**APP CARTA MOBILE**

1. **RESUMEN DEL PROYECTO**

**Nuestro Proyecto es una carta para bares disponible como aplicación web ya que se encontraría alojada en algún servidor que disponibilize la app en internet. Básicamente seria como tener la carta de cualquier bar al alcance de las manos en el celular con simplemente escanear un código Qr. Allí se presentarían los menues y productos del bar que utilice nuestra aplicación, además el cliente podría realizar el pedido o la orden desde la app sabiendo cuanto gastaría y la cantidad de cada producto a consumir. Por detrás de esta aplicación habría un panel de administrador el cual permitiría visualizar las ordenes entrantes e ir marcándolas como ya entregadas en medida de que estas órdenes se conciban como recibidas por el cliente. No buscamos que el mozo de un bar sea sustituido por la aplicación, ya que el mozo es esencial en cualquier tipo de bar, simplemente permitirle al cliente poder visualizar lo que el bar ofrece como negocio sin tener la necesidad de que alguien le acerque una carta física o le diga todas las opciones que el bar ofrezca.**

1. **TEMA A DESARROLLAR**

**Planeamiento del problema**

**El Proyecto consiste principalmente en cubrir la necesidad del cliente al entrar a un bar y saber que ofrece este mismo ya sea como menú, bebida, postre o cualquier tipo de producto gastronómico que este ofrezca. Hay veces en la que los mozos se ven saturados de la cantidad de gente que hay en determinados momentos y no pueden acercarles las cartas a todos los clientes. Ante el problema anterior surgió nuestra idea de desarrollar esta carta app para que el cliente vea los productos y realice un pedido desde la misma.**

1. **OBJETIVOS**

**Problemáticas a resolver**

**Principalmente el problema que tratamos de cubrir es la atención tardía que a veces sucede en un bar cualquier sea este en horarios de desayuno, almuerzo, merienda o cena. Sinceramente sabemos lo que es para el cliente tener que esperar a veces a ser atendido para tan solo poder ver que es lo que ofrece el bar en cuestión, por eso esta app de carta móvil permitirá al cliente realizar dicha acción sin la necesidad de que el mozo le disponga la carta física. También el control sobre las ordenes pedidas y realizadas por otra parte desde el panel de administrador. Esto le permite al administrador de pedidos y de poder llevar un seguimiento sobre los pedidos entregados, los montos, las cantidades vendidas, la mesa que pidió, etc.**

1. **ESTADO DEL ARTE**

**Hay proyectos que se asemejan a este tipo de aplicación que nosotros desarrollamos, como por ejemplo ni más ni menos que cuando uno entra a una página web de algún negocio a comprar algo, realiza un pedido y este le llega al administrador para que el mismo pueda ver todos los detalles acerca del pedido realizado por el cliente.**

**ANALISIS DEL MERCADO**

**En el mercado por el momento no vimos tantos bares que posean carta web salvo los más conocidos, por eso nosotros creemos que es un mercado abierto por el momento a más competencia.**

1. **ACTIVIDADES**

**Tareas a realizar y tiempo empleado en cada una**

**Paso 1: Planificación (4 horas)**

**Definir Requisitos:**

**Identificar las funciones clave de la aplicación, como la visualización de menús, búsqueda y filtros.**

**Definir los requisitos de usuario y los casos de uso.**

**Diseño de la Base de Datos:**

**Diseñar la estructura de la base de datos utilizando Prisma Client y PostgreSql.**

**Definir las relaciones entre las tablas.**

**Paso 2: Configuración del Proyecto (6 horas)**

**Configuración de Next.js:**

**Iniciar un nuevo proyecto Next.js.**

**Configurar las rutas y la estructura de carpetas.**

**Configuración de Prisma Client:**

**Instalar Prisma Client y configúralo con MySQL.**

**Generar los modelos basados en el diseño de la base de datos.**

**Configuración de SWR y Axios:**

**Integrar SWR para la gestión del estado y Axios para realizar peticiones HTTP.**

**Utilizamos Cloudinary para gestionar el almacenamiento y la entrega optima de archivos multimedia al usuario final, con archivos multimedia hacemos referencia a las imágenes.**

**Utilizamos las librerías zod y zustand del ecosistema javascript/typescript:**

**Zod permite validar esquemas y tipados de los datos de la aplicación. Esta diseñada para definir,Validar y transformer datos estructurados.**

**Zustand gestiona el estado de las aplicaciones realizadas con react. Ofrece una api simple y sencilla para el manejo de estados en react.**

**Paso 3: Desarrollo del Frontend (20 horas)**

**Componentes React:**

**Crear componentes reutilizables para la interfaz de usuario.**

**Implementar la navegación entre páginas.**

**Integración con SWR:**

**Utilizar SWR para gestionar la caché y las solicitudes de datos.**

**Implementar la carga y visualización de menús.**

**Búsqueda y Filtros:**

**Implementar funciones de búsqueda y filtros para mejorar la experiencia del usuario.**

**Estilos y Diseño:**

**Aplica estilos CSS para mejorar la apariencia de la aplicación.**

**Paso 4: Desarrollo del Backend (16 horas)**

**Endpoints API:**

**Crear endpoints API en Next.js para gestionar las solicitudes del frontend.**

**Conectar con Prisma Client para interactuar con la base de datos.**

**Lógica de Negocio:**

**Implementar la lógica de negocio necesaria para manejar la búsqueda y filtrado de menús.**

**Paso 5: Pruebas y Optimización (10 horas)**

**Pruebas Unitarias:**

**Escribir pruebas unitarias para los componentes y funciones clave.**

**Optimización de Rendimiento:**

**Identificar y abordar posibles cuellos de botella en el rendimiento.**

**Implementar técnicas de optimización, como la carga diferida de imágenes.**

**Paso 6: Documentación y Despliegue (4 horas)**

**Documentación del Código:**

**Documentar el código para facilitar futuras actualizaciones y colaboraciones.**

**Despliegue:**

**Configurar la aplicación para el entorno de producción.**

**Desplegar la aplicación en un servidor.**

**Desplegamos la aplicación de la siguiente manera:**

**Todo lo que es respecto a imágenes, directamente lo maneja cloudinary.**

**Lo que respecta a la base de datos, tenemos una base de datos con postgreSql desplegada en render.**

**Y por ultimo el frontend en si de la aplicación desplegado en vercel.**

**Paso 7: Pruebas Finales y Ajustes (10 horas)**

**Pruebas Finales:**

**Realizar pruebas exhaustivas para asegurar el correcto funcionamiento de la aplicación.**

**Ajustes y Correcciones:**

**Realizar cualquier ajuste necesario según los resultados de las pruebas.**

**Paso 8: Entrega y Mantenimiento (2 horas)**

**Entrega del Proyecto:**

**Preparar la aplicación para ser entregada al cliente o para su implementación.**

**Mantenimiento:**

**Establecer un plan de mantenimiento para futuras actualizaciones y correcciones de errores.**

REQUERIMIENTOS

**REQUSITOS FUNCIONALES:**

**\_Los botones de pedido deben de enviar el pedido de cliente al menú administrador**

**\_Los menues deben tener el botón que permita agregarlos al pedido del cliente**

**\_Los datos del cliente deben poder llenarse a través del input correspondiente**

**REQUISITOS NO FUNCIONALES:**

**\_Un requisito no funcional seria que la pagina tenga https para que no puedan tirarla abajo algunos malintencionados**

**\_Otro requisito no funcional seria que el cliente no pierda el estado del pedido**

**\_Y por último un requisito no funcional es que se debe cifrar la comunicación entre la aplicación del cliente y el panel de control del administrador para proteger la integridad de los datos y prevenir el acceso no autorizado**.

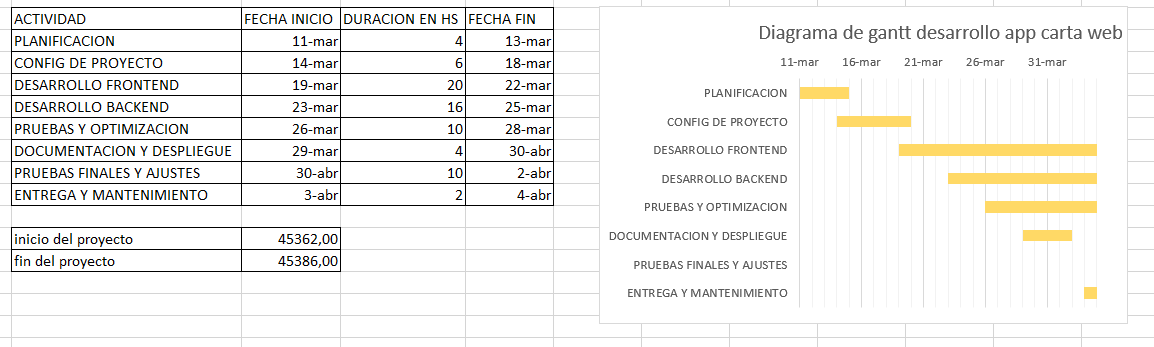
**REQUISITOS DE INTERFAZ:**

**\_La interfaz debe ser amigable para el usuario**

**\_El usuario debe poder visualizar clara y nítidamente los menu que hay disponibles**

**\_Los placeholder de los botones deben ser simples y concisos así el cliente entienda antes de apretarlos.**

**DIAGRAMA DE GANTT**

****

**CUANTIFICACION DEL EQUIPO**

**En el desarrollo de la aplicación tuvimos la necesidad de integrar 3 personas, que fuimos:**

**Williams Neme Scheij Tomas, Albarado Lucas Miguel y Bulacia Andres.**

**Nos dividimos las tareas de una manera concreta:**

**Albarado se encargo de ser el SCRUM MASTER, quien básicamente organizaba los tiempos de trabajo, las reuniones diarias, los sprint y todo aquello que sea fundamental para el buen desarrollo de la aplicación. Se encargo de definir bien los requerimientos de nuestro cliente, planificar cuanto tiempo iba a tardar estimativamente.**

**Williams se encargo del desarrollo del backend de la aplicación. Todo lo que conforma el desarrollo de la base de datos, modelado de datos, conexiones con los servidores de despliegue y lógica de negocio.**

**Bulacia se encargo del desarrollo frontend. Siempre tratando de cumplir los requerimientos de nuestro cliente, los requisitos y demás detalles estéticos que nuestro cliente nos fue pidiendo.**

**Al fin de cuentas, los tiempos empleados por cada uno fueron masomenos entre 20 y 25 horas cada uno, logrando así un poco mas de 60 horas que era el mínimo.**

**DESARROLLO DEL PROYECTO**

1. **EJECUCION DE LA SOLUCION**

**Al principio teníamos un stack definido para poder llevar a cabo el desarrollo, que era: Mysql para base de datos, React para frontend, Bootstrap para estilos de la pagina, node y express para manejo de servidores.**

**Estas ideas fueron cambiando a lo largo del tiempo por ciertas cuestiones de manejo de las tecnologías, y terminamos usando las siguientes:**

**Resumen del Stack**

**Framework principal: Next.js**

**Librería de UI: React**

**CSS: Tailwind CSS**

**Íconos: Heroicons, React Icons**

**Notificaciones: React Toastify**

**Estado: SWR, Zustand**

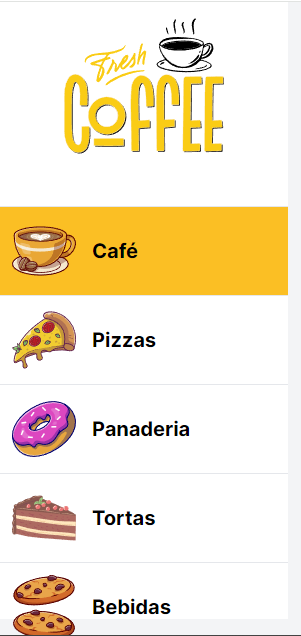
**Base de datos: Prisma y postgresql**

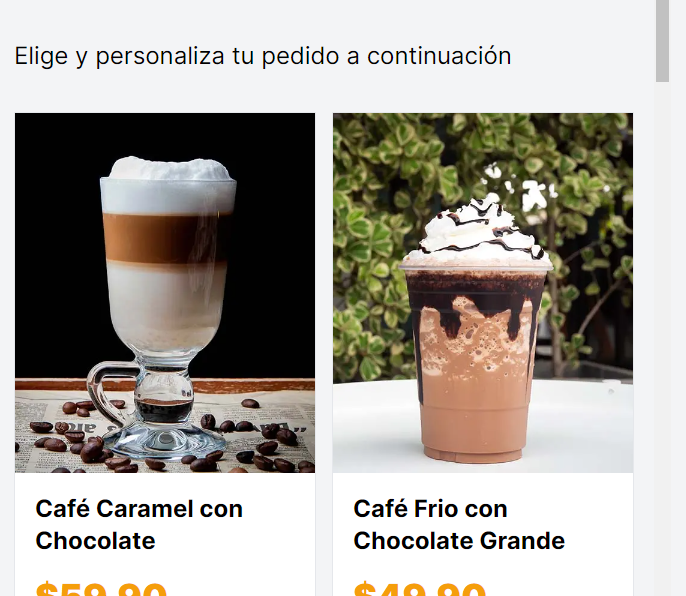
**Lenguaje: TypeScript**

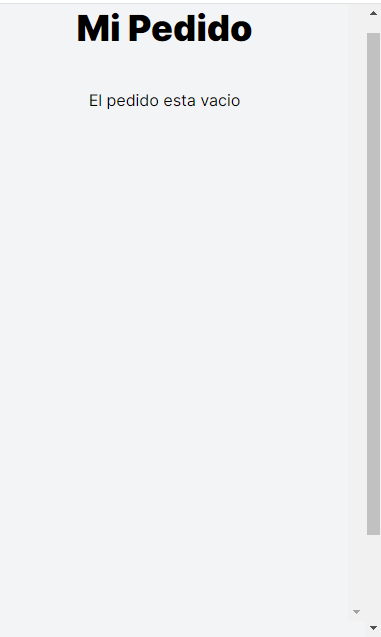
**Herramientas de desarrollo: ESLint, PostCSS, Autoprefixer**

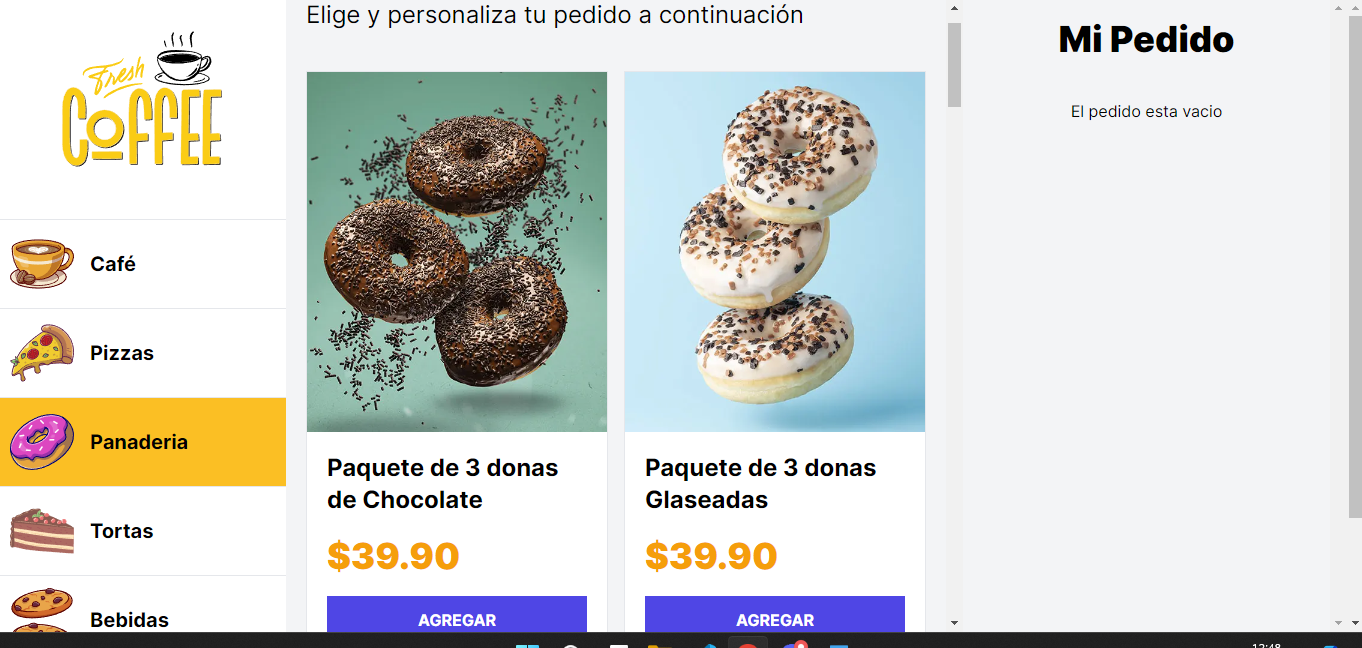
**Gestión de medios: Cloudinary**

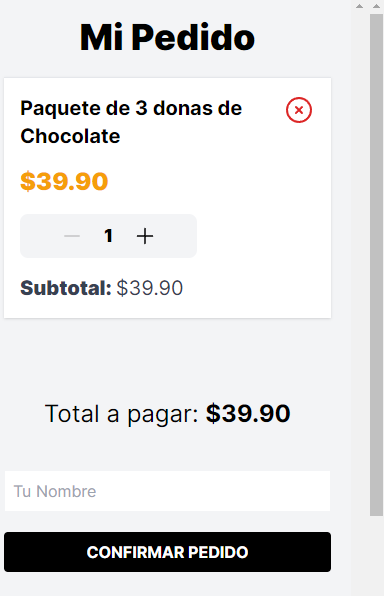
**CAPTURAS DE LA APLICACIÓN**

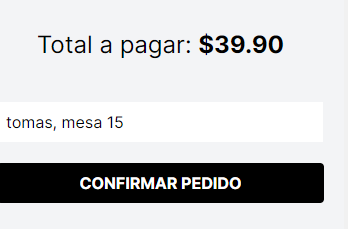
****

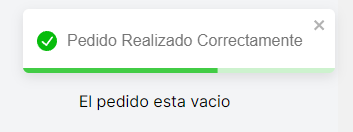
****

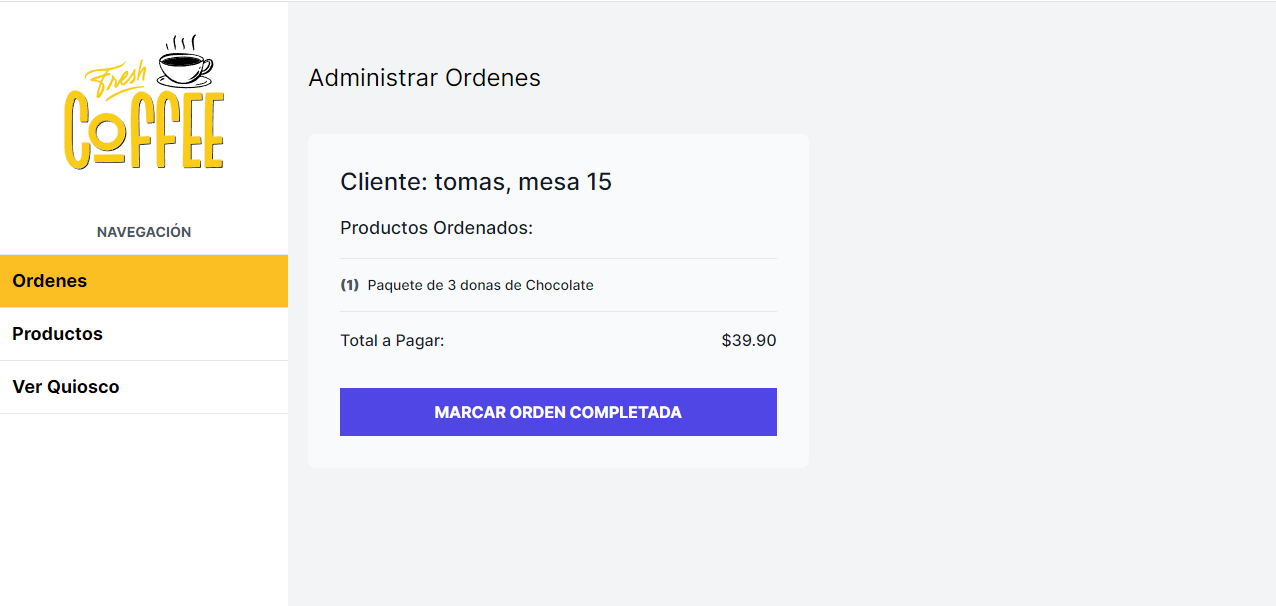
****

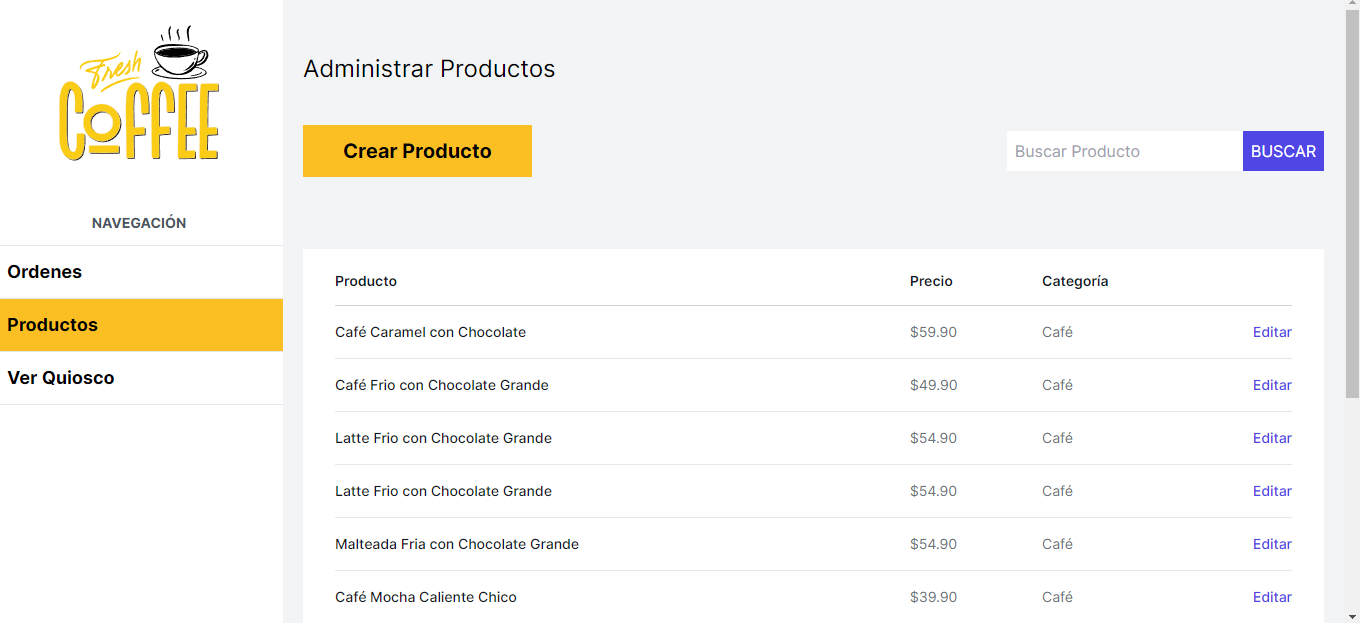
****

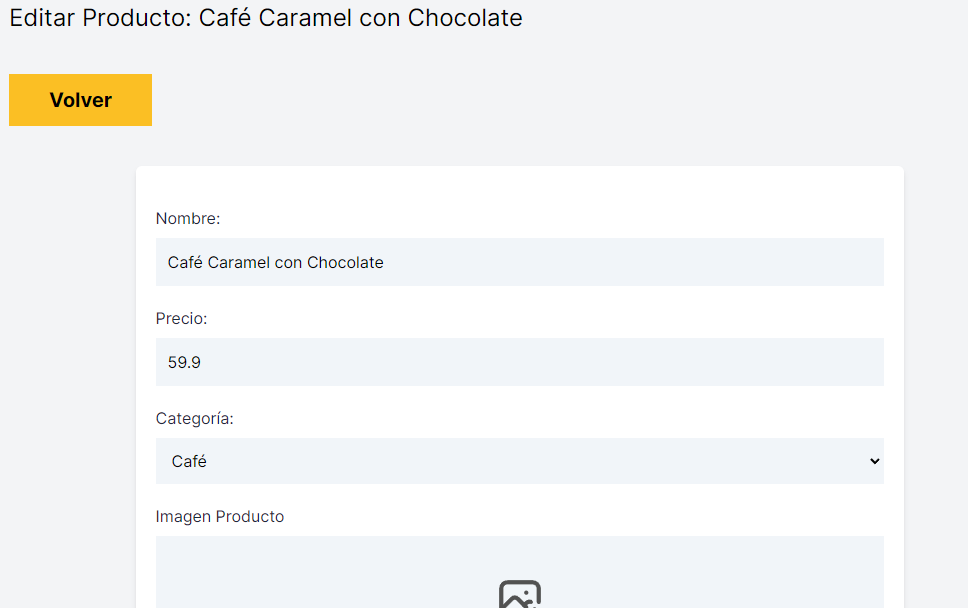
****

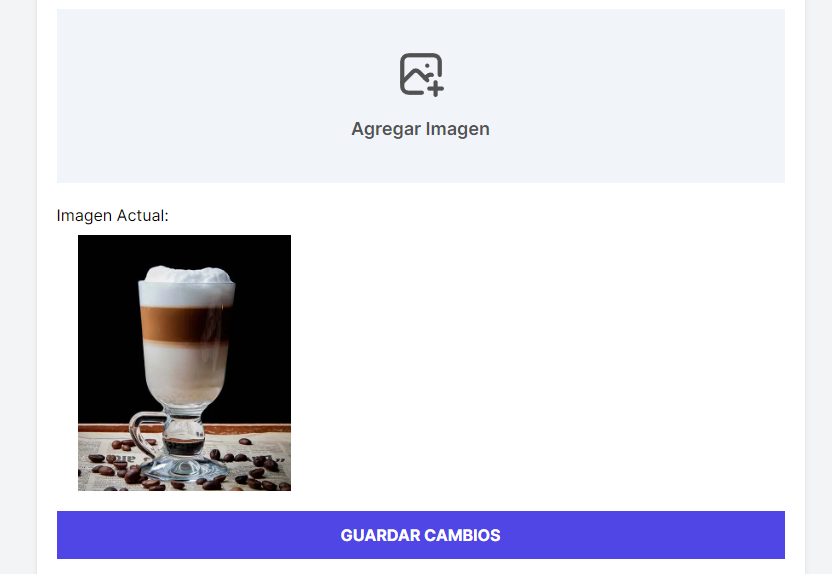
****

****

****

****

****

****

****

****

**REQUERIMIENTOS DE HARDWARE:**

**Para un entorno de desarrollo, una máquina con CPU de 2 núcleos, 8 GB de RAM y 50 GB de almacenamiento SSD es suficiente. Para producción, ajusta los recursos en función del tamaño de tu aplicación y la cantidad de tráfico que esperas manejar, comenzando con al menos 2-4 núcleos de CPU, 8-16 GB de RAM, y 100 GB de almacenamiento SSD, y escalando según sea necesario.**

**REQUERIMIENTOS DE SOFTWARE**

**Entorno de Desarrollo: Un sistema operativo compatible, Node.js, npm/yarn, una base de datos local, un entorno de desarrollo como VSCode, y herramientas adicionales para el control de versiones, contenedores, y pruebas de API.**

**Entorno de Producción: Una distribución de Linux, un servidor web (Nginx o Apache), Node.js, una base de datos, herramientas de gestión de procesos, almacenamiento y CDN, medidas de seguridad, y herramientas de monitoreo y automatización para despliegue continuo y gestión de aplicaciones**

**MANUAL DE USUARIO:**

**Primero que nada generaremos un código qr para que el usuario escanee y pueda ingresar al link donde este subida la aplicación web. Desde ahí el usuario podrá ver las categorías de los productos que ofrezca el bar y en cada categoría poder elegir lo que deseen consumir.**

**Al agregar un producto el carrito ira mostrando lo que el usuario este seleccionando y el total a pagar que tendrá. Al confirmar el pedido, el trayecto de uso del cliente habrá finalizado.**

**Para quien este desde el lado de administrador de ordenes y productos, es decir manejando el panel, podrá visualizar las ordenes que estén hechas y no entregadas. Desde allí tiene un menú con tres links: uno que es el de ordenes, desde donde las podrá marcar como entregadas y estas serán retiradas de esa sección y enviadas a ordenes ya entregadas.**

**Otra que es donde podrá ver todos los productos que tiene en la base de datos, y desde allí podrá:**

**Cargar mas productos, nombre, precio, descripción e imagen del mismo.**

**Editar ciertos productos que ya posea.**

**Eso seria con respecto al panel de productos.**

**Con todo esto, desde otra ruta, que seria /orders, podrá ver todas aquellas ordenes que haya marcado como ya entregadas. Que una orden este entregada, quiere decir que el mozo ya acerco el pedido del cliente hacia la mesa y que este ya lo esta consumiendo o lo consumió y se retiró. El tema de pagos no es manejado desde esta app.**

**Con todo esto queda resumido concretamente el modus operanti de esta aplicación tanto para cliente consumidor como para el cliente administrador.**

**CODIGO QR PARA ESCANEAR E INGRESAR A LA APP**

****