

TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA (IFC302)

MÓDULO: PROYECTO

CURSO: 2023-2024

CONVOCATORIA: (Marzo 2024)

PitsaJaus

PROFESOR-COORDINADOR: GUILLÓ PITARQUE, JORGE

AUTOR: PAN CHEN, GUO RUI

# ÍNDICE:

[**ÍNDICE:**](#_dj9rv0uu869v) **1**

[**1) INTRODUCCIÓN:**](#_m0s0elske8qr) **1**

[1.1) MOTIVACIÓN:](#_lszylibcf6k4) 1

[1.2) DEFINICIÓN:](#_o6a41rn9kk1l) 2

[1.3) OBJETIVOS:](#_op95tfiqhq8a) 3

[**2) RELACIÓN CON PROYECTOS SIMILARES:**](#_cru6mvdbum4v) **5**

[**3) TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS**](#_eh21zdkvtjlh) **6**

[3.1) TECNOLOGÍAS](#_43om3ib8jhh4) 6

[Framework y lenguajes](#_er7gqnxm2k84) 6

[Base de datos](#_8hxc63laxhmd) 6

[Librerías](#_5djynovxf8xq) 6

[3.2) HERRAMIENTAS](#_bwt24kj2mlbb) 7

[**4) PLANIFICACIÓN TEMPORAL**](#_axv2fglphx4j) **7**

[**5) DESARROLLO DEL PROYECTO**](#_rrxuapr25zrt) **8**

[**6) CONCLUSIONES**](#_252ntdwvtbpm) **9**

[**7) GLOSARIO**](#_n08rbw5jwubq) **10**

[**8) BIBLIOGRAFÍA/WEBGRAFÍA**](#_6i7t39l2g975) **11**

[Documentación oficial](#_2h5p1dl1hgi3) 11

[Otros enlaces de interés](#_d9tcig9k6pyl) 11

[**ANEXO A. Manual de instalación**](#_4od9a5obgi52) **11**

[Prerrequisitos](#_5a6jmj4y47ts) 12

[Instalación del proyecto](#_dhkg1ssq9nnb) 12

[Despliegue de la web](#_dp7pydmrbxi1) 13

[Configuración de la base de datos](#_w3e26qf0jucl) 15

[**ANEXO B. Manual de usuario**](#_47srcnifsb4r) **17**

[Invitado (sin registro):](#_211jabbx2rut) 17

[Usuario general](#_r7w73smsl6q2) 25

[Cliente:](#_ye7bbi345itb) 28

[Gerente:](#_cxwujzpru0h5) 35

[Administrador:](#_w9l47gltpcpm) 39

# 1) INTRODUCCIÓN:

## 1.1) MOTIVACIÓN:

La pizzeria PitsaJaus necesita una web para expandir su presencia en línea y facilitar a sus clientes la realización de pedidos desde la comodidad de sus hogares o dispositivos móviles. Además, quiere que los empleados de sus numerosas franquicias puedan usar la web para realizar tareas de gestión interna, como es la gestión de los pedidos que reciben.

El proyecto busca satisfacer los requisitos del cliente para presentar un MVP antes del plazo límite (31 de mayo del 2024)

## 1.2) DEFINICIÓN:

Este proyecto será una aplicación web para la gestión de una cadena de pizzerías, que permitirá a los usuarios acceder a contenido y realizar diferentes tareas según el “rol” que tengan.

La página está desplegada en producción y es accesible desde el siguiente [enlace](https://pitsa-jaus.vercel.app/)

Hay cuatro tipos de usuarios previstos:

* **Invitado (sin inicio de sesión)**: podrá acceder únicamente a las páginas públicas, como son la página de inicio, la carta, el listado de locales, etc.
* **Cliente**: podrán hacer pedidos a las tiendas cercanas a ellos, así como realizar seguimiento de los pedidos pasados y guardar su información personal para futuras compras.
* **Gerente**: este tipo de usuario es para los gerentes de las tiendas de PitsaJaus. Su función es la de gestionar los pedidos que recibe su tienda, así como otras tareas de gestión interna.
* **Administrador**: realizarán tareas de gestión de los productos que se muestran en la web, así como de los usuarios que están registrados

El proyecto se realizará utilizando Next.js, un popular framework para React, lo que nos permitirá aprovechar sus diversas ventajas para crear una aplicación web moderna, eficiente y escalable.

A continuación, se detallan algunos de los beneficios específicos que aporta Next.js al proyecto:

1. **Rendimiento mejorado**: Next.js ofrece renderizado del lado del servidor (SSR) y renderizado estático (SSG), lo que se traduce en tiempos de carga de página más rápidos y una mejor experiencia de usuario. Esto es especialmente importante para sitios web con mucho tráfico o contenido pesado.
2. **SEO optimizado**: Next.js facilita la creación de sitios web que optimicen SEO, ya que genera automáticamente etiquetas meta, mapas de sitio y otros datos relevantes para los motores de búsqueda. Esto ayuda a que el sitio web se clasifique mejor en los resultados de búsqueda y obtenga más tráfico orgánico.
3. **Desarrollo simplificado**: Next.js proporciona una serie de características integradas que simplifican el desarrollo de aplicaciones web React, como enrutamiento automático, administración de estado y precarga de datos. Esto permite a los desarrolladores concentrarse en el desarrollo de la web en lugar de la infraestructura subyacente.
4. **Amplia comunidad**: Next.js cuenta con una comunidad grande y activa de desarrolladores, así como con una abundancia de librerías y herramientas de terceros.
5. **Facilidad de implementación**: Next.js facilita la implementación de aplicaciones web en una variedad de plataformas de alojamiento, como Vercel, Netlify y AWS Amplify. Esto permite a los desarrolladores poner sus aplicaciones en producción rápidamente y sin problemas.

## 1.3) OBJETIVOS:

A continuación se define una serie de objetivos a cumplir durante el desarrollo del proyecto:

* **OG1: Interfaz de usuario.**
  + OE1.1: Debe ser atractiva
  + OE1.2: Debe incluir la marca de la empresa (colores, logos, etc.)
  + OE1.3: Debe ser accesible e intuitiva
  + OE1.4: Debe ser responsiva
* **OG2: La carta (/menu)**
  + OE2.1: Debe mostrar listado de pizzas
  + OE2.2: Debe incluir imagen, nombre, descripción, precio
  + OE2.3: Debe permitir añadir pizzas al carrito a los usuarios Cliente
  + OE2.4: El listado debe estar paginado
* **OG3: Listado de locales (/find-us)**
  + OE3.1: Debe mostrar todos los locales que hay registrados
  + OE3.2: El listado debe estar paginado
  + OE3.3: Debe permitir filtrado mediante barra de búsqueda
  + OE3.4: La tabla debe actualizarse automáticamente al escribir
* **OG4: Inicio de sesión (/auth/login)**
  + OE4.1: Debe permitir inicio de sesión con credenciales válidas
  + OE4.2: Si el usuario y contraseña no coinciden con un registro en la base de datos, no permitir ingreso
  + OE4.3: Debe incluir un enlace a la página de registro.
  + OE4.4: Si el usuario que intenta acceder está registrado, pero está pendiente de validación, no debe permitir acceso.
* **OG5: Registro de usuario (/auth/register)**
  + OE5.1: Debe permitir registrar en la base de datos al usuario usando los datos del formulario
  + OE5.2: Debe validar la obligatoriedad de los campos requeridos, así como otros requisitos como son la longitud de los campos de tipo texto.
  + OE5.3: Si el registro es de un usuario Cliente, debe mostrar pantalla de éxito al completar registro
  + OE5.4: Si el registro es de un usuario Gerente o Administrador, debe mostrar pantalla de “Esperando validación” al completar registro
* **OG6: Mi perfil (/dashboard/my-account)**
  + OE6.1: Debe permitir al usuario visualizar y editar sus datos personales
  + OE6.2. Debe permitir al usuario editar su contraseña
    - OE6.2.1:Dado que un usuario quiere editar su contraseña, debe primero introducir la contraseña vieja, y entonces introducir la nueva dos veces.
    - OE6.2.2: Dado que un usuario cambia su contraseña, debe de cerrar la sesión automáticamente.
  + OE6.5: Debe permitir a un usuario borrar su cuenta.
    - OE6.5.1: Dado que un usuario quiere borrar su cuenta, debe exigir que introduzca primero su contraseña
    - OE6.5.2: Dado que un usuario ha borrado su cuenta con éxito, debe cerrar sesión inmediatamente.
* **OG7: Carrito (/dashboard/cart)**
  + OE7.1: Debe ser accesible sólo para los usuarios Cliente.
  + OE7.2: El usuario debe ver las diferentes fases del pedido, claramente diferenciadas entre sí
  + OE7.3: Selección de tienda
    - OE7.3.1: El usuario debe poder seleccionar una tienda de un listado
    - OE7.3.2: Para poder completar el pedido, el usuario debe haber seleccionado una tienda.
  + OE7.4: Carrito:
    - OE7.4.1: El usuario debe poder ver un enlace a la página de La carta para poder añadir pizzas a su carrito
    - OE7.4.2: El usuario debe poder añadir o quitar unidades a los productos que ha seleccionado previamente
    - OE7.4.3: El usuario debe poder quitar productos del carrito, así como vaciarlo completamente
    - OE7.4.4: Para poder completar el pedido, el usuario debe haber seleccionado al menos una pizza.
  + OE7.5: Completar la compra
    - OE7.5.1 El usuario debe poder revisar la información de su pedido antes de completar la compra
    - OE7.5.2: Para poder completar el pedido, el usuario debe haber introducido su dirección de domicilio en su perfil previamente.
    - OE7.5.3: Dado que el usuario ha completado la compra, debe ser visible en la página de Mis pedidos
* **OG8: Mis pedidos (/dashboard/my-orders)**
  + OE8.1: Debe ser accesible sólo para los usuarios Cliente.
  + OE8.2: Debe mostrar un listado paginado de los pedidos del cliente.
  + OE8.3: Debe mostrar los datos de cada pedido, incluyendo la tienda, dirección de envío, precio total, fecha del pedido, fecha de entrega, y un desglose de las pizzas compradas.
* **OG9: Mi tienda (/dashboard/my-store)**
  + OE9.1. Debe ser accesible sólo para los usuarios Gerente
  + OE9.2 Debe mostrar un formulario para registrar una tienda si no existe ya.
  + OE9.3. En caso de que exista, mostrar un formulario para editar sus datos.
  + OE9.4: El nombre debe ser único, para distinguir a las diferentes tiendas de la cadena.
* **OG10: Gestión de pedidos (/dashboard/manage-orders)**
  + OE10.1. Debe ser accesible sólo para los usuarios Gerente
  + OE10.2. Si el usuario no ha creado tienda todavía, mostrar mensaje de advertencia y redirigir a la página de registro de la tienda.
  + OE10.3. Debe poder ver un listado paginado con los pedidos que ha recibido su tienda
  + OE10.4. Debe poder ver un desglose con las pizzas que componen cada pedido
  + OE10.4. Debe poder marcar como completado un pedido, guardándose la fecha en la que realiza esta acción.
* **OG11: Gestión de usuarios (/dashboard/manage-users)**
  + OE11.1: Debe ser accesible sólo para los usuarios Administrador
  + OE11.2. Debe poder ver un listado paginado con los usuarios que hay registrados en la web
  + OE11.3. Debe poder filtrar el listado utilizando una barra de búsqueda.
  + OE11.4. Debe poder interactuar con los registros de usuarios pendientes de validación, y aceptar o rechazar su solicitud
* **OG12: Gestión de pizzas**
  + OE12.1: Debe ser accesible sólo para los usuarios Administradorç
  + OE12.2. Debe poder ver un listado paginado con las pizzas que hay registradas en la web
  + OE12.3: Debe poder crear pizzas nuevas, editar las existentes o incluso eliminarlas

# 2) RELACIÓN CON PROYECTOS SIMILARES:

Este proyecto web se asemeja a otras páginas web ya existentes, creadas para otras cadenas de restaurantes, como KFC, McDonald 's, Telepizza, etc.

El proyecto se diferencia de los ya existentes al estar completamente personalizado a las necesidades específicas de este cliente.

* **Diseño a medida**: La web no solo debe tener una estética atractiva, sino que también debe reflejar la identidad única de la marca de la pizzería. Esto incluye el uso de una paleta de colores personalizada, tipografía, imágenes y elementos gráficos.
* **Experiencia de usuario adaptada**: La navegación, las funcionalidades y el contenido de la web deben estar diseñados específicamente para satisfacer las necesidades y preferencias de los clientes de la pizzería.
* **Comunicación personalizada**: El tono de voz y el estilo de comunicación utilizados en la web deben ser coherentes con la imagen de marca de la pizzería y resonar con su público objetivo.
* **Escalabilidad**: una de las grandes ventajas de este proyecto para el cliente es que la web está pensada para crecer según sus necesidades. Hay implementaciones futuras planeadas que se ajustarán a sus requerimientos específicos.

# 3) TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS

## 3.1) TECNOLOGÍAS

### Framework y lenguajes

Este proyecto utiliza **Next.js**, un framework de React de código abierto utilizado para construir aplicaciones web modernas de manera eficiente y escalable. El uso de este framework implica el uso de las siguientes tecnologías:

* **React**: es una librería de JavaScript de código abierto utilizada para construir interfaces de usuario interactivas y dinámicas en aplicaciones web
* **TypeScript**: es un lenguaje de programación de código abierto que se basa en JavaScript y agrega tipos estáticos opcionales. Utilizado principalmente para agregar interactividad y dinamismo a las páginas web.
* **JSX**: es una extensión de sintaxis JavaScript utilizada en React que permite escribir código HTML dentro de JavaScript. TypeScript, al ser un superconjunto de JavaScript, también admite esta sintaxis.

### Base de datos

* **PostgreSQL**: es un sistema de gestión de bases de datos relacional de código abierto. También está alojada en Vercel.

### Librerías

Para la lista completa de dependencias del proyecto, consultar el archivo /package.json. A continuación se incluyen las principales librerías utilizadas en este proyecto.

* **Material UI**: es una librería de componentes de interfaz de usuario para React, basada en el diseño de Material Design. Utiliza **Emotion** por defecto para crear estilos.
* **Auth.js**: es una librería para manejar autenticación en aplicaciones JavaScript, proporcionando flujos de autenticación, gestión de usuarios e integración con proveedores de identidad.
* **Formik**: es una librería de gestión de formularios en React que simplifica la validación, el manejo de estados y envío de datos.
* **Yup:** es una libreríapara la validación de datos de formularios, utilizada en conjunto con Formik.
* **Zod**: es otra librería de validación de datos. Es útil para la validación de datos del lado del servidor
* **ESLint**: es una herramienta de linting para JavaScript que ayuda a identificar y corregir errores de código automáticamente.
* **Prettier**: es una herramienta de formateo de código, que ayuda a que el proyecto sea más legible.
* **Bcrypt:** es una librería para encriptar contraseñas en aplicaciones, proporcionando funciones de hashing seguro y verificación de contraseñas para proteger datos sensibles.
* **Dotenv**: es una librería que carga variables de entorno desde un archivo .env en aplicaciones Node.js, facilitando la gestión de configuraciones sensibles y credenciales.

## 3.2) HERRAMIENTAS

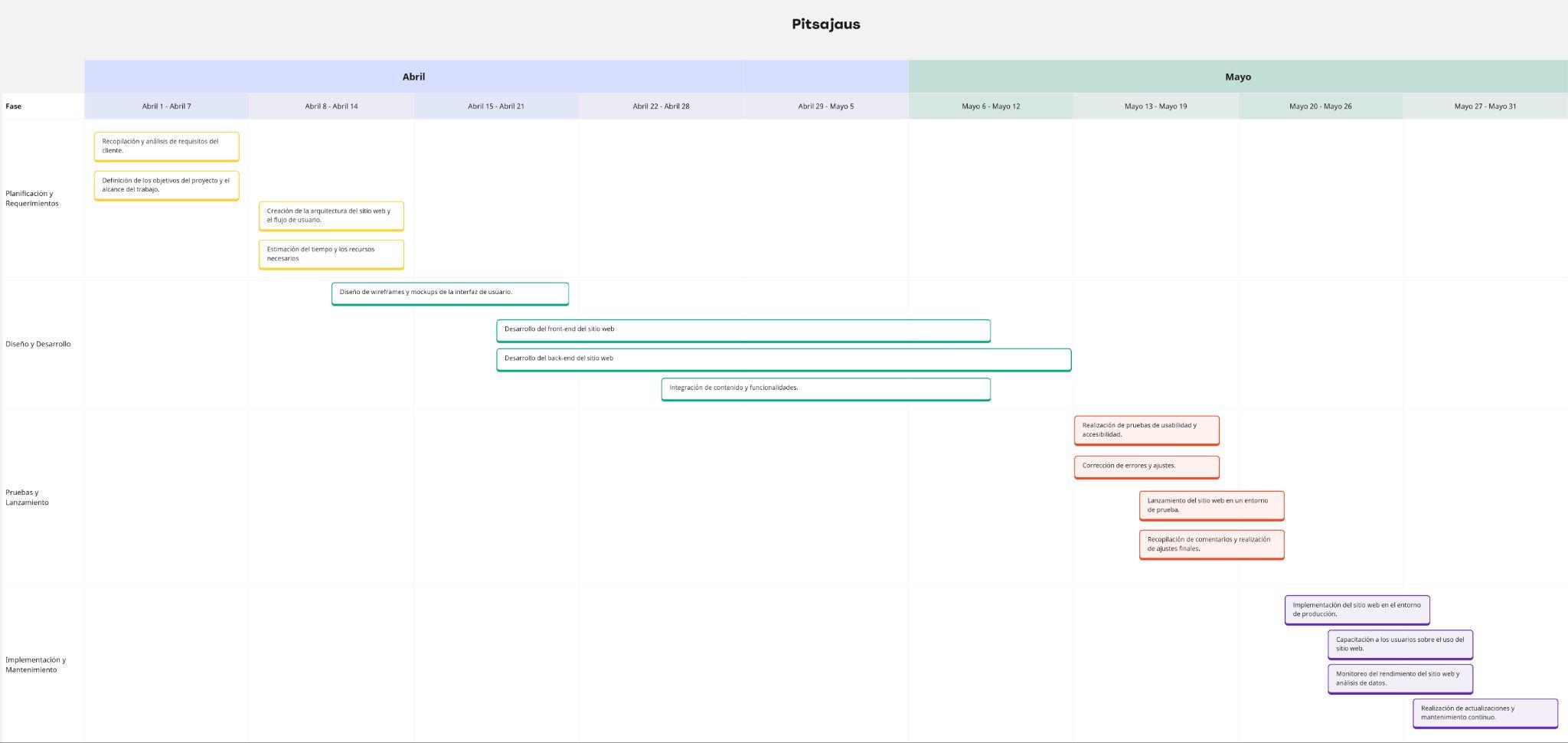
* **Windows 11:** es el sistema operativo usado
* **Visual Studio Code** (versión 1.87.2). es un IDE gratuito y de código abierto, conocido por su velocidad, extensibilidad y amplia gama de características. Se han utilizado las siguientes extensiones:

1. **ESLint**: es una extensión de linting para JavaScript que ayuda a identificar y corregir errores, así como a mantener un código limpio y consistente.
2. **Prettier**: es una extensión para formatear mejor el código

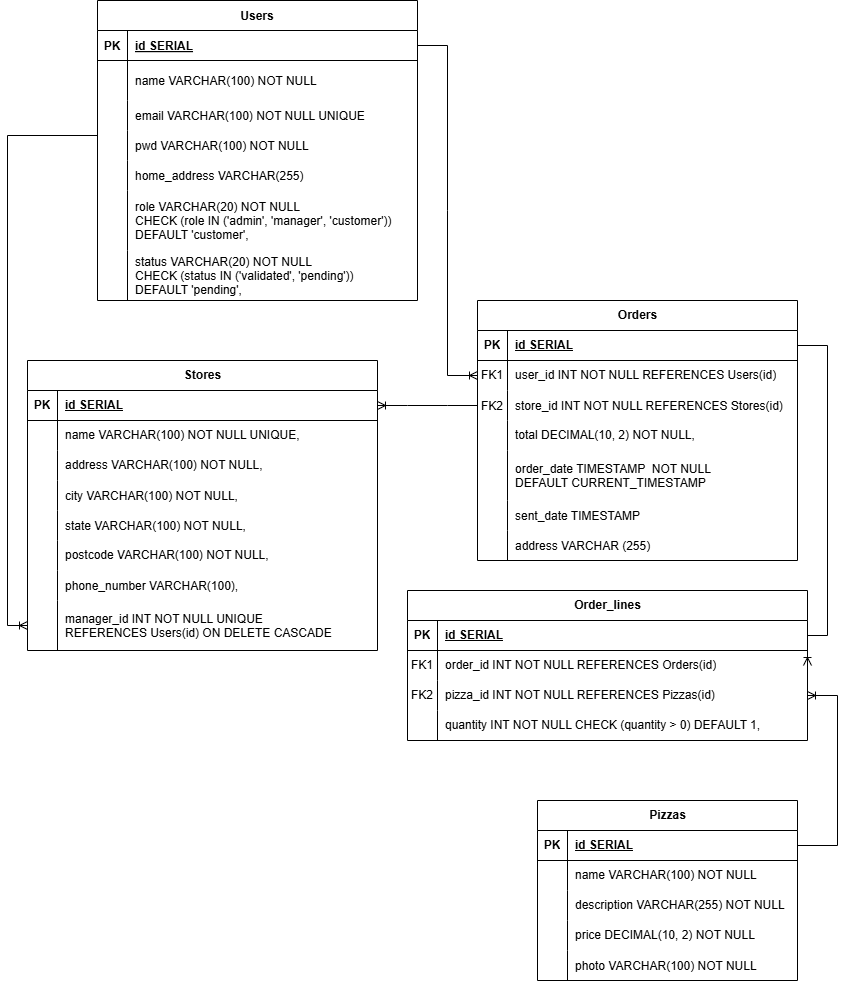
* **Node.js** (versión 20.11.1): es un entorno de ejecución de JavaScript de código abierto y basado en el motor V8 de Google Chrome. Permite ejecutar código JavaScript fuera del navegador, en el lado del servidor. Utilizo **npm** para gestionar los paquetes.
* **Git** (versión 2.44.0.windows.1): es un sistema de control de versiones. Es necesario conocer algunos comandos de terminal.
* **Github**: es una plataforma de desarrollo colaborativo basada en la nube que utiliza el sistema de control de versiones Git.
* **GitHub Desktop**: es una aplicación de escritorio diseñada para facilitar el flujo de trabajo de utilizando GitHub. Proporciona una interfaz gráfica de usuario intuitiva
* **Vercel**: es una plataforma de despliegue y alojamiento para aplicaciones web, optimizada para proyectos frontend y serverless, ofreciendo despliegue continuo y escalabilidad automática. Utilizada en este proyecto también para el alojamiento de la base de datos.

# 4) PLANIFICACIÓN TEMPORAL

A continuación se adjunta la estimación por horas dedicadas al proyecto mediante diagrama de Gantt:



# 5) DESARROLLO DEL PROYECTO



# 6) CONCLUSIONES

Habiendo finalizado este proyecto de desarrollo web, me gustaría reflexionar sobre los objetivos propuestos y los resultados obtenidos. El propósito principal era crear una plataforma funcional y atractiva que atendiera a las necesidades específicas del cliente, y en gran medida, he logrado cumplir con esta meta.

No obstante, ha habido algunas posibles funcionalidades que fueron sugeridas pero nunca se llegaron a implementar.

* **La posibilidad de crear pizzas personalizadas**. Esta idea se desechó muy pronto, ya que añadía una complejidad que se consideró innecesaria en su momento..
* **Un formulario de contacto**. Era una funcionalidad más avanzada y no era esencial, por lo que su desarrollo se iba a realizar una vez los demás objetivos fueran completados. Pero debido a limitaciones de tiempo no se llegó a implementar.
* Otras funcionalidades, como la posibilidad de tener **promociones**.

Dicho esto, estoy bastante contento con el resultado del proyecto. Ha sido una experiencia enriquecedora que me ha permitido aprender nuevas tecnologías y herramientas que no conocía antes. El aprendizaje ha sido completamente autodidacta, lo que ha resultado muy desafiante.

# 7) GLOSARIO

A continuación se incluyen algunos términos técnicos incluidos en este documento, así como sus definiciones:

* **MVP**: Producto Viable Mínimo es un producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro.
* **Logueado**: se refiere a un usuario de la web, que ha iniciado sesión.
* **Framework:** es una plataforma de software que proporciona herramientas y librerías predefinidas para facilitar el desarrollo de aplicaciones web de manera eficiente y estructurada.
* **Sitio web**: es una colección de páginas web relacionadas y comunes a un dominio de internet.
* **Página web**: es un documento de la World Wide Web «con dirección propia»
* **Páginas públicas**: páginas web que son visibles sin haber un usuario logueado.
* **Páginas privadas**: páginas web que requieren que haya un usuario logueado con los permisos necesarios para ver la página.:
* **Sesión**: es un período de interacción entre un usuario y una aplicación web, iniciado al abrir la web y finalizado al cerrar el navegador o tras un tiempo de inactividad.
* **Renderizado**: es el proceso de convertir el código HTML, CSS y JavaScript en contenido visual interactivo que los usuarios ven y utilizan en el navegador
* **SEO**: se refiere a las técnicas y estrategias utilizadas para mejorar la posición de un sitio web en los resultados de búsqueda orgánicos. El objetivo es aumentar la visibilidad del sitio web y atraer más tráfico de usuarios.
* **Enrutamiento**: se refiere al proceso de dirigir a los usuarios a la página correcta en función de la URL que ingresan.
* **Librería**: se refiere a una colección de código reutilizable que facilita el desarrollo de páginas web.
* **Responsivo:** un diseño responsivo se refiere a la capacidad de un sitio web de adaptarse automáticamente a diferentes tamaños de pantalla

# 8) BIBLIOGRAFÍA/WEBGRAFÍA

## Documentación oficial

* [Referencia de Javascript en MDN](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
* [Documentación oficial TypeScript](https://www.typescriptlang.org/docs/)
* [Aprender React](https://react.dev/learn)
* [Documentación oficial React](https://react.dev/reference/react)
* [Tutorial de React en nextjs.org](https://nextjs.org/learn/react-foundations): enseña conceptos básicos de React, y cómo migrar un proyecto React a Next.js.
* [Página oficial de Next.js](https://nextjs.org/)
* [Tutorial de Next.js](https://nextjs.org/learn/dashboard-app)
* [Documentación oficial de Node.js](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)
* [Documentación Material UI](https://mui.com/material-ui/getting-started/)
* [Documentación Formik](https://formik.org/docs/overview)
* [Página de ESLint](https://eslint.org/)
* [Formik con Material UI](https://formik.org/docs/examples/with-material-ui)
* [Documentación oficial de Auth.js](https://authjs.dev/getting-started) (antes conocido como NextAuth.js)

## Otros enlaces de interés

* [Next.js Cheatsheet - handy snippets and tips (gourav.io)](https://gourav.io/blog/nextjs-cheatsheet)
* [Nextjs Authentication](https://nextjs.org/learn/dashboard-app/adding-authentication)
* [Nextjs Authentication 2](https://nextjs.org/docs/app/building-your-application/authentication)
* [Auth.js Credentials Provider](https://authjs.dev/getting-started/authentication/credentials)
* [Auth.js installation](https://authjs.dev/getting-started/installation?framework=next.js)
* [Auth.js and Next.js 14](https://medium.com/@renanleonel/how-to-set-up-nextauth-v5-authentication-with-middleware-and-jest-configuration-in-next-js-14-ca3e64bfb7d5)
* [(YT) Auth.js and Next.js](https://www.youtube.com/watch?v=qDRQ2EaWsNM)
* [(YT) Next.js Credentials Provider](https://www.youtube.com/watch?v=splh0ZIo-s4)
* [Next.js 14 Tutorial Part 4: File Upload with Server Actions (strapi.io)](https://strapi.io/blog/epic-next-js-14-tutorial-part-5-file-upload-using-server-actions)
* [How to Load Data from a File in Next.js (vercel.com)](https://vercel.com/guides/loading-static-file-nextjs-api-route)
* [Server Actions - File Upload and parsing : r/nextjs (reddit.com)](https://www.reddit.com/r/nextjs/comments/139jbsf/server_actions_file_upload_and_parsing/)
* [Server Uploads with Vercel Blob](https://vercel.com/docs/storage/vercel-blob/server-upload)
* [Next.js File Uploads: Server-Side Solutions | Pro NextJS](https://www.pronextjs.dev/next-js-file-uploads-server-side-solutions)

# ANEXO A. Manual de instalación

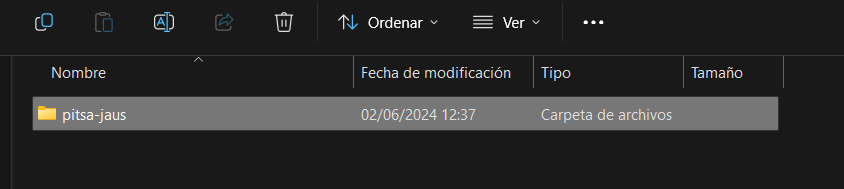
En este manual se explica cómo se puede instalar este proyecto en un entorno local. Para una guía completa y comprensiva, consultar la [documentación oficial](https://nextjs.org/learn/dashboard-app)

## Prerrequisitos

* Sistema operativo: Windows (también WSL), Linux o macOS
* Node.js 18.17.0 o posterior. Descárgalo [aquí](https://nodejs.org/en)
* npm (node package manager): puede ser instalado junto a Node.js
* Cuenta de [Github](https://github.com/join/) y de [Vercel](https://vercel.com/signup) (requeridas para el despliegue de la web y el alojamiento de la base de datos en Vercel)
* Copia del proyecto (adjunta como zip)

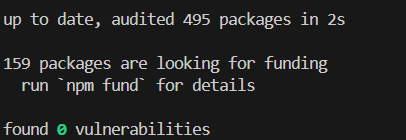
## Instalación del proyecto

1. Extraer el zip en el directorio deseado:



1. En una terminal (en este caso powershell), ejecuta “npm install”. Instalará todas las dependencias definidas en package.json.





* Ejecuta “npm run dev”



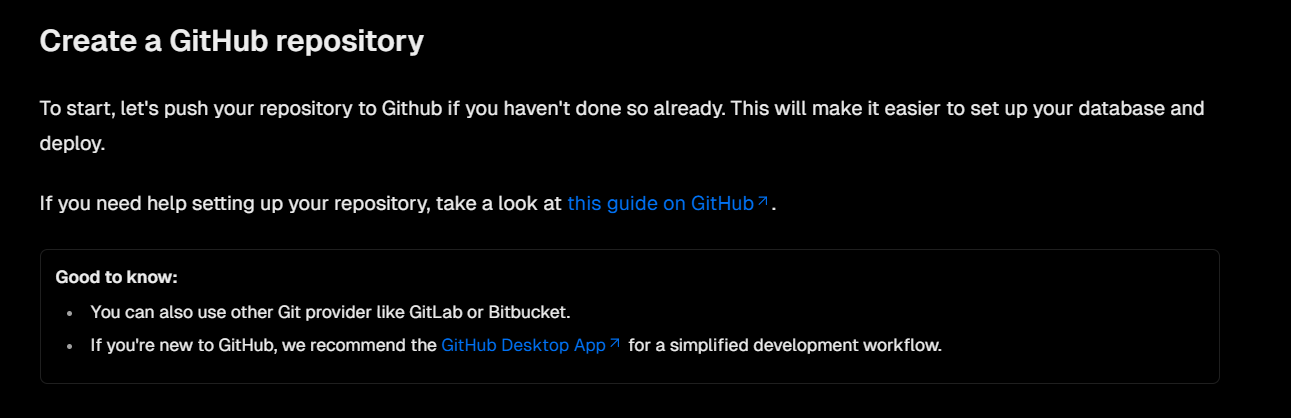


* Abre <http://localhost:3000> en tu navegador para ver el proyecto

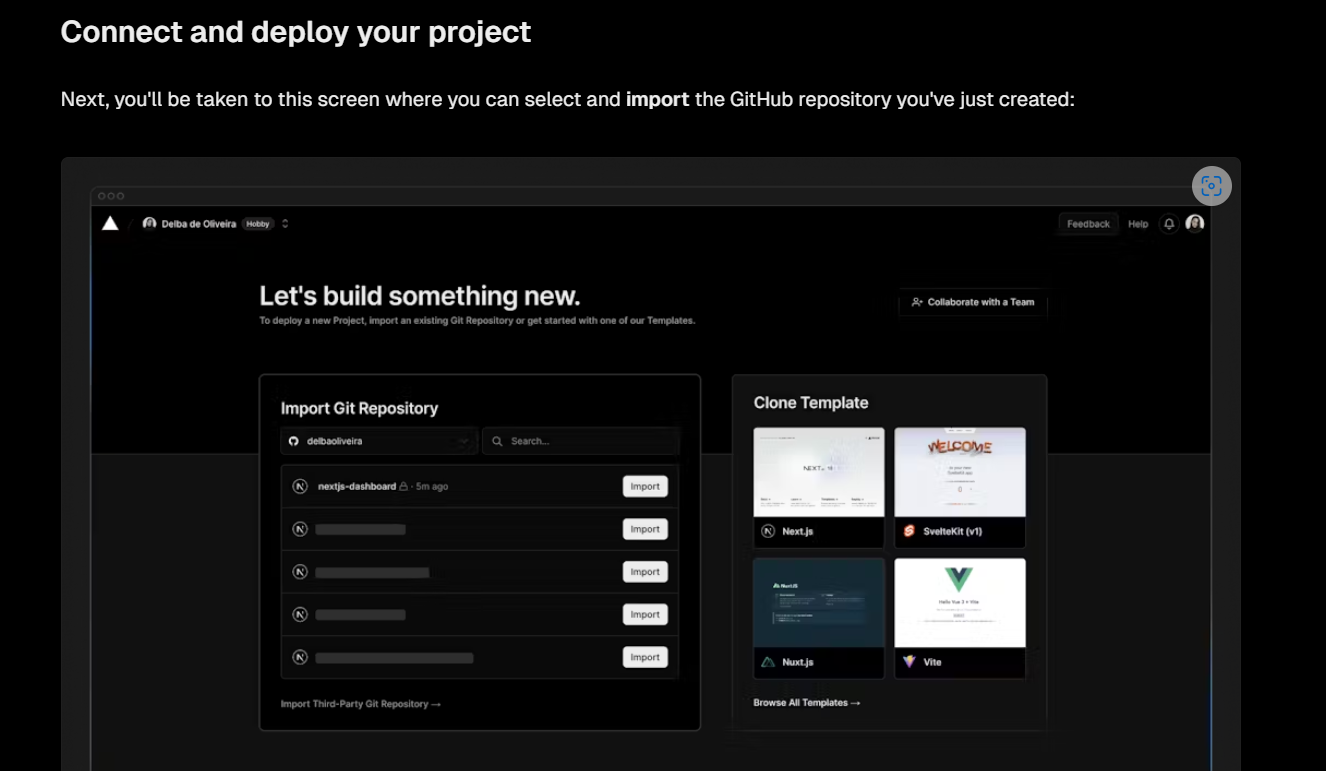


## Despliegue de la web

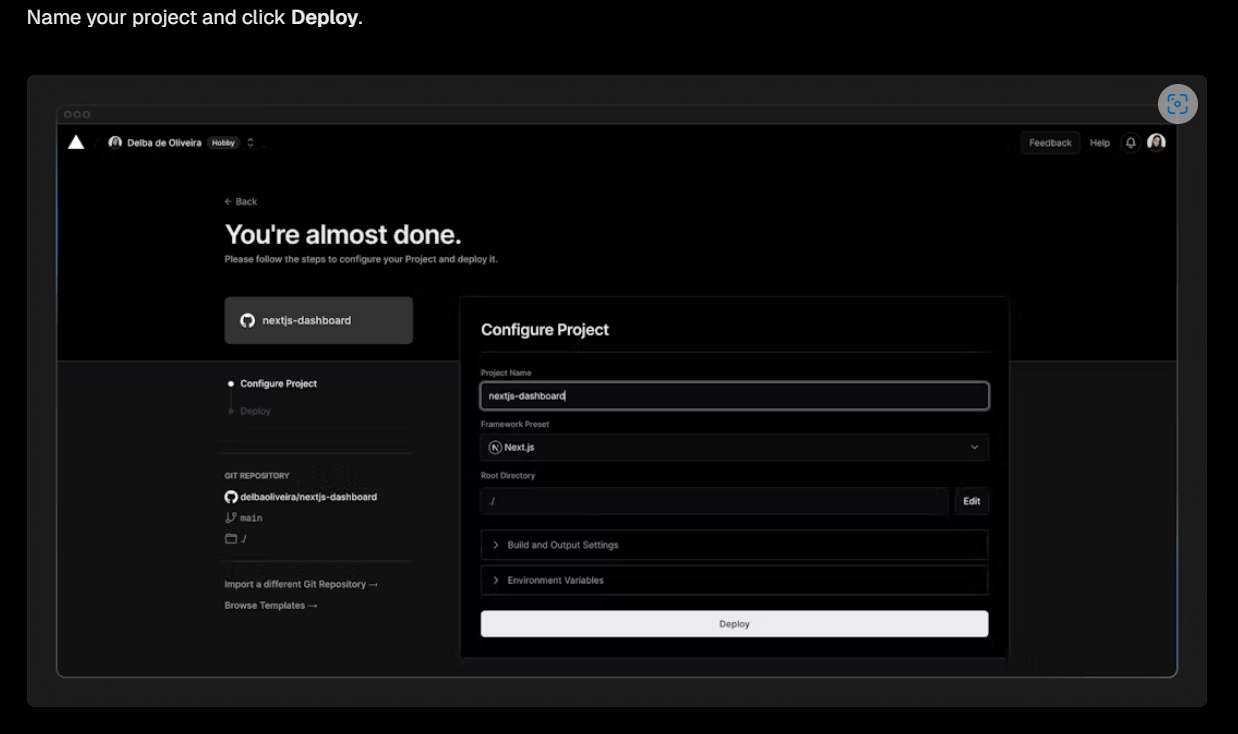
1. Para este paso, es obligatorio subir el repositorio a Github. En esta [guía](https://help.github.com/en/github/getting-started-with-github/create-a-repo) podrás encontrar los pasos a seguir.



1. También es necesario crear una cuenta de [Vercel](https://vercel.com/signup). Selecciona el plan gratuito “Hobby”.
2. Conecta las cuenta de Vercel con la de Github. Entonces, podrás importar el repositorio que acabas de crear



1. Despliégalo:

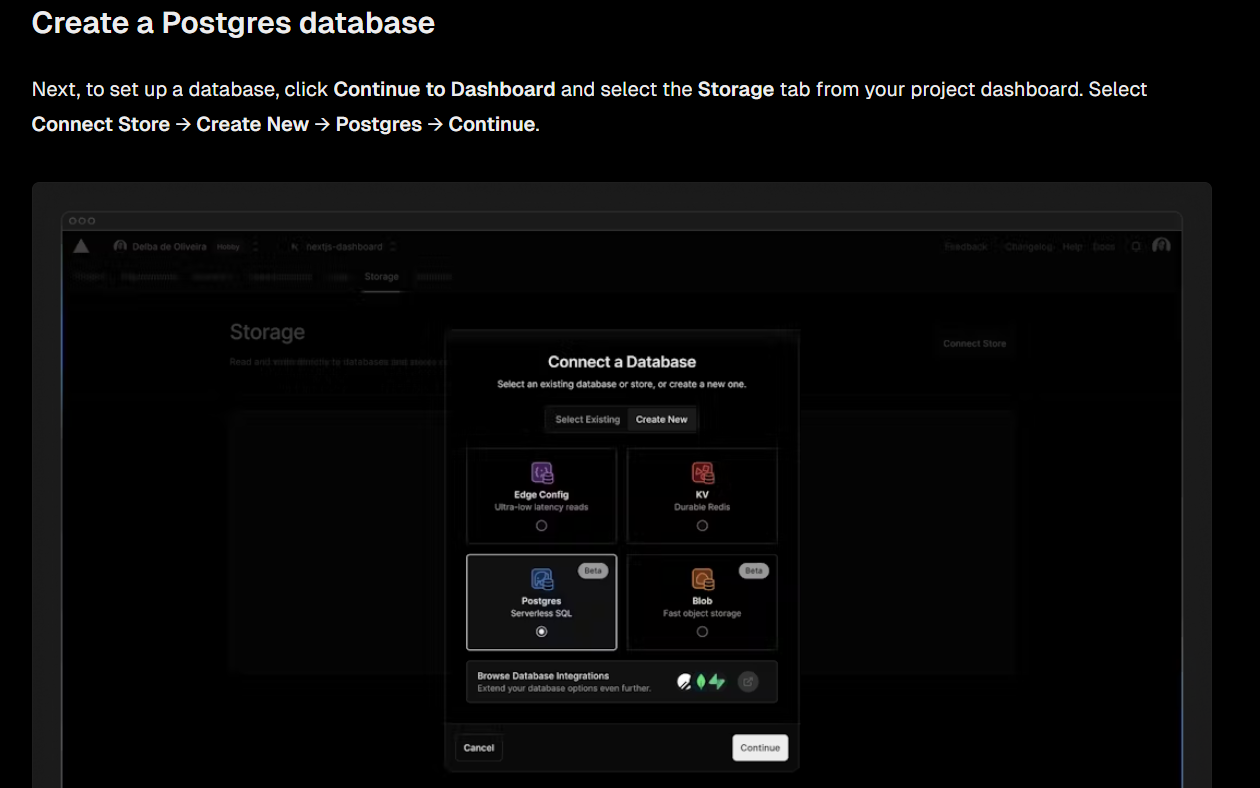




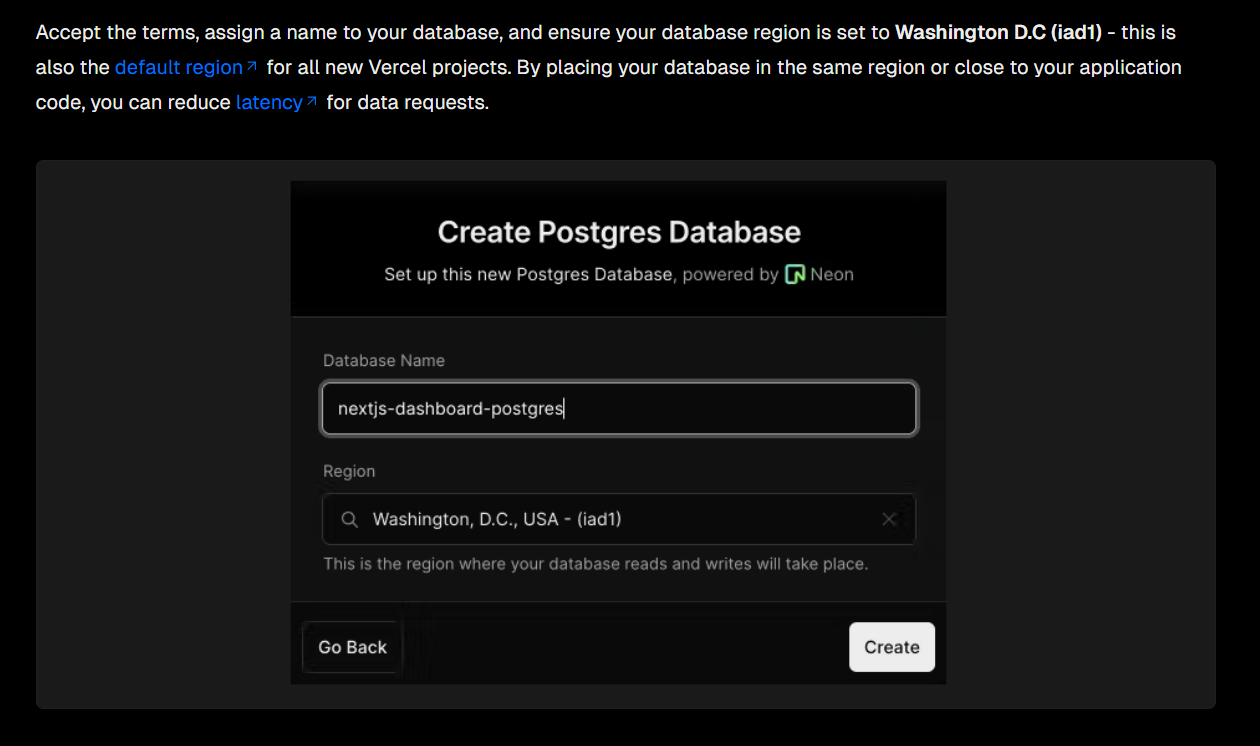
1. Al conectar la cuenta de Github, cuando la rama main se actualice, Vercel desplegará esos cambios automáticamente, sin configuración adicional por parte del usuario.

## Configuración de la base de datos

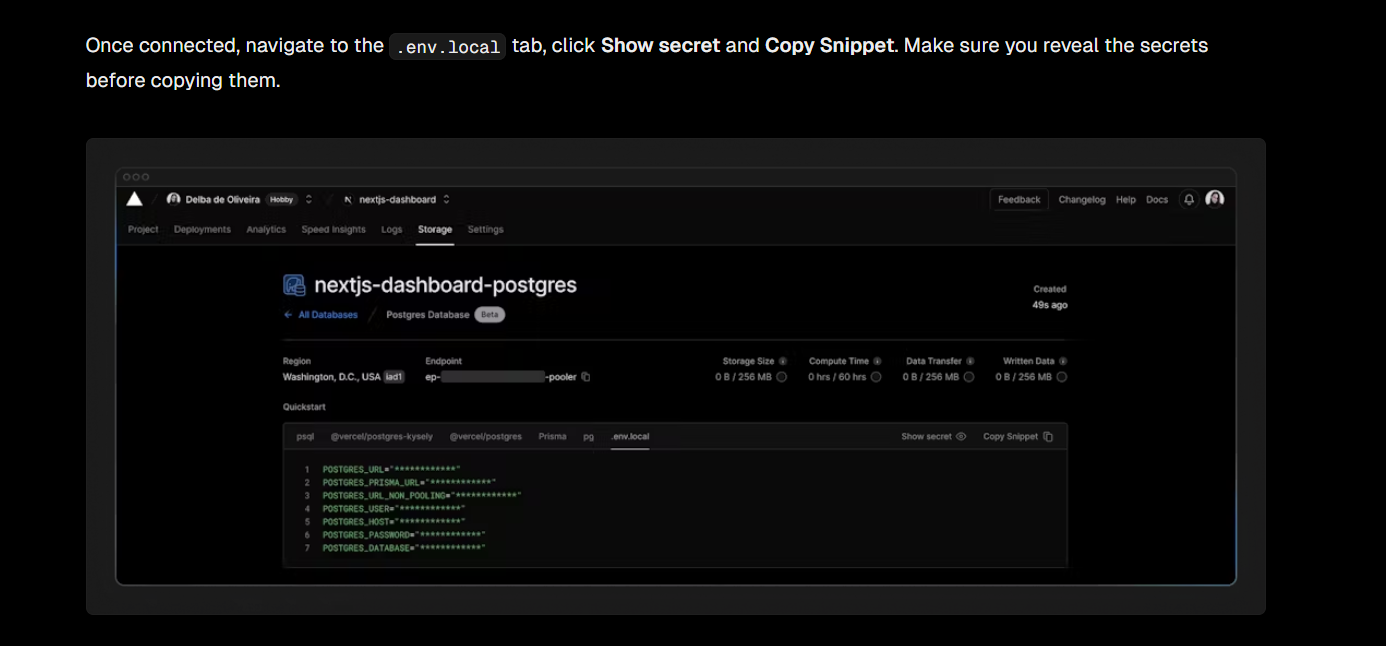
1. Para crear una base de datos Postgres, dentro del dashboard en [Vercel](https://vercel.com/), selecciona la pestaña de Almacenamiento. Entonces Conectar Store -> Crear Nueva -> Postrgres -> Continuar



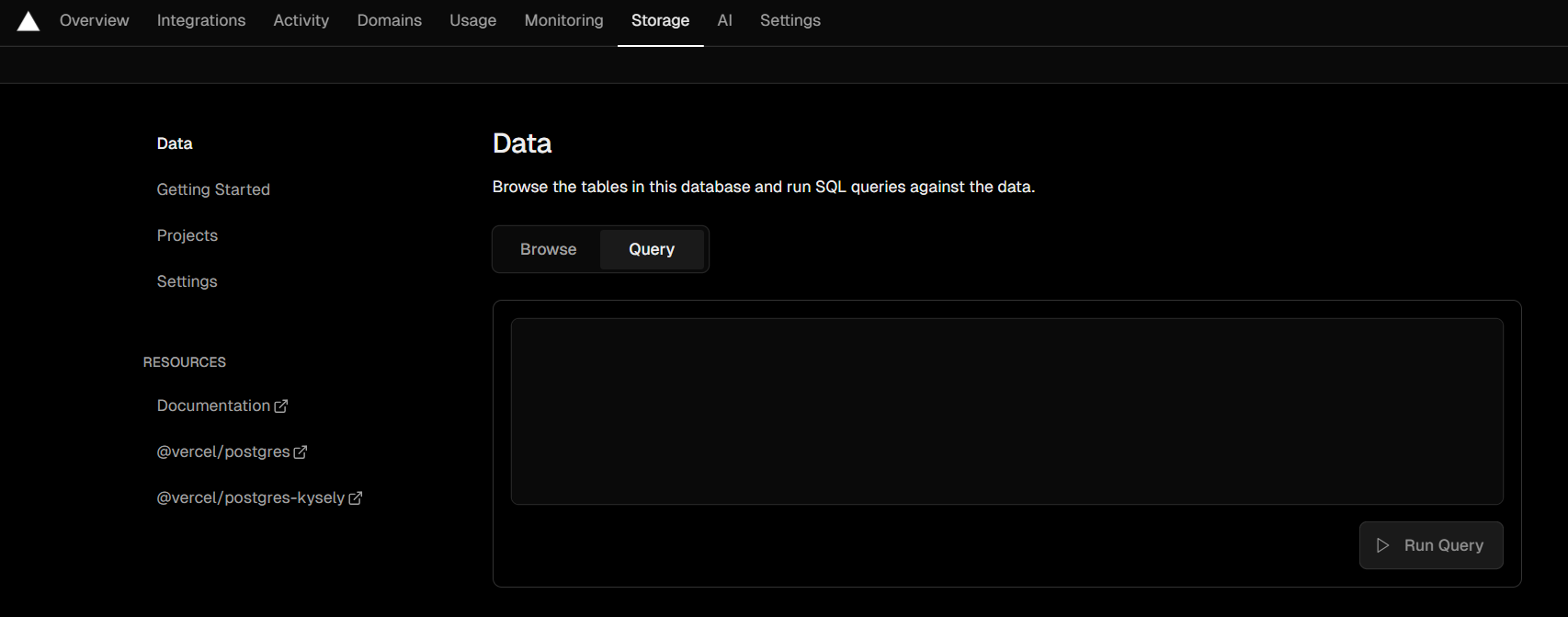
1. Acepta los términos y dale un nombre a la base de datos. Asegúrate de que la región es Washington DC (iad1). Es la región por defecto. Esto mejorará la velocidad de las peticiones.



1. Una vez creada, ve a la pestaña “.env.local”. Pulsa “Mostrar secreto” y “Copia Snippet”.



1. En la raíz del proyecto, crea un archivo .env.local y pega el contenido ahí
2. Para poder utilizar ciertas funcionalidades, como el registro de usuarios gerente y administrador, es necesario que haya al menos 1 usuario de tipo administrador ya registrado. Esto puede lograrse insertando directamente en la base de datos:



1. O utilizando el comando “npm run seed”, que ejecutará el script localizado en /src/scripts/seed.js, insertando algunos datos iniciales.

# ANEXO B. Manual de usuario

Hay 4 tipos de usuarios que pueden acceder a la web. A continuación se describen los posibles casos de uso:

## Invitado (sin registro):

* Este es el punto de entrada de la web



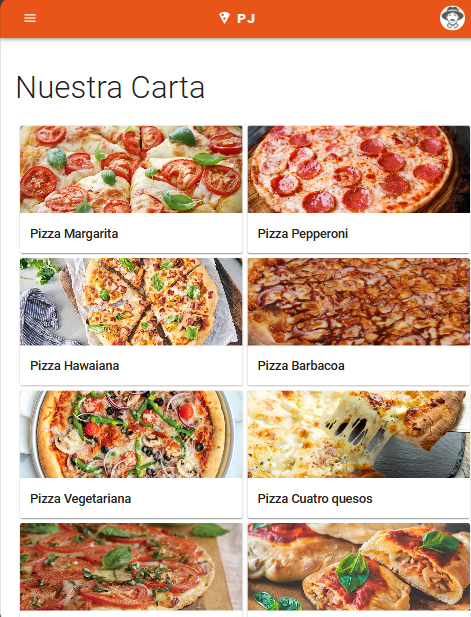
* Hay un menú superior que permite navegar a las principales páginas públicas.



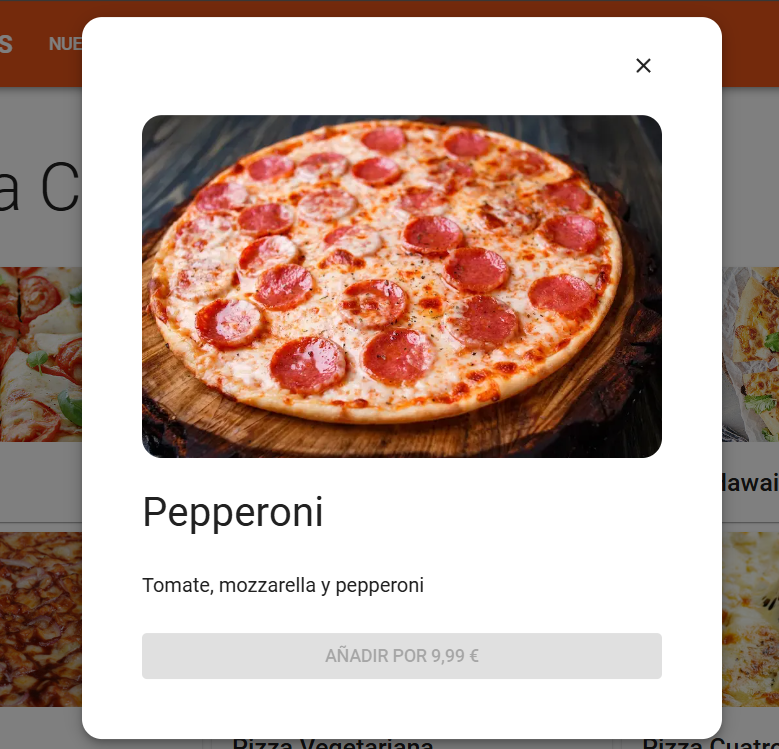
* Puede pulsar en este enlace para acceder al menú.



* Verá este listado:



* Al pulsar en una de las pizzas, verá en detalle sus datos. El botón de añadir carrito permanece desactivado hasta que se registre como cliente.



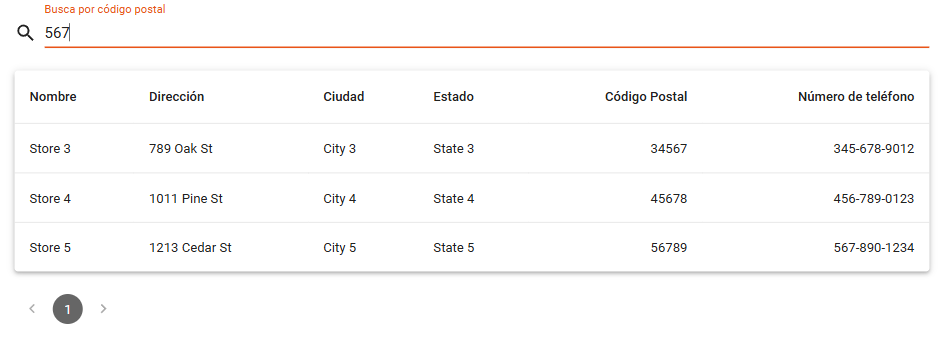
* Pulsa aquí para ver el listado de las tiendas PitsaJaus



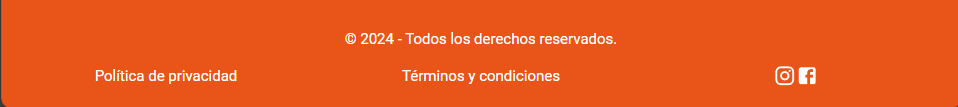
* Verá esta tabla



* Puede filtrar por código postal para encontrar una pizzería que se encuentre cerca



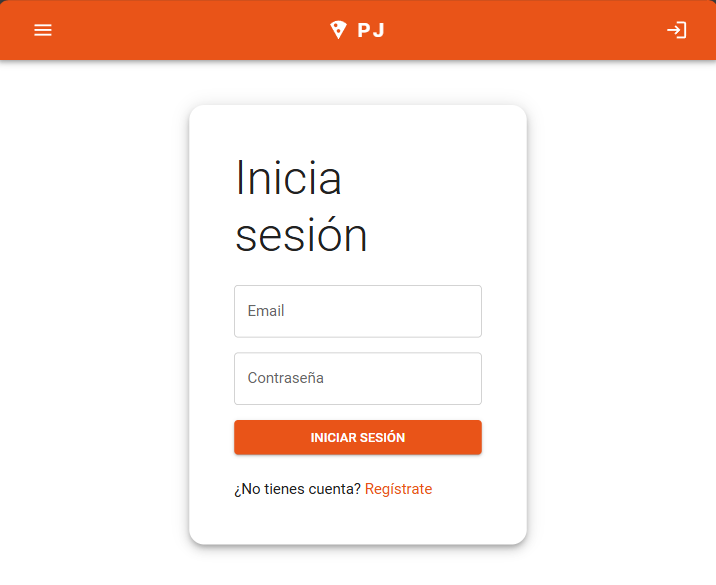
* El footer, que es común a todas las páginas de la web, contiene enlaces a las páginas de Política de privacidad, Términos y condiciones, y las redes sociales de la empresa



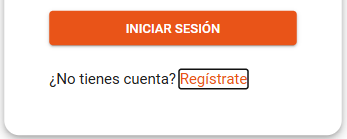
* Puede pulsar este enlace para iniciar sesión



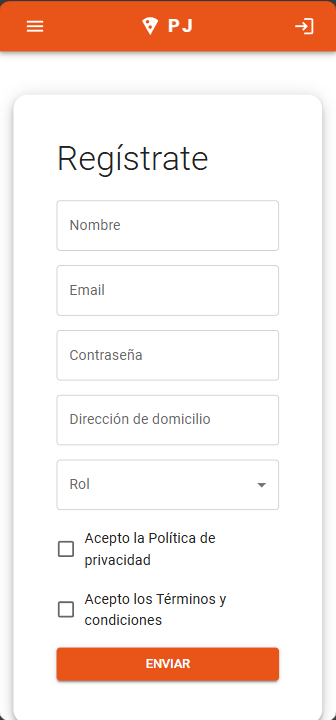
* Verá esta pantalla



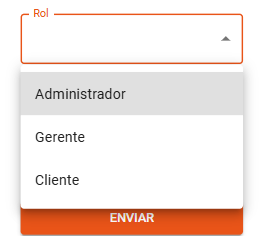
* Puede pulsar en este enlace para registrarse



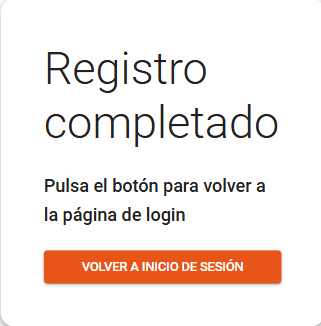
* Completa el formulario



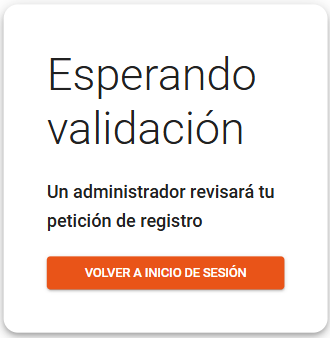
* Hay 3 roles que puede elegir.



* Si se registra como Cliente, verá esta pantalla

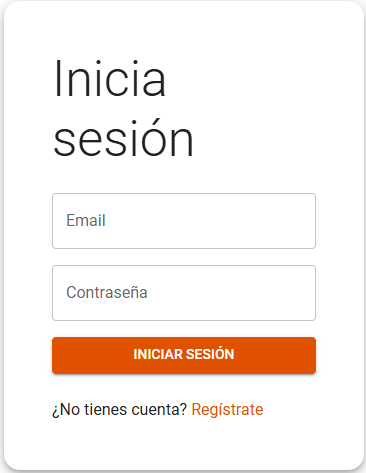


* Si quiere ser Gerente o Administrador, deberá ser validado por un administrador.

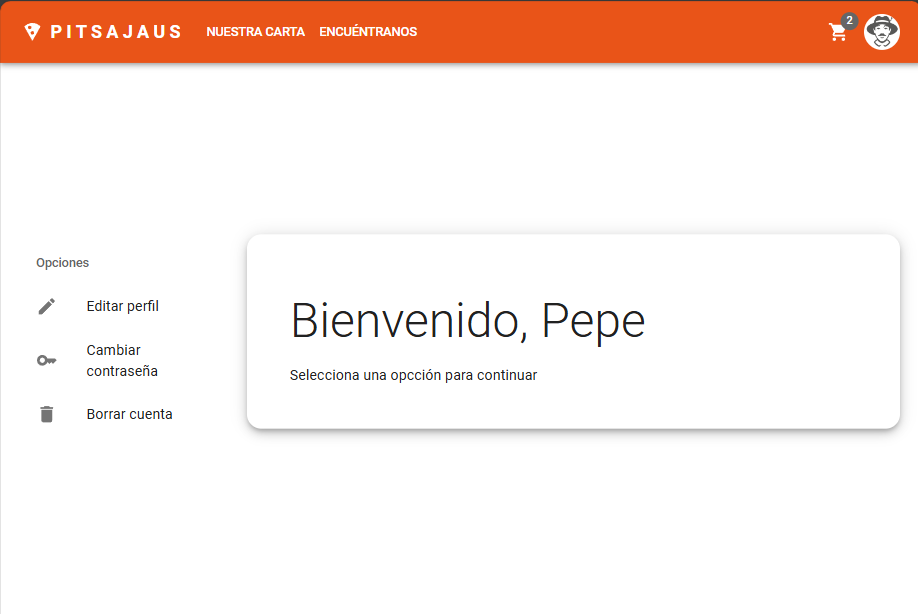


## Usuario general

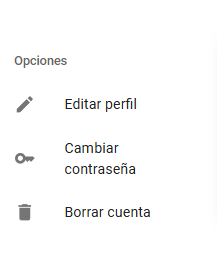
* Inicia sesión utilizando credenciales válidas



* Todos los usuarios que hayan iniciado sesión (cliente, gerente, administrador), verán esta pantalla de inicio

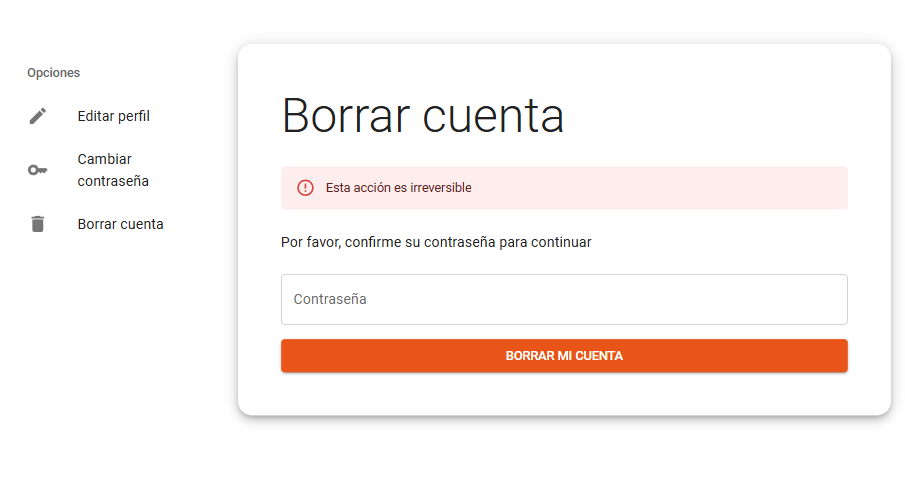


* Pueden utilizar este menú para editar sus datos, su contraseña, o borrar su perfil



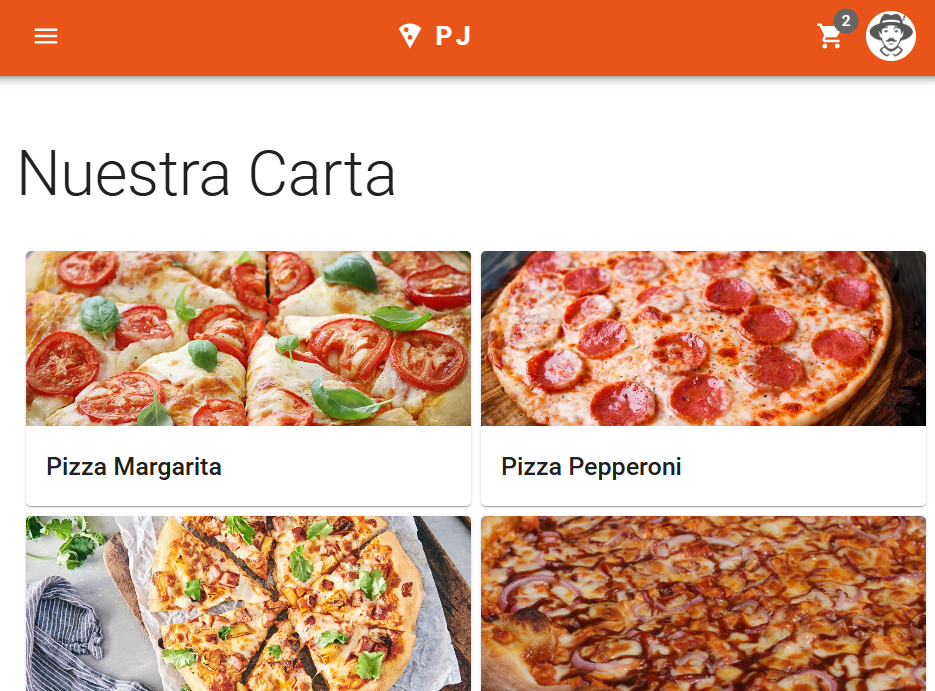


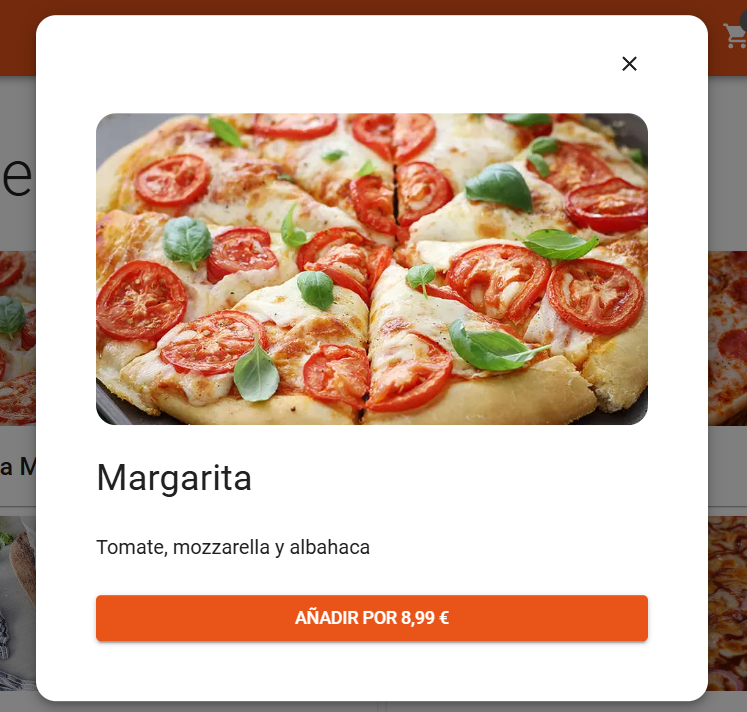




## Cliente:

* Un cliente puede añadir pizzas al carrito desde La carta





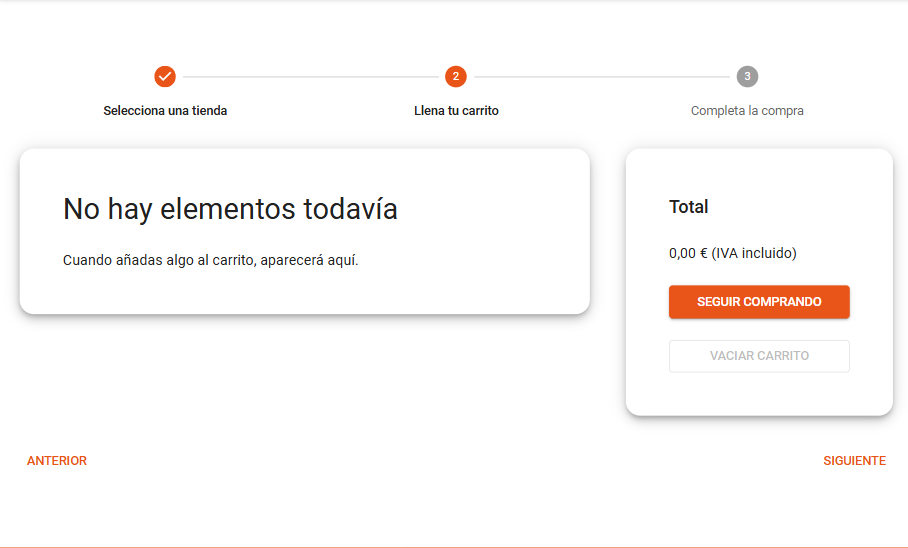
* El carrito es accesible desde el menú superior

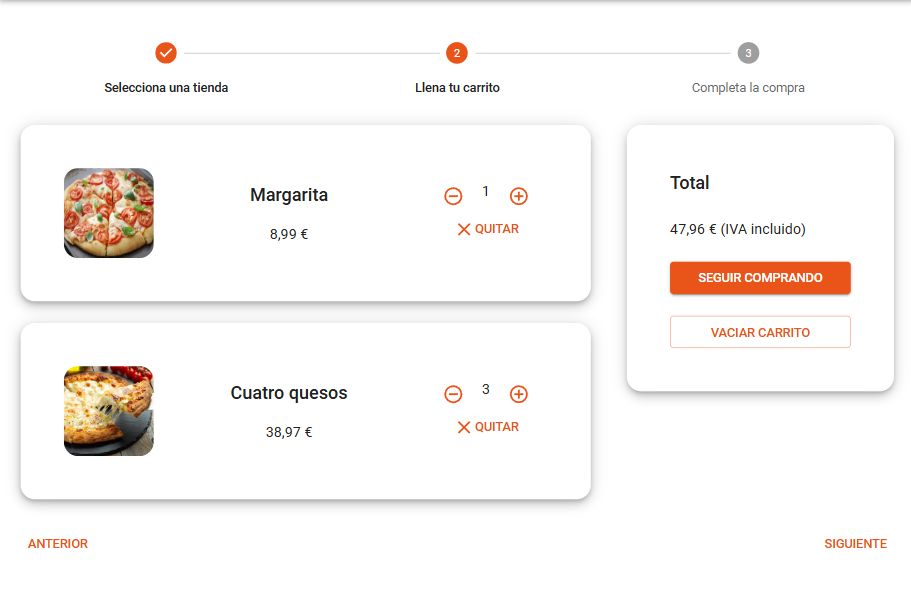


* Debe seleccionar una tienda

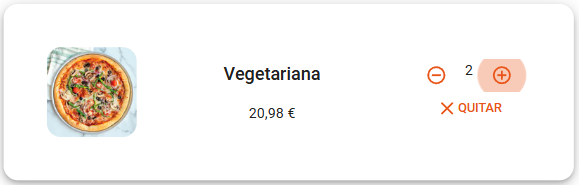


* Debe seleccionar al menos una pizza para poder completar el pedido

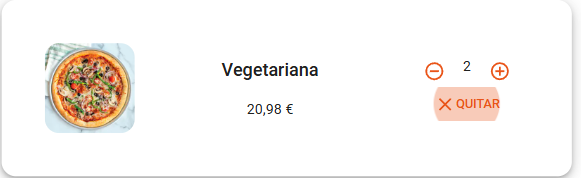




* Puede usar los botones con los símbolos de + y - para añadir o quitar unidades a la pizza



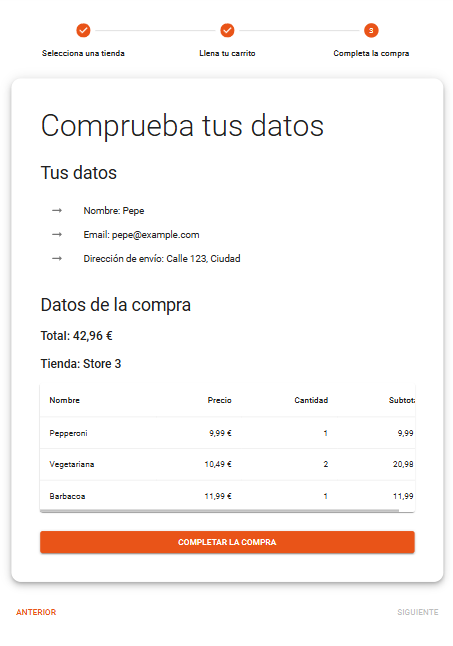
* O incluso quitar todas las unidades con un solo clic



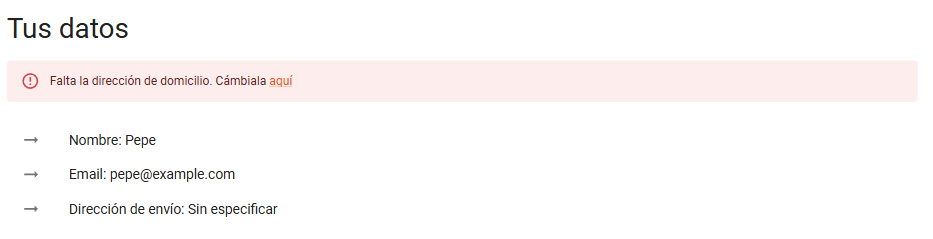
* Aquí puede ver el total de su pedido, vaciar el carrito, o navegar a la página de La Carta



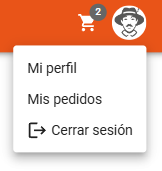
* Tendrá la oportunidad de revisar su pedido antes de completarlo

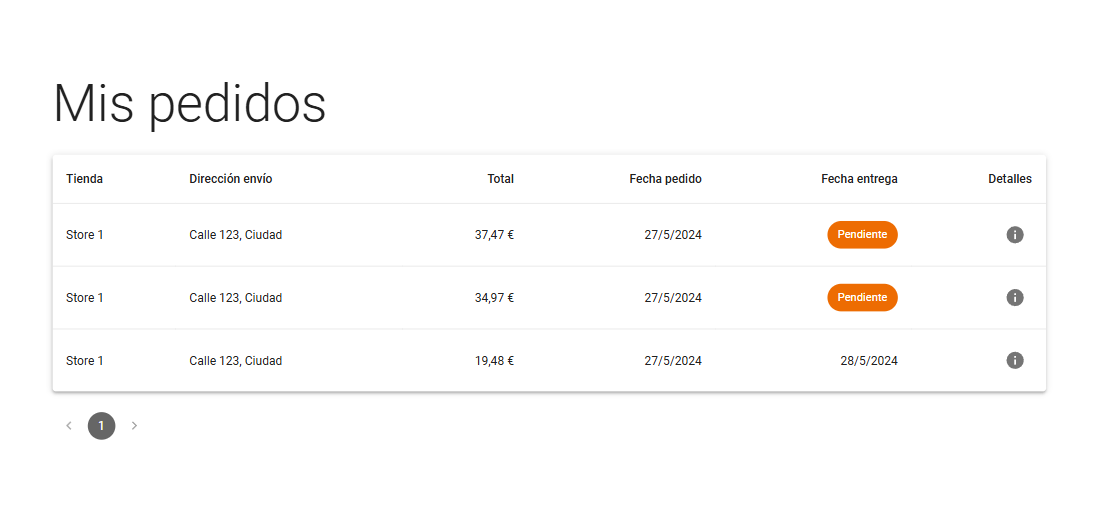


* Recuerde especificar la dirección de envío

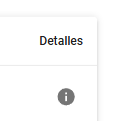


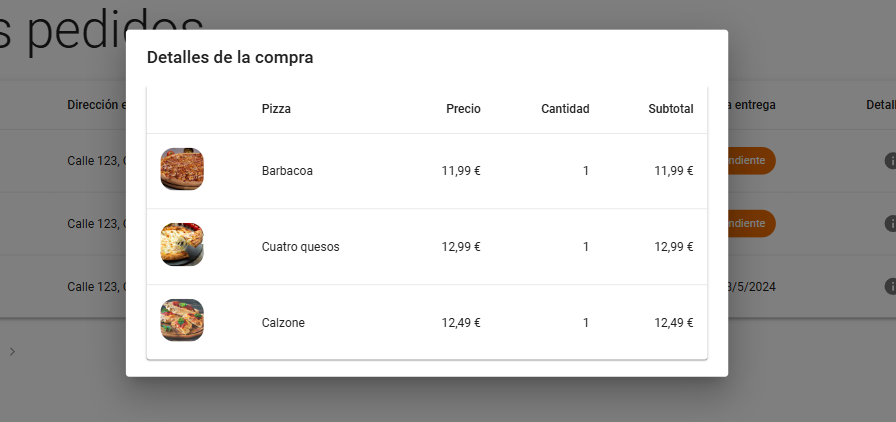
* Desde el menú de usuario puede revisar sus pedidos:





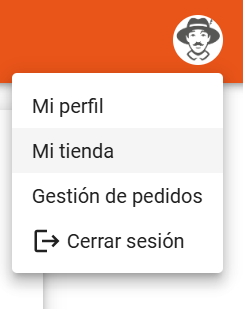
* Pulse en el icono de información para ver los detalles de la compra

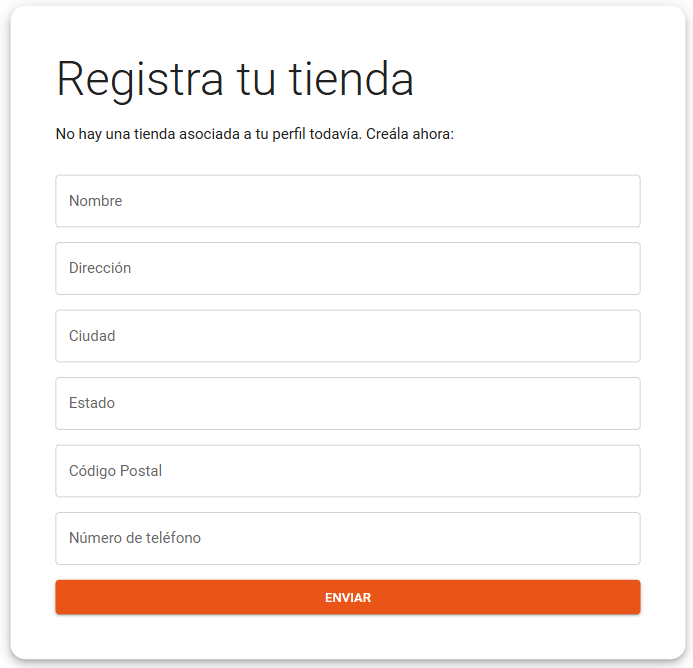


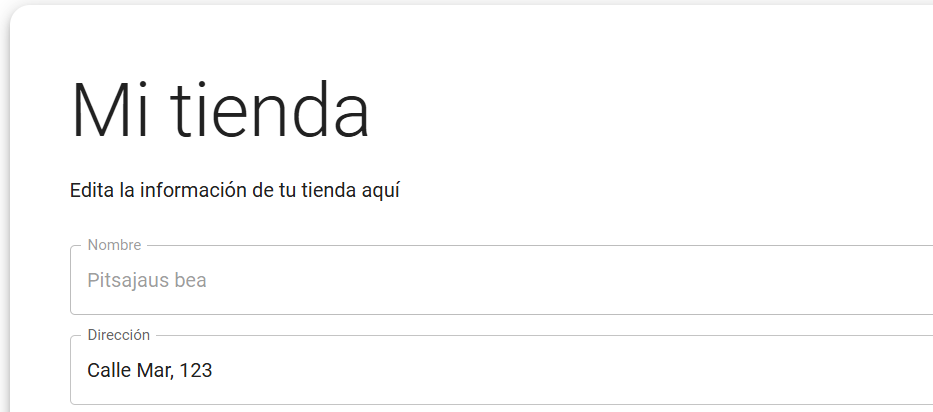


## Gerente:

* Los gerentes podrán registrar una tienda nueva, o editar sus datos si ya existe







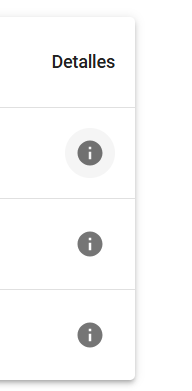
* Para poder gestionar sus pedidos, es obligatorio que haya creado una tienda primero:



* Una vez la haya creado, podrá ver los pedidos que hayan llegado a su tienda

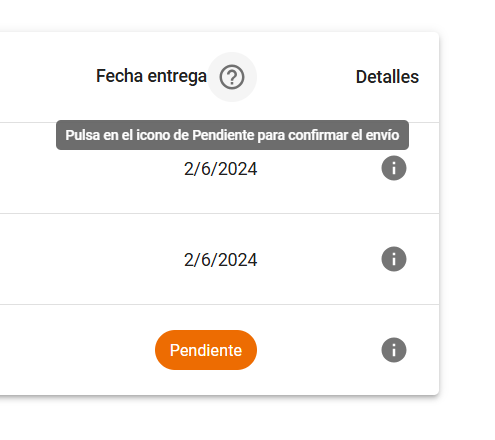


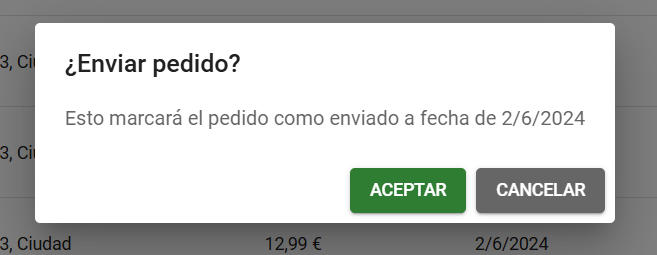
* Pulsa en el icono de detalles para ver los detalles de la compra





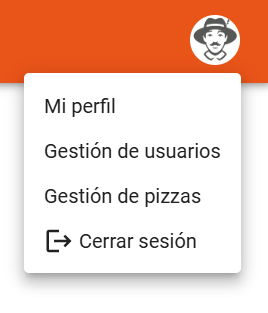
* Para marcar un envío como completado, pulsar en el icono de Pendiente:



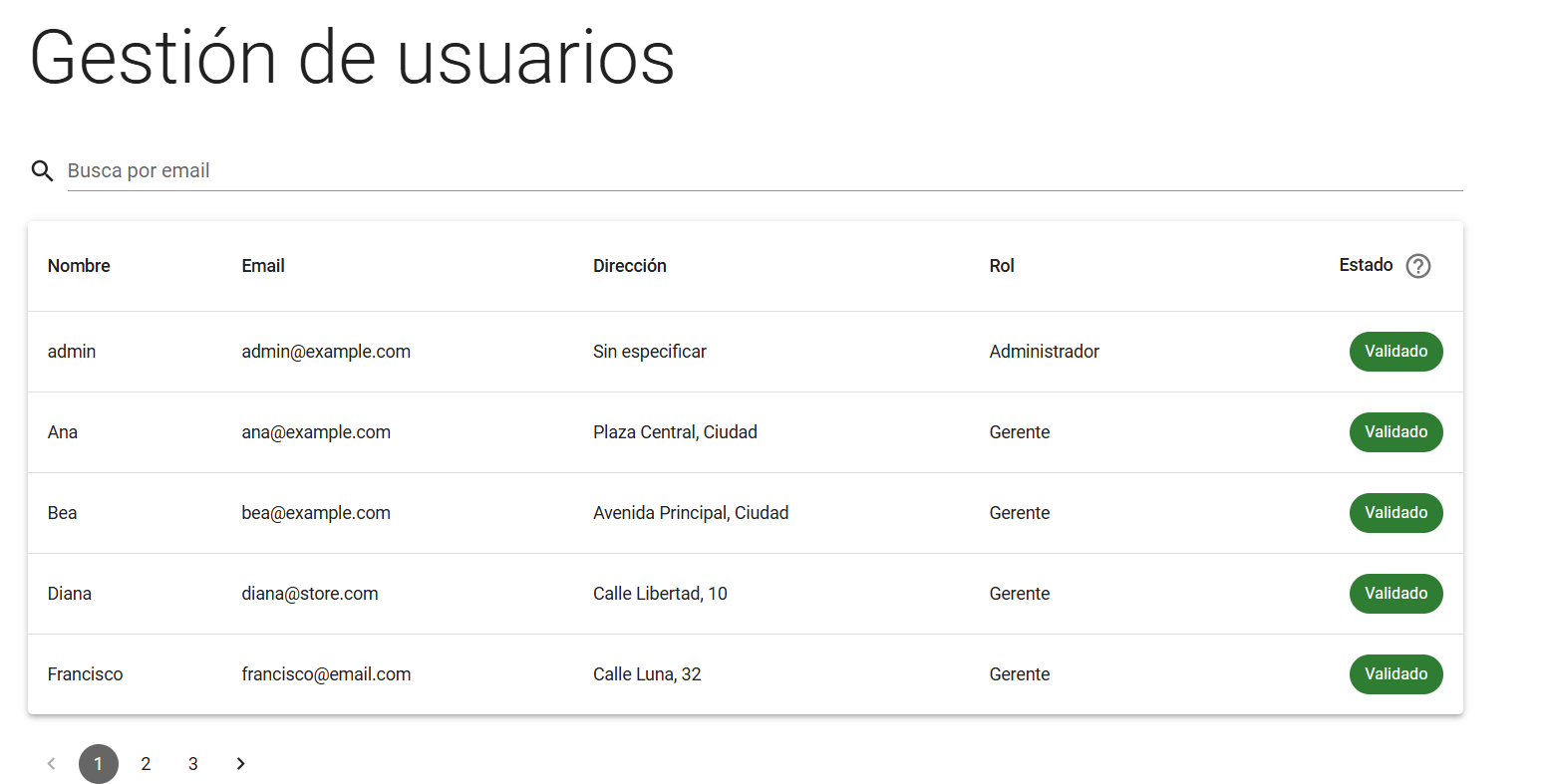


## Administrador:

* Un administrador podrá gestionar usuarios y pizzas:

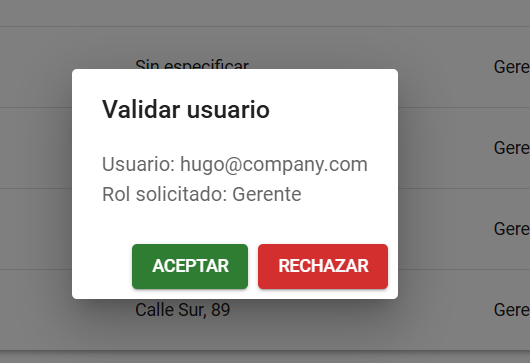


* En gestión de usuarios podrá ver un listado de todos los usuarios que hay registrados:

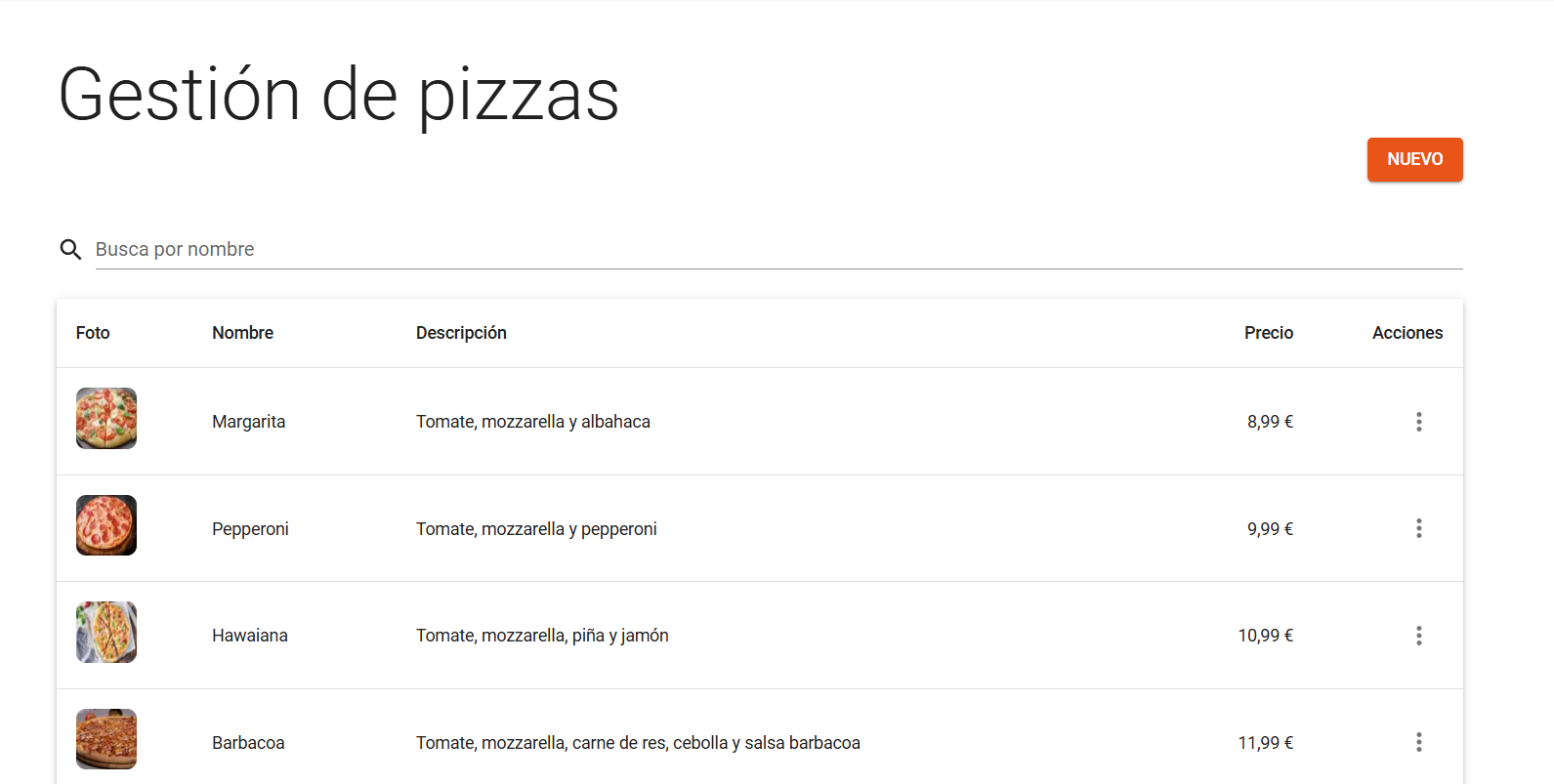


* Pulsa en el icono de Pendiente para aceptar o rechazar la solicitud de validación (las peticiones para registrar Gerente o Administrador lo requieren)

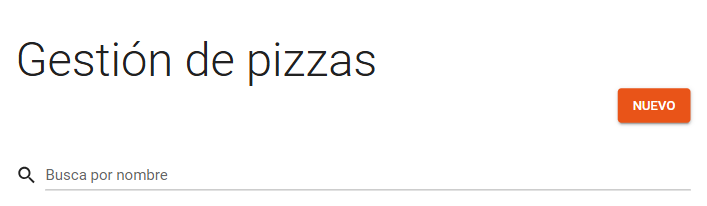


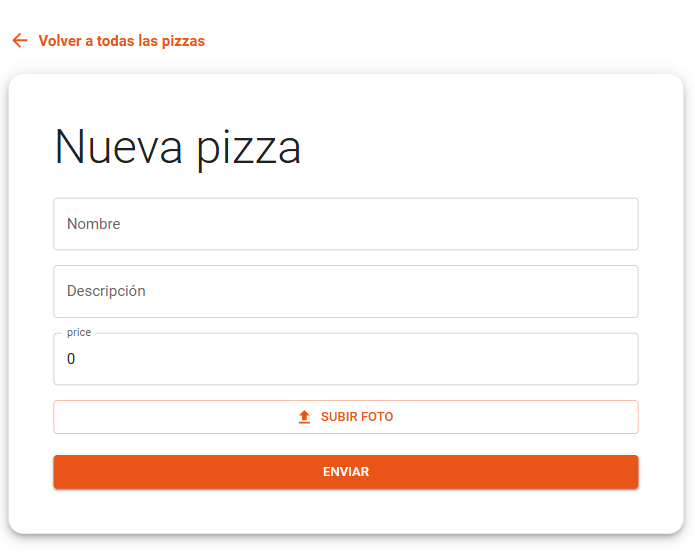


* En la gestión de pizzas podrá ver un listado completo de las pizzas que hay en la carta:

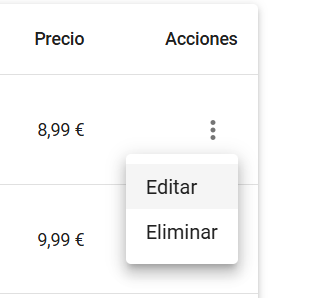


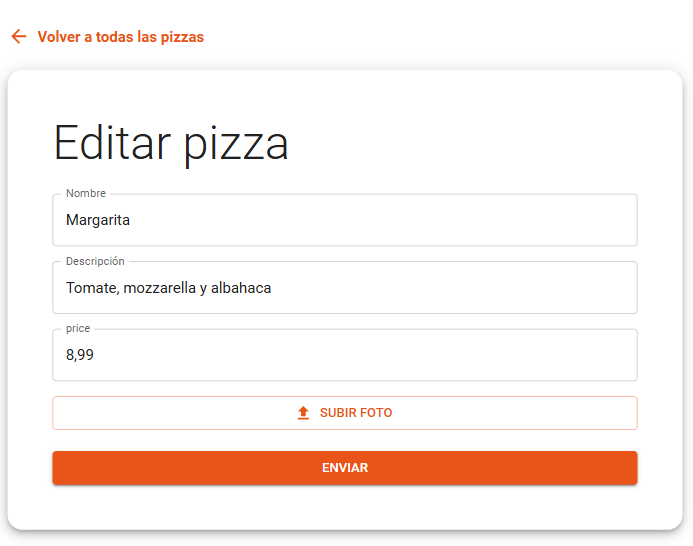
* Podrá añadir pizzas:





* O editar los datos de las existentes





* También podrá eliminarlas:

