

Documentación del Módulo de Proyecto: DAILYGO

Ignacio Gil Torres, Gonzalo Oliver Alfonso
y Carlos Almendros Arellano

Tutora: Laura González Pastor



Índice

1. Introducción	3.
Descripción de la aplicación	3.
Visión general del proyecto	3.
2. Planificación y análisis	4.
Idear el proyecto	4.
Requisitos y estructura	4.
Diagrama de Gantt	8.
Decisión de tecnologías	6.
Mapa del sitio	7.
3. Cuerpo del documento	8.
Modelo de bases de datos	8.
Diagrama de clases	11.
Casos de Uso	12.
Diagramas de secuencias	13.
Diagrama de transición de estados	17.
Diagramas de colaboración	18.
Funciones por tipo de usuario	19.
Fases de prueba	21.
Creación teórica de la pasarela de pago	24.
Despliegue	26.
Potenciales usuarios: “UX Personas”	27.
Viaje del usuario	29.
4. Conclusiones	30.
5. Trabajo futuro	31.
6. Anexos	33.
Herramientas	33.
Identidad de la marca	35.
7. Bibliografía	38.

¿QUÉ ES DAILYGO?

1. Introducción

DESCRIPCIÓN DE LA APLICACIÓN

DailyGo es una aplicación web creada para ofrecer pedidos de comida y productos de parafarmacia al instante, recibiendo estos en tiempos récord.

Esta aplicación está dirigida tanto para usuarios que realicen compras que podrán beneficiarse de ofertas, riders para poder ganar dinero transportando pedidos a clientes y proveedores que pondrán los productos haciendo que cualquiera puede disfrutar del mismo servicio que pueda ofrecer cualquier local en presencial, en cualquier lugar.

DailyGo es un proyecto innovador que busca revolucionar la forma en que las personas acceden a servicios y productos locales de manera rápida y conveniente. En un mundo cada vez más digitalizado, DailyGo se posiciona como una solución integral que conecta a proveedores y clientes de forma eficiente, brindando beneficios tanto para aquellos que buscan servicios como para aquellos que los ofrecen.

VISIÓN GENERAL DEL PROYECTO

El objetivo principal de DailyGo es proporcionar una plataforma fácil de usar que permita a los usuarios encontrar y contratar servicios locales de manera rápida y confiable. Ya sea que necesites un servicio de entrega, un servicio de limpieza, un servicio de cuidado personal o cualquier otro tipo de servicio local, DailyGo te conecta con proveedores de confianza que se ajustan a tus necesidades.

Además, DailyGo también brinda a los proveedores una oportunidad única para ampliar su alcance y llegar a nuevos clientes. Mediante la creación de un perfil en la plataforma, los proveedores pueden promocionar sus servicios y llegar a una audiencia más amplia, lo que resulta en una mayor visibilidad y oportunidades de negocio.

SEGUIMIENTO

2. Planificación y análisis

IDEAR EL PROYECTO

DailyGo fue concebido para mejorar la experiencia de pedir comida a domicilio. Nos dimos cuenta de las dificultades que enfrentan los usuarios al utilizar las plataformas existentes y decidimos crear una aplicación que fuera más eficiente y conveniente. Nuestro enfoque fue proporcionar a los usuarios una amplia variedad de opciones gastronómicas y productos variados y la capacidad de realizar pedidos de forma rápida y sencilla.

También consideramos a los riders y proveedores, asegurándonos de que tuvieran herramientas para gestionar las entregas y administrar los productos de manera intuitiva. Durante el proceso de desarrollo, nos centramos en aspectos como la seguridad de los pagos(teóricamente) , la precisión de la información de los pedidos y la facilidad de búsqueda y filtrado de alimentos. Queríamos que DailyGo fuera fácil de usar, seguro y sin complicaciones. En resumen, DailyGo nació para simplificar y agilizar el proceso de pedir comida a domicilio. Nuestra aplicación se enfoca en brindar a los usuarios una experiencia fluida, a la vez que proporciona herramientas eficientes para los riders y proveedores.

REQUISITOS Y ESTRUCTURA

Desde el inicio del desarrollo de DailyGo, nos planteamos una serie de requisitos clave que la aplicación debía cumplir para satisfacer las necesidades de nuestros usuarios. Estos requisitos incluían:

Capacidad para que los usuarios pudieran realizar pedidos: Uno de los principales objetivos de DailyGo era permitir a los usuarios realizar pedidos de comida de forma rápida y sencilla. Implementamos un flujo de pedido intuitivo y una interfaz de usuario amigable que facilita la selección de productos y la finalización del proceso de compra.

Funcionalidad para que los riders pudieran cambiar el estado de los pedidos: Reconocimos la importancia de mantener a los usuarios informados sobre el estado de sus pedidos y brindar a los riders la capacidad de actualizar dichos estados. Implementamos un sistema que permite a los riders cambiar el estado de los pedidos, como "en camino", "entregado" o "cancelado", para mantener a todos los involucrados actualizados en tiempo real.

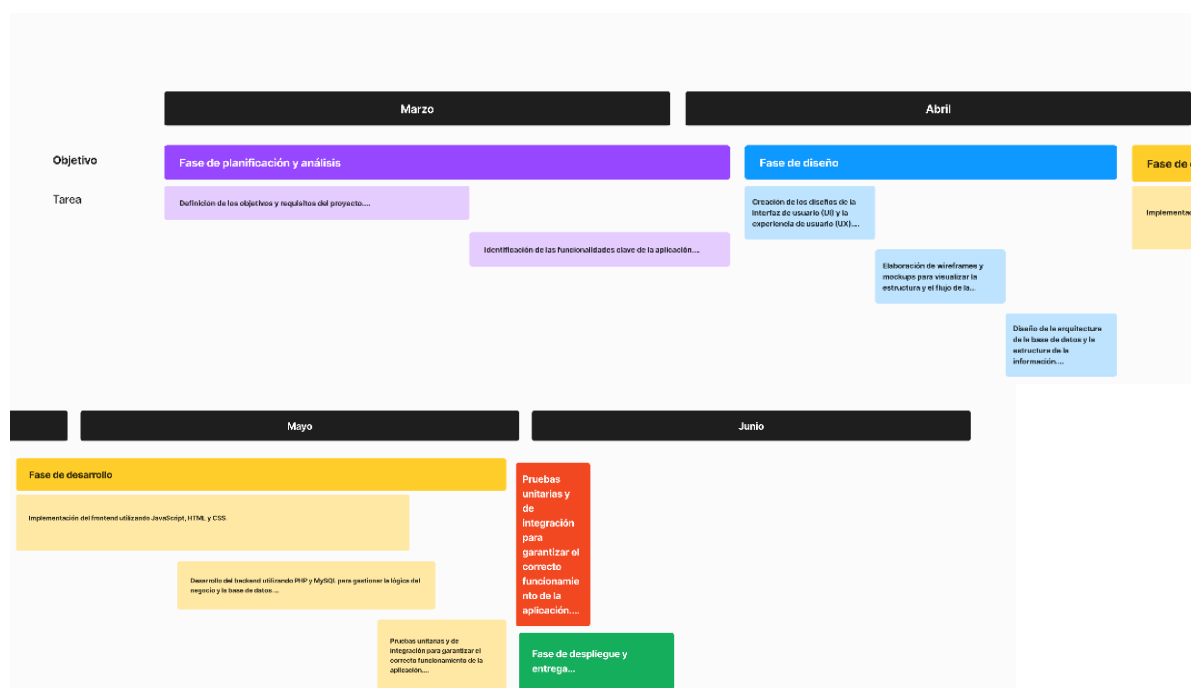
Posibilidad para que los proveedores pudieran incluir nuevos productos: Para garantizar una amplia variedad de opciones para los usuarios, era fundamental permitir a los proveedores agregar nuevos productos a la plataforma. Desarrollamos una funcionalidad que permite a los proveedores administrar su catálogo de productos, agregar imágenes, descripciones, precios y gestionar su disponibilidad en tiempo real.

Sistema de gestión de pagos seguro (teórica): Implementamos una pasarela de pago confiable y segura que permite a los usuarios realizar transacciones de forma segura y protegida. Utilizamos las mejores prácticas de seguridad y encriptación de datos para garantizar la confidencialidad de la información financiera de nuestros usuarios.

Funcionalidad de búsqueda y filtrado: Reconocimos la importancia de facilitar a los usuarios la búsqueda y selección de productos. Implementamos un sistema de búsqueda y filtrado que permite a los usuarios encontrar rápidamente los productos deseados según categorías, precios, valoraciones y otras opciones de personalización.

En resumen, DailyGo fue diseñado para cumplir con una serie de requisitos clave desde el inicio del desarrollo. Nos enfocamos en permitir que los usuarios realicen pedidos de comida, brindar a los riders la capacidad de actualizar el estado de los pedidos, permitir a los proveedores agregar nuevos productos, garantizar un sistema de pagos seguro y ofrecer funcionalidades de búsqueda y filtrado. Cumplir con estos requisitos fue fundamental para proporcionar una experiencia satisfactoria a nuestros usuarios y lograr el éxito de DailyGo.

DIAGRAMA DE GANTT



DECISIÓN DE TECNOLOGÍAS

En el proceso de desarrollo de DailyGo, hemos tomado decisiones importantes en cuanto a las tecnologías que utilizaremos para construir el front-end y el back-end de la aplicación. Después de un análisis exhaustivo, hemos seleccionado JavaScript y PHP como nuestras principales opciones tecnológicas para el desarrollo web.

JavaScript es un lenguaje de programación versátil y ampliamente utilizado en el desarrollo web. Su amplia adopción y la disponibilidad de numerosas bibliotecas y frameworks lo convierten en una elección sólida para crear una experiencia interactiva y dinámica en el front-end de DailyGo. Con JavaScript, podemos manipular el contenido de la página en tiempo real, interactuar con los usuarios y gestionar eventos de forma eficiente.

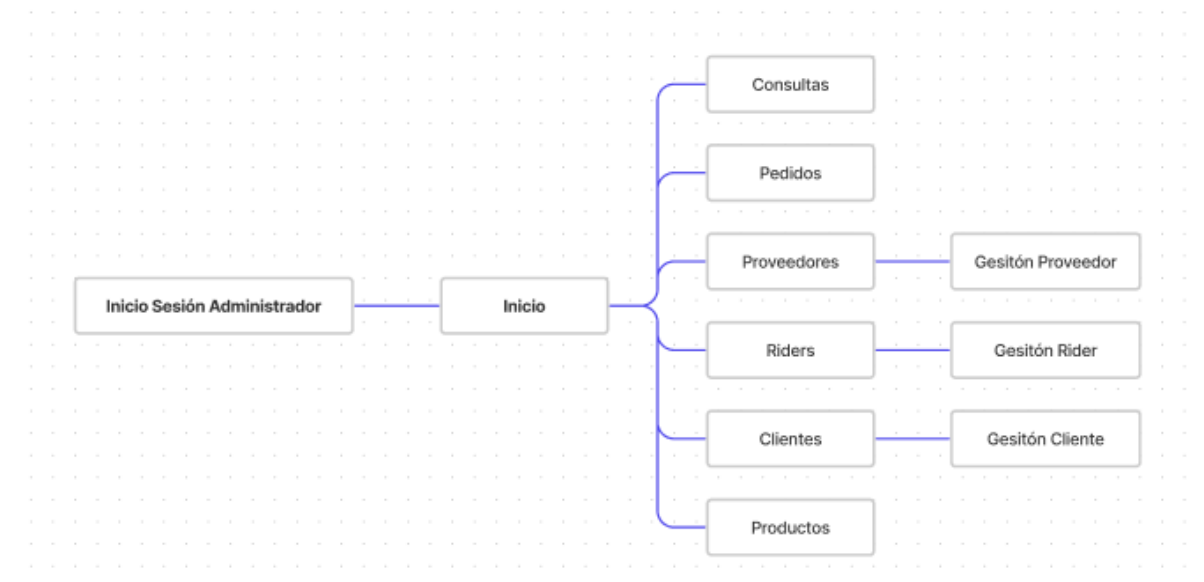
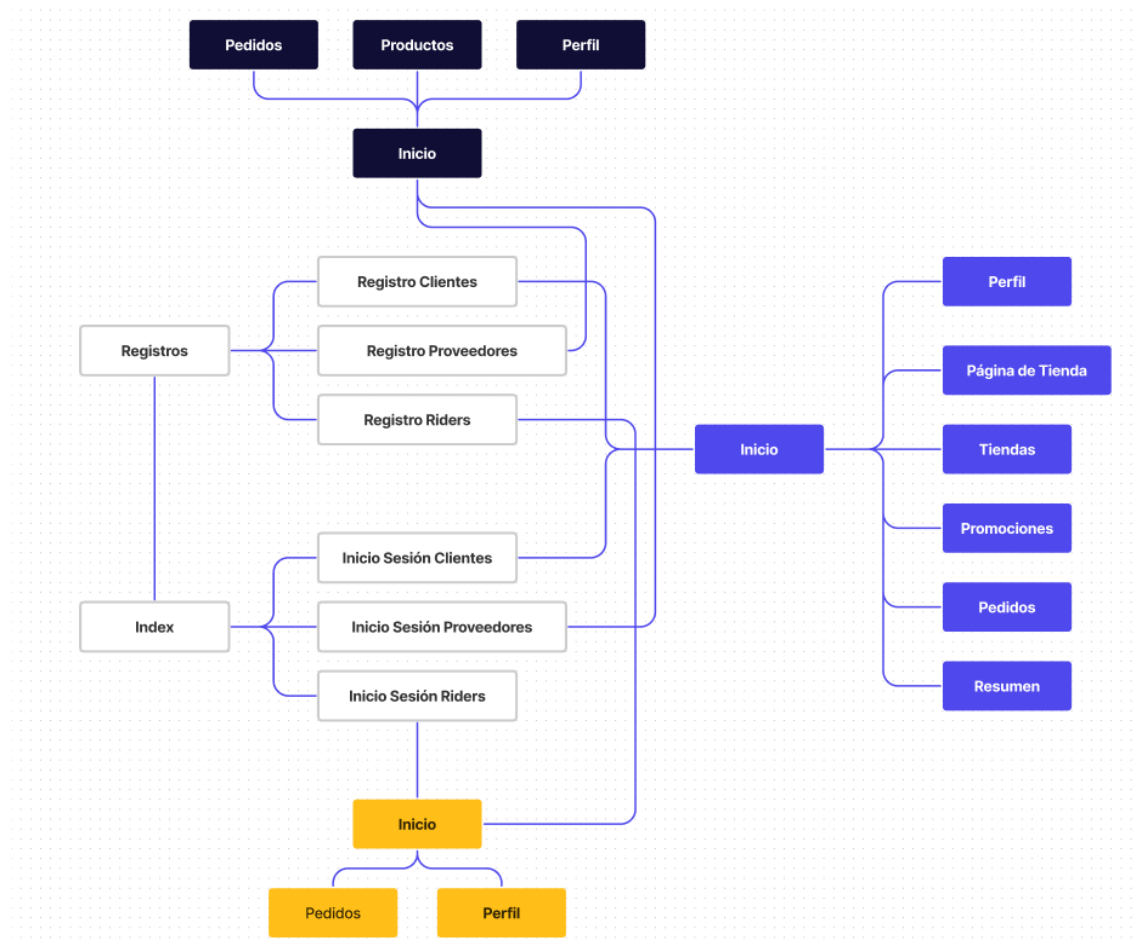
Por otro lado, PHP es un lenguaje de programación del lado del servidor que se ha establecido como una opción confiable y escalable para el desarrollo web. Con PHP, podemos manejar la lógica de negocio y la interacción con la base de datos de DailyGo de manera eficiente. Su integración con los sistemas de gestión de bases de datos y su amplia comunidad de desarrolladores hacen que PHP sea una elección sólida para nuestras necesidades.

En cuanto a las tecnologías de diseño, hemos optado por utilizar Tailwind CSS. Este framework de CSS moderno y altamente popular que nos permite crear interfaces de usuario atractivas y responsivas de manera eficiente. Tailwind CSS, nos brinda una forma más granular y personalizable de diseñar la interfaz, lo que nos permite adaptarla a nuestras necesidades específicas.

En el proceso de toma de decisiones, hemos descartado el uso de Java y Python en nuestro stack tecnológico. Aunque ambos son lenguajes sólidos y ampliamente utilizados en el desarrollo web, hemos optado por centrarnos en JavaScript y PHP debido a su compatibilidad y experiencia previa en nuestro equipo de desarrollo. Además, consideramos que JavaScript y PHP cumplen con todos los requisitos de Daily Go de manera efectiva y eficiente.

En resumen, hemos seleccionado JavaScript y PHP como nuestras principales opciones para el desarrollo web en DailyGo, Tailwind CSS se encarga del diseño y la apariencia de la aplicación. Al descartar Java y Python, hemos tomado decisiones basadas en la compatibilidad y la experiencia de nuestro equipo, asegurando que las tecnologías seleccionadas cumplan con nuestras necesidades de manera óptima.

MAPA DEL SITIO



ESTRUCTURA INTERNA

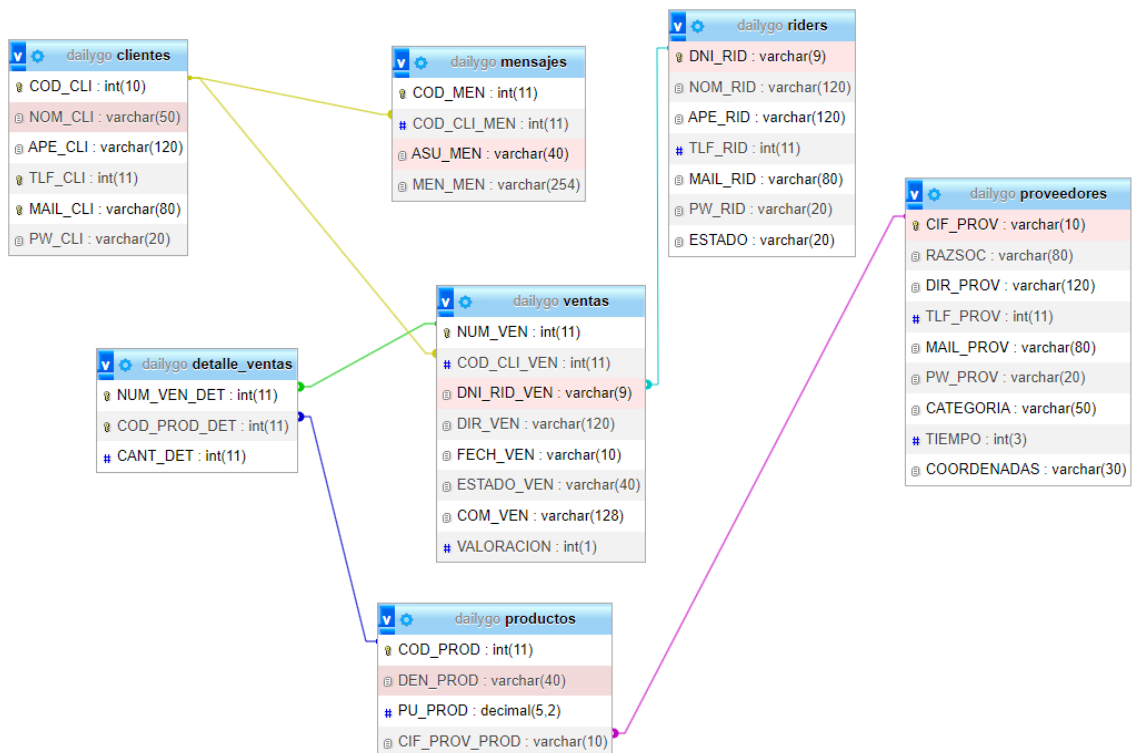
3. Cuerpo del documento

MODELO DE BASES DE DATOS

1.- Modelo conceptual:



2.- Modelo lógico:



3.- Modelo físico:

CLIENTES				
Campo	Tipo	Tamaño	Restricción	Descripción
COD_CLI	INT	10	PK	Código del cliente
NOM_CLI	VARCHAR	50		Nombre del cliente
APE_CLI	VARCHAR	120		Apellidos del cliente
TLF_CLI	INT	11		Teléfono del cliente
MAIL_CLI	VARCHAR	80		Correo electrónico
PW_CLI	VARCHAR	20		Contraseña

PROVEEDORES				
Campo	Tipo	Tamaño	Restricción	Descripción
CIF_PROV	INT	10	PK	CIF del proveedor
RAZSOC	VARCHAR	80		Razón social
DIR_PROV	VARCHAR	120		Dirección
TLF_PROV	INT	11		Teléfono
MAIL_PROV	VARCHAR	80		Correo electrónico
PW_PROV	VARCHAR	20		Contraseña
CATEGORÍA	VARCHAR	50		Categoría del negocio
COORDENADAS	VARCHAR	30		Coordenadas del local

DETALLE VENTAS				
Campo	Tipo	Tamaño	Restricción	Descripción
NUM_VEN_DET	INT	11	PK	Número de venta
COD_PROD_DET	INT	11	PK FK (Productos)	Código de producto
CANT_DET	INT	11		Cantidad de producto

RIDERS				
Campo	Tipo	Tamaño	Restricción	Descripción
DNI_RID	VARCHAR	9	PK	DNI del rider
NOM_RID	VARCHAR	120		Nombre
APE_RID	VARCHAR	120		Apellidos
TLF_RID	INT	11		Teléfono
MAIL_RID	VARCHAR	80		Correo electrónico
PW_RID	VARCHAR	20		Contraseña
ESTADO	VARCHAR	20		Estado de disponibilidad del Rider

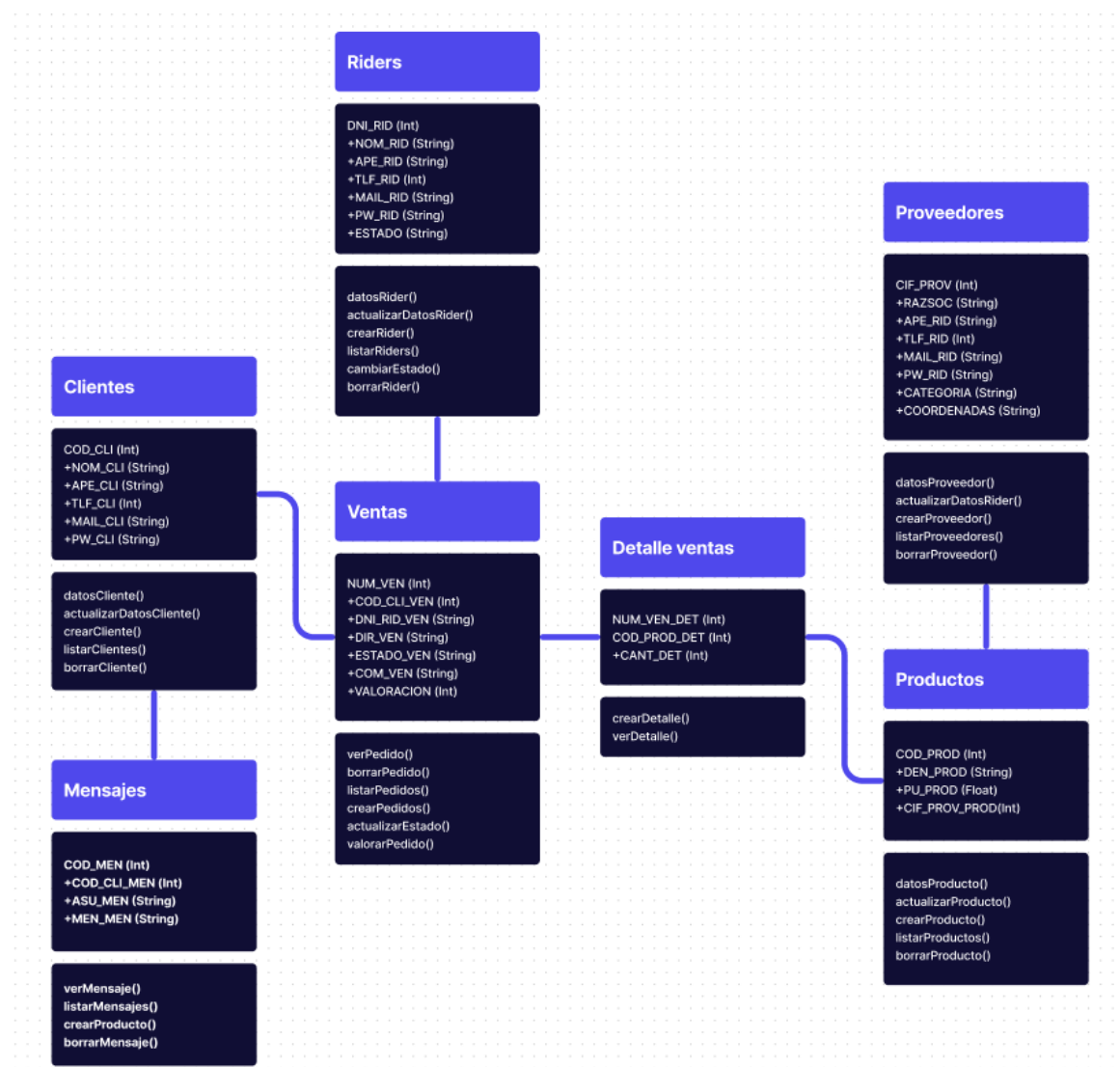
PRODUCTOS				
Campo	Tipo	Tamaño	Restricción	Descripción
COD_PROD	INT	11	PK	Código de producto
DEN_PROD	VARCHAR	40		Denominación
PU_PROD	DECIMAL	5,2		Precio unitario
CIF_PROV_PROD	VARCHAR	120	FK (Proveedores)	CIF del Proveedor del producto

MENSAJES				
Campo	Tipo	Tamaño	Restricción	Descripción
COD_MEN	INT	11	PK	Código de mensaje
COD_CLI_MEN	INT	11	FK (Clientes)	Código del cliente que envía el mensaje
ASU_MEN	VARCHAR	540		Asunto del mensaje
MEN_MEN	VARCHAR	254		Cuerpo del mensaje

VENTAS				
Campo	Tipo	Tamaño	Restricción	Descripción
NUM_VEN	INT	11	PK	Número de venta
COD_CLI_VEN	INT	11	FK (Clientes)	Código de cliente
DNI_RID_VEN	VARCHAR	9	FK (Riders)	DNI de rider
DIR_VEN	VARCHAR	120		Dirección de destino
FECH_VEN	VARCHAR	10		Fecha del pedido
ESTADO_VEN	VARCHAR	40		Estado actual del pedido
COM_VEN	VARCHAR	128		Comentarios al rider o al proveedor
VALORACION	INT	1		Nota del pedido (1 al 5)

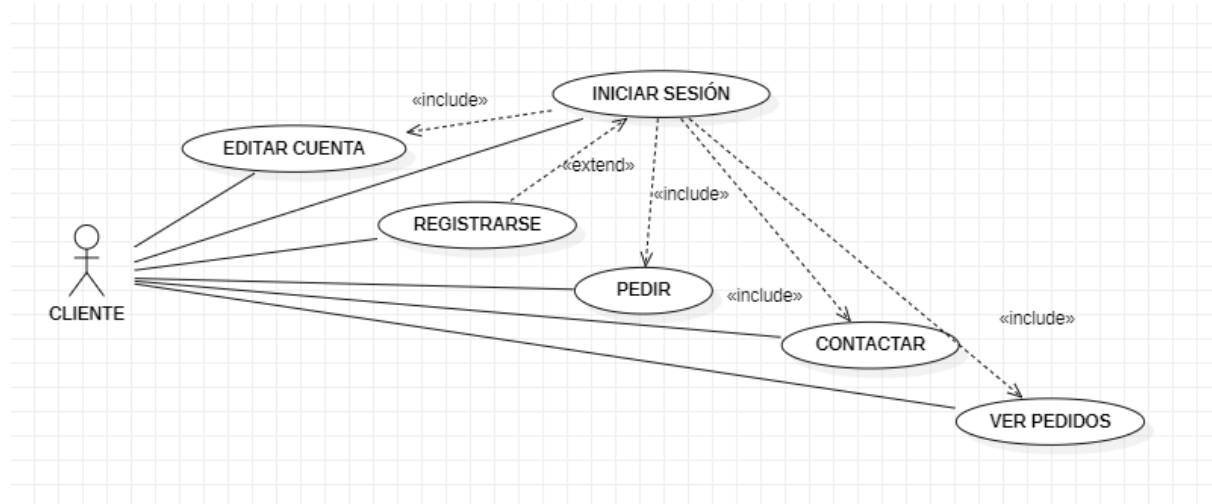
DIAGRAMA DE CLASES

(Diagrama de Clases teórico)

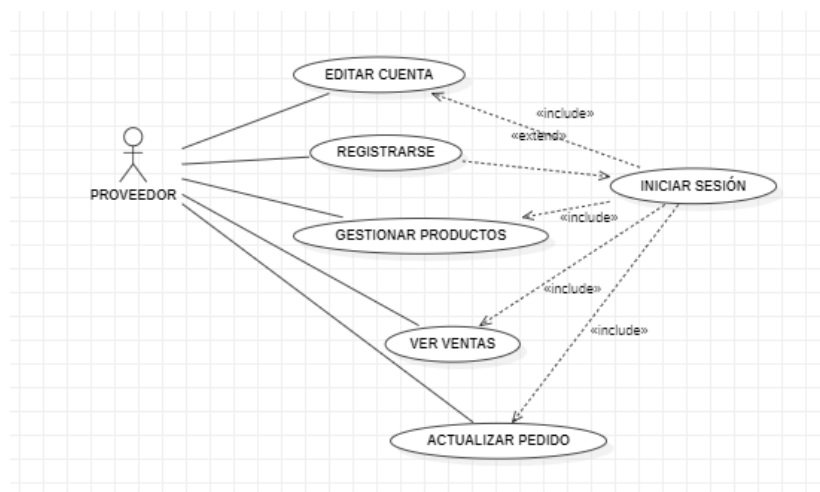


CASOS DE USO

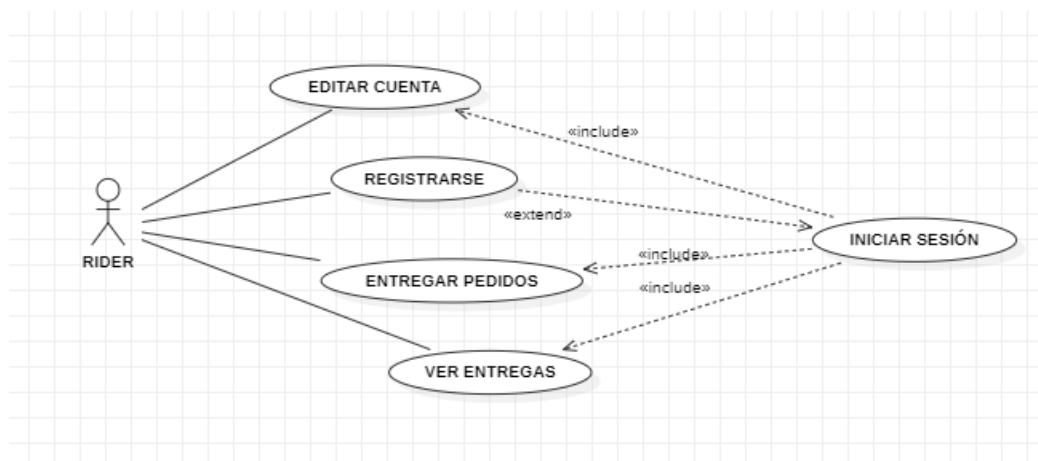
Cientes



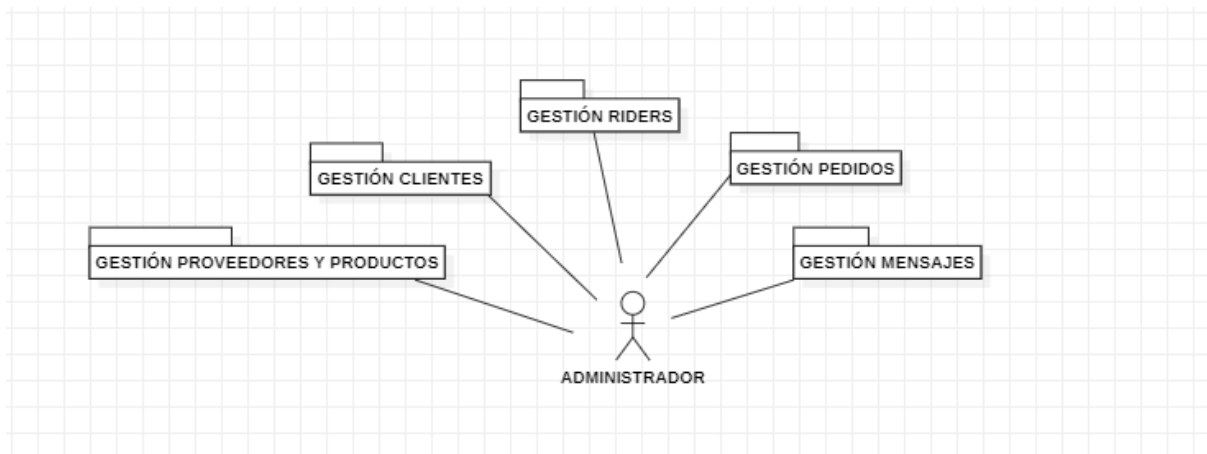
Proveedores



Rider

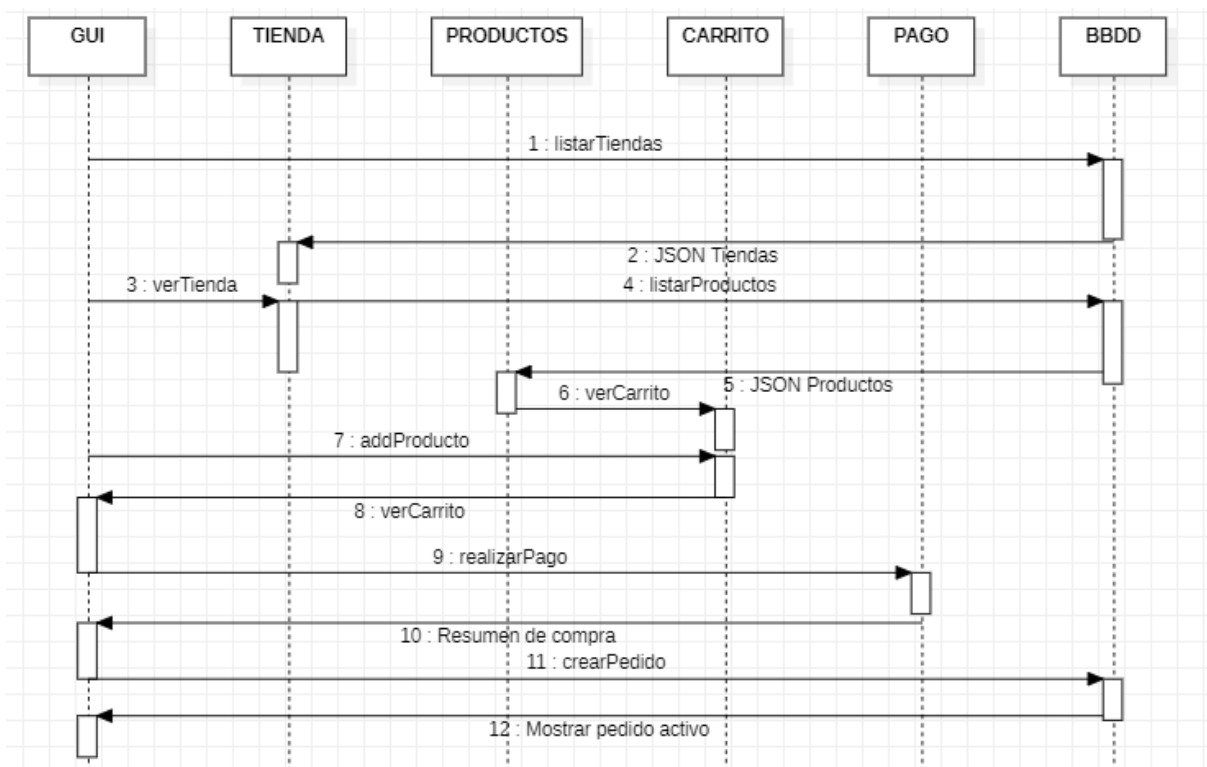


Administrador

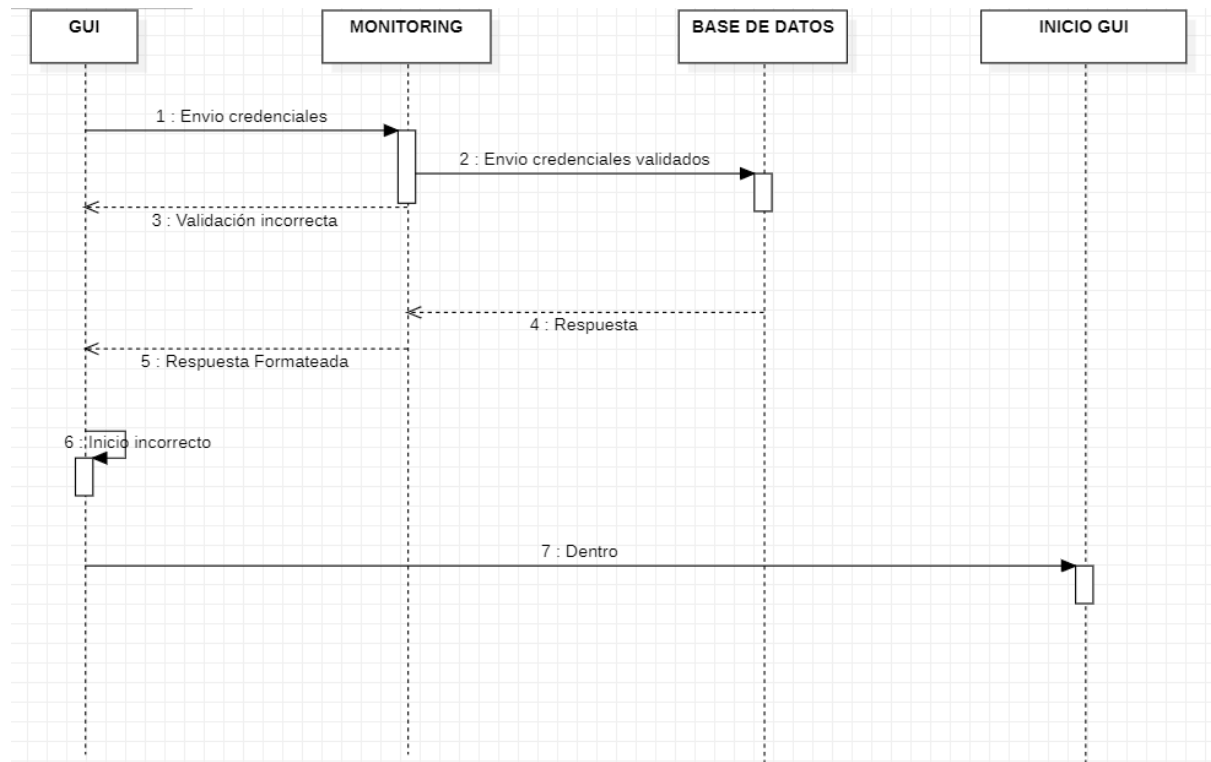


DIAGRAMAS DE SECUENCIA

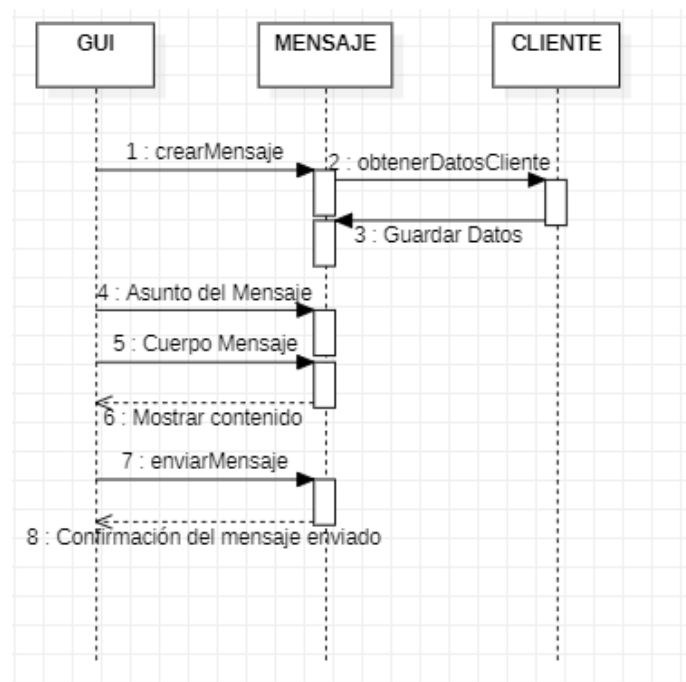
Crear pedido



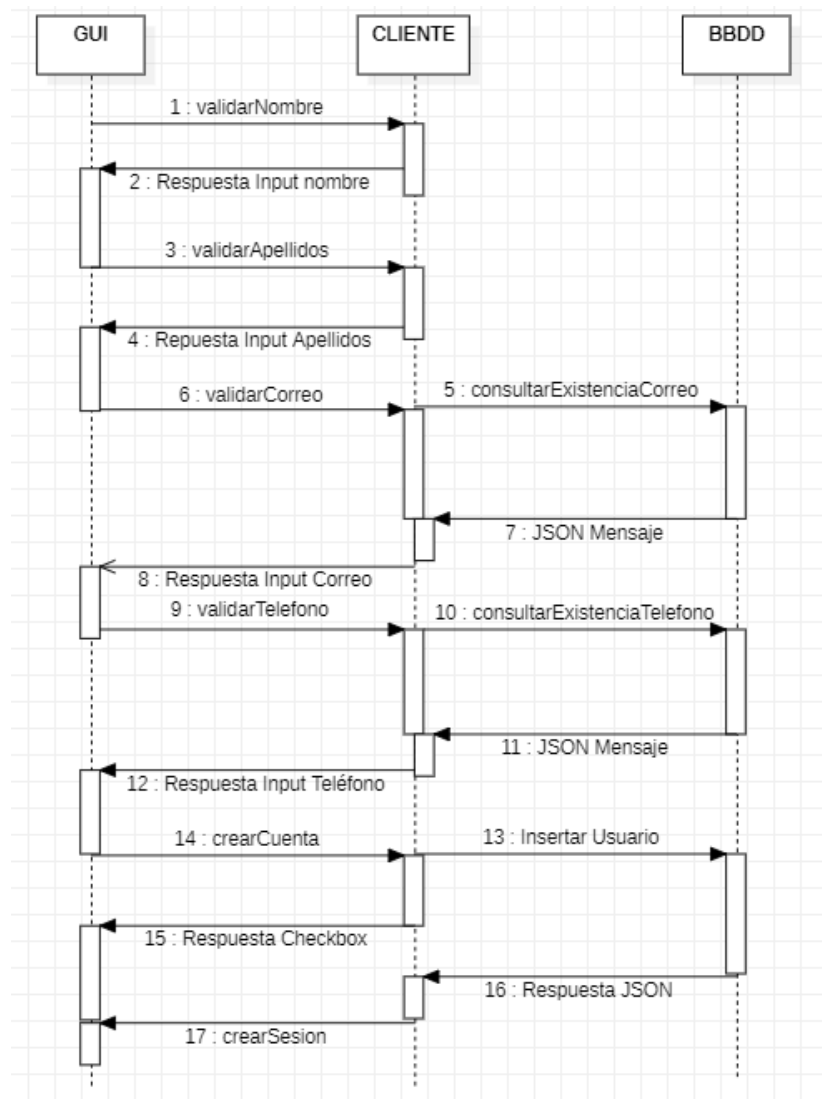
Inicio de sesión



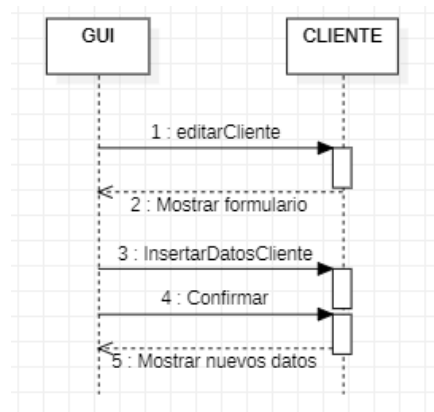
Crear mensaje



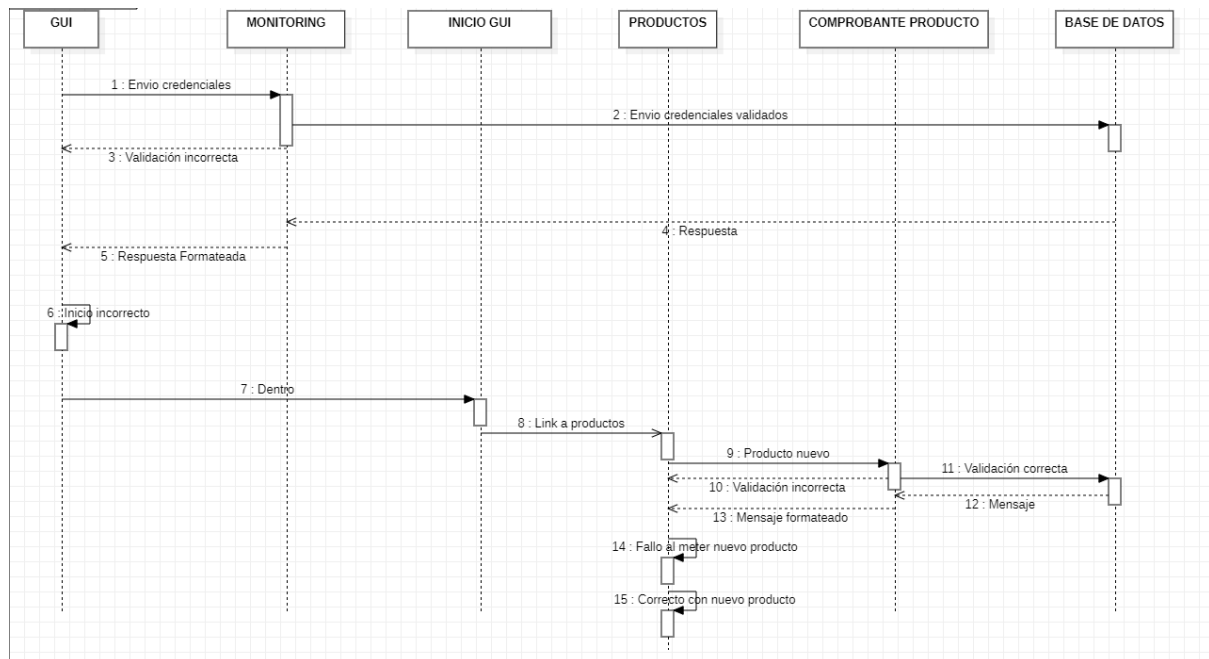
Registro



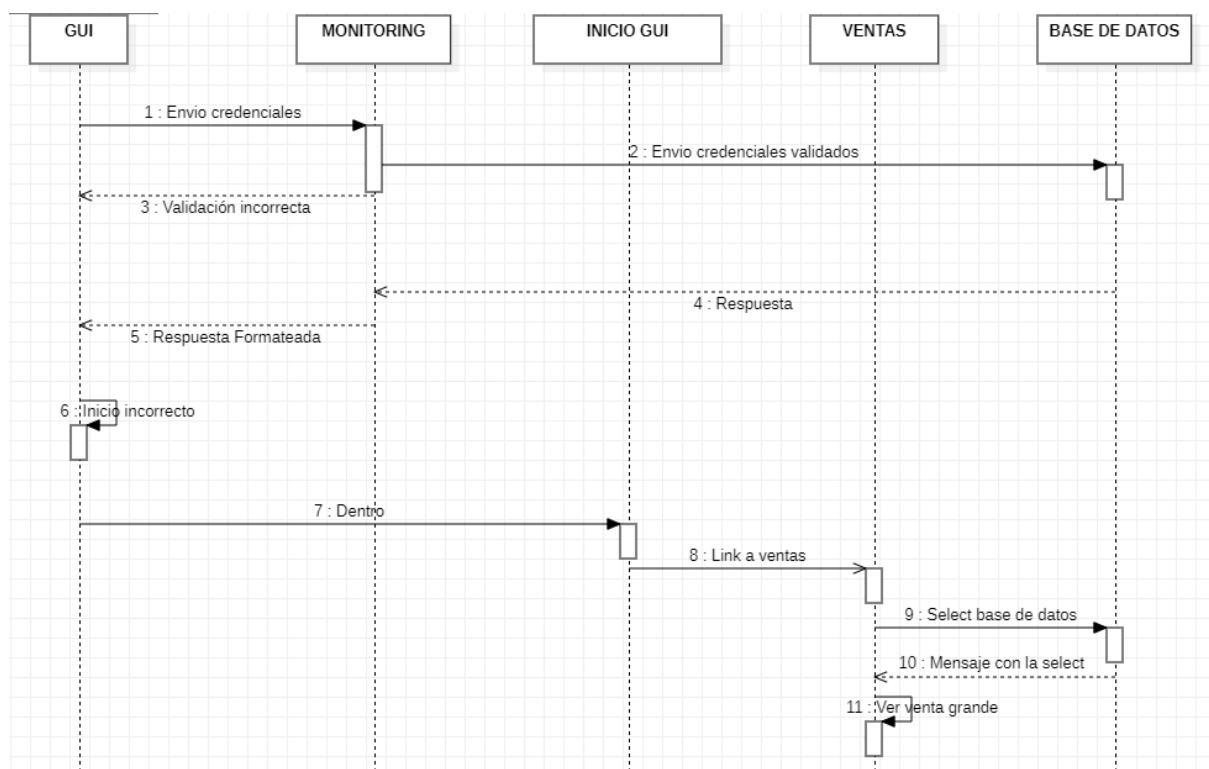
Editar datos de usuario



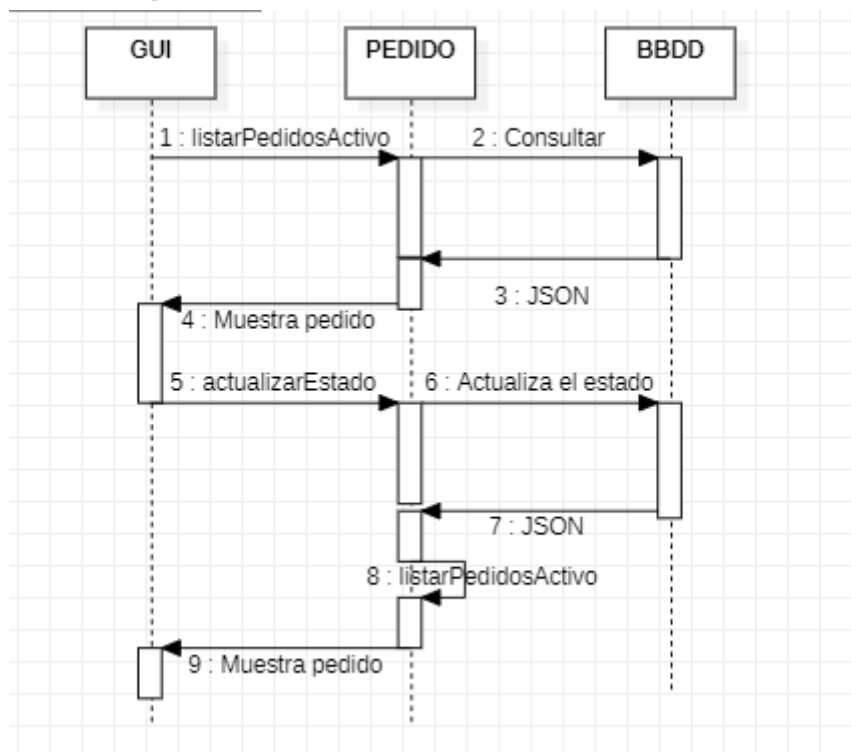
Añadir producto



Ver ventas



Actualizar estado del pedido



TRANSICIÓN DE ESTADOS

Transición de estados del pedido

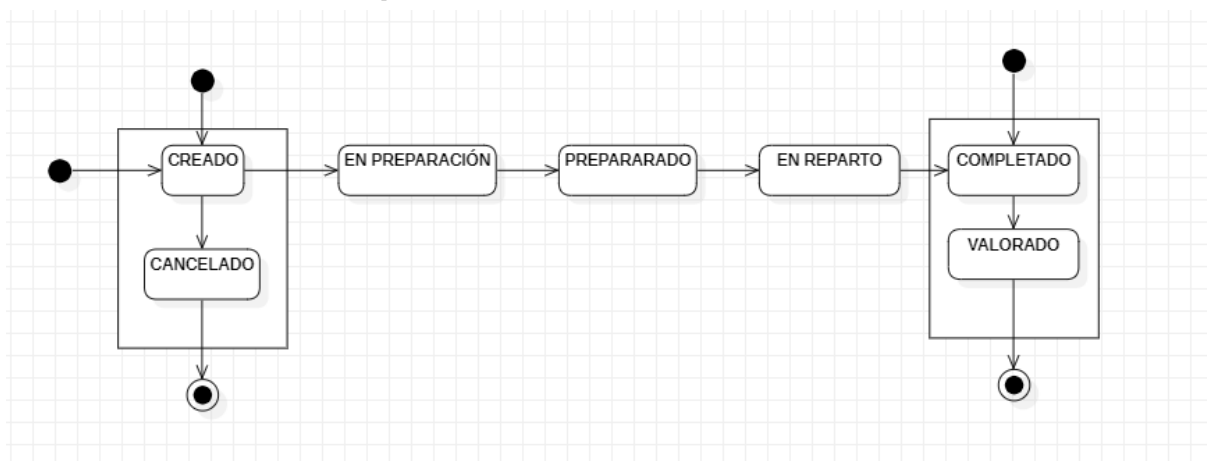
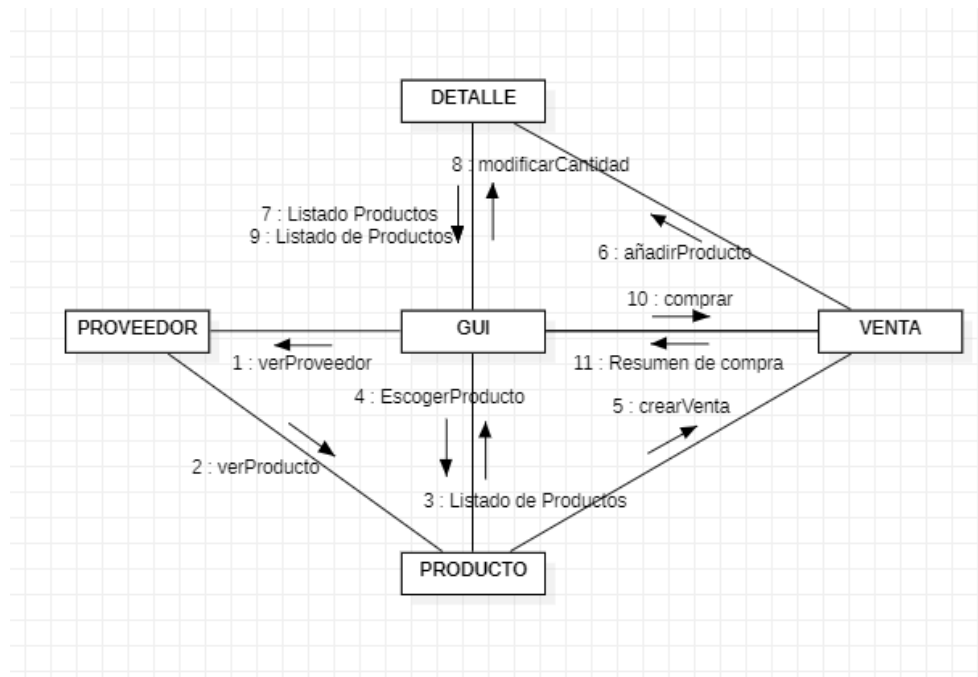
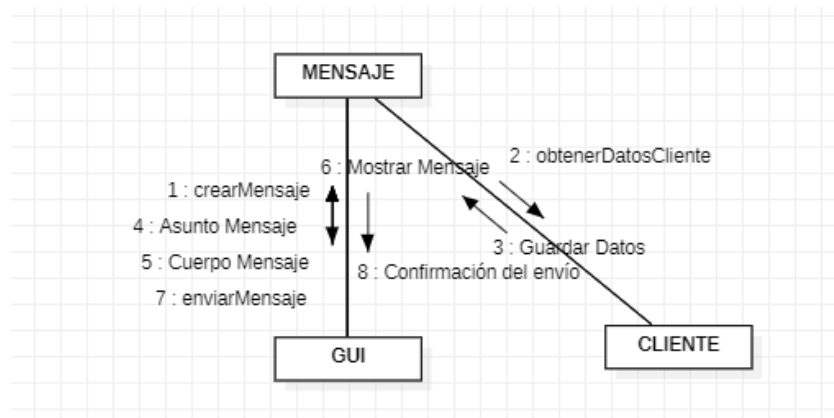


DIAGRAMA DE COLABORACIÓN

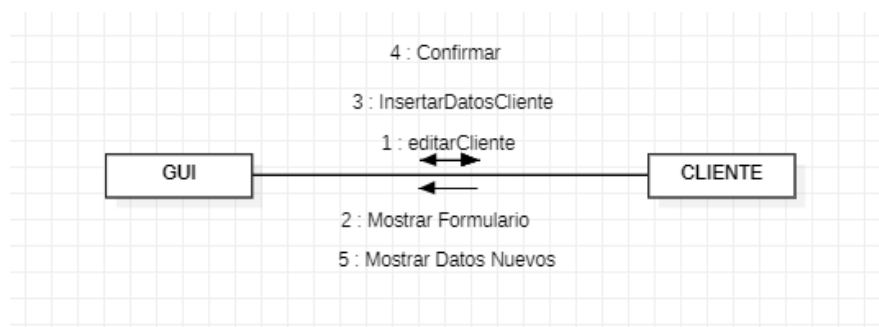
Crear pedido



Crear mensaje



Editar perfil



FUNCIONES POR TIPO DE USUARIO

Clientes

Los clientes son usuarios registrados en nuestra plataforma que tienen acceso a una variedad de funciones para facilitar su experiencia de compra. Pueden realizar compras de productos de diferentes tiendas, explorar y filtrar tiendas según categorías y tiempo.

Pueden ver el historial de pedidos, permitiéndoles hacer un seguimiento de sus compras anteriores junto a las actuales. Incluso disponen de una sección para usar códigos promocionales.

Este perfil puede actualizar su información personal como lo son sus datos de contacto. Además, pueden realizar pagos en línea de manera segura, solicitar devoluciones o reembolsos gracias al envío de mensajes al soporte.

Proveedores

Nuestra plataforma ofrece a los proveedores la oportunidad de mostrar y vender sus productos a través de un medio digital. Los proveedores pueden agregar nuevos productos a la plataforma, así como actualizar la información de los productos existentes, como su precio.

También son responsables de gestionar los pedidos de los clientes, asegurándose de cumplir con ellos de manera eficiente ya que estos pueden aceptar o no los pedidos por motivos de la empresa (falta de stock) y modificar el estado del pedido. Además pueden visualizar los pedidos que han recibido en un historial por fechas.

Riders

Los riders desempeñan un papel crucial en nuestro sistema de entregas. Una vez que se les asigna un pedido, los riders pueden acceder a información detallada sobre la entrega, como la dirección de entrega tanto del proveedor como del cliente. Para poder realizar entregas deben estar en un estado disponible que puede cambiar.

Pueden comunicarse directamente con los clientes para coordinar la entrega y actualizar el estado de la misma en tiempo real. Los riders también reciben notificaciones sobre cambios en el estado de la entrega o nuevas asignaciones. Además de su papel en la entrega, los riders pueden acceder a su historial de entregas, lo que les permite evaluar su rendimiento y mejorar continuamente su servicio.

Administrador

El administrador puede gestionar usuarios y otro tipo de gestiones como ventas, mensajes y productos. Los usuarios pueden ser modificados algunos de sus campos, siempre respetando su privacidad, además de ser eliminados.

Las ventas disponen de un filtro avanzado para ver en detalle la información de cada venta y una búsqueda exacta de la venta. Se puede eliminar para solucionar errores.

Las consultas o mensajes se pueden ver y responder por medio de correo electrónico personalizado. La forma más común de resolver consultas.

Los productos se pueden visualizar o eliminar en el caso de no cumplir las normas de DailyGo o por otros motivos.

FASES DE PRUEBAS

VALIDACIÓN DE DATOS

En primer lugar se han hecho pruebas de la validación de datos que el usuario puede introducir en los formularios para evitar que entren caracteres que puedan suponer un peligro para la base de datos o no cumpla los requisitos del diseño de esta. También hemos tenido en cuenta por medio de expresiones regulares que los datos coinciden con lo pedido. Datos validados con requisitos específicos:

- Correos electrónicos.
- DNIs.
- CIFs.
- Teléfonos.
- Direcciones.
- Tarjetas de crédito (VISA y MasterCard).

VALIDAR CONSULTAS

Cada consulta ha sido validada previamente por medio de pruebas usando extensiones de Visual Studio Code o por medio PHPMYADMIN para comprobar que la devolución de información es la esperada.

También se ha comprobado que su formato a JSON de las respuestas cumple con las necesidades y no genera errores al transformar arrays a JSON. Incluyendo mensajes en caso de error.

CORRECTO FUNCIONAMIENTO DE CADA FUNCIÓN

A nivel global:

- ☒ Validación de sesiones para limitar el acceso.
- ☒ Inicio de sesión de cada tipo de usuario.
- ☒ Registro de cada tipo de usuario.
- ☒ Correcto funcionamiento de cada enlace (navbars, footers y otro tipos de enlaces).

Clientes:

- ☒ Búsqueda correcta de las tiendas.
- ☒ Búsqueda por filtros de las tiendas.
- ☒ Envío de mensajes al administrador
- ☒ Obtener datos de la tienda de forma personalizada (productos, información y mapa).
- ☒ Funcionamiento correcto del carrito (cambiar productos, cálculo y añadir nuevos).

- ☒ Creación de pedidos, cálculos de costes y uso de descuentos por medio de códigos.
- ☒ Mostrar historial de pedidos con filtro de fecha.
- ☒ Valorar pedidos al ser entregados.
- ☒ Ver en tiempo real los estados del pedido.
- ☒ Modificar datos del usuario (nombre, apellidos, correo, teléfono y contraseña)
- ☒ Permitir cerrar sesión.

Riders:

- ☒ Modificar la disponibilidad para permitir recibir encargos de entrega de pedidos. Además de cambiar a ocupado cuando esté realizando un pedido.
- ☒ Visualizar datos del pedido.
- ☒ Modificar el estado del pedido: 'Completado' y 'En camino'.
- ☒ Visualizar mapa para encontrar la localización del proveedor.
- ☒ Ver historial de pedidos realizados con filtro de fecha.
- ☒ Modificar datos del usuario (nombre, apellidos, correo, teléfono y contraseña)
- ☒ Cerrar sesión.

Proveedores:

- ☒ Aceptar o rechazar pedidos
- ☒ Visualizar datos sobre el pedido: productos, riders, estado...
- ☒ Poder cambiar el estado del pedido (preparado y aceptado/cancelado).
- ☒ Visualizar los productos y pedidos que han sido adquiridos en local y filtrar por fecha.
- ☒ Actualizar los datos relativos al proveedor (razón social, teléfono, dirección, contraseña, correo e imágenes)
- ☒ Añadir nuevos productos y modificar los existentes.
- ☒ Cerrar sesión.

Administrador:

- ☒ Iniciar sesión.
- ☒ Ver pedidos.
- ☒ Filtrar pedidos.
- ☒ Eliminar pedidos.
- ☒ Ver mensajes.
- ☒ Responder mensajes por correo.
- ☒ Ver productos.
- ☒ Filtrar productos.
- ☒ Borrar productos.
- ☒ Ver clientes.

- ☒ Filtrar clientes.
- ☒ Borrar clientes.
- ☒ Modificar clientes.
- ☒ Ver proveedores.
- ☒ Filtrar proveedores.
- ☒ Borrar proveedores.
- ☒ Modificar proveedores.
- ☒ Ver riders.
- ☒ Filtrar riders.
- ☒ Borrar riders.
- ☒ Modificar rider.
- ☒ Cerrar sesión.

USER TESTING

Para ayudar al correcto desarrollo de la aplicación hemos utilizado personas ajenas al proyecto para que prueben la aplicación y descubrir los problemas que supone utilizarla Hemos corregido:

- ☒ Cambio en el formato responsive de las navbar.
- ☒ Modificación de los tamaños de letra para que no afecte el responsive.
- ☒ Mejoras en el feedback del carrito a la hora de adquirir productos de otras tiendas. Generando un aviso al usuario.
- ☒ Contraste para diferenciar significativamente entre el administrador y el resto de usuarios.

CREACIÓN TEÓRICA DE PASARELA DE PAGO

Guía de creación de la pasarela de pago de DailyGo

1. Crear una cuenta en PayPal: Si aún no tienes una cuenta en PayPal, debes crear una en su sitio web oficial. Sigue los pasos proporcionados para completar el registro.
2. Obtener las credenciales de API: Una vez que hayas creado tu cuenta de PayPal, necesitarás obtener las credenciales de la API para integrar la pasarela de pago en tu proyecto de DailyGo. Inicia sesión en tu cuenta de PayPal y sigue estos pasos:
 - Haz clic en el ícono de engranaje en la esquina superior derecha de la página de inicio de PayPal.
 - Selecciona "Perfil y configuración" en el menú desplegable.
 - Haz clic en "Mis herramientas de venta" en la sección "Herramientas de venta" del menú izquierdo.
 - En la sección "API de PayPal", haz clic en "Actualizar" junto a "Credenciales de API".
 - Selecciona "Crear credenciales de API" y sigue las instrucciones para generar un conjunto de credenciales de API.
 - Una vez creadas las credenciales, podrás encontrarlas en la sección "Credenciales de API" en la página "Herramientas de venta".
3. Integrar la pasarela de pago de PayPal: Existen varias formas de integrar la pasarela de pago de PayPal en tu proyecto web de DailyGo, pero la opción más sencilla es utilizar una biblioteca o un SDK de PayPal. Sigue estos pasos:
 - Elige el SDK adecuado para tu proyecto. PayPal ofrece bibliotecas de SDK para varios lenguajes de programación, como PHP, Java, Python, Ruby, entre otros. Asegúrate de seleccionar el SDK compatible con DailyGo.
 - Descarga e instala el SDK en tu proyecto, siguiendo las instrucciones de instalación proporcionadas por PayPal. Asegúrate de configurarlo correctamente utilizando las credenciales de API que obtuviste anteriormente.
 - Crea una instancia de la pasarela de pago de PayPal en tu proyecto. Este paso te permitirá configurar y controlar las opciones de pago.
4. Configurar las opciones de pago: Una vez que hayas creado la instancia de la pasarela de pago de PayPal en tu proyecto, es hora de configurar las opciones de pago. Asegúrate de establecer el monto de la transacción, la moneda y el método de pago que desees admitir, como tarjeta de crédito o cuenta de PayPal.

5. Implementar la pasarela de pago en tu proyecto: En esta etapa, deberás implementar la pasarela de pago de DailyGo en tu sitio web o aplicación. Dependiendo del SDK que hayamos elegido, esto puede implicar agregar código de programación o utilizar una interfaz de usuario predefinida proporcionada por PayPal.
6. Prueba la pasarela de pago en un entorno de pruebas: Antes de implementar la pasarela de pago en producción, es fundamental realizar pruebas exhaustivas en un entorno de pruebas. PayPal ofrece un entorno de pruebas que te permite simular transacciones y asegurarte de que todo funcione correctamente.
7. Sigue las mejores prácticas de seguridad: Para proteger la seguridad de los datos de tus clientes y prevenir posibles fraudes, asegúrate de seguir las mejores prácticas de seguridad recomendadas por PayPal. Esto incluye mantener seguras tus credenciales de API, validar la entrada de datos, implementar medidas de seguridad adicionales como la autenticación de dos factores y mantener actualizado y seguro tu sitio web.
8. Monitorea el rendimiento de la pasarela de pago: Una vez que hayas implementado la pasarela de pago de DailyGo, es importante monitorear regularmente su rendimiento. Esto te permitirá identificar cualquier problema o mejorar el proceso de pago para brindar una experiencia fluida a los usuarios.
9. Lanza DailyGo: Después de realizar pruebas exhaustivas y asegurarte de que la pasarela de pago funciona correctamente, estás listo para lanzar tu proyecto DailyGo y comenzar a aceptar pagos a través de PayPal.

DESPLIEGUE

Utilizaremos dos máquinas virtuales un Windows Server y un Ubuntu:

WINDOWS SERVER

Esta máquina cumple la función de resolver los nombres de dominio para acceder a la aplicación. Usaremos de nombre de dominio:

- “dailygo.com” que accederá al “index.html”
- “admin.dailygo.com” para acceder al “inicio_sesion_administrador.html”.

Creamos una zona de búsqueda directa y configuraremos un servidor caché. Además de crear una zona de resolución directa para resolver las direcciones anteriores. También crearemos alias, puntos y CNAMEs.

UBUNTU

Es la encargada de alojar nuestro servidor web y el conjunto de documentos que componen la aplicación. Se instalarán los siguientes paquetes para que funcione el sitio:

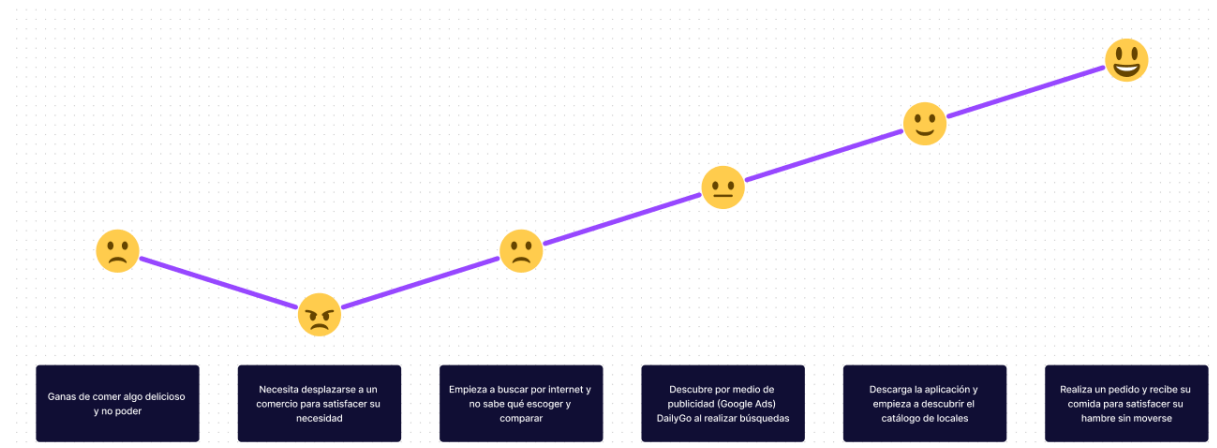
- Servidor web Apache para gestionar el sitio y poder guardar la aplicación
- PHP 8 para ejecutar este lenguaje y conectarnos con la base de datos con el fin de realizar consultas.
- Gestor de bases de datos MySQL/MariaDB para almacenar los datos de la tienda.
- PHPMYADMIN nos ayudará a agilizar consultas en la base de datos y a importar datos. Además de gestionar usuarios.

POTENCIALES USUARIOS: “UX PERSONAS”

Nombre	Marta García
Edad	28 años
Ocupación	Diseñadora gráfica freelance
Descripción personal	Marta García es una diseñadora gráfica freelance dedicada y creativa. Como profesional independiente, pasa la mayor parte de su tiempo trabajando desde su estudio en casa o en espacios de coworking. Marta valora la eficiencia y la calidad en cada aspecto de su vida, incluyendo la comida que consume diariamente.
Motivaciones y objetivos	Marta busca una solución que le permita ahorrar tiempo y esfuerzo en la preparación de sus comidas diarias. Desea tener acceso a una variedad de opciones gastronómicas de calidad y poder disfrutar de una experiencia de pedido a domicilio rápida y conveniente. Para ella, es importante poder elegir entre una amplia gama de restaurantes y recibir la comida en la comodidad de su hogar o estudio de trabajo. .
Frustraciones	Se siente frustrada cuando no puede encontrar opciones de comida a domicilio que se adapten a sus gustos y preferencias. Le molesta cuando la plataforma de pedido no muestra información detallada sobre los platos, los ingredientes o las opciones de personalización. Además, le preocupa recibir pedidos incorrectos o con retrasos, lo cual afecta su productividad y la calidad de su experiencia gastronómica. Marta desea una plataforma de pedidos a domicilio fácil de usar, confiable y que le brinde una experiencia gastronómica satisfactoria.
Conclusión	<p>DailyGo se esfuerza por satisfacer las necesidades de Marta proporcionándole una experiencia de pedido a domicilio eficiente y conveniente. La plataforma ofrece una amplia selección de restaurantes de calidad, con información detallada sobre los platos, ingredientes y opciones de personalización.</p> <p>Marta puede navegar fácilmente por la aplicación, elegir sus comidas favoritas y realizar pedidos con tan solo unos pocos clics. Además, DailyGo se preocupa por la puntualidad y la calidad de los pedidos, asegurándose de que Marta reciba exactamente lo que solicitó y en el tiempo acordado. La aplicación también brinda opciones de pago seguras y diversas para mayor comodidad. Con DailyGo, Marta puede disfrutar de una experiencia gastronómica satisfactoria y ahorrar tiempo en la preparación de sus comidas diarias.</p>

Nombre	Pedro Pérez
Edad	34 años
Ocupación	Informático
Descripción personal	Pedro Pérez es un informático dedicado y entusiasta que trabaja de forma remota. Como teletrabajador, pasa la mayor parte de su tiempo en casa, concentrado en sus tareas y proyectos. Pedro valora la eficiencia y la comodidad en su rutina diaria, incluyendo la alimentación durante su jornada laboral.
Motivaciones y objetivos	Pedro busca una solución que le permita acceder a comidas rápidas y saludables mientras trabaja desde casa. Quiere tener la opción de elegir entre una variedad de platos y disfrutar de una experiencia de pedido a domicilio rápida y conveniente. La comodidad y la calidad de los alimentos son fundamentales para él.
Frustraciones	Se siente frustrado cuando tiene que interrumpir su trabajo para preparar comida o buscar opciones de entrega a domicilio. Le molesta cuando las plataformas de pedidos son complicadas de usar o no ofrecen una selección adecuada de comidas saludables. También le preocupa recibir pedidos incorrectos o con retrasos, lo cual afecta su productividad. Pedro busca una plataforma de pedidos a domicilio confiable y eficiente, que le brinde una experiencia gastronómica satisfactoria.
Conclusión	DailyGo se esfuerza por satisfacer las necesidades de Pedro proporcionándole una experiencia de pedido a domicilio eficiente y conveniente. La aplicación le ofrece una amplia variedad de opciones gastronómicas saludables, con platos adecuados para su estilo de vida y preferencias. Pedro puede acceder fácilmente a la aplicación, explorar diferentes restaurantes y realizar pedidos de forma rápida y sencilla, sin interrumpir su trabajo.

VIAJE DEL USUARIO



¿QUÉ HEMOS APRENDIDO?

4. Conclusiones

Durante el desarrollo de DailyGo, nos enfrentamos a diversos desafíos y aprendimos muchas lecciones importantes en el ámbito del desarrollo web. Aquí hay algunas de las cosas que aprendimos durante el proceso:

Arquitectura y tecnologías: Durante el desarrollo de DailyGo, tuvimos que tomar decisiones clave sobre la arquitectura y las tecnologías a utilizar. Aprendimos a evaluar diferentes opciones y seleccionar las más adecuadas para construir una aplicación web eficiente y escalable. Exploramos frameworks y bibliotecas y comprendimos cómo aprovechar al máximo estas herramientas para crear una interfaz de usuario dinámica y receptiva.

Gestión de datos: Un componente fundamental de DailyGo fue la gestión de datos. Aprendimos a diseñar y desarrollar una base de datos sólida para almacenar información sobre restaurantes, menús, usuarios y pedidos. Utilizamos tecnologías de base de datos como MySQL y aprendimos a optimizar consultas y garantizar un rendimiento eficiente.

Integración de servicios de terceros: DailyGo se benefició de la integración con servicios externos, como sistemas de pago en línea y servicios de geolocalización para el seguimiento de las entregas. Aprendimos a utilizar APIs y SDKs para integrar estos servicios en nuestra aplicación y asegurarnos de que funcionaran de manera confiable y segura.

Seguridad: Durante el desarrollo de DailyGo, la seguridad fue una preocupación constante. Aprendimos a implementar prácticas de seguridad recomendadas, como la protección contra ataques de inyección de código, la autenticación y autorización de usuarios, y el cifrado de datos sensibles. También nos familiarizamos con las vulnerabilidades comunes en el desarrollo web y aprendimos a mitigarlas adecuadamente.

Mejora continua: Durante el desarrollo de DailyGo, aprendimos a recopilar comentarios de los usuarios y utilizarlos para iterar y optimizar la aplicación. Esto implicó realizar ajustes en la interfaz de usuario, agregar nuevas funcionalidades y optimizar el rendimiento general del sistema.

En resumen, el desarrollo de DailyGo fue una oportunidad para aprender sobre la arquitectura y tecnologías web, la gestión de datos, la integración de servicios de terceros, la seguridad y la mejora continua. Estas lecciones nos han permitido adquirir habilidades valiosas en el desarrollo web y nos han preparado para enfrentar nuevos desafíos en proyectos futuros.

MEJORAS EN DAILYGO

5. Trabajo futuro

Desde la perspectiva de desarrollo web, hay varias formas en las que podríamos enfocar el futuro de DailyGo para mejorar su funcionalidad y experiencia de usuario. Aquí hay algunas ideas:

Mejora de la velocidad y rendimiento: La optimización del rendimiento es un aspecto esencial en el desarrollo web. Podríamos realizar análisis exhaustivos para identificar y mejorar cualquier cuello de botella en el rendimiento de la aplicación. Esto podría incluir la reducción de tiempos de carga, la optimización de consultas de bases de datos y la implementación de técnicas de almacenamiento en caché. Mejorar la velocidad y la capacidad de respuesta garantiza una experiencia fluida para los usuarios y fomentaría su participación continua.

Integración de pagos digitales: Una funcionalidad adicional que podríamos considerar es la integración de pasarelas de pago digitales populares. Esto permitiría a los usuarios realizar pagos en línea de forma segura y conveniente al realizar sus pedidos en DailyGo. Podríamos explorar la integración con servicios como PayPal, Stripe u otros métodos de pago digital ampliamente utilizados siempre en una perspectiva de posible mantenimiento económico de estos sistemas.

Personalización de la experiencia del usuario: Ser capaz de ofrecer una experiencia personalizada a los usuarios es algo muy valorado. Podríamos implementar funcionalidades como recomendaciones personalizadas de restaurantes o platos basadas en las preferencias y el historial de pedidos de cada usuario. Esto implicaría la recopilación y el análisis de datos para generar recomendaciones precisas y relevantes, mejorando así la experiencia general del usuario en DailyGo.

Implementación de notificaciones en tiempo real: Para mantener a los usuarios actualizados sobre el estado de sus pedidos y promociones especiales, podríamos considerar la implementación de notificaciones en tiempo real. Esto permitiría enviar notificaciones instantáneas a través de mensajes push o correos electrónicos, manteniendo a los usuarios informados sobre el progreso de sus pedidos y ofreciendo incentivos o descuentos exclusivos.

Expansión de la plataforma: Si bien DailyGo se centra actualmente en el servicio de entrega de comida, podríamos explorar oportunidades para expandir la plataforma. Considerando la incorporación de nuevos servicios, convirtiendo a DailyGo en una plataforma integral para satisfacer diversas necesidades de entrega.

En conclusión, para enfocar el futuro de DailyGo desde una perspectiva escalable, es crucial mejorar la experiencia del usuario (UX), optimizar el rendimiento con sistemas más modernos, implementar funcionalidades personalizadas, incorporar servicios de pago digitales, utilizar notificaciones en tiempo real y explorar oportunidades de expansión. Estas mejoras impulsarían el crecimiento y la satisfacción de los usuarios, posicionando a la marca como una plataforma líder en el sector de entregas en línea.

INFORMACIÓN ADICIONAL

6. Anexos

HERRAMIENTAS

Para ampliar la utilidad de la aplicación y mejorar su desarrollo hemos utilizado algunas herramientas y tecnologías externas: Figma, Github, TailwindCSS y Leaflet API.

FIGMA

Este programa nos permite diseñar toda la interfaz de usuario y prototipar la aplicación. Esto nos sirve para tener una estructura UI para implementarla a código y hacer buenas prácticas. Prototipar el sitio ayuda a mejorar la experiencia de usuario y a obtener un feedback para realizar testeos antes de codificar.

Enlace: <https://www.figma.com/>

GITHUB

Es el repositorio donde hemos almacenado el código. Dado a su coste y facilidad de uso ha sido la mejor opción para poder mantener versiones del código y colaborar entre los integrantes del grupo.

Enlace: <https://github.com/>

TAILWIND CSS

Este framework de CSS nos permite desarrollar la parte Front-End más rápida y personalizada. Pudiendo personalizar cada propiedad de CSS, colores y otros atributos para aplicar la identidad exacta de DailyGo.

Su funcionamiento es muy similar a Bootstrap con la diferencia de incluir más propiedades, mayor rango de color y posibilidad de customizar todo. Pudiendo crear paletas de colores infinitas y usar una sintaxis más similar a CSS que otros frameworks o librerías de la competencia.

El uso principal de esta herramienta ha sido tener un responsive adecuado y maquetar el sitio lo más idéntico a los mock-ups realizados. Usando grid para

Enlace a la documentación: <https://tailwindcss.com/docs/installation>

LEAFLET API

Esta API es la encargada de poder mostrar los mapas de las tiendas. Funciona por medio un Host Virtual al cual se le envía coordenadas del sitio y devuelve un mapa con la posición. En nuestro caso hemos editado la respuesta por el DOM para personalizar los botones del mapa a la identidad de la marca y además incluir la imagen de cada proveedor correspondiente en el mapa

Enlace a la documentación: <https://leafletjs.com/download.html>

SWIPERJS

SwiperJS es una biblioteca JavaScript de código abierto que proporciona un conjunto completo de funcionalidades para crear y controlar componentes de slider (deslizadores) y carruseles interactivos en aplicaciones web. SwiperJS facilita la creación de presentaciones de contenido que se pueden desplazar horizontal o verticalmente, lo que permite mostrar una serie de elementos (como imágenes, tarjetas, textos, etc.) de manera atractiva y dinámica.

Enlace a la documentación: <https://swiperjs.com/get-started>

StarUML

StarUML es una herramienta de modelado UML (Unified Modeling Language) que proporciona una interfaz gráfica intuitiva para crear y visualizar diferentes tipos de diagramas. Con StarUML, los desarrolladores y analistas pueden representar de manera clara y concisa la estructura, comportamiento y relaciones de un sistema de software.

StarUML permite crear y editar diagramas de clases, diagramas de objetos, diagramas de casos de uso, diagramas de secuencia, diagramas de actividad, diagramas de estado, diagramas de componentes y diagramas de despliegue. Cada tipo de diagrama proporciona una representación visual única de diferentes aspectos del sistema, lo que facilita la comprensión y la comunicación entre los miembros del equipo de desarrollo.

Enlace a la documentación: <https://docs.staruml.io/>

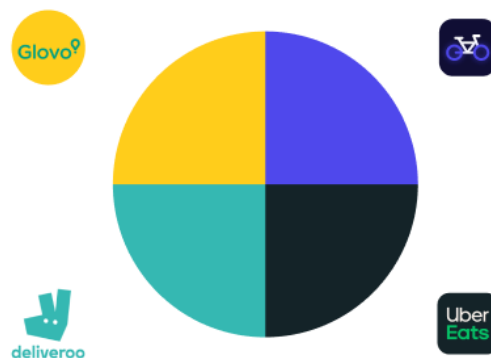
IDENTIDAD DE LA MARCA

Para la identidad de DailyGo hemos querido representar una marca alegre, activa y positiva. Su eslogan “Pedidos ultrarrápidos” ya marca el camino de la entidad. Para ello hemos utilizado este “moodboard” como guía:



COLOR

Usamos colores fuertes para mostrar la energía y para diferenciarnos de la identidad de la competencia.



Para ello usamos los siguientes:

GoNavy

#100E34
#1C1A3E
#282648
#403E5D
#585671
#706E85
#9F9FAE
#B7B7C2
#CFCFD6

GoIndigo

#312E81
#3730A3
#4338CA
#4F46E5
#6366F1
#818CF8
#A5B4FC
#C7D2FE
#E0E7FF

GoLight

#EAB308
#FACC15
#FDE047
#FEF08A
#FEF9C3

(Paleta principal)

GoGreen

#16A34A
#22C55E
#4ADE80
#86EFAC
#BBF7D0
#DCFCE7

GoRed

#DC2626
#EF4444
#F87171
#FCA5A5
#FECACA
#FEE2E2

(Paleta complementaria)

TIPOGRAFÍA

La tipografía utilizada es la ‘Work Sans’ una tipografía sin serifas (sans-serif) para transmitir limpieza y dinamismo. Esta fuente tiene unas leves curvas ligeras para mostrar esa diversión y evitar un camino. Además al usar manchas más grandes como ocurre en “bold” o “semi-bold” podemos ver que el texto toma más fuerza y ayuda a destacar con energía los elementos que deseemos.

Work Sans

 (por Google)

¡Pedidos ultrarrápidos!

Más información en www.dailygo.com

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
12345678910?¡!'"`+-=º/

LOGOTIPOS E IDENTIFICADORES

The logo for DailyGo, featuring the word 'Daily' in a dark blue sans-serif font and 'Go' in a lighter blue sans-serif font.

(Logotipo principal)

RECURSOS



(Icono-logotipo)

7. Bibliografía

PHP: Documentación oficial de PHP - <https://www.php.net/manual/es/>

JavaScript: Documentación oficial de JavaScript (MDN) - <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>

HTML: Documentación oficial de HTML (MDN) - <https://developer.mozilla.org/es/docs/Web/HTML>

Leaflet: Documentación oficial de Leaflet - <https://leafletjs.com/reference-1.7.1.html>

Stackoverflow: Foro de ayuda - <https://stackoverflow.com/>

Tailwind CSS: Documentación oficial de Tailwind CSS - <https://tailwindcss.com/docs>

CSS: Documentación oficial de CSS (MDN) - <https://developer.mozilla.org/es/docs/Web/CSS>

SwiperJS: Documentación oficial de swiperJS - <https://swiperjs.com/get-started>

JSDoc: vídeo para aprender a usar JSDoc y documentar código JavaScript - <https://www.youtube.com/watch?v=U329pKWKqWw>

Python para backend: vídeo explicativo de Python - <https://www.youtube.com/watch?v=NacGWWmLxos>

Curso de TailwindCSS: vídeo explicativo de TailwindCSS - <https://www.youtube.com/watch?v=h5HQVHTpeHs>

Tutorial API Fetch de Ajax: tutorial del uso de la Api Fetch - <https://www.youtube.com/watch?v=dYB1ULGGmfk>

Unsplash: banco de imágenes - <https://unsplash.com/es>

Pexels: banco de imágenes - <https://www.pexels.com/es-es/>

Adobe stock: banco de imágenes - <https://stock.adobe.com/es>