
UNIVERSIDAD TECNOLÓGICA DE XICOTEPEC DE JUÁREZ

Sistema de Gestión de Farmacia(Front-end).

Alumnos:

(220296) José Daniel Loza Marín.

(220186) Luis Abdiel Rivera Gayosso.

(220336) Griselda Cabrera Franco.

(220864) Esaú Vargas Álvarez.

(220418) Jareni Gómez Juan.

Grado: Octavo Cuatrimestre

Grupo: A

Carrera: Ingeniería en Desarrollo y Gestión de Software.

Materia: Administración de Base de datos.

Asesor: Héctor Valderrabano González.

Xicotepec de Juárez Puebla 09 de abril de 2025.

Indicé

Introducción.....	2
Tecnologías Utilizadas.	3
Características Clave.....	4
Estructura del Proyecto.....	8
Carpetas	9
Componentes:	10

Introducción.

El presente documento detalla el desarrollo del frontend del módulo de **Farmacia** del **Hospital Las Lomas**, un sistema diseñado para optimizar la gestión de medicamentos y servicios farmacéuticos, brindando una experiencia eficiente y accesible tanto para los profesionales de la salud como para los pacientes. Este proyecto tiene como objetivo principal mejorar la atención al paciente mediante una interfaz de usuario intuitiva y moderna que facilite el acceso a la información y la gestión de productos farmacéuticos.

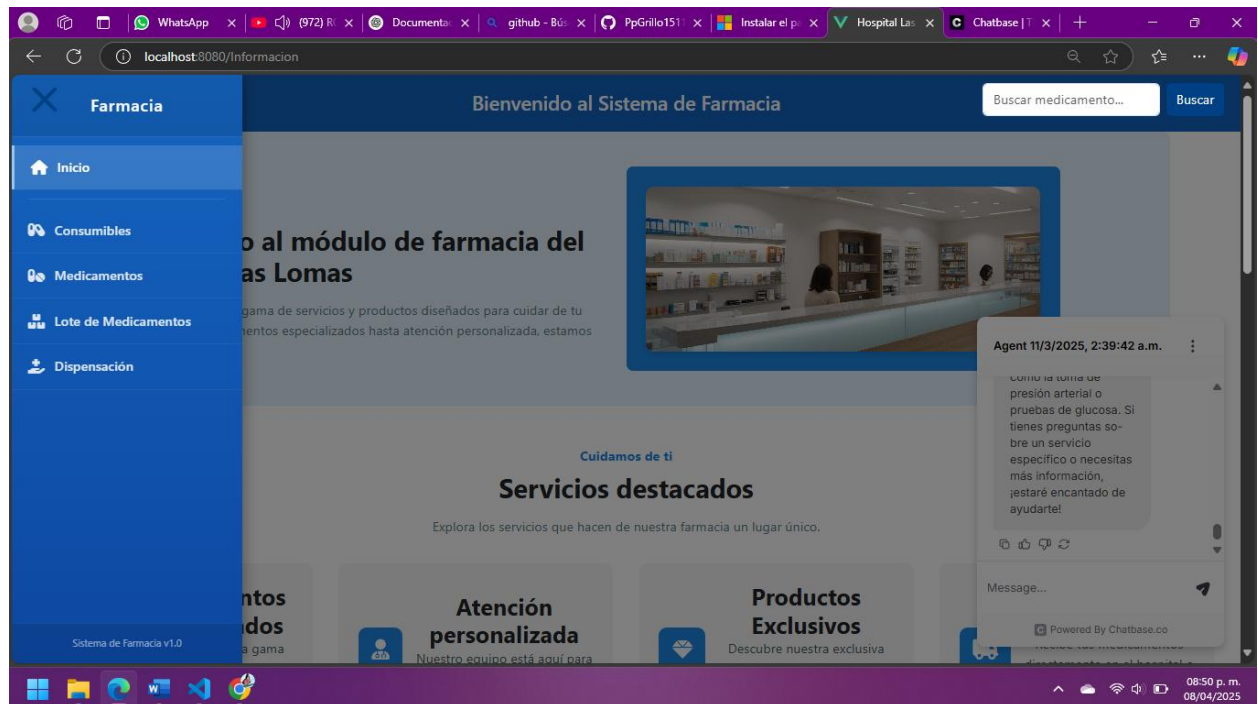
El módulo de Farmacia está construido utilizando **Vue.js**, un framework progresivo de JavaScript que permite crear interfaces de usuario interactivas y reactivas. Además, se han implementado componentes reutilizables y un diseño responsivo que garantiza una experiencia de usuario fluida en dispositivos de diferentes tamaños, desde computadoras de escritorio hasta dispositivos móviles.

Entre las principales características del sistema se encuentran:

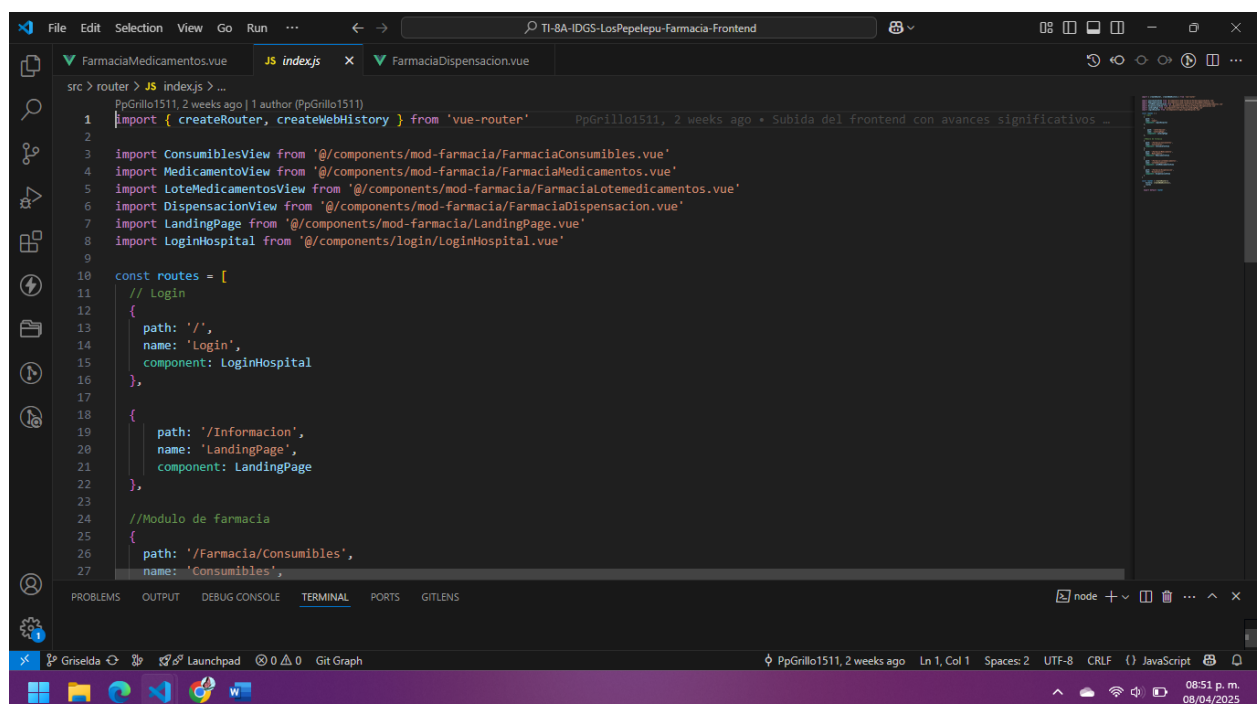
- **Visualización de Medicamentos y Consumibles:** Un catálogo exhaustivo de productos disponibles, organizado por categorías, que permite a los usuarios buscar y filtrar medicamentos según diferentes criterios.
- **Gestión de Lotes y Dispensaciones:** Herramientas para administrar y supervisar los lotes de medicamentos y sus respectivas dispensaciones, asegurando la correcta gestión del inventario.
- **Interacción con un Chatbot:** Un chatbot integrado que ofrece asistencia en tiempo real a los usuarios, facilitando respuestas a preguntas frecuentes y mejorando la atención al cliente.
- **Gráficos y Estadísticas:** Visualización de datos a través de gráficos que permiten un análisis fácil de la información relacionada con los medicamentos y el inventario, ayudando en la toma de decisiones informadas.

- **Vue.js:** Framework progresivo para construir interfaces de usuario.
- **Vue Router:** Biblioteca para gestionar las rutas de la aplicación.
- **Vuex:** Patrón de gestión de estado para aplicaciones Vue.
- **Axios:** Cliente HTTP para realizar solicitudes a la API.
- **HTML/CSS:** Para la estructura y estilo de los componentes.
- **JavaScript ES6+:** Para la lógica y funcionalidades interactivas.
- **MySQL:** Sistema de gestión de bases de datos relacional para almacenar datos estructurados.
- **MongoDB Atlas:** Servicio de base de datos en la nube para gestionar datos NoSQL.
- **Bootstrap:** Framework de código abierto para el desarrollo de interfaces web responsivas y móviles.

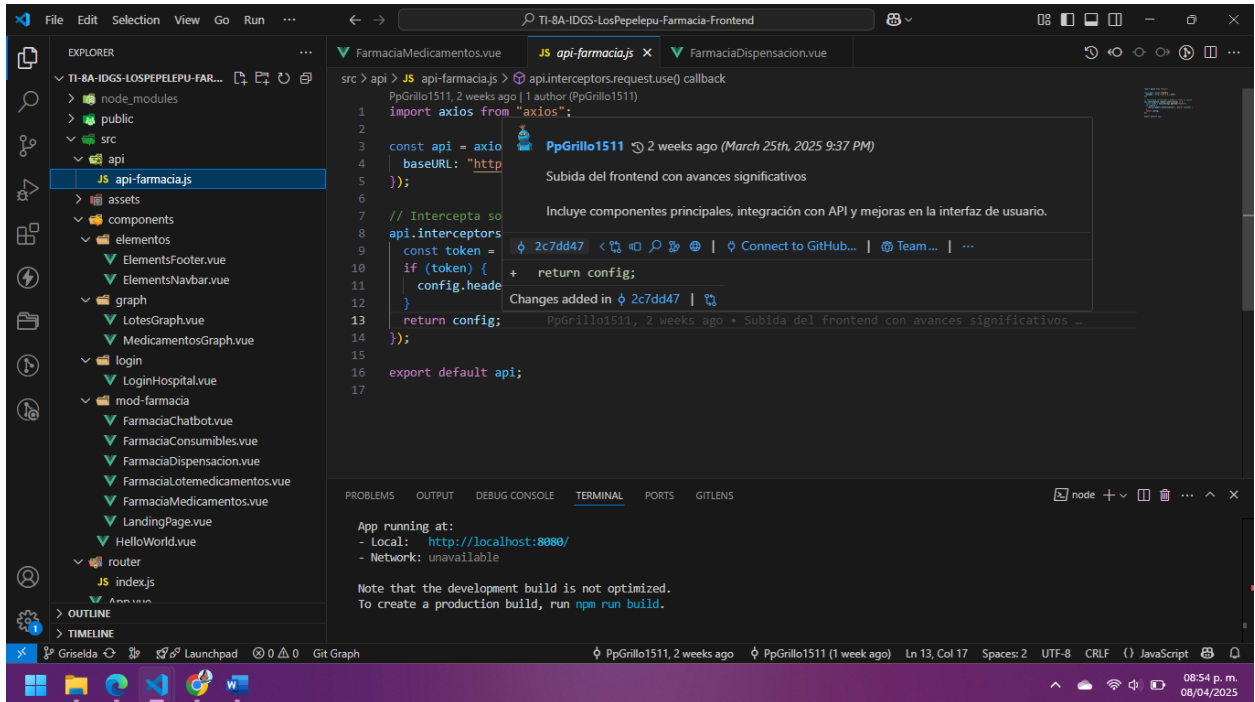
- **Interfaz Intuitiva:** Diseñada para facilitar la navegación y usabilidad.



- **Rutas Dinámicas:** Uso de Vue Router para cargar componentes dinámicamente según la ruta.



- **Gestión de Estado:** Implementación de Vuex para manejar el estado de la aplicación de manera centralizada.
- **Consumo de API:** Integración de Axios para realizar peticiones HTTP, garantizando la interacción con la base de datos del sistema.

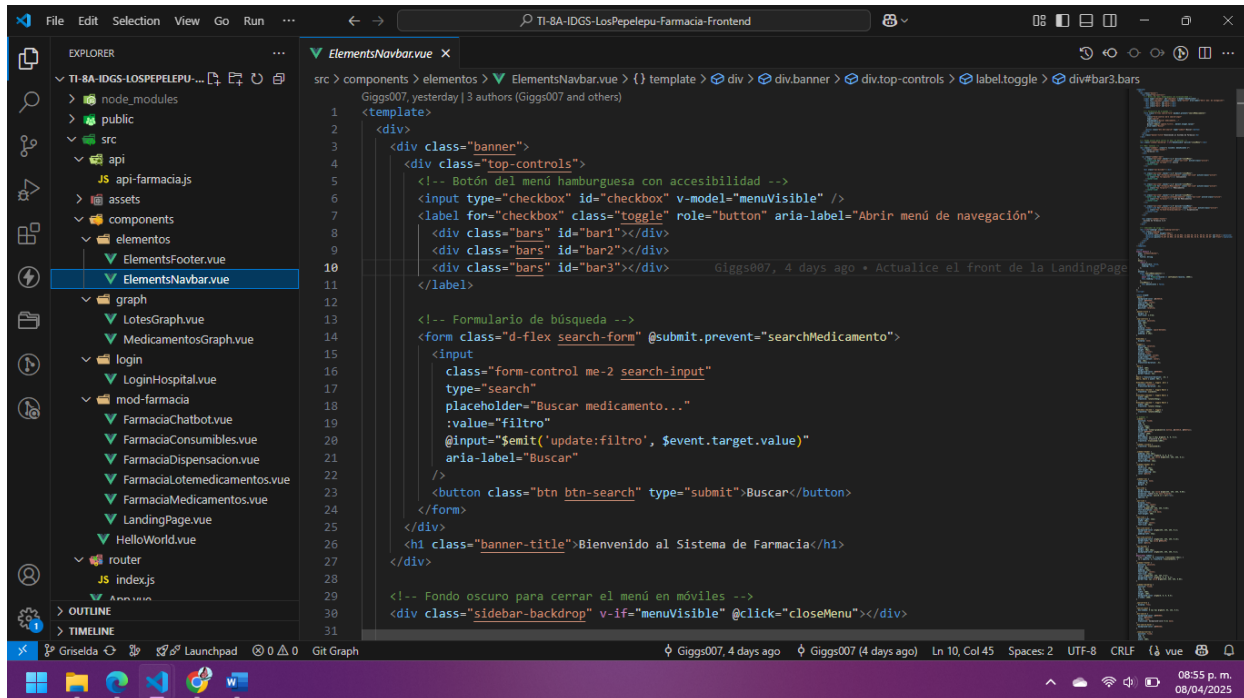


The screenshot shows a Visual Studio Code editor window with the file explorer on the left, the editor in the center, and the terminal at the bottom. The file explorer shows a project structure for 'TI-8A-IDGS-LosPepelu-Farmacia-Frontend'. The editor displays the 'api-farmacia.js' file, which contains the following code:

```
src > api > JS api-farmacia.js > api.interceptors.request.use() callback
PpGrillo1511, 2 weeks ago | 1 author (PpGrillo1511)
1 import axios from "axios";
2
3 const api = axios
4   .baseUrl("http")
5   .headers({});
6
7 // Intercepta so
8 api.interceptors
9   .const token =
10     if (token) {
11       config.head
12     }
13   .return config;
14   .});
15
16 export default api;
17
```

The terminal output shows the application running at the local address `http://localhost:8080/` and the network being unavailable. It also includes a note that the development build is not optimized and that a production build can be created by running `npm run build`.

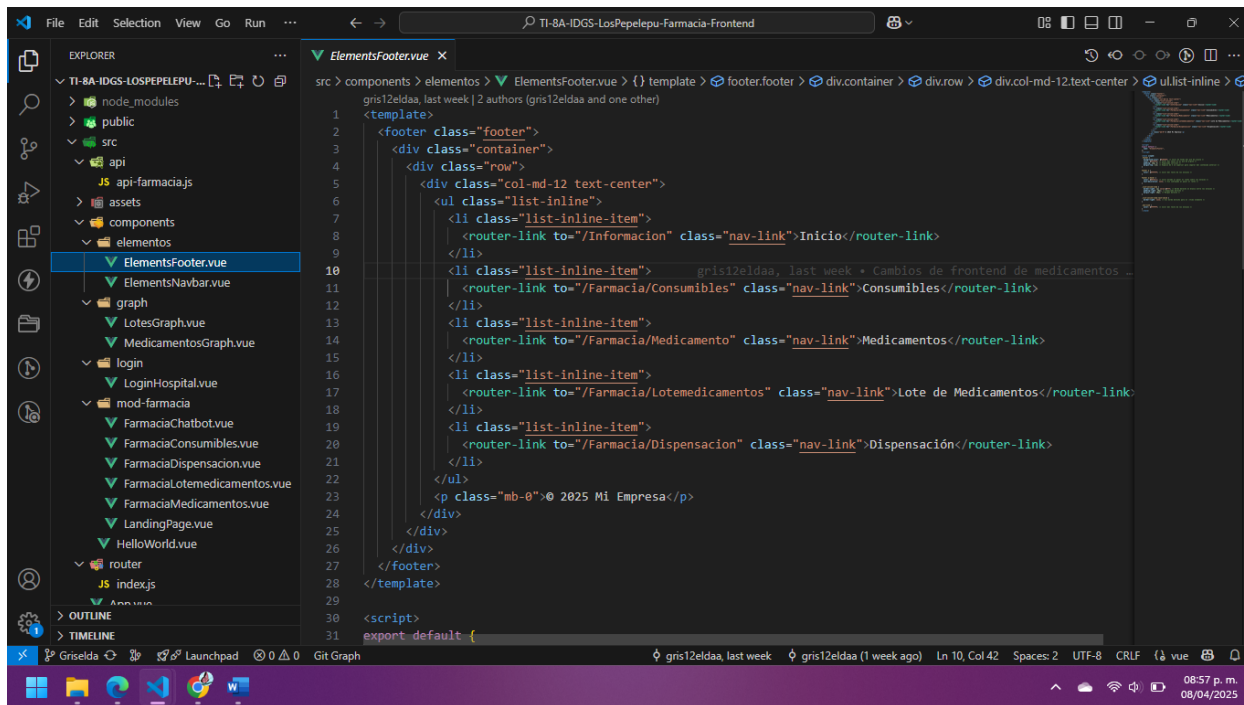
- **Componentes Reutilizables:** Estructura modular con componentes como ElementsFooter.vue, ElementsNavbar.vue, y otros que permiten un mantenimiento y escalabilidad más simples.



```

src > components > elementos > ElementsNavbar.vue > {} template > div > div.banner > div.top-controls > label.toggle > div#bar3.bars
1  Giggis007, yesterday | 3 authors (Giggis007 and others)
2  <template>
3  </template>
4  <div class="banner">
5  <div class="top-controls">
6  <!-- Botón del menú hamburguesa con accesibilidad -->
7  <input type="checkbox" id="checkbox" v-model="menuVisible" />
8  <label for="checkbox" class="toggle" role="button" aria-label="Abrir menú de navegación">
9  <div class="bars" id="bar1"></div>
10 <div class="bars" id="bar2"></div>
11 <div class="bars" id="bar3"></div>
12 </label>
13
14 <!-- Formulario de búsqueda -->
15 <form class="d-flex search-form" @submit.prevent="searchMedicamento">
16 <input
17   class="form-control me-2 search-input"
18   type="search"
19   placeholder="Buscar medicamento..."
20   :value="filtro"
21   @input="$emit('update:filtro', $event.target.value)"
22   aria-label="Buscar"
23 />
24 <button class="btn btn-search" type="submit">Buscar</button>
25 </form>
26 </div>
27 <h1 class="banner-title">Bienvenido al Sistema de Farmacia</h1>
28 </div>
29
30 <!-- Fondo oscuro para cerrar el menú en móviles -->
31 <div class="sidebar-backdrop" v-if="menuVisible" @click="closeMenu"></div>

```

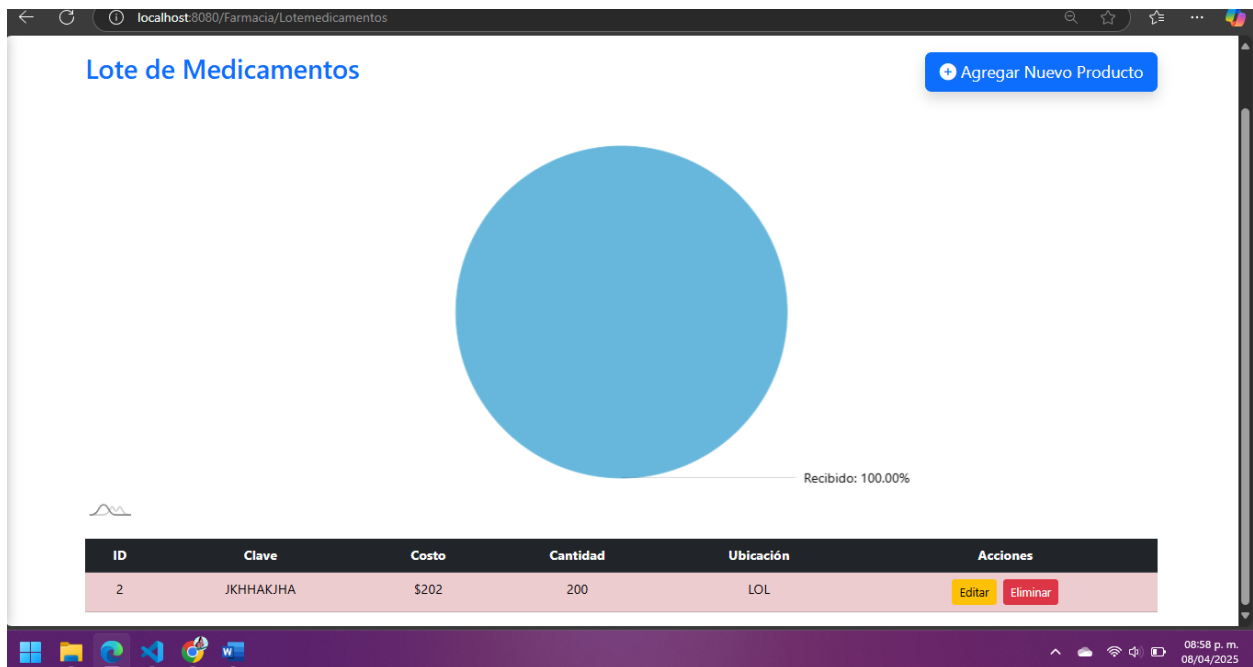
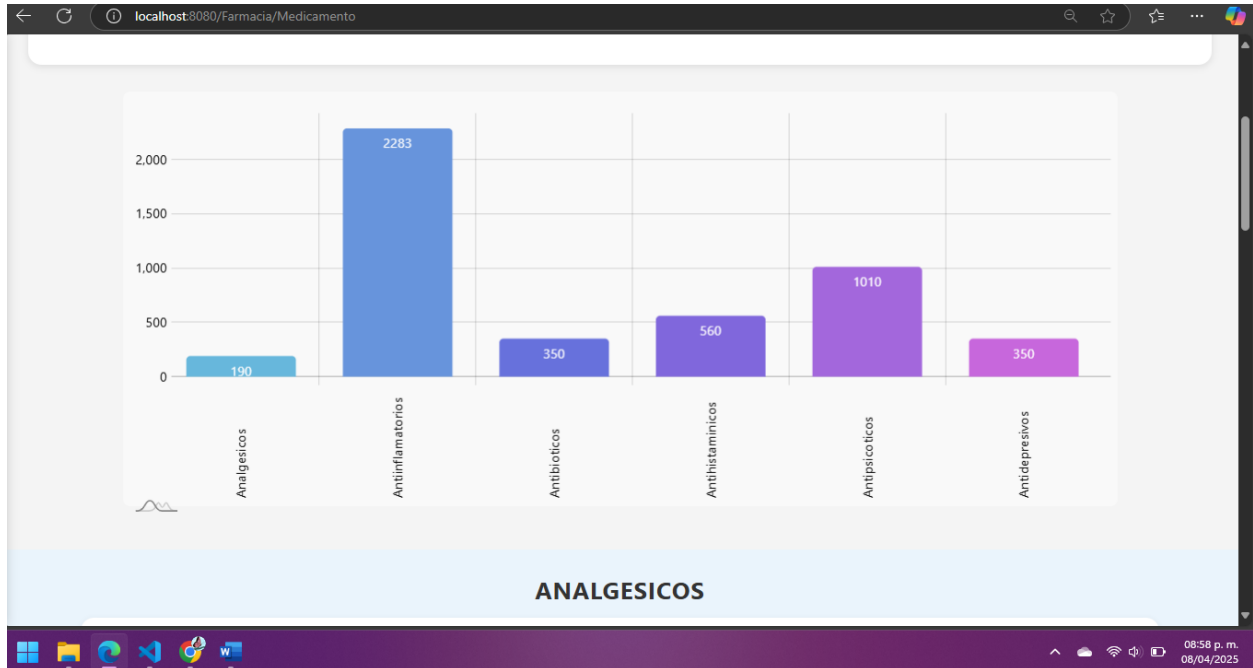


```

src > components > elementos > ElementsFooter.vue > {} template > footer.footer > div.container > div.row > div.col-md-12.text-center > ul.list-inline > li
1  gris12elda, last week | 2 authors (gris12elda and one other)
2  <template>
3  </template>
4  <div class="footer">
5  <div class="container">
6  <div class="row">
7  <div class="col-md-12 text-center">
8  <ul class="list-inline">
9  <li class="list-inline-item">
10 <router-link to="/" class="nav-link">Inicio</router-link>
11 </li>
12 <li class="list-inline-item">
13 <router-link to="/Farmacia/Consumibles" class="nav-link">Consumibles</router-link>
14 </li>
15 <li class="list-inline-item">
16 <router-link to="/Farmacia/Medicamento" class="nav-link">Medicamentos</router-link>
17 </li>
18 <li class="list-inline-item">
19 <router-link to="/Farmacia/Lotemedicamentos" class="nav-link">Lote de Medicamentos</router-link>
20 </li>
21 <li class="list-inline-item">
22 <router-link to="/Farmacia/Dispensacion" class="nav-link">Dispensación</router-link>
23 </li>
24 </ul>
25 <p class="mb-0">© 2025 Mi Empresa</p>
26 </div>
27 </div>
28 </div>
29 </div>
30 </template>
31 <script>
32 export default {

```

- **Gráficas y Análisis:** Componentes como LotesGraph.vue y MedicamentosGraph.vue para visualización de datos.



Estructura del Proyecto.

La estructura del frontend del proyecto es la siguiente:

Carpeta/Archivo	Descripción
TI-8A-IDGS-LosPeeplepu-Farmacia/	Raíz del proyecto.
src/	Código fuente del frontend.
api/	Métodos de API para consumir datos.
api-farmacia.js	Archivo de conexión a la API.
assets/	Archivos estáticos (imágenes, estilos).
components/	Componentes reutilizables.
elementos/	Componentes de elementos.
ElementsFooter.vue	Pie de página.
ElementsNavbar.vue	Barra de navegación.
graph/	Componentes de gráficos.
LotesGraph.vue	Gráfica de lotes.
MedicamentosGraph.vue	Gráfica de medicamentos.
login/	Componentes de inicio de sesión.
LoginHospital.vue	Componente de inicio de sesión.
mod-farmacia/	Módulos relacionados con la farmacia.
FarmaciaChatbot.vue	Chatbot para soporte.
FarmaciaConsumibles.vue	Gestión de consumibles.
FarmaciaDispensacion.vue	Proceso de dispensación.
FarmaciaLotemedicamentos.vue	Gestión de lotes.
FarmaciaMedicamentos.vue	Gestión de medicamentos.
router/	Configuración de rutas.
index.js	Definición de rutas de la aplicación.
App.vue	Componente raíz.
main.js	Archivo de entrada para Vue.
HelloWorld.vue	Componente de ejemplo.

API-FARMACIA:

Descripción:

Este archivo configura una instancia personalizada de Axios, una herramienta para hacer peticiones HTTP. Se utiliza para conectarse con el backend del proyecto, especialmente con los servicios relacionados al módulo de farmacia.

Funcionalidad principal:


Define la dirección base del servidor al que se harán las peticiones. Esto evita repetir la misma URL en cada solicitud.

Implementa un interceptor que se activa antes de enviar cualquier petición. Este interceptor revisa si hay un token de autenticación guardado en el navegador (en localStorage) y, si existe, lo agrega automáticamente a los encabezados de la petición.

Gracias a este proceso, todas las solicitudes que requieren autenticación se manejan de forma automática y segura, sin necesidad de añadir manualmente el token cada vez.

Uso general:

Este archivo se importa en los componentes Vue que necesiten comunicarse con el backend, permitiendo hacer peticiones como obtener datos, guardar registros, actualizar o eliminar información del sistema de farmacia.

```
src > api >  api-farmacia.js > ...  
1   import axios from "axios";  
2  
3   const api = axios.create({  
4     baseURL: "http://127.0.0.1:8000",  
5   });  
6  
7   // Intercepta solicitudes y añade el token si existe  
8   api.interceptors.request.use((config) => {  
9     const token = localStorage.getItem("token");  
10    if (token) {  
11      config.headers.Authorization = `Bearer ${token}`;  
12    }  
13    return config;  
14  });  
15  
16  export default api;  
17
```

ElementsFooter.vue

Descripción:

Este componente representa el pie de página (footer) de la aplicación. Está diseñado para mantenerse visible en la parte inferior de la interfaz, ofreciendo accesos rápidos a secciones clave del sistema y enlaces a redes sociales.

Funcionalidad principal:

Estructura del contenido:

Contiene una barra de navegación con enlaces que permiten al usuario moverse entre diferentes vistas de la aplicación, como la página de inicio, consumibles y medicamentos.

Muestra un mensaje de derechos reservados, indicando que el contenido pertenece al Hospital Las Lomas.

Incluye íconos de redes sociales (Twitter, Facebook e Instagram), que en una versión final podrían enlazar a las cuentas oficiales del hospital.

Estilos personalizados:

El diseño usa colores suaves y un formato centrado para mantener la coherencia visual con el resto de la aplicación.

Se aplican efectos de hover en los enlaces de navegación y los íconos sociales para mejorar la experiencia del usuario.

Se utiliza scoped para que los estilos afecten únicamente a este componente, evitando conflictos con otros estilos de la app.

Uso general:

Este componente se importa y se integra en la vista principal o en cada vista del sistema donde se desea mostrar el footer. Su función es brindar navegación accesible y reforzar la identidad visual de la aplicación con información institucional y enlaces externos.



ElementsNavbar:

Descripción:

Este componente ofrece navegación fluida entre diferentes vistas del sistema (como medicamentos, consumibles, dispensación, etc.) y además integra accesibilidad y animaciones para una mejor experiencia de usuario.

Funcionalidad Principal:

Menú Hamburguesa (Sidebar):

Controlado por una casilla (checkbox) que activa o desactiva la visibilidad del menú lateral.

Usa animaciones en las barras del icono hamburguesa para ofrecer retroalimentación visual.

El fondo oscuro (sidebar-backdrop) permite cerrar el menú haciendo clic fuera del mismo.

Ofrece navegación entre diferentes secciones del sistema (Inicio, Medicamentos, Dispensación, etc.) usando router-link.

Formulario de Búsqueda:

Permite al usuario buscar medicamentos escribiendo en un campo y presionando "Buscar".

La acción de búsqueda simula un retardo de 2 segundos con un indicador de carga para representar la espera de una petición al backend.

Usa v-model a través de v-bind y \$emit para mantener el valor sincronizado con un filtro externo (probablemente definido en un componente padre).

Indicador de Carga:

Se muestra un spinner SVG animado mientras se realiza la búsqueda simulada.

Ayuda al usuario a saber que algo está ocurriendo en segundo plano.

Diseño Visual y Accesibilidad:

El menú está animado, organizado por secciones (sidebar-nav, nav-divider) y utiliza íconos con Font Awesome.

Incluye atributos de accesibilidad como role="button" y aria-label.

Uso General:

Este componente se utiliza en el layout principal de la aplicación, actuando como cabecera del sistema de farmacia. Es ideal para integrarlo como parte del App.vue o como componente hijo en cualquier vista que requiera navegación lateral y búsqueda centralizada.



Componente Login:

Este componente proporciona un formulario de inicio de sesión que permite a los usuarios ingresar sus credenciales (usuario, contraseña, correo electrónico y teléfono). Al enviar el formulario, se valida la información y, si es correcta, se otorga acceso al sistema, almacenando un token de sesión.

Estructura del componente:

1. Diseño general:

El formulario de inicio de sesión está dividido en dos secciones principales:

Sección izquierda: Muestra una imagen (por ejemplo, el logo de la empresa) que está alineada en el centro verticalmente.

Sección derecha: Contiene el formulario donde los usuarios ingresan sus datos para iniciar sesión.

2. Formulario:

El formulario tiene los siguientes campos:

Usuario: Un campo de texto para que el usuario ingrese su nombre de usuario.

Contraseña: Un campo de contraseña para que el usuario ingrese su clave.

Correo Electrónico: Un campo para el correo electrónico del usuario.

Teléfono: Un campo para el número telefónico del usuario.

Estos campos están diseñados con estilos básicos y responsivos para ajustarse correctamente en pantallas de diferentes tamaños.

3. Botón de envío:

El formulario incluye un botón de envío para iniciar sesión, que al ser presionado ejecuta la función de login.

Componente LandingPage:

Este componente representa la página principal o de bienvenida del módulo de farmacia del "Hospital Las Lomas". Su objetivo es ofrecer una primera impresión atractiva, amigable y funcional, proporcionando información general sobre los servicios de farmacia, accesos a categorías, y presentación del equipo médico.

Estructura del componente:

1. Navbar (barra de navegación):

Se utiliza un componente reutilizable (ElementsNavbar) para mostrar la barra de navegación en la parte superior, que probablemente contiene el logo, navegación principal y acceso a otras secciones del sistema.

2. Sección Hero:

Incluye un título de bienvenida, una breve descripción de los servicios de farmacia y una imagen representativa del hospital. Esta sección funciona como una introducción visual y textual para el usuario.

3. Botón flotante de chatbot:

En la parte inferior derecha, se muestra un botón flotante que permite al usuario mostrar u ocultar un chatbot incrustado mediante un iframe. Esta funcionalidad mejora la interacción y brinda atención automatizada.

4. Sección de servicios destacados:

Muestra una cuadrícula con servicios clave que ofrece la farmacia, como:

Medicamentos especializados

Atención personalizada

Productos exclusivos

Entrega rápida

Cada servicio está representado con un icono, título y breve descripción, brindando un vistazo rápido de las funcionalidades.

5. Sección de categorías:

Presenta tarjetas con diferentes categorías de productos y servicios disponibles en la farmacia. Cada tarjeta incluye:

Imagen representativa

Título

Botón para navegar a la vista específica de la categoría (por ejemplo: Medicamentos, Consumibles, Lote de medicamentos y Dispensación)

6. Sección del equipo médico:

Sección que muestra tarjetas interactivas de médicos destacados. Cada tarjeta tiene dos caras:

Frente: Foto, nombre, especialidad y botón para ver más detalles.

Reverso: Descripción más detallada del médico y botón para volver a la vista frontal.

Esta interacción (efecto de "flip") brinda una forma dinámica de mostrar información adicional.

7. Footer (pie de página):

Al final se incluye un componente reutilizable (ElementsFooter) con la información de contacto, links útiles o cualquier otro contenido estándar del pie de página.



Componente: FarmaciaConsumible

Este componente representa una vista visualmente atractiva e interactiva para mostrar productos farmacéuticos y consumibles agrupados por categorías, utilizando carruseles horizontales. Además, incluye funcionalidades de filtrado y una ventana modal para ver más detalles de cada producto.

Estructura general del componente:

Navbar y Footer

Se utilizan los componentes reutilizables ElementsNavbar y ElementsFooter.

Navbar permite aplicar filtros dinámicos sobre los productos visibles.

Listado de categorías y carruseles

Cada categoría (Consumibles, Otros, etc.) contiene un grupo de productos.

Cada grupo se muestra en un carrusel horizontal con botones para desplazarse (moverIzquierda y moverDerecha).

Tarjetas de productos

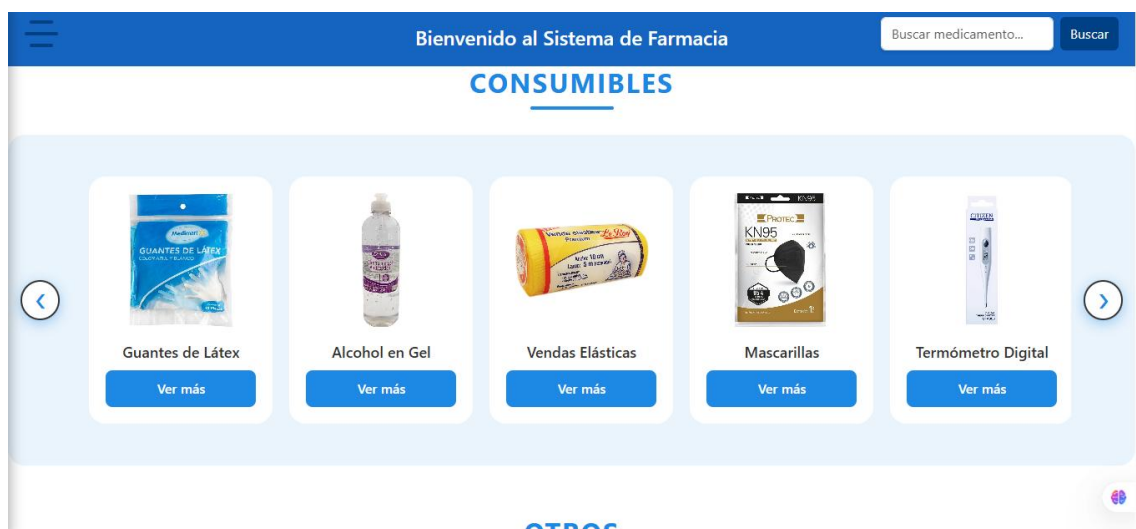
Cada producto se presenta como una tarjeta con imagen, nombre y un botón "Ver más".

Las tarjetas son responsivas y animadas al hacer hover.

Modal de detalle

Al hacer clic en "Ver más", se abre una ventana modal con información detallada del producto (nombre, folio, laboratorio).

El modal se puede cerrar haciendo clic en el fondo o en el botón "Cerrar".



Bienvenido al Sistema de Farmacia

Buscar medicamento...

Buscar

CONSUMIBLES



Guantes de Látex

Ver más

Alcohol en Gel

Folio: 67890

Laboratorio: XYZ Labs

Cerrar



Termómetro Digital

Ver más

Ver más

Ver más

Ver más

OTROS

Componente FarmaciaDispensacion:

Este componente forma parte del módulo de farmacia del sistema y tiene como objetivo principal mostrar una tabla con todos los medicamentos disponibles, así como alertar visualmente cuando alguno está a punto de agotarse o ya no está disponible. Este componente hace uso de notificaciones visuales, navegación y estilos para mejorar la experiencia del usuario.

Estructura general del componente

ElementsNavbar

Barra de navegación reutilizable. Tiene una propiedad v-model: filtro que permite filtrar los medicamentos por nombre comercial.

Notificaciones:

notificacionVisible: aparece si un medicamento tiene menos de 6 unidades (color amarillo).

Menú hamburguesa (icono de tres líneas):

Estructura visual para mostrar un menú en pantallas pequeñas, aunque no se observa su funcionalidad conectada en este componente.

Tabla de medicamentos:

Se muestra una tabla con los siguientes campos:

Imagen del medicamento

Nombre comercial

Nombre genérico

Vía de administración

Presentación

Tipo

Cantidad

Además, las filas se colorean automáticamente dependiendo del estado del medicamento:

Rojo translúcido si está agotado (cantidad = 0)

Amarillo translúcido si tiene menos de 6 unidades disponibles

Bienvenido al Sistema de Farmacia

Lista de Medicamentos

¡Alerta! El medicamento Infiximab está por acabarse.

Imagen	Nombre del Medicamento	Genérico	Vía	Presentación	Tipo	Cantidad
	Paracetamol	Acetaminofén	Oral	Comprimidos	Analgesicos	50
	Ibuprofeno	Ibuprofeno	Oral	Capsulas	Antiinflamatorios	30
	Aspirina	Ácido acetilsalicílico	Oral	Comprimidos	Antiinflamatorios	40
	Amoxicilina	Amoxicilina	Oral	Capsulas	Antibioticos	25
	Omeprazol	Omeprazol	Oral	Capsulas	Antiinflamatorios	60
	Loratadina	Loratadina	Oral	Comprimidos	Antihistaminicos	40
	Metformina	Metformina	Oral	Comprimidos	Antiinflamatorios	50
	Diazepam	Diazepam	Oral	Grageas	Antipsicoticos	30
	Fluoxetina	Fluoxetina	Oral	Capsulas	Antidepresivos	25

Componente LoteMedicamentos:

Este componente forma parte del módulo de farmacia del sistema y tiene como objetivo gestionar el registro, visualización y eliminación de lotes de medicamentos. Además, proporciona una representación visual del inventario mediante una gráfica. Permite al usuario registrar nuevos productos, visualizar el estado del inventario en una tabla y detectar visualmente si un medicamento está en niveles bajos de existencia.

Estructura general del componente

ElementsNavbar

Barra de navegación reutilizable que se muestra en la parte superior del componente.

Encabezado y botón de acción

Muestra el título principal: "Lote de Medicamentos".

Botón "Agregar Nuevo Producto" que al hacer clic despliega un formulario de registro para un nuevo lote.

Gráfica del inventario

Componente LotesGraph

Recibe como prop la lista de medicamentos y muestra una gráfica que representa visualmente los datos.

Útil para monitorear rápidamente la cantidad y variedad de medicamentos almacenados.

Formulario de registro

Aparece cuando mostrarFormulario es true.

Contiene los campos necesarios para registrar un nuevo medicamento:

Nombre

Clave

Costo Total

Cantidad

Ubicación

El formulario incluye dos botones:

Guardar: para registrar el nuevo medicamento.

Cancelar: para cerrar el formulario y limpiar los campos.

Tabla de medicamentos

Se despliega una tabla que contiene los siguientes campos:

ID del lote

Clave del medicamento

Costo total

Cantidad disponible

Ubicación del producto

Acciones (editar o eliminar)

Además, las filas de la tabla cambian de color dependiendo del nivel de stock:

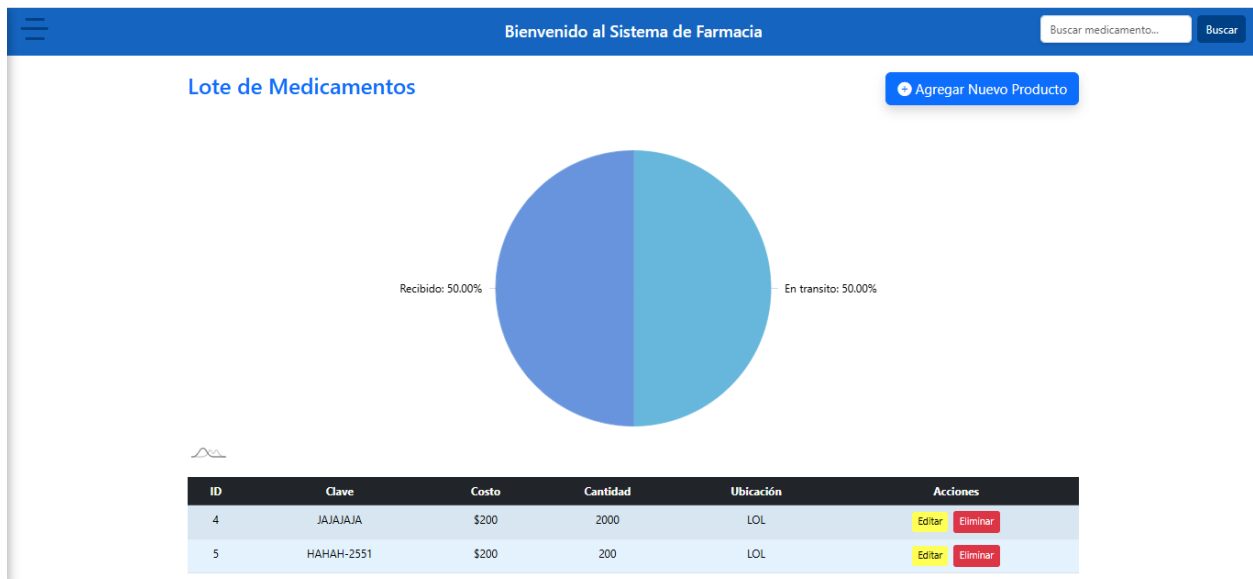
Amarillo: si la cantidad del medicamento es menor a 10 unidades.

Azul: si la cantidad es igual o mayor a 10 unidades.

Acciones sobre los medicamentos

Eliminar: El botón llama a la función eliminarMedicamento, que realiza una petición al backend para eliminar el registro y lo retira de la tabla localmente.

Editar: Se muestra el botón, pero no tiene funcionalidad implementada aún.





Componente Medicamentos:

Este componente representa la vista principal del área de medicamentos. Permite visualizar una descripción general del área, una gráfica informativa y un carrusel de productos por categoría. También muestra un modal con detalles cuando se selecciona un producto.

Estructura general del componente:

1. Encabezado (Navbar):

Contiene una barra de navegación reutilizable, que incluye un campo de búsqueda para filtrar productos de medicamentos según su nombre comercial.

2. Sección de descripción del área:

Presenta una breve explicación sobre el área de medicamentos, con subtítulos, párrafos informativos y una imagen ilustrativa. Esta sección ayuda al usuario a entender el propósito del área y cómo está organizada.

3. Gráfica informativa:

Debajo de la descripción se muestra un componente de gráfico personalizado que representa visualmente información estadística sobre los medicamentos. Esto puede incluir distribución por tipo, vía de administración, etc.

4. Carrusel por categoría (Tipo de medicamento):

Cada categoría representa un tipo de medicamento (por ejemplo, antibióticos, analgésicos, etc.).

Dentro de cada categoría:

Se visualizan tarjetas de productos con nombre, tipo, presentación, etc.

Las tarjetas se organizan en una fila tipo carrusel horizontal.

Se puede navegar hacia adelante o hacia atrás con botones de flechas laterales.

El carrusel es independiente por cada categoría, es decir, cada uno tiene su propio control de navegación.

5. Modal de información detallada:

Cuando un usuario hace clic sobre una tarjeta de producto, se abre un modal emergente con información completa sobre ese medicamento: nombre comercial, genérico, vía de administración, tipo, cantidad, precio, presentación, etc. También se muestra una imagen representativa del medicamento.

6. Pie de página (Footer):

Incluye el componente de pie de página del sitio, reutilizado para mantener la coherencia en el diseño general de la aplicación.

7. Lógica del componente:

Carga inicial de datos:

Al montarse el componente, se realiza una llamada a la API para obtener todos los medicamentos disponibles. Estos se agrupan por categoría (Tipo) y se guardan en una estructura interna.

Filtrado de productos:

Cuando el usuario escribe en el campo de búsqueda, se filtran las categorías y productos que coincidan con el texto ingresado, permitiendo una experiencia dinámica.

Navegación del carrusel:

Se controla la posición actual del carrusel por categoría, permitiendo avanzar o retroceder en los productos visibles.

Selección de producto:

Al seleccionar un producto, se activa el modal para mostrar su información detallada. Al hacer clic fuera del modal o en cerrar, este se oculta.



ANALGESICOS

Paracetamol

Ver detalles

Ketorolaco

Ver detalles

Tramadol

Ver detalles

Metamizol

Ver detalles

Ibuprofeno

Ver detalles

Ver detalles

Ver detalles

Ver detalles

Metformina

Ver detalles

Ketorolaco

Genérico: Ketorolaco

Vía: Intravenoso

Presentación: Viales

Tipo: Analgesicos

Cantidad: 20

Precio: \$50.00

ANTIBIOTICOS

Componente: Chatbot Interactivo

Este componente permite abrir un chatbot incrustado en la página, brindando interacción directa con los usuarios. Además, proporciona la funcionalidad de mover la ventana del chatbot, haciendo que el componente sea dinámico y fácil de usar.

1. Botón de apertura del chatbot:

Un botón fijo en la esquina inferior derecha de la pantalla, que al ser clicado, abre la ventana del chatbot. El botón tiene un estilo destacado con un fondo azul y texto blanco.

2. Ventana del chatbot:

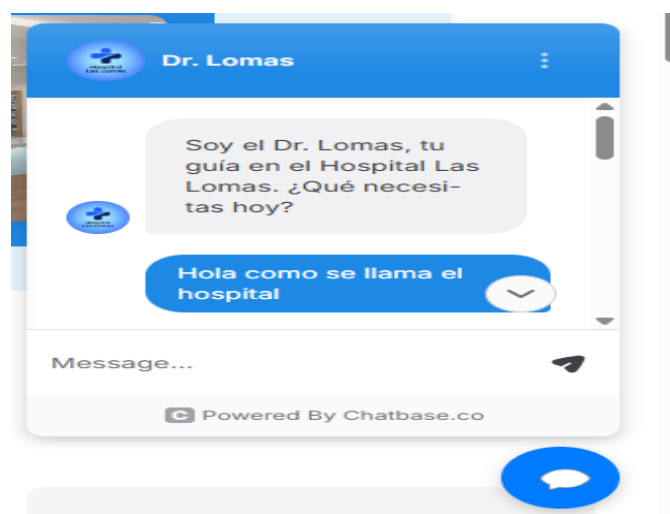
Una vez el usuario hace clic en el botón, aparece una ventana emergente que contiene el iframe del chatbot. La ventana tiene un diseño moderno, con bordes redondeados y una sombra suave para destacar visualmente.

Cabecera de la ventana: La cabecera tiene un fondo azul, donde se muestra el título "Chatbot" y un botón de cierre que permite al usuario cerrar la ventana del chatbot.

Iframe del chatbot: Dentro de la ventana, se incrusta un iframe que carga el chatbot desde un enlace externo. Este iframe ocupa todo el espacio disponible dentro de la ventana.

3. Funcionalidad de arrastre:

El usuario puede mover la ventana del chatbot por la pantalla al hacer clic y mantener presionada la cabecera de la ventana. Esto se logra con una función de drag-and-drop (arrastrar y soltar), que permite desplazar la ventana libremente.



Router/index.js:

Descripción:

Este archivo se encarga de definir y configurar todas las rutas de navegación del frontend de la aplicación. Utiliza el sistema de enrutamiento de Vue Router para controlar qué componente se muestra según la URL que el usuario visite.

Funcionalidad principal:

Importación de vistas o componentes:

Se importan los componentes principales que representan cada vista de la aplicación, como la página de login, la página de inicio y las distintas secciones del módulo de farmacia.

Definición de rutas:

Se establece un arreglo con las rutas disponibles en el sistema. Cada ruta contiene:

Una ruta de acceso (path), que es la URL que el usuario visitará.

Un nombre identificador (name), útil para navegar de forma programada.

El componente que se mostrará cuando se acceda a esa ruta.

Agrupación por módulos:

Las rutas están organizadas por secciones, como:

Inicio de sesión (Login)

Página informativa principal (LandingPage)

Módulo de farmacia, que incluye subrutas para:

Consumibles

Medicamentos

Lote de medicamentos

Dispensación

Creación del router:

Finalmente, se configura el router para que utilice el historial del navegador (sin hash en la URL), lo cual mejora la experiencia del usuario y mantiene URLs limpias.

Uso general:

Este archivo es clave para la navegación dentro de la aplicación. Permite cambiar de una vista a otra sin recargar la página, respetando el enfoque de una Single Page Application (SPA).

```
src > router > index.js > routes
1 import { createRouter, createWebHistory } from 'vue-router'
2
3 import ConsumiblesView from '@/components/mod-farmacia/FarmaciaConsumibles.vue'
4 import MedicamentoView from '@/components/mod-farmacia/FarmaciaMedicamentos.vue'
5 import LoteMedicamentosView from '@/components/mod-farmacia/FarmaciaLoteMedicamentos.vue'
6 import DispensacionView from '@/components/mod-farmacia/FarmaciaDispensacion.vue'
7 import LandingPage from '@/components/mod-farmacia/LandingPage.vue'
8 import LoginHospital from '@/components/login/LoginHospital.vue'
9
10 const routes = [
11   // Login
12   {
13     path: '/',
14     name: 'Login',
15     component: LoginHospital
16   },
17
18   {
19     path: '/Informacion',
20     name: 'LandingPage',
21     component: LandingPage
22   },
23
24   //Modulo de farmacia
25   {
26     path: '/Farmacia/Consumibles',
27     name: 'Consumibles',
28     component: ConsumiblesView
29   },
30   {
31     path: '/Farmacia/Medicamento',
32     name: 'Medicamento',
```

Conclusión:

En este proyecto, se ha logrado desarrollar un sistema de navegación y búsqueda funcional para la aplicación de farmacia, implementando una barra lateral interactiva, una barra de búsqueda eficiente y un manejo dinámico de rutas. A lo largo del desarrollo, se han considerado aspectos clave como la accesibilidad, la responsividad y la optimización de la experiencia de usuario.

Con la integración de rutas dinámicas, el componente de navegación ofrece una transición fácil entre las diferentes secciones del sistema, lo que mejora la usabilidad del sistema de administración de la farmacia.

En resumen, este proyecto cumple con los requisitos establecidos, ofreciendo una solución robusta y escalable que puede adaptarse a futuras funcionalidades y mejoras. Se ha priorizado la experiencia de usuario, asegurando un sistema intuitivo, accesible y eficiente.