***STACK A UTILIZAR :***

Next js para el diseño y el frontend.

Bootstrap (Por el sistema de grid más que nada) / CSS

Tener en cuenta que el producto debe ser lo mas liviano posible porque no es una plataforma con animaciones si no que es un medio para comprar.

**A futuro:**

MongoDB.

Mongo atlas se usara para almacenar nuestra base de datos.

Express para la API.

Averiguar precios y características de Mongo atlas con aws (53usd mes) firebase(FireStore) y DigitalOcean (me dijeron que es de las mejores relación precio/calidad 15usd mes)

**Funcionalidades**

* **Esenciales MPV:**

🔹 **Catálogo de productos**:

* Mostrar productos con imagen, descripción y precio.
* Filtros por categoría arriba.
* Botón para añadir al carro (se repite en la pagina del producto)

🔹 **Carrito de compras**:

* Agregar productos con el botón desde el listado.
* Ver resumen del carrito con cantidades de productos y total a pagar.
* Opción para modificar cantidad o eliminar productos.

🔹 **Proceso de compra**:

* Formulario de datos del comprador (nombre y teléfono).
* Selección de método de pago (transferencia, débito, cash).
* Elección de modalidad de entrega (retiro en local o delivery).
* Confirmación del pedido con detalles.

🔹 **Gestión de pedidos (para el administrador)**:

* Listado de pedidos en tiempo real.
* Estado del pedido (pendiente, en preparación, entregado).
* Opción para marcar un pedido como completado.

### **Funcionalidades avanzadas (futuro)**

**🔹 Sistema de notificaciones (email o WhatsApp para confirmaciones).**

🔹 **Pasarela de pagos** para procesar tarjetas automáticamente.  
🔹 **Registro de usuarios** para que los clientes guarden pedidos previos.  
🔹 **Historial de compras** para clientes recurrentes.  
🔹 **Sistema de descuentos y promociones.**

Los productos a su vez internamente tienen una key con el stock, por lo que cuando agreguemos la base de datos esta podría ser una funcionalidad que sea añadible por 1500 pesos o algo así.

Documentacion:   
Navbar: Se realizo el con un libre diseño el cual consta de una barra superior horizontal en dispositiivos grandes con una imagen del cliente en el medio, a su costado derecho presenta un avatar para logearse y a la izquierda presenta un sidebar desplegable responsive el cual se encarga de mostrar toda la informacion del cliente. Aqui se aplico estados para manejar dentro del navbar la informacion y una vez obtenidos estos datos necesarios para el navbar, utilizamos zustand para alojar un estado global el cual contiene el nombre de la empresa porque lo necesitamos mostrar en el footer. Recordemos que zustand nos permite este control de estados globales y renderiza globalmente en tiempo real si el estado cambia.

10/07/2025: Fix! Carrito y navbar. Modificamos el enfoque, a partir de ahora todos los datos recogidos de la DB por la api (datos de la empresa) se guardaran en localStorage una vez recibidos y renderizo con los datos del localStorage. Con respecto al carrito vamos a realizar las siguientes modificaciones: Una vez agregado el producto al carrito este se guarda en localStorage por 20 minutos con un timeStamp y utilizaremos un estado global que lea el localStorage para renderizar esos datos. Esto permite que cuando los productos en localStorage cambien el estado global también cambie, y ademas que cuando la pagina se recarge se lea primero el localStorage y el estado Global renderizara los datos xq si nosotros guardamos solo en el estado global los datos en cada refresh del sitio se perderían.

**Mejoras:**

El carrito debe eliminarse después de 30 minutos, no importa lo que pase.

En la página del pedido el total y el precio deben estar con un justify content between.