

Universidad Centroamericana José Simeón Cañas  
San Salvador, El Salvador



**Ingeniería y Arquitectura**

## **Propuesta de proyecto Código Abierto**

### **Docente:**

Metzi Rutilia Aguilar Munguia

### **Integrantes:**

Erick Rickelmy Vasquez Alfaro 00065520

David Alejandro Iglesias Moreno 00047920

Andrés Josué Mendoza Alvarado 00305422

Erick Fernando Quinteros Lizama 00220619

Antiguo Cuscatlan, El Salvador 10 Junio 2024

## 1. Descripción General

El Geoportal de Centros de Asistencia Médica es una aplicación web diseñada para ayudar a las personas a ubicar clínicas y centros de asistencia médica cercanos a sus viviendas. A través de una interfaz intuitiva y fácil de usar, los usuarios pueden encontrar rápidamente servicios médicos disponibles en su área, mejorando así el acceso a la atención médica oportuna y eficiente. Además, la aplicación está diseñada para turistas que no conocen el país, ayudándoles a encontrar asistencia médica en ubicaciones desconocidas.

Una característica destacada del Geoportal es su asistente virtual, que brinda ayuda respondiendo preguntas frecuentes de los usuarios, facilitando la obtención de información relevante sobre los centros médicos disponibles.

### Repositorio Original.

El proyecto original se encuentra alojado en el repositorio de GitHub:

GitHub - Geoportal de Vacunación

<https://github.com/MaryBenitez/Geoportal-Vacunacion>

Este proyecto se ha bifurcado (fork) del repositorio mencionado anteriormente, con el objetivo de adaptarlo para proporcionar información sobre centros de asistencia médica y añadir un asistente virtual.

GitHub- Geoportal de Centros de Asistencia Médica.

<https://github.com/AndresMendoza0030/Geoportal.git>

### Licencia.

Este proyecto está licenciado bajo la Licencia MIT. La Licencia MIT es una licencia de software libre permisiva que permite a los usuarios usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y/o vender copias del software, siempre que se incluya la declaración de derechos de autor y la licencia original en todas las copias o partes sustanciales del software.

### Modificaciones en Curso

Actualmente, se están realizando las siguientes modificaciones y mejoras en el proyecto:

Integración de Nuevas API de Datos: Implementación de nuevas fuentes de datos para mejorar la precisión y actualización de la información de los centros de asistencia médica.

Mejoras en la interfaz de usuario: Rediseño de la interfaz gráfica para mejorar la experiencia del usuario, incluyendo nuevas funcionalidades como búsqueda optimizada.

Optimización del Rendimiento: Refactorización del código para mejorar la eficiencia y velocidad de carga del

Mejoras de Seguridad: Implementación de mejores prácticas de seguridad para proteger los datos de los usuarios y la integridad del sistema.

### **Tecnologías Utilizadas**

- Lenguaje de Programación: Java.
- Frameworks y Biblioteca:
- Spring Boot: Para la creación de aplicaciones Java robustas y escalables.
- Leaflet: Biblioteca de JavaScript para mapas interactivos compatibles con dispositivos móviles.
- Thymeleaf: Biblioteca Java que implementa un motor de plantillas de XML/XHTML/HTML5.
- Base de Datos: PostgreSQL para almacenar datos de ubicación y servicios médicos.
- Desarrollo Frontend:
- Bootstrap: HTML, CSS y JavaScript para desarrollar proyectos móviles con capacidad de respuesta en la web.
- Tailwind CSS: Framework de CSS para diseñar interfaces de usuario modernas y responsivas.

## **2. Identificación de roles en el grupo.**

**Coordinador del Grupo:** Fernando Quinteros.

**Backend Developers:** Andrés Mendoza y Erick Vasquez.

### **Responsabilidades:**

- Desarrollar y mantener la lógica del servidor.
- Crear y gestionar bases de datos.
- Desarrollar APIs (Interfaces de Programación de Aplicaciones) para que el frontend pueda comunicarse con el backend.
- Implementar la seguridad y la autenticación.
- Optimizar el rendimiento del servidor y la base de datos.
- Integrar servicios y sistemas externos.

**Frontend Developers:** Fernando Quinteros y David Moreno.

### **Responsabilidades:**

- Diseñar y desarrollar la interfaz de usuario de la aplicación.
- Implementar diseños UX/UI asegurando una experiencia de usuario intuitiva y atractiva.
- Integrar el frontend con las APIs del backend para que los datos se muestren correctamente.

- Asegurar la responsividad del diseño para que la aplicación funcione bien en diferentes dispositivos y navegadores.
- Optimizar el rendimiento del frontend para una carga rápida y eficiente.
- Realizar pruebas de usabilidad y hacer ajustes según sea necesario.

### **3. Comunicación para el proyecto.**

para asegurar una comunicación efectiva entre los miembros del equipo, hemos establecido los siguientes canales de comunicación:

#### **WhatsApp.**

- Utilizamos WhatsApp para la comunicación diaria y rápida entre los miembros del equipo. Esta herramienta facilita la coordinación de tareas, la resolución inmediata de dudas y la planificación de actividades. Su accesibilidad y facilidad de uso permiten mantener una comunicación constante y efectiva, asegurando que todos los integrantes del equipo estén al tanto de las actualizaciones y cambios en el proyecto.

#### **Google Meet.**

- Google Meet es nuestra plataforma elegida para las reuniones semanales del equipo. Durante estas sesiones, discutimos el progreso del proyecto, planificamos las próximas etapas y resolvemos cualquier problema o desafío que surja. Estas reuniones son esenciales para mantener una colaboración fluida, asegurándonos de que todos los miembros estén alineados con los objetivos y avances del proyecto.

#### **Comunicación Externa.**

- Hasta el momento, no hemos establecido comunicación con comunidades de desarrollo o personas externas. Nos hemos centrado en la coordinación interna para avanzar en el desarrollo del proyecto, priorizando la cohesión y eficiencia del equipo. Sin embargo, a medida que el proyecto avance, consideraremos la posibilidad de ampliar nuestra red de contactos y colaborar con desarrolladores y expertos externos para enriquecer el desarrollo del Geoportal de Centros de Asistencia Médica.

### **4. Análisis de licencias en el repositorio.**

## File info - Geoportal-Centros-de-Asistencia-Medica

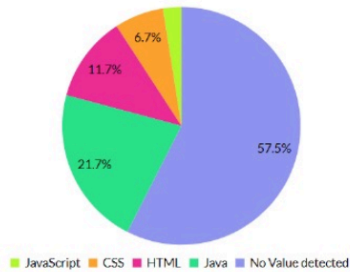
120

Total Number of Files

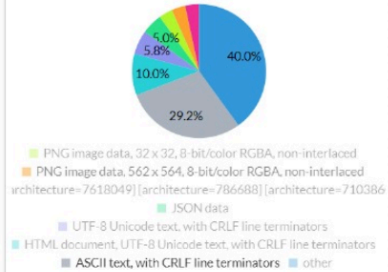
51

Total Number of Directories

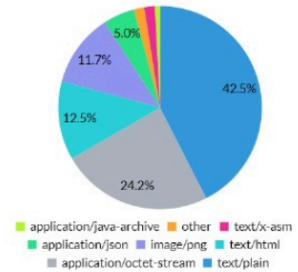
Programming languages



File types



Mime types



## Copyright info - Geoportal-Centros-de-Asistencia-Medica

1

Unique holders

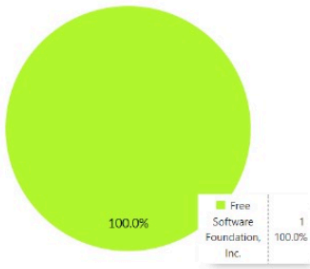
1

Unique notices

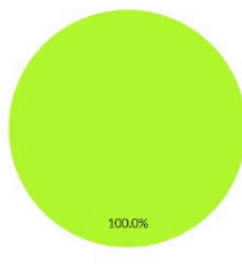
0

Unique authors

Copyright Holders



Copyright Notices



Copyright Authors

No Data

## License info - Geoportal-Centros-de-Asistencia-Medica

3

Total licenses

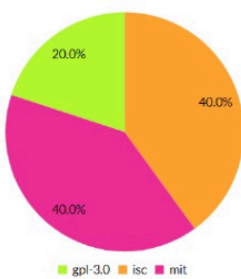
5

Total files with licenses

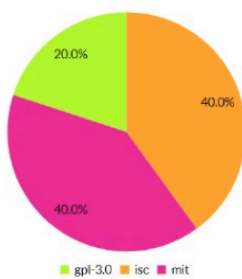
3

Total SPDX licenses

License expression



License keys



License policy

No Data

Default columns

Show all

Hide all

Clear Filters

Search ...

File cols

Origin cols

Copyright cols

License cols

Package cols

Custom Columns

Path	Copyright Statements	Copyright Holder	Copyright Author	Copyright Start Line
		All		All
Geoportal-Centros-de-Asistencia-Medica				
Geoportal-Centros-de-Asistencia-Medica/tern-project				
Geoportal-Centros-de-Asistencia-Medica/hs_err_pid13280.log				
Geoportal-Centros-de-Asistencia-Medica/hs_err_pid20452.log				
Geoportal-Centros-de-Asistencia-Medica/hs_err_pid7028.log				
Geoportal-Centros-de-Asistencia-Medica/LICENSE	Copyright (c) 2007 Free Software Foundation, Inc. <https://fsf.org/>	Free Software Foundation, Inc.		4
Geoportal-Centros-de-Asistencia-				

1 to 25 of 172

<

>

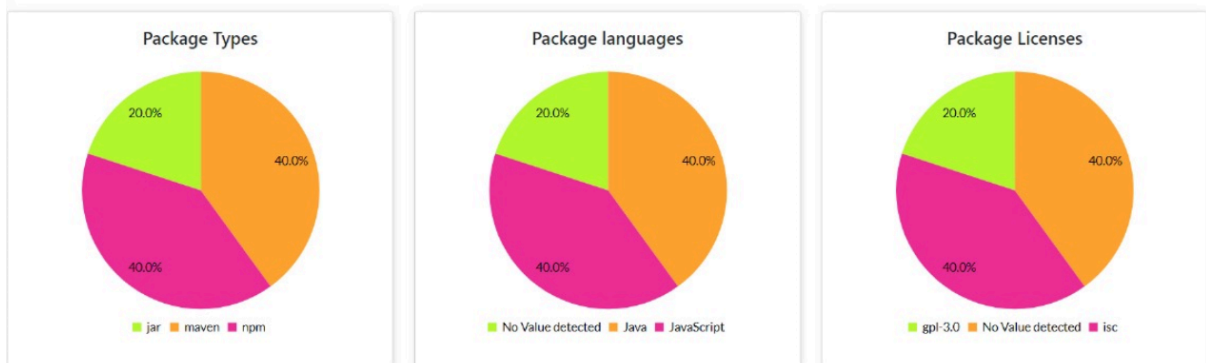
Page 1 of 7

>>

#### Package info - Geoportal-Centros-de-Asistencia-Medica

1

Total Number of packages

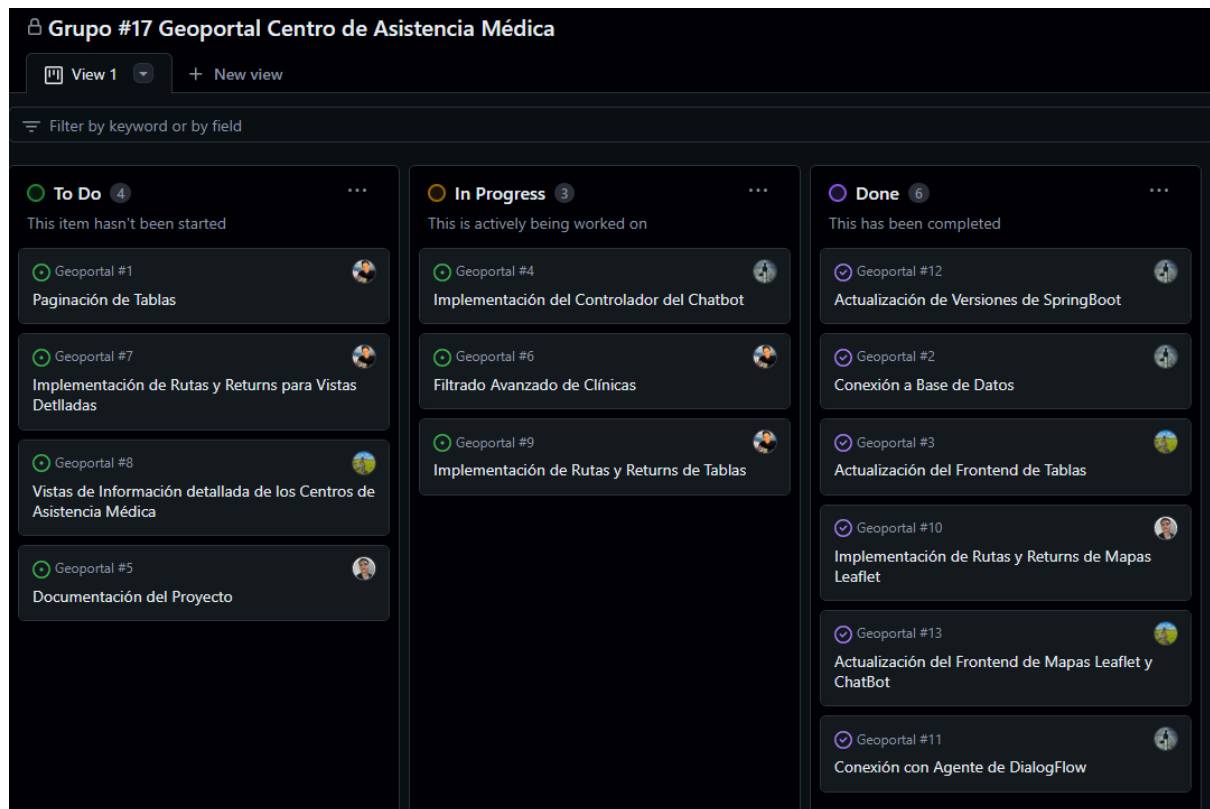


#### 5. Control de versiones que se está utilizando y forma en que se está usando

#### Gestión de Issues

Para organizar y rastrear el progreso de nuestro proyecto, utilizamos la funcionalidad de Issues de GitHub. Cada tarea o error se crea como un issue, que se etiqueta y asigna a los miembros del equipo correspondientes. Los issues se categorizan en diferentes columnas dentro de nuestro tablero de proyectos de GitHub Projects:

- **To Do:** Aquí se colocan las tareas pendientes que están por comenzar.
- **In Progress:** Incluye las tareas en las que se está trabajando actualmente.
- **Done:** Incluye las tareas completadas.



Cada issue tiene una descripción detallada, pasos a seguir, comportamiento esperado y notas adicionales para asegurar que todos los miembros del equipo tengan una comprensión clara de la tarea a realizar. Además, los issues se vinculan a Pull Requests (PRs) para facilitar el seguimiento del progreso y la integración del código.

## Gestión de Ramas

Decidimos dividir el flujo de trabajo en 4 ramas principales para mantener una estructura organizada y facilitar el desarrollo paralelo. Estas ramas son:

- **main**: Contiene la versión estable y de producción del proyecto.
- **mapaleaflet**: Rama para el desarrollo y mantenimiento de funcionalidades relacionadas con Leaflet y mapas.
- **chatbot**: Rama para el desarrollo y mantenimiento del chatbot.
- **frontend**: Rama para el desarrollo y mantenimiento del frontend.

## Descripción de Ramas y su Uso

### main

- **Propósito**: Contiene la versión estable y lista para producción del proyecto.
- **Uso**: Solo se fusionan cambios que han sido completamente testeados y aprobados en las ramas de desarrollo específicas (**mapaleaflet**, **chatbot**, **frontend**).

### mapaleaflet

- **Propósito:** Desarrollo y mantenimiento de todas las funcionalidades relacionadas con Leaflet y mapas.
- **Uso:** Todas las funcionalidades y actualizaciones relacionadas con mapas se desarrollan en esta rama antes de fusionarse en **main**.

### chatbot

- **Propósito:** Desarrollo y mantenimiento del chatbot.
- **Uso:** Todas las funcionalidades y actualizaciones del chatbot se desarrollan en esta rama antes de fusionarse en **main**.

### frontend

- **Propósito:** Desarrollo y mantenimiento del frontend.
- **Uso:** Todas las funcionalidades y actualizaciones del frontend se desarrollan en esta rama antes de fusionarse en **main**.

6. Informe de errores encontrados desde la instalación del repositorio original hasta el avance actual (bitácora y seguimiento)

## Bitácora de Errores

### Error: Necesidad de Actualización de la Versión de Spring Boot

**Fecha:** 06/06/2024

**Descripción:** Al clonar el repositorio original y ejecutar **mvn clean install**, se encontró un error que impedía la construcción del proyecto debido a problemas de compatibilidad con la versión de Spring Boot utilizada.

**Mensaje de Error:** **Failed to resolve dependencies for project**

### Acciones Tomadas:

1. **Identificación del Problema:** Se identificó que el problema estaba relacionado con la versión de Spring Boot utilizada en el proyecto.
2. **Investigación:** Se revisaron las notas de la versión de Spring Boot y se consultaron recursos en línea para determinar la mejor manera de resolver el problema.
3. **Decisión:** Se decidió actualizar la versión de Spring Boot a la última versión estable disponible.

## Resolución

### Pasos de Resolución:

1. **Identificar la Última Versión Estable de Spring Boot:**
  - La última versión estable de Spring Boot se identificó como [nueva versión].
2. **Actualizar el Archivo **pom.xml**:**
  - Se actualizó el archivo **pom.xml** para utilizar la nueva versión de Spring Boot



```
<parent>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>2.7.0</version>

    <relativePath/>

</parent>
```

### 3. Compilar y Construir el Proyecto:

- Se ejecutó `mvn clean install` para asegurar que todas las dependencias estuvieran resueltas correctamente.

### 4. Ejecución de Pruebas:

- Se ejecutaron todas las pruebas unitarias y de integración para asegurar que el proyecto funcionara correctamente con la nueva versión de Spring Boot.

### 5. Corrección de Errores Adicionales:

- Se realizaron pequeños ajustes en la configuración que surgieron debido a la actualización de la versión de Spring Boot.

## 2. Nombre de Proyecto Demasiado Largo para Railway

**Fecha:** 08/06/2024

**Descripción:** Durante el despliegue en Railway, se encontró un error debido a que el nombre del proyecto original, "Geoportal Centros de Asistencia Médica", era demasiado largo.

**Mensaje de Error:** 'invalid service name'

**Resolución:**

- Se acortó el nombre del proyecto a "Geoportal".
- Esto permitió el despliegue exitoso en Railway sin problemas adicionales.

## 7. Conclusiones hasta el avance actual

Hemos avanzado significativamente en el proyecto Geoportal de Centros de Asistencia Médica. Desde el inicio, enfrentamos un desafío crucial con la versión de Spring Boot, que solucionamos actualizando a la última versión estable. Esto nos permitió construir y desplegar el proyecto correctamente.

Logramos establecer una conexión con nuestra base de datos PostgreSQL y hemos implementado funcionalidades clave utilizando Leaflet para los mapas y Dialogflow para el chatbot. Además, actualizamos el frontend para mejorar la experiencia del usuario, y hemos desplegado la primera versión del proyecto en Railway, haciéndolo accesible para pruebas y uso. [\[Geoportal Centros de Asistencia Médica\]](#)

A pesar de estos logros, enfrentamos desafíos en la compatibilidad de versiones y la gestión de dependencias, lo que nos enseñó la importancia de mantener las dependencias

actualizadas y bien gestionadas. La configuración y el despliegue del entorno también presentaron algunos desafíos, que superamos investigando las causas del problema.

Los próximos pasos incluyen completar la documentación del proyecto, implementar funcionalidades de filtrado avanzado para una búsqueda más precisa, implementar métodos para el controlador Webhook para gestionar las respuestas del chatbot y terminar de alimentar completamente la base de datos con todos los centros de asistencia médica. Continuaremos mejorando el frontend y añadiendo nuevas funcionalidades para ofrecer una mejor experiencia al usuario.