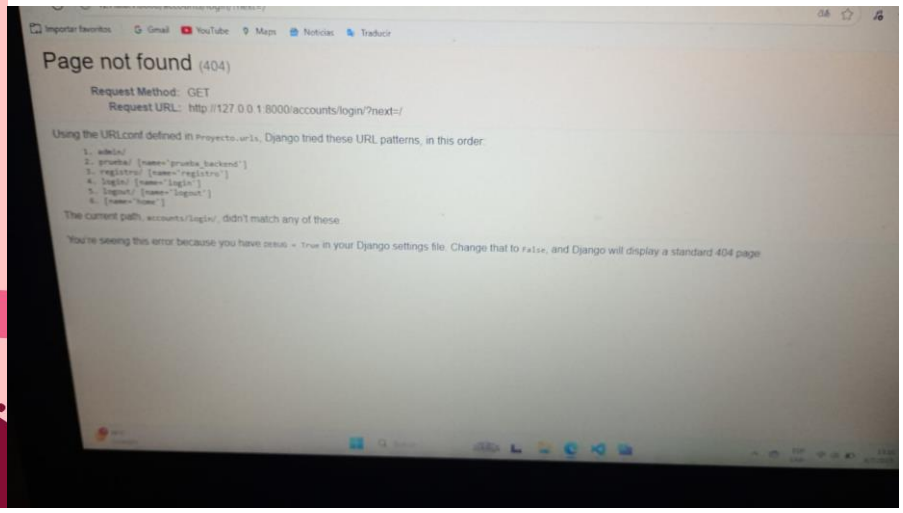
Anexos

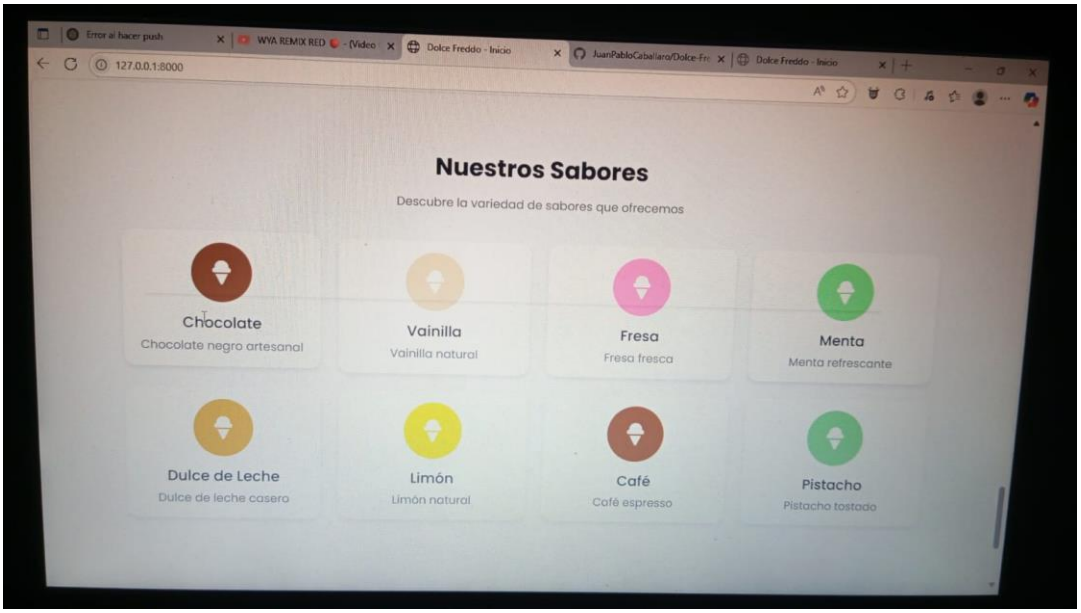

Dolce Freddo
HELADERÍA
FANTASY CREAM
HELADERIA

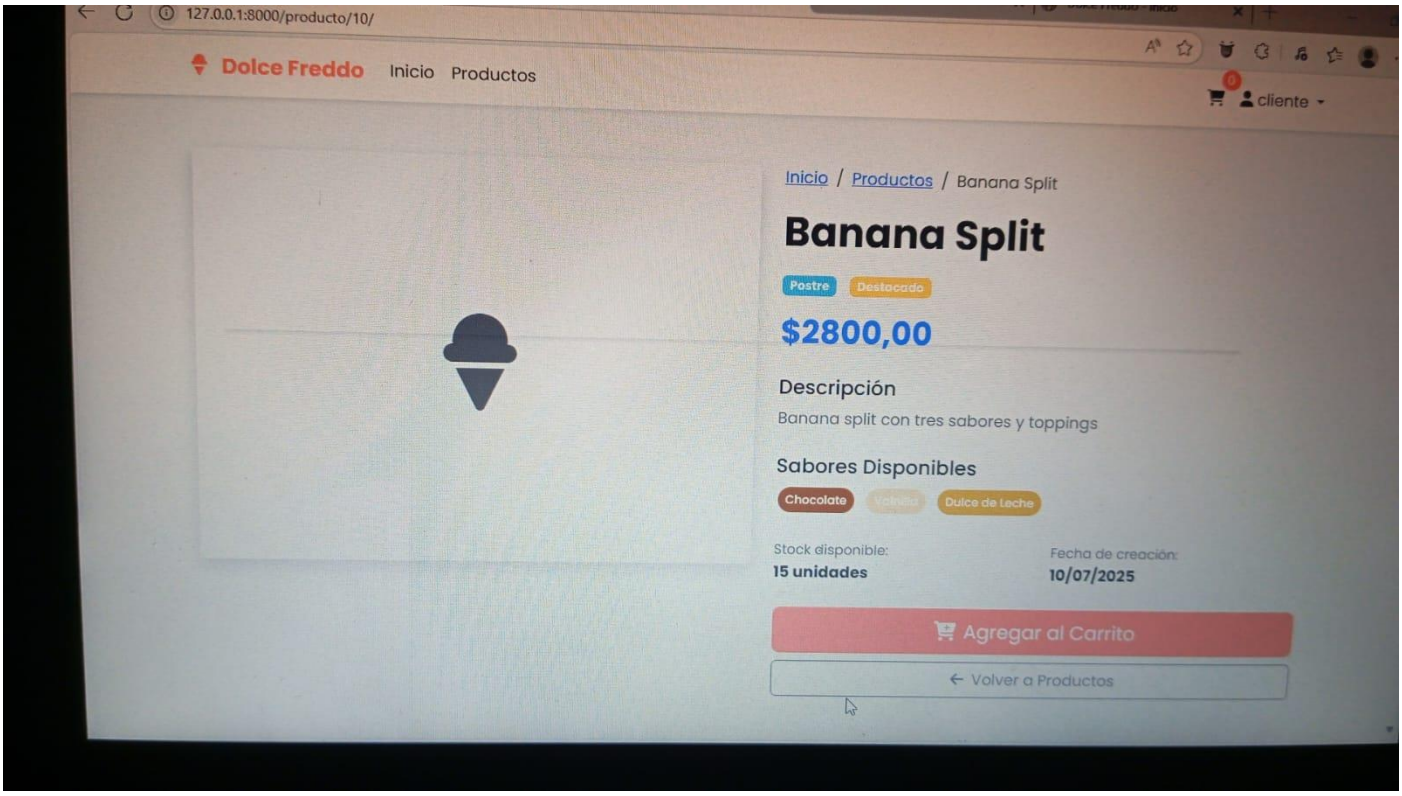

Éxito
USUARIO → INDEX → LOGIN / REGISTRO
PRODUCTOS
CARRITO
*PARA AGREGAR AL CARRITO, TIENE QUE INICIAR SESIÓN.
2 ROLES
- ADMIN → DASHBOARD CON EL CRUD
- USUARIO
INDEX, CON LA OPCIÓN DEL CARRITO


USUARIO
INDEX
PARA AÑADIR AL CARRITO, TIENE QUE REGISTRARSE O INICIAR SESION
PRODUCTOS
ADMINISTRADO
INICIO DE SESIÓN CONSTA DE ROLES
CRUD
CARRITO
DASHBOARD


Combo
"Triple
Tentación"
Dolce Freddo
Llevas 3
Pagas 2
DELIVERY
+11-6203-4571
PEDILA


Page not found (404)
Request Method: GET
Request URL: http://127.0.0.1:8000/accounts/login/?next=/
Using the URLconf defined in Proyecto.urls, Django tried these URL patterns, in this order:
1. admin/
2. prueba/ [name='prueba_backend']
3. registro/ [name='registro']
4. login/ [name='login']
5. logout/ [name='logout']
6. [name='home']
The current path, accounts/login/, didn't match any of these.
You're seeing this error because you have DEBUG = True in your Django settings file. Change that to False, and Django will display a standard 404 page.


127.0.0.1:8000
Nuestros Sabores
Descubre la variedad de sabores que ofrecemos
Chocolate
Chocolate negro artesanal
Vainilla
Vainilla natural
Fresa
Fresa fresca
Menta
Menta refrescante
Dulce de Leche
Dulce de leche casero
Limón
Limón natural
Café
Café espresso
Pistacho
Pistacho tostado


127.0.0.1:8000/producto/10/
Dolce Freddo
Inicio
Productos
cliente
Inicio / Productos / Banana Split
Banana Split
Postre
Destacado
$2800,00
Descripción
Banana split con tres sabores y toppings
Sabores Disponibles
Chocolate
Dulce de Leche
Stock disponible:
15 unidades
Fecha de creación:
10/07/2025
Agregar al Carrito
Volver a Productos