



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE TECNOLOGÍAS EN LA INFORMACIÓN

DESARROLLO DE UNA APLICACIÓN WEB Y MÓVIL MAPA
PARLANTE EN EL CENTRO DE SALUD CHAMBO

Trabajo de Titulación para optar al título de Ingeniero en la especialidad
de Tecnologías de la Información

Autor:

Lucio Barragan Noralma Alexandra

Tutor:

Ing. Gonzalo Allauca Peñafiel MgS.

Riobamba, Ecuador - 2024

DECLARATORIA DE AUTORÍA

Yo, Lucio Barragan Noralma Alexandra, con cédula de ciudadanía 0605532357, autor del trabajo de investigación titulado: Desarrollo de una aplicación web y móvil mapa parlante en el Centro de Salud Chambo, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 18 de marzo del 2024.



Lucio Barragan Noralma Alexandra

0605532357

DICTAMEN FAVORABLE DEL PROFESOR TUTOR



Dirección
Académica
VICERRECTORADO ACADÉMICO



UNACH-RGF-01-04-08.11
VERSIÓN 01: 06-09-2021

ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 15 días del mes de Marzo de 2024, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por la estudiante **Noralma Alexandra Lucio Barragán** con CC: **0605532357**, de la carrera Ingeniería en Tecnologías de la Información y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE** DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN titulado "**DESARROLLO DE UNA APLICACIÓN WEB Y MÓVIL MAPA PARLANTE EN EL CENTRO DE SALUD CHAMBO**", por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



Mgs. Gonzalo Allauca Peñafiel
TUTOR(A)

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación “Desarrollo de una aplicación web y móvil mapa parlante en el Centro de Salud Chambo”, presentado por Noralma Alexandra Lucio Barragan, con cédula de identidad número 0605532357, bajo la tutoría de Mgs. Luis Gonzalo Allauca Peñafiel; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba a los 12 días del mes de abril de 2024.

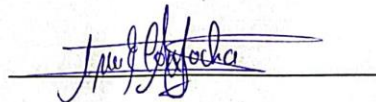
Jorge Delgado, Mgs.
PRESIDENTE DEL TRIBUNAL DE GRADO



Diego Reina, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO



Ana Congacha, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO



CERTIFICADO ANTIPLAGIO



Dirección
Académica
VICERRECTORADO ACADÉMICO



UNACH-RGF-01-04-08.15
VERSIÓN 01: 06-09-2021

CERTIFICACIÓN

Que, **Noralma Alexandra Lucio Barragán** con CC: **0605532357**, estudiante de la Carrera de Ingeniería en **Tecnologías de la Información**, Facultad de **Ingeniería**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**DESARROLLO DE UNA APLICACIÓN WEB Y MÓVIL MAPA PARLANTE EN EL CENTRO DE SALUD CHAMBO**", cumple con el **8 %**, de acuerdo al reporte del sistema Anti plagio Turnitin, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 2 de abril de 2024



Mgs. Gonzalo Allauca Peñafiel
TUTOR

DEDICATORIA

A Dios por ser el eje de todas mis fuerzas y el que paso a paso me ayuda a cumplir mis metas con dedicación y esfuerzo. A mis padres y hermanos, quienes siempre me han brindado su amor, apoyo y motivación en todas mis metas académicas y personales. Agradezco su paciencia y comprensión durante los momentos difíciles que tuve que atravesar durante este proceso.

A mi mentor Isco Xavier, por su orientación experta y por brindarme la libertad para explorar y descubrir. Su paciencia en cada instante y su compromiso con mi desarrollo académico han sido fundamentales para la finalización de esta tesis.

Lucio Barragan Noralma Alexandra

AGRADECIMIENTO

Agradezco infinitamente a Dios quien me ha brindado salud, energía, inteligencia y las fuerzas necesarias para seguir adelante en cada paso de mi vida, por proporcionarme familia, amigos, docentes, quienes me impulsaron a luchar y cumplir mis metas, siempre con humildad y perseverancia, ante todo.

A mis padres quienes me apoyaron en las dificultades a lo largo de este camino, me brindaron su cariño incondicional, apoyo emocional y económico en toda mi etapa estudiantil, siendo los motores que me ayudan a cumplir mis metas y enseñándome valores como respeto, perseverancia y responsabilidad.

Inmensa gratitud a mi asesor de Investigación, profesores y colaboradores por su valiosa contribución y por brindarme los recursos necesarios para llevar a cabo mi proyecto.

Esta tesis es una expresión de mi agradecimiento y aprecio por todo lo que han hecho por mí. Espero que mi investigación pueda contribuir de manera positiva a la sociedad y honrar así el tiempo y esfuerzo que han dedicado en mi formación académica.

Lucio Barragan Noralma Alexandra

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

CERTIFICADO ANTIPLAGIO

DEDICATORIA

AGRADECIMIENTO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

ABSTRACT

CAPÍTULO I. INTRODUCCIÓN.....	15
1.1. Problema y Justificación	16
1.2. Objetivos	17
CAPÍTULO II. MARCO TEÓRICO	18
2.1. El Manual del Modelo de Atención Integral de Salud (MAIS)	18
2.2. Diseño web responsivo (RWD)	18
2.3. Aplicaciones responsivas con Angular	19
2.4. Aplicación del lado del servidor con Nest.js	21
2.5. Gestión de bases de datos	24
2.6. Comunicación entre cliente y servidor	26
2.7. Servidores de mapas en línea.....	26
2.8. Modelo de Calidad ISO/IEC 25010.....	31
2.9. Metodología Scrum.....	32
CAPÍTULO III. METODOLOGÍA	33
3.1. Tipo de Investigación.....	33
3.2. Identificación de variables	34
3.3. Métodos de análisis y procesamiento de datos	36
3.4. Procedimiento	36
3.5. Desarrollo de la propuesta	37
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN	55
4.1. Resultados.....	55

4.2. Discusión	72
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	73
5.1. Conclusiones	73
5.2. Recomendaciones	74
BIBLIOGRAFÍA	75
ANEXOS	78
Anexo 1. Bosquejo para la aplicación responsiva	78
Anexo 2. Guía rápida para la implementación del servidor de Geoserver	79
Anexo 3. Pantallas de la aplicación con acceso web y móvil Mapa Parlante y la descripción de sus funciones.	81
Anexo 4. Cuestionario aplicado a los profesionales de la salud de Chambo para evaluar la usabilidad de la aplicación.	86

ÍNDICE DE TABLAS

Tabla 1. Operacionalización de variables	35
Tabla 2. Equipo Scrum Mapa Parlante	37
Tabla 3. Requerimientos funcionales.....	38
Tabla 4. Requerimientos no funcionales.....	39
Tabla 5. Product Backlog.....	39
Tabla 6. Sprint Backlog	42
Tabla 7. Herramientas utilizadas para el desarrollo de la aplicación Mapa Parlante.....	46
Tabla 8. Análisis de herramientas para el Desarrollo de la Aplicación Mapa Parlante	55
Tabla 9. Promedio obtenido del criterio de aprendizabilidad	62
Tabla 10. Promedio obtenido del criterio de operabilidad.....	65
Tabla 11. Promedio obtenido del criterio de protección contra errores de usuario	68
Tabla 12. Promedio obtenido del criterio de la estética de la interfaz de usuario	71
Tabla 13. Herramientas necesarias para la implementación de Geoserver.	79

ÍNDICE DE FIGURAS

Figura 1. Ciclo de vida de Angular.....	20
Figura 2. Arquitectura básica de Angular	21
Figura 3. Arquitectura de Node.js	22
Figura 4. Estructura de carpetas de un proyecto en Nest.js.....	23
Figura 5. Arquitectura de PostgreSQL.	25
Figura 6. Servidor de Geoserver.....	28
Figura 7. Estructura de manejo de entrada y salida de datos en Geoserver	29
Figura 8. Características del modelo de calidad de software ISO/IEC 25010.	31
Figura 9. Diagrama de caso de uso para el administrador.....	40
Figura 10. Diagrama de caso de uso para usuario EAIS	41
Figura 11. Arquitectura Cliente/Servidor con microservicios.	44
Figura 12. Wireframes de la aplicación	45
Figura 13. Modelo relacional de la base de datos	46
Figura 14. Estructura organizativa de carpetas en el módulo principal de Nest	48
Figura 15. Configuración de PostgreSQL en el proyecto de Nest.....	48
Figura 16. Uso de TypeORM para conexión con la base de datos	49
Figura 17. Implementación de SQL para Interactuar con la BD Ficha Familiar.....	49
Figura 18. Vista previa del desarrollo de la aplicación Mapa Parlante	50
Figura 19. Importación de Tailwind CSS	50
Figura 20. Espacio de trabajo en Geoserver.....	51
Figura 21. Almacenes de datos en Geoserver.....	51
Figura 22. Capas publicadas en Geoserver.....	51
Figura 23. Configuración de estilos SLD en Geoserver.....	52
Figura 24. Conexión de Geoserver a Angular mediante OpenLayers	52
Figura 25. Consumo del API de Bing Maps.....	53
Figura 26. Integración de Openrouteservice en Angular.....	53
Figura 27. Diseño web de la aplicación.....	57
Figura 28. Diseño móvil de la aplicación	58
Figura 29. Población de estudio	59
Figura 30. Aprendizabilidad - Facilidad de aprendizaje del uso de la aplicación	59
Figura 31. Resultado del indicador aprendizaje – Pregunta 1	60
Figura 32. Aprendizabilidad - Intuición del aplicativo.....	60
Figura 33. Resultado del indicador aprendizaje – Pregunta 2.....	61
Figura 34. Aprendizabilidad - Navegabilidad entre las funciones	61
Figura 35. Resultado del indicador aprendizaje – Pregunta 3.....	61
Figura 36. Operabilidad - Eficiencia de la aplicación	62
Figura 37. Resultado del indicador operabilidad– Pregunta 4	63
Figura 38. Operabilidad - Acceso a funciones.....	63
Figura 39. Resultado del indicador operabilidad– Pregunta 5	64
Figura 40. Operabilidad - Consistencia de elementos	64
Figura 41. Resultado del indicador operabilidad– Pregunta 6	65
Figura 42. Protección contra errores de usuario - Mensajes de error	65

Figura 43.	Protección contra errores de usuario - Mensajes de confirmación	66
Figura 44.	Resultado del indicador protección contra errores - Pregunta 8	66
Figura 45.	Protección contra errores de usuario - Notificaciones	67
Figura 46.	Resultado del indicador protección contra errores - Pregunta 9	67
Figura 47.	Estética de la interfaz de usuario - Visualmente atractiva	68
Figura 48.	Resultado del indicador estética de la interfaz de usuario – Pregunta 10	69
Figura 49.	Estética de la interfaz de usuario - Distribución	69
Figura 50.	Resultado del indicador estética de la interfaz de usuario – Pregunta 11	70
Figura 51.	Estética de la interfaz de usuario - Utilización de elementos	70
Figura 52.	Resultado del indicador estética de la interfaz de usuario – Pregunta 12	71
Figura 53.	Diseño responsivo de la aplicación	78
Figura 54.	Proceso de instalación de Apache Tomcat	79
Figura 55.	Implementar Geoserver dentro del gestor de aplicaciones de Apache.....	80
Figura 56.	Servidor de Geoserver activo	80
Figura 57.	Inicio de sesión	81
Figura 58.	Módulo de visualización del mapa.....	82
Figura 59.	Módulo para generación de rutas	82
Figura 60.	Módulo de gestión del perfil del usuario	83
Figura 61.	Módulo de gestión de brigada EAIS	83
Figura 62.	Módulo de gestión de profesionales.....	84
Figura 63.	Módulo de gestión de profesión.....	84
Figura 64.	Módulo de gestión de información EAIS	85

RESUMEN

Este proyecto de investigación se enfocó en el desarrollo de la aplicación con acceso web y móvil Mapa Parlante en el Centro de Salud Chambo para la gestión de factores de riesgo siendo estos: grupos vulnerables, grupos prioritarios, riesgos biológicos, socioeconómicos y sanitario ambiental, mediante la visualización gráfica geoespacial de datos obtenidos de la Ficha Familiar.

La metodología ágil SCRUM guió el proceso de desarrollo de software, empleando una arquitectura cliente-servidor con microservicios para garantizar eficiencia y adaptabilidad. En la construcción del aplicativo se empleó metodologías y herramientas estratégicamente seleccionadas, PostgreSQL como sistema gestor para la base de datos, Nest.js en el backend y Angular junto con Tailwind CSS para el desarrollo del frontend. Geoserver y OpenLayers enriquecieron la gestión de datos geoespaciales y la interactividad del mapa, mientras que Bing Maps y Openrouteservice proporcionaron acceso a mapas detallados y servicios de enrutamiento.

La evaluación de usabilidad, basada en el modelo ISO/IEC 25010 empleando un enfoque cuantitativo, destacó altas puntuaciones en aprendizabilidad (92.56 puntos), operabilidad (93.54 puntos), protección contra errores de usuario (93.54 puntos) y estética de la interfaz (94.00 puntos). Este enfoque ágil, combinado con la tecnología seleccionada, culminó en una aplicación funcional, dinámica y altamente interactiva.

Palabras claves: Aplicaciones responsivas, sistemas de información geográfico, servidores de mapas en línea, Mapa Parlante, salud digital, Scrum, usabilidad, ISO/IEC 25010.

ABSTRACT

ABSTRACT

This research project embarked on a groundbreaking journey, focusing on the development of an innovative application with web and mobile access to the Mapa Parlante at the Centro de Salud Chambo. This application revolutionizes the management of risk factors, vulnerable groups, priority groups, and biological, socioeconomic, and environmental health risks through the geospatial visualization of data obtained from the Familiar Register.

The agile SCRUM methodology, a testament to our commitment to efficiency and adaptability, guided the software development process. Every step was carefully planned, with a client-server architecture with microservices chosen to ensure optimal performance. Strategically selected methodologies and tools were used in the construction of the application: PostgreSQL as the database management system, Nest.js in the backend, and Angular together with Tailwind CSS for the frontend development. Geo server and Open Layers enriched the geospatial data management and map interactivity, while Bing Maps and Open route service provided access to detailed maps and routing services.

The usability evaluation, based on the ISO/IEC 25010 model using a quantitative approach, highlighted high scores in learnability (92.56 points), operability (93.54 points), user error protection (93.54 points), and interface aesthetics (94.00 points). Combined with the selected technology, this agile approach culminated in a functional, dynamic, and highly interactive application.

Keywords: Responsive applications, geographic information systems, online map servers, (Talking Map) Mapa Parlante, digital health, Scrum, usability, ISO/IEC 25010.



Reviewed by:
Ms.C. Ana Maldonado León
ENGLISH PROFESSOR
C.I.0601975980

CAPÍTULO I. INTRODUCCIÓN

Las tecnologías de la información y comunicación (TIC) desempeñan un papel trascendental en el ámbito de la salud, ofreciendo una serie de beneficios que abarcan desde el acceso hasta la eficacia, eficiencia y generación de conocimiento [1]. No obstante, es crucial destacar que su impacto va más allá de estos aspectos tangibles. La Organización Mundial de la Salud (OMS) subraya la importancia de aprovechar plenamente el potencial de las TIC, y un ejemplo destacado es el empleo de sistemas de información geográfica (SIG). Estos sistemas, compuestos por hardware, software y procedimientos, tienen como objetivo resolver problemas de gestión y planificación en el ámbito de la salud [2]. Su capacidad para representar información mediante mapas se convierte en una herramienta esencial en el sector sanitario, permitiendo, por ejemplo, la visualización de la distribución de enfermedades y la geolocalización de centros de salud.

La ejecución de proyectos SIG en América Latina ha sido significativa, con enfoques dirigidos a identificar problemas relacionados con las condiciones de vida y a compilar y estandarizar la accesibilidad a los servicios de salud [3]. Este progreso se complementa con un crecimiento notable en el desarrollo web orientado a mejorar las necesidades en el área de la salud. La sinergia entre la informática y la salud no solo introduce nuevos procesos de trabajo que aportan agilidad y colaboración en proyectos, sino que también abre la puerta a una comprensión más profunda de los desafíos específicos que enfrentan países como Ecuador.

Ecuador presenta desafíos notables en la accesibilidad a servicios de salud inclusivos y de calidad, manifestados en problemáticas como el embarazo en adolescentes, el consumo de drogas y la escasez de insumos médicos. Este escenario subraya la urgencia de proporcionar una atención oportuna y de calidad a los pacientes [1].

En este contexto, la valoración de los mapas como guías cognitivas y representaciones gráficas de territorios adquiere una relevancia aún mayor. La evolución de los mapas hacia la categoría de "mapas parlantes" destaca su papel esencial en la cartografía de la salud. Estos mapas, al capturar no solo las representaciones visuales de territorios, sino también elementos temporales clave (pasado, presente y futuro), se erigen como guías cognitivas que fortalecen la identidad comunal y desempeñan un papel crucial en la toma de decisiones fundamentada en salud [3].

No obstante, para que esta información sea efectivamente utilizada en el ámbito de la salud, es esencial contar con infraestructuras sólidas para la visualización y gestión de datos geoespaciales [4]. Los servidores de mapas en línea, particularmente GeoServer, desempeñan un papel crucial al proporcionar esta infraestructura. Al adoptar estándares abiertos, como el Web Map Service (WMS), GeoServer no solo genera mapas visualmente atractivos, sino que también garantiza la interoperabilidad [5]. Su integración con tecnologías web amplía la accesibilidad y colaboración en la gestión de información geográfica en el sector de la salud, respaldando la toma de decisiones informada [3]. Dentro

del contexto del desarrollo del sistema Mapa Parlante, la capacidad de GeoServer para generar mapas de manera dinámica y su integración con tecnologías web permiten una mayor accesibilidad y colaboración en la gestión de la información geográfica en el ámbito de la salud.

1.1. Problema y Justificación

En Ecuador, los profesionales de la salud pública llevan a cabo visitas domiciliarias mediante brigadas de salud para atender integralmente a comunidades alejadas, ofreciendo servicios que abarcan desde la atención curativa hasta la promoción de la salud y la entrega de insumos definidos por el Ministerio de Salud Pública [6]. Este enfoque, centralizado en el Modelo de Atención Integral en Salud (MAIS), implica un intercambio de información entre el equipo de salud y las familias, permitiendo conocer aspectos como la situación familiar, el entorno y sus recursos [6].

Las visitas domiciliarias, esenciales en este modelo, demandan un abordaje complejo por parte de los profesionales de la salud. Este abordaje incluye la observación de las necesidades individuales en el contexto familiar, el establecimiento de relaciones colaborativas, y el fortalecimiento del papel de la familia en su propia salud. A través de la recolección de información, se busca identificar riesgos y daños, implementando planes de atención e intervención con el apoyo logístico, trabajo interdisciplinario, coordinación entre niveles asistenciales y educación para la salud.

En el Centro de Salud Chambo Tipo B, actualmente, la elaboración manual de planes de acción, especialmente utilizando mapas parlantes, representa un desafío que demanda recursos considerables, como tiempo, personal y transporte. Este enfoque, aunque valioso en la identificación de factores de riesgo mediante simbología representativa, requiere una optimización.

En este contexto, la presente investigación busca desarrollar una aplicación con acceso web y móvil mapa parlante diseñada para el Centro de Salud Chambo de la provincia de Chimborazo. Este enfoque permitirá gestionar factores de riesgo, como grupos vulnerables, grupos prioritarios, riesgos biológicos, socioeconómicos y ambientales, mediante la representación geoespacial de datos de las fichas familiares. La aplicación ofrecerá funcionalidades como gestión de Brigadas de los Equipos Integrales de Salud (E AIS), filtros personalizados, generación de reportes y rutas entre profesionales de salud y pacientes, contribuyendo así a la eficiencia y eficacia en la atención de la salud.

La usabilidad de esta aplicación representa una respuesta directa al desafío identificado en el Centro de Salud Chambo, brindando al personal de salud herramientas avanzadas para la gestión de factores de riesgo mediante representaciones gráficas geoespaciales. Esta investigación busca responder a la pregunta central: ¿Cómo la usabilidad de la aplicación mapa parlante con acceso web y móvil contribuirá a la gestión de factores de riesgo en el Centro de Salud de Chambo utilizando datos geográficos obtenidos de las fichas familiares?

1.2. Objetivos

1.2.1. Objetivo General

- Desarrollar una aplicación web y móvil mapa parlante en el Centro de Salud Chambo.

1.2.2. Objetivos Específicos

- Analizar metodologías y herramientas para la representación del mapa parlante en el Centro de Salud Chambo.
- Implementar una aplicación con acceso web y móvil mapa parlante en el Centro de Salud Chambo utilizando la metodología Scrum.
- Evaluar la usabilidad de la aplicación con acceso web y móvil mapa parlante en el Centro de Salud Chambo aplicando el modelo de calidad ISO/IEC 25010.

CAPÍTULO II. MARCO TEÓRICO

2.1. El Manual del Modelo de Atención Integral de Salud (MAIS)

El Manual del Modelo de Atención Integral de Salud (MAIS) adopta la concepción de salud conforme a la definición de la OMS en su Constitución de 1946. Según esta perspectiva, la salud no se limita a la mera ausencia de enfermedad, sino que implica un estado de bienestar completo que abarca lo físico, mental, espiritual, emocional y social. En este enfoque, se reconoce que la salud comprende la satisfacción de todas las necesidades esenciales de las personas, incluyendo aspectos afectivos, sanitarios, nutricionales, sociales y culturales [3].

La salud, según el MAIS, es considerada como un resultado de las condiciones sociales y biológicas, al mismo tiempo que actúa como un factor determinante de contextos que facilitan el desarrollo integral a nivel individual y colectivo. Se entiende que la salud se construye en el marco de diversas condiciones, tales como las económicas, sociales, políticas, culturales y ambientales, así como en las formas de interacción con los grupos sociales. También se destaca la importancia de la distribución equitativa de bienes y servicios de salud en cada sociedad [3].

La Constitución Ecuatoriana respalda esta visión integral de la salud, reconociéndola como un derecho fundamental en su Capítulo Segundo, Artículo 32. Este documento establece que el Estado tiene la responsabilidad de garantizar el derecho a la salud, y su realización se encuentra vinculada al ejercicio de otros derechos que sustentan el concepto de Buen Vivir [3].

2.2. Diseño web responsivo (RWD)

Este enfoque innovador, conocido como Diseño Web Responsivo (RWD), fue propuesto por el diseñador web estadounidense Ethan Marcotte en 2010. Marcotte sugirió alejarse de diseños a medida y aislados para cada dispositivo web, proponiendo en su lugar considerarlos como facetas de una única experiencia. La esencia de esta perspectiva radica en diseñar para una experiencia óptima de observación, integrando tecnologías basadas en estándares en los diseños para hacerlos no solo más flexibles, sino también más adaptables al medio que los muestra [7].

RWD se configura como un enfoque de desarrollo web que efectúa cambios dinámicos en la apariencia de un producto web según el tamaño de la pantalla y la orientación del dispositivo que lo visualiza. Este método utiliza puntos de interrupción para determinar la apariencia del diseño en función del ancho del navegador. A diferencia de crear productos y códigos separados para distintos dispositivos, RWD utiliza una base de código única, lo que puede resultar en un desarrollo más rápido y fácil de mantener a lo largo del tiempo. Además, RWD es adaptable y "a prueba" para el futuro, ya que puede admitir nuevos puntos de interrupción en cualquier momento, brindando flexibilidad sin vincularse a un dispositivo específico [7].

2.1.1. Ventajas del RWD respecto a la accesibilidad

- Separación de contenido y presentación

Los estilos se definen en CSS, no se usan tablas para maquetar, lo que beneficia a personas con diversas discapacidades al permitir adaptar el contenido según sus necesidades [7].

- Mayor respeto por estándares web

Maximiza la compatibilidad con aplicaciones de usuario actuales y futuras, incluyendo ayudas técnicas [7].

- Estructuración y jerarquización de información

Favorece la accesibilidad al tener información estructurada y jerarquizada correctamente [7].

- Uso de elementos semánticos

Permite definir estilos en CSS, facilitando la accesibilidad al determinar la función estructural del contenido [7].

- Diseño flexible y tamaños relativos

Facilita la ampliación de texto y zoom sin desbordamientos, mejorando la experiencia para usuarios con baja visión [7].

- Orden de lectura correcto

Diseños fluidos presentan el contenido en el mismo orden que el DOM, favoreciendo la lectura para lectores de pantalla [7].

- Visualización en distintos dispositivos

Considera la variabilidad en las formas de interactuar y mejora el contraste de color para adaptarse a dispositivos móviles [7].

- Focalización en lo necesario

Prioriza y simplifica, resultando en productos más fáciles de navegar y entender, mejorando la accesibilidad [7].

2.3. Aplicaciones responsivas con Angular

Angular es un marco de desarrollo de código abierto respaldado por Google, diseñado específicamente para la creación de páginas web de tipo SPA (Single Page Application). En sus inicios en 2010, se lanzó como AngularJS, pero experimentó una reescritura completa en 2016, adoptando el nombre Angular. Para utilizar Angular de manera efectiva, es recomendable tener conocimientos sólidos en HTML, CSS, JavaScript y programación orientada a objetos (POO). Su lenguaje de programación principal es TypeScript, desarrollado por Microsoft y de código abierto [8].

Esta plataforma proporciona a los desarrolladores un conjunto integral de herramientas para la creación de aplicaciones web, móviles y de escritorio al combinar plantillas declarativas, inyección de dependencias y utilidades integradas. Angular permite a los desarrolladores escribir código en JavaScript moderno, haciendo uso de inyección de dependencia y decoradores para metadatos [9].

El ciclo de vida de los componentes de Angular se presenta de manera detallada en la figura 1, brindando una visión clara de cómo interactúan y evolucionan los elementos dentro de la estructura de Angular.

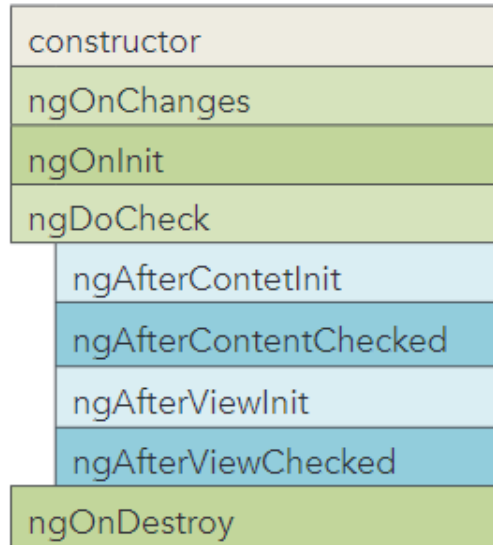


Figura 1. Ciclo de vida de Angular
Fuente: [9]

Una SPA, acrónimo de Single Page Application (Aplicación de Página Única), es un tipo específico de aplicación web en la que todas las interfaces de usuario se presentan en una única página, eliminando la necesidad de recargar el navegador. Desde una perspectiva técnica, una SPA se configura como un sitio web con un solo punto de entrada, típicamente identificado como el archivo index.html. En este contexto, no existe ningún otro archivo HTML accesible de manera independiente dentro de la aplicación que pueda mostrar contenido o componentes de la aplicación por separado [10].

2.3.1. Arquitectura de Angular

Angular se adhiere a una estructura orientada a componentes conocida como "Arquitectura de Componentes". Esta metodología guarda similitudes con el patrón MVC (Modelo-Vista-Controlador), aunque incorpora conceptos específicos propios del framework [8]. Los elementos principales de este framework son:

- Módulos
- Componentes
- Servicios
- Directivas
- Inyección de dependencias
- Plantillas
- Metadatos
- Vinculación de datos

La representación visual que se muestra en la figura 2, da a conocer la relación de estos conceptos esenciales para la construcción de una aplicación en Angular [11].

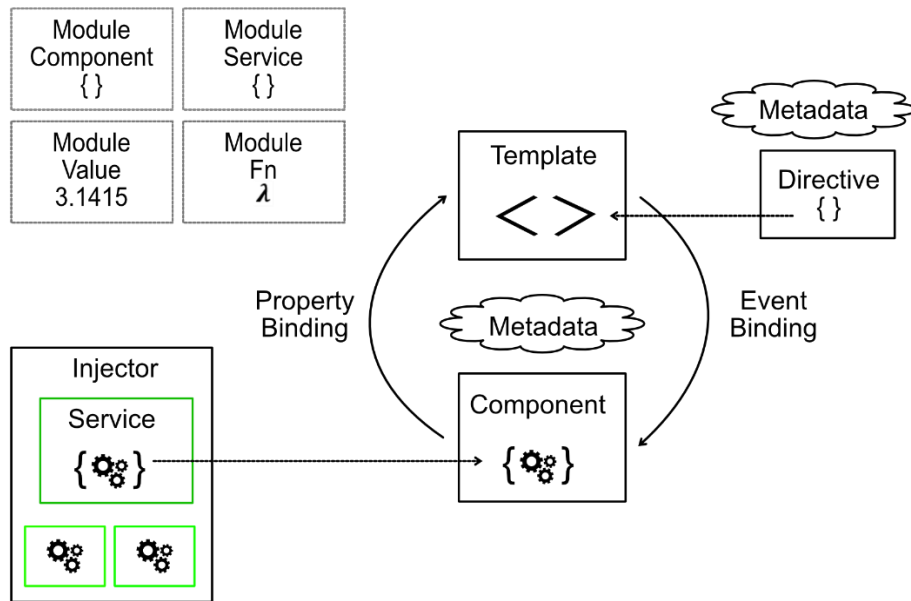


Figura 2. Arquitectura básica de Angular
Fuente: [11]

2.3.2. Lenguaje de desarrollo de Angular: TypeScript

TypeScript es un lenguaje de programación de código abierto creado por Microsoft en el año 2012, que implementa muchos mecanismos de programación orientada a objetos. Se trata de un superconjunto de JavaScript, lo que significa que extiende la sintaxis de JavaScript. Al compilarlo, se genera código JavaScript, ya que el navegador no puede interpretar TypeScript. Este proceso se llama transpilar, que significa generar código en un lenguaje específico a partir de otro [12].

2.4. Aplicación del lado del servidor con Nest.js

2.4.1. Node Js

Es un entorno de ejecución orientado a eventos de JavaScript desarrollado para construir aplicaciones, una de las principales características es desarrollar aplicaciones de red escalables, impulsados a eventos asíncronos, si no existe trabajo que realizar Node.js permanece dormido hasta que haya una conexión que active el proceso, esto gracias al uso de JavaScript del lado del servidor. Node.js un entorno de ejecución desarrollado con el motor JavaScript V8 de Chrome, diseñado para generar aplicaciones web de manera óptima y ágil [13].

Node.js emplea un modelo de evaluación de un solo hilo de ejecución que se basa en un esquema de entrada/salida asíncrona conocido como Event Loop. Esta característica posibilita el procesamiento eficiente de numerosas llamadas sin recurrir a cambios de contexto, que pueden ser costosos a nivel del sistema operativo. Además, Node.js incorpora una amplia variedad de módulos y componentes específicamente optimizados para operaciones de redes. Estos elementos brindan soporte para la gestión de estándares y protocolos de Internet convencionales, tales como DNS, HTTP, TCP, TLS/SSL y UDP [14].

Una destacada particularidad de Node.js es su enfoque en el desarrollo de aplicaciones de red escalables impulsadas por eventos asíncronos. En otras palabras, cuando no hay tareas pendientes, Node.js permanece inactivo hasta que se activa una conexión que demanda una respuesta. Esta capacidad se fundamenta en el uso de JavaScript en el lado del servidor [15].

En la figura 3 se puede observar la estructura de creación de un restApi, utilizando Node.js, servicio integrado para móviles y ordenadores.

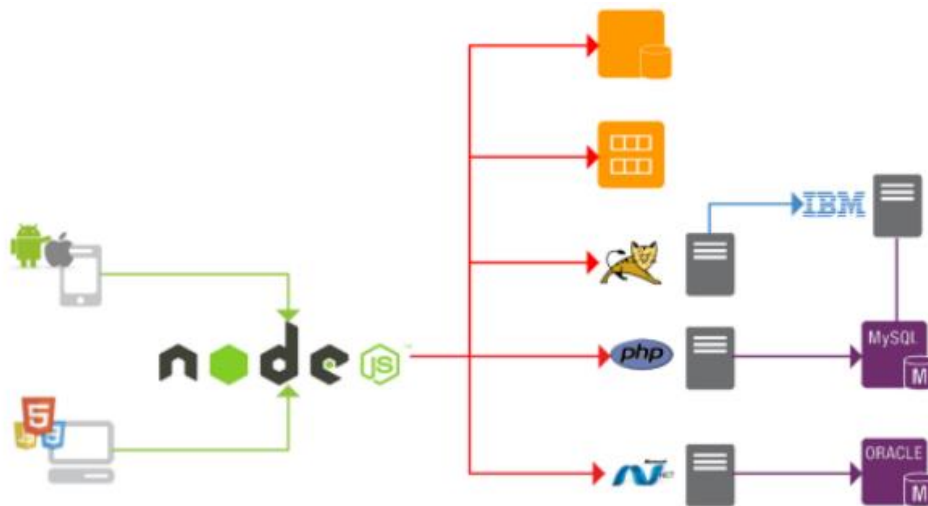


Figura 3. Arquitectura de Node.js
Fuente: [13]

2.4.2. Nest Js

Nest (NestJS) es un framework para construir aplicaciones Node.js eficientes y escalables del lado del servidor. Utiliza JavaScript progresivo, está construido y es totalmente compatible con TypeScript (aunque permite a los desarrolladores codificar en JavaScript puro) y combina elementos de POO (Programación Orientada a Objetos), PF (Programación Funcional) y FRP (Programación Reactiva Funcional). Nest hace uso de sólidos marcos de servidores HTTP como Express (por defecto) y, opcionalmente, puede configurarse para utilizar también Fastify. Nest proporciona un nivel de abstracción por encima de estos frameworks Node.js comunes (Express/Fastify), pero también expone sus APIs directamente al desarrollador. Esto da a los desarrolladores la libertad de utilizar la miriada de módulos de terceros que están disponibles para la plataforma [16].

2.4.3. Estructura del Framework Nest.js

Nest.js, por defecto, utiliza Express, un framework de servidor HTTP ampliamente reconocido. No obstante, brinda la flexibilidad de configurar otros frameworks comunes de Node.js, como Fastify. La arquitectura de Nest se inspira en gran medida en la estructura de Angular, lo que simplifica la administración de datos en el servidor y permite una respuesta eficiente al cliente. Esta arquitectura se basa en tres capas fundamentales: controladores, proveedores y módulos, que son esenciales para la construcción de aplicaciones en Nest.js [16].

A continuación, se detallan los conceptos fundamentales que conforman la base para desarrollar una aplicación en Nest.js.

- Nest CLI

El Nest CLI es una herramienta de interfaz de línea de comandos que facilita la inicialización, desarrollo y mantenimiento de aplicaciones Nest.js. Su funcionalidad abarca diversas áreas, como el establecimiento del proyecto, la ejecución en modo de desarrollo, la construcción y el empaquetado para la distribución en producción. Además, integra patrones arquitectónicos de mejores prácticas para fomentar la creación de aplicaciones bien estructuradas [16].

La estructura de carpetas típica de una aplicación Nest.js se ilustra en la figura 4.

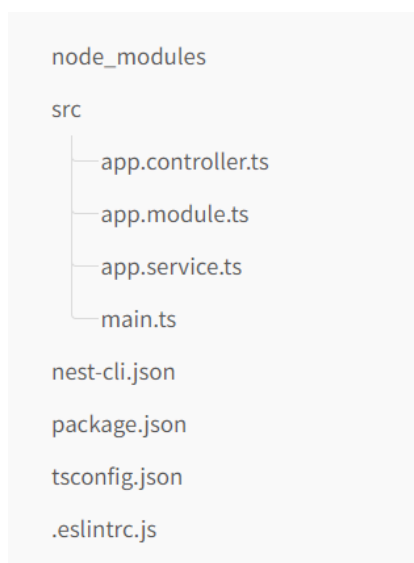


Figura 4. Estructura de carpetas de un proyecto en Nest.js

Fuente: [16]

- Módulos

Los módulos representan el componente esencial de Nest.js, sirviendo para organizar el código en unidades lógicas que facilitan la gestión de la aplicación. Cada unidad lógica conforma un módulo, y cada aplicación Nest debe tener al menos un módulo inicial, denominado módulo raíz. Un módulo se define como una clase TypeScript simple, decorada con el decorador `@Module ()`, incluido en el paquete `@nestjs/common` [17].

- Controladores

Los controladores son responsables de manejar las solicitudes entrantes y proporcionar respuestas al cliente. Utilizan el decorador `@Controller ()` y se centran en recibir solicitudes específicas para la aplicación. El mecanismo de enrutamiento coordina las solicitudes que un controlador recibe, y generalmente, cada controlador cuenta con múltiples rutas, donde cada ruta puede ejecutar acciones distintas [16].

- Proveedores

En Nest.js, los proveedores son conceptos fundamentales, representados por bloques que pueden inyectarse como dependencias. Son clases TypeScript simples, decoradas con `@Injectable ()`, parte del paquete `@nestjs/common` [17].

- Servicios

Los servicios se encargan de la lógica empresarial de la aplicación y pertenecen al grupo de proveedores. Actúan como intermediarios entre el controlador y la base de datos. Utilizan el decorador `@Injectable ()` y, por lo general, gestionan el almacenamiento y la recuperación de datos [18].

- Inyección de dependencias

Gracias a las capacidades de TypeScript, la gestión de dependencias se simplifica, ya que se resuelven por tipo. La inyección de dependencias es un modelo de programación en el que una clase, en lugar de crear sus propias dependencias, las requiere de fuentes externas. Esta aproximación permite que la clase reciba las dependencias necesarias directamente en el constructor [18].

2.5. Gestión de bases de datos

2.5.1. Base de datos

Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización, como, por ejemplo, una universidad o un hospital. Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios [19].

2.5.2. El modelo relacional

El modelo de datos relacional es una representación tabular de una base de datos, siguiendo la estructura de las tablas generadas a partir del diagrama Entidad-Relación (E-R). Este enfoque establece una correspondencia directa entre el concepto de tabla y la noción matemática de una relación, donde cada columna de la tabla representa una relación entre un conjunto de elementos. En este sentido, una tabla se concibe como un conjunto de relaciones, y la denominación "relacional" se deriva del hecho de que cada registro de la base de datos contiene información vinculada a un tema específico [20].

Este modelo se encarga de representar datos y sus relaciones mediante una colección de tablas, donde las filas (tuplas) se equiparán a los registros de la base de datos, y las columnas reflejan las características de cada registro localizado en la tupla [21].

2.5.3. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional que opera bajo la licencia BSD, y su código fuente está accesible de manera gratuita. En la actualidad, se destaca como uno de los sistemas de gestión de bases de datos de código abierto más robustos

y ampliamente utilizados. Este sistema adopta un modelo cliente-servidor y opta por el uso de múltiples procesos en lugar de múltiples hilos para asegurar la estabilidad del sistema. En caso de que un proceso experimente un fallo, dicho incidente no afectará al resto, y el sistema continuará funcionando de manera ininterrumpida [22].

A continuación, en la figura 5 se puede observar la estructura y funcionamiento de PostgreSQL.

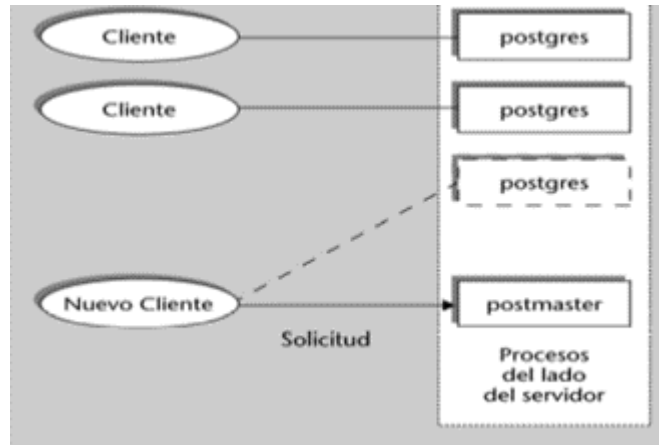


Figura 5. Arquitectura de PostgreSQL.

Fuente: [23]

PostgreSQL ofrece una serie de características destacadas diseñadas para asistir a los desarrolladores en la creación de aplicaciones, proporcionar a los administradores herramientas para asegurar la integridad de los datos, establecer entornos tolerantes a fallos y gestionar datos, independientemente de su tamaño. Además de ser gratuito y de código abierto, PostgreSQL es altamente adaptable, permitiendo la definición personalizada de datos, la creación de funciones a medida, y la escritura de código en varios lenguajes de programación sin la necesidad de recompilar la base de datos. PostgreSQL se esfuerza por seguir el estándar SQL, siempre y cuando esta conformidad no entre en conflicto con sus características tradicionales o conduzca a decisiones arquitectónicas inapropiadas. Por estas razones, esta base de datos relacional es elogiada por su estabilidad, potencia, robustez y su facilidad tanto de administración como de implementación [24].

2.5.4. TypeORM

TypeORM se presenta como un mapeador objeto-relacional (ORM, por sus siglas en inglés) compatible con diversas plataformas, como Node.js, Browser, Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo y Electron, y es compatible con TypeScript y JavaScript (ES2021). Su propósito es respaldar las últimas características de JavaScript y ofrecer funcionalidades adicionales para el desarrollo de aplicaciones que interactúan con bases de datos, desde proyectos pequeños hasta aplicaciones empresariales de gran envergadura. Lo que diferencia a TypeORM de otros ORMs en JavaScript es su soporte para los patrones Active Record y Data Mapper, posibilitando la creación de aplicaciones de alta calidad, con un acoplamiento flexible, escalables y fácilmente mantenibles. Además, TypeORM ha sido

influenciado por otros ORMs destacados, como Hibernate, Doctrine y Entity Framework [25].

2.6. Comunicación entre cliente y servidor

2.6.1. API REST

Una API, que significa Interfaz de Programación de Aplicaciones, constituye un conjunto de definiciones y protocolos destinados a diseñar e integrar el software de las aplicaciones. Por consiguiente, se percibe comúnmente como un acuerdo entre el proveedor de información y el usuario, estableciendo el contenido necesario tanto para el consumidor (petición) como para el productor (respuesta) [26].

Siguiendo esta línea, una API REST (Transferencia de Estado Representacional) es un servicio web que utiliza la arquitectura de REST para atender una solicitud proveniente de una aplicación cliente (frontend). En otras palabras, cuando un cliente realiza una solicitud mediante una API de tipo RESTful, esta traslada una representación del estado del recurso solicitado al que la requirió. La información de respuesta se transmite mediante el protocolo HTTP y adopta uno de estos formatos: JSON (JavaScript Object Notation), HTML, XML o texto sin formato [27].

2.7. Servidores de mapas en línea

2.7.1. Mapa Parlante

Empleados como una herramienta técnica y metodológica para la comunicación e intercambio de diversas perspectivas, los mapas parlantes identifican tres dimensiones temporales: pasado, que implica un análisis de las relaciones previas en el espacio que han llevado a una condición particular explicativa de la situación actual; presente, que consiste en un diagnóstico de la realidad territorial en relación con la temática seleccionada; y futuro, que representa una visión prospectiva del territorio al buscar integrar una perspectiva compartida sobre las proyecciones de la comunidad para su espacio. El Mapa Parlante, una técnica de manejo sencillo, al centrarse en la colectivización del conocimiento comunitario, facilita un flujo de información intergeneracional y horizontal de gran importancia educativa [28].

Los mapas parlantes forman parte de las técnicas del Diagnóstico Rápido Participativo (DRP) utilizadas para recopilar información específica sobre un evento en relación con un territorio; esta información se orienta hacia percepciones construidas de manera colectiva. Estos mapas forman parte de diversas cartografías empleadas en procesos de recopilación de información, ya sea con fines diagnósticos o investigativos, especialmente en técnicas interactivas. La técnica implica la creación de un bosquejo de un área geográfica y se utiliza principalmente en el análisis de la situación y en el desarrollo de la visión. El Mapa Parlante constituye una primera fuente de información sobre temas relevantes en el contexto del análisis participativo, como mapas de recursos, mapas de conflictos y mapas de la estructura social [29].

2.7.2. Sistemas de Información Geográfica en la salud

Un SIG es un sistema utilizado para describir y categorizar la Tierra y otras geografías, con el propósito de mostrar y analizar información referida espacialmente, principalmente a través de mapas. La meta del SIG es crear, compartir y aplicar productos de información basada en mapas que respalden las operaciones de las organizaciones y gestionar información geográfica relevante. Los mapas representan conjuntos lógicos de información geográfica organizados como capas de mapas, sirviendo como una metáfora eficaz para modelar y organizar la información geográfica en forma de capas temáticas. Además, los mapas SIG interactivos constituyen la interfaz principal con la que se accede y utiliza la información geográfica [30].

Los Sistemas de Información Geográfica (SIG) están siendo cada vez más empleados a nivel global como parte de programas de vigilancia de salud pública. En contraste con otros sistemas de información utilizados para actividades de vigilancia, las herramientas SIG poseen la capacidad excepcional de vincular datos de salud, población y medio ambiente, posibilitando a investigadores y autoridades sanitarias mapear enfermedades y evaluar cuantitativamente posibles riesgos socioambientales para la salud en diversas escalas [4].

Definido como un sistema computacional dedicado a capturar, almacenar, manipular, analizar y visualizar datos referenciados espacialmente, el software SIG se encuentra comúnmente almacenado en computadoras de escritorio o portátiles, y su operación requiere usuarios con experiencia. No obstante, debido al avance tecnológico y la rápida expansión de Internet en los últimos tiempos, el desarrollo de los SIG está orientado hacia su integración con las tecnologías de la World Wide Web (web). Los SIG que aprovechan las tecnologías web para la comunicación entre sus componentes, como datos, recursos e interfaz gráfica, son conocidos como SIG basados en web o simplemente SIG-Web, ampliando las funcionalidades de los SIG tradicionales al permitir la gestión, análisis y representación dinámica de la geo información entre usuarios con diferentes niveles de conocimiento [4].

2.7.3. Geoserver

Geoserver se presenta como un servidor web de software libre desarrollado en Java que posibilita la visualización y edición de datos geoespaciales. Al adoptar estándares abiertos definidos por el Consorcio Geoespacial Abierto (OGC), Geoserver brinda una amplia flexibilidad para la creación de mapas y el intercambio de datos [5]. Permite publicar conjuntos de datos garantizando así su interoperabilidad, GeoServer es una implementación de referencia a la hora de crear servicios web de mapas (WMS).

Gracias a la implementación del estándar Web Map Service (WMS), GeoServer puede generar mapas en diversos formatos de salida. Además, cuenta con la integración de OpenLayers, una biblioteca cartográfica gratuita que simplifica y acelera el proceso de creación de mapas. GeoServer se basa en GeoTools, un conjunto de herramientas SIG Java de código abierto. Más allá de la creación de mapas estéticos, GeoServer cumple con los estándares Web Feature Service (WFS) y Web Coverage Service (WCS), lo que posibilita

compartir y editar los datos utilizados para la generación de mapas. También hace uso del estándar Web Map Tile Service para dividir los mapas en mosaicos, facilitando su integración en aplicaciones móviles y de cartografía web [5].

GeoServer se presenta como una aplicación modular con la capacidad de ampliar sus funcionalidades mediante extensiones. Las extensiones para el Servicio de Procesamiento Web ofrecen diversas opciones de procesamiento, incluso la posibilidad de desarrollar extensiones personalizadas. Facilitando que otros incorporen sus datos en sus sitios web y aplicaciones, liberando la información y promoviendo la transparencia [5].

Organización interna de las capas en GeoServer

Para comprender la estructura de GeoServer, se presenta la siguiente figura 6, que ofrece una visión detallada de su configuración.

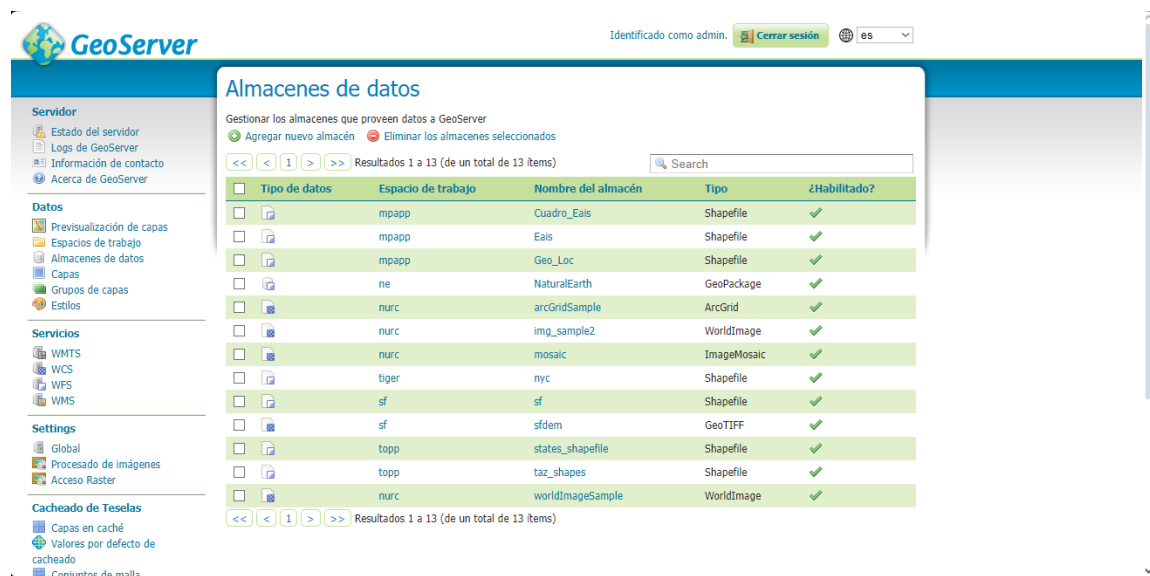


Figura 6. Servidor de Geoserver

Fuente: El Autor

- El espacio de trabajo – Works pace

Un espacio de trabajo, también conocido como workspace, actúa como un contenedor destinado a organizar capas afines, proporcionando un entorno separado y aislado vinculado a un proyecto específico. La utilización de espacios de trabajo posibilita el uso de capas con nombres idénticos en otros espacios de trabajo sin que surjan conflictos. En GeoServer, cada capa y almacén de datos deben tener un espacio de trabajo asociado, a excepción de los grupos de capas [31].

- Almacén de datos – store

Un almacén de datos, también conocido como store, se refiere al recipiente que contiene datos geográficos. Este almacén, o store, se conecta a una fuente de datos que engloba información ya sea ráster o vectorial. La fuente de datos puede ser un archivo, como un geopackage o conjunto de archivos, una tabla dentro de una base de datos o incluso un directorio. Es esencial destacar que cada almacén de datos debe estar vinculado a un espacio de trabajo (workspace), limitándose a asociarse únicamente con uno [31].

- Capas

Una capa hace referencia a los datos geográficos, ya sean ráster o vectoriales, que incluyen los elementos geográficos. Cada objeto geográfico, conocido como "Feature", se representa mediante capas para su visualización en un mapa. Geoserver guarda información relacionada con la capa, como la proyección, la extensión geográfica o el estilo vinculado [31].

Fuentes de datos y servicios

GeoServer tiene la capacidad de acceder a diversas fuentes de datos, que abarcan desde archivos almacenados en el disco local en formato vectorial, como shapefiles de ESRI y GeoPackage, hasta bases de datos como PostGIS. Además, puede manejar archivos ráster en formatos como ArcGrid, GeoPackage ráster, Geo TIFF, ImageMosaic o WorldImage [31].

GeoServer implementa los protocolos estándares open web que establece el Open Geospatial Consortium (OGC), y es un servidor de alto rendimiento compatible con la certificación Web Map Service (WMS), y de hecho es la implementación de referencia de las normas OGC Web Feature Service (WFS) y Web Coverage Service (WCS) [31], véase el acceso de los datos y el consumo que este nos permite, en la figura 7.



Figura 7. Estructura de manejo de entrada y salida de datos en Geoserver

Fuente: [31]

- **Estándar Web Map Service**

El estándar de Interfaz de Servicio de Mapas Web (WMS) ofrece una interfaz HTTP sencilla para solicitar imágenes de mapas georreferenciadas desde una o varias bases de datos geoespaciales distribuidas. Al realizar una solicitud WMS, se especifican la(s) capa(s) geográfica(s) y el área de interés a procesar. La respuesta a esta solicitud es una o varias imágenes cartográficas georreferenciadas (en formatos como JPEG, PNG, etc.) que pueden visualizarse en una aplicación de navegador. Además, la interfaz permite indicar si las imágenes devueltas deben ser transparentes, facilitando la combinación de capas de diversos servidores [32].

2.7.4. Open Layers

OpenLayers es una destacada librería de alto rendimiento diseñada para la creación de mapas interactivos en entornos web. Esta versátil herramienta tiene la capacidad de exhibir mapas teselados, datos vectoriales y marcadores provenientes de diversas fuentes, otorgando a los desarrolladores la flexibilidad de incorporar esta funcionalidad en cualquier página web. OpenLayers ha sido creada con el propósito de promover el aprovechamiento de información geográfica diversa. Aunque no es posible identificar de manera precisa quiénes son los usuarios de esta librería en la web, se sabe que es la elección preferida en reconocidos paquetes de software como GeoShape, Cartaro o GeoMoose, todos ellos destinados a la construcción de soluciones completas para Sistemas de Información Geográfica (GIS) en la web [33].

Características de uso de OpenLayers

OpenLayers destaca en la comunidad gracias a diversas ventajas que ofrece, entre las cuales se incluyen:

- Simplicidad de Uso

OpenLayers se distingue por su interfaz de fácil manejo, lo que facilita tanto a usuarios como desarrolladores la creación y manipulación de mapas interactivos de manera intuitiva [33].

- Independencia de Servidores de Mapas

Esta librería no está limitada a servidores de mapas específicos, brindando la flexibilidad de trabajar con diferentes fuentes y proveedores de datos geográficos [33].

- Compatibilidad con Servicios WMS y WFS

OpenLayers aborda de manera transparente los servicios Web Map Service (WMS) y Web Feature Service (WFS), ofreciendo una integración sencilla para usuarios y desarrolladores [33].

- Soporte de Tiles y Caché

La capacidad de trabajar con tiles (teselas) y la gestión eficiente de caché contribuyen a mejorar el rendimiento, proporcionando una experiencia de usuario más rápida y eficaz [33].

- Acceso a Mapas de Google Maps y Yahoo Maps

OpenLayers facilita la integración y acceso a mapas de reconocidos proveedores como Google Maps y Yahoo Maps, ampliando las opciones disponibles para el desarrollo de aplicaciones basadas en mapas [33].

2.7.5. Open Route Service

Openrouteservice, un servicio de enrutamiento de código abierto de acceso gratuito, se sustenta en los datos de OpenStreetMap (OSM). Sus servicios abarcan la planificación de rutas, generación de matrices de distancia y tiempo de viaje, e isolíneas para diversos perfiles, como peatón, bicicleta, automóvil o silla de ruedas. Además, ofrece funciones de geo codificación y correspondencia de mapas. En la actualidad, openrouteservice utiliza una bifurcación altamente modificada de Graphhopper para llevar a cabo el enrutamiento [34].

La API openrouteservice ofrece servicios espaciales globales al procesar datos geográficos gratuitos generados por usuarios y recopilados colaborativamente desde OpenStreetMap. Este servicio, altamente adaptable y eficiente, está implementado en Java [35].

A través de una interfaz HTTP proporcionada por Tomcat, se accede a los siguientes servicios:

- Indicaciones

Proporciona una ruta entre dos o más ubicaciones para un perfil específico, con la opción de ajustes personalizados y direcciones adicionales [35].

- Isócronas

Genera áreas de accesibilidad desde ubicaciones dadas, mostrando las zonas alcanzables en un determinado tiempo [35].

- Matriz

Calcula rutas uno-a-muchos, muchos-a-uno o muchos-a-muchos para cualquier modo de transporte admitido por openrouteservice [35].

2.8. Modelo de Calidad ISO/IEC 25010

El modelo de calidad representa la piedra Angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas [36].

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la figura 8:



Figura 8. Características del modelo de calidad de software ISO/IEC 25010.

Fuente: [36]

2.8.1. Usabilidad

ISO / IEC 25010 define la calidad en uso como "el grado con el cual un producto es utilizado por usuarios específicos satisfaciendo sus necesidades para lograr objetivos específicos con efectividad, eficiencia, seguridad y satisfacción en contextos específicos de uso". La

usabilidad es una característica planteada por la norma ISO 25010, esta característica representa la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones (ISO/IEC 25000, 2017), esta característica se subdivide a su vez en las siguientes subcaracterísticas [36]:

- Aprendizabilidad.

Capacidad del producto que permite al usuario aprender su aplicación.

- Operabilidad.

Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

- Protección contra errores de usuario

Capacidad del sistema para proteger a los usuarios de hacer errores.

- Estética de la interfaz de usuario

Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

2.9. Metodología Scrum

La aplicación de metodologías ágiles, como SCRUM, ha demostrado ser altamente eficiente en el desarrollo de software, según diversos autores. Un estudio se enfocó en la estandarización de procesos de desarrollo mediante prácticas de programación y SCRUM en departamentos de TI, con el objetivo de mejorar la funcionalidad del software. Este autor subraya que la aplicación de SCRUM facilita la integración del usuario en el proceso de desarrollo, generando un compromiso en el equipo para obtener un producto de calidad que cumpla con las expectativas del usuario [37].

Entre los beneficios destacados para la implementación de SCRUM se encuentran la entrega rápida de resultados, la gestión de expectativas del cliente, la flexibilidad y adaptación a nuevas prioridades, el retorno de inversión (ROI) mediante la priorización de requisitos, la definición temprana de riesgos para su mitigación, la mejora continua en productividad y calidad, y el trabajo colaborativo entre el cliente y el equipo desde la concepción de requisitos [37].

En cuanto a los componentes de SCRUM, se dividen en fases y roles. Las fases, también conocidas como Sprints, incluyen la planificación del Backlog, el seguimiento diario del sprint y la revisión del incremento al finalizar el sprint. Los roles comprometidos con el proyecto incluyen al Product Owner, SCRUM Master y el Equipo de Desarrollo, mientras que los usuarios, stakeholders y managers son parte del segundo grupo, proporcionando realimentación y tomando decisiones finales [37].

En resumen, SCRUM destaca por su capacidad para ofrecer resultados rápidos, gestionar expectativas, adaptarse a cambios, maximizar el ROI y promover la colaboración entre los diversos actores involucrados en el proceso de desarrollo de software [37].

CAPÍTULO III. METODOLOGÍA

Esta investigación se llevó a cabo utilizando un enfoque cuantitativo, donde la evaluación de la usabilidad se realizó conforme a los parámetros definidos por el modelo de calidad ISCO/EC 250100. En el marco de desarrollo de este proyecto, se implementó la metodología ágil de software SCRUM, este es un enfoque ágil que proporciona un respaldo efectivo para la organización y gestión del trabajo a través de un conjunto de principios, valores y prácticas. Esta metodología de desarrollo de software fomenta el trabajo en equipo con una estrecha colaboración con el cliente, aplicando buenas prácticas que permiten mejoras continuas y aceptan los cambios que puedan surgir en distintas etapas [37]. Se centra en lograr el mejor resultado posible mediante la retroalimentación constante sobre tareas y obstáculos, especialmente cuando los requisitos del usuario pueden cambiar.

3.1. Tipo de Investigación

3.1.1. Diseño de la Investigación

Esta investigación adoptó un enfoque cuantitativo, utilizando la norma ISO/IEC 25010 para evaluar la usabilidad de la aplicación Mapa Parlante con acceso web y móvil. Durante esta evaluación, se aplicaron diversas métricas contempladas en dicha norma.

- **Según el objeto de estudio (Usabilidad)**

Aplicada: Se implementó la aplicación responsiva Mapa Parlante mediante la metodología SCRUM, abordando de manera efectiva un problema identificado. Durante este proceso, se identificaron y atendieron las necesidades específicas que surgieron.

Cuantitativa Experimental: Es fundamental obtener conclusiones estadísticas que permitan recopilar información procesable, para determinar el grado de usabilidad de la aplicación responsiva Mapa Parlante por parte de los profesionales de la salud que utilizaran la misma. En este tipo de diseño, se observa el efecto que produce la variable independiente de la investigación “Aplicación con acceso web y móvil” sobre la variable dependiente “Usabilidad de la aplicación con acceso web y móvil”.

- **Según la fuente de investigación**

Bibliográfica: Analizar metodologías y herramientas para la representación del Mapa Parlante en el Centro de Salud Chambo, la investigación se apoya en la revisión documental durante la fase teórica y elaboración del documento, obteniendo la información de: libros, artículos científicos, publicaciones, tesis, revistas, etc.

3.1.2. Técnicas de recolección de Datos

- **Encuesta (Cuestionario Cerrado)**

Esta técnica de recolección de datos permitió obtener información directamente de los usuarios finales de la aplicación para conocer el grado de usabilidad del sistema.

3.1.3. Instrumentos de recolección de datos

- **Google FORMS**

Herramienta utilizada para la creación de encuestas mediante formularios de manera muy accesible y didáctica que permitió la obtención y captación de datos aplicados a los profesionales de salud en el Centro de Salud Chambo, posterior a una capacitación sobre el manejo del sistema.

3.1.4. Población de estudio y tamaño de muestra

La población de estudio en esta investigación incluye a 13 profesionales de la salud del Centro de Salud Chambo. Dado que esta población se considera finita y cuenta con menos de 50 individuos, se optará por una estrategia que aprovecha la totalidad de la población en lugar de aplicar un muestreo.

Para la evaluación del criterio de usabilidad de la aplicación con acceso web y móvil, se ha optado por utilizar el método de evaluación ISO/IEC 25010. La selección de la muestra se realizará de manera no probabilística intencional, focalizándonos en la participación de trece profesionales de la salud pertenecientes al Centro de Salud Chambo Tipo B. La omisión de nombres se realiza por consideraciones de privacidad de la información de los participantes. Este enfoque permitió una elección deliberada de los profesionales que, según ciertos criterios específicos, se consideran más relevantes para la evaluación de la usabilidad de la aplicación en el contexto de la investigación.

3.2. Identificación de variables

3.2.1. Variable Dependiente

Usabilidad de la aplicación con acceso web y móvil.

3.2.2. Variable Independiente

Aplicación con acceso web y móvil mapa parlante.

3.2.3. Operacionalización de variables

La operacionalización de variables es un procedimiento esencial que contribuye a garantizar la confiabilidad y validez de los datos recabados en esta investigación, tal como se observa en la tabla 1, así como la precisión y relevancia de las conclusiones que se extraen de ellos.

Tabla 1. Operacionalización de variables

PREGUNTA DE INVESTIGACIÓN	TEMA	OBJETIVOS	VARIABLES	CONCEPTUALIZACIÓN	DIMENSIÓN	INDICADORES
¿Cómo la usabilidad de la aplicación mapa parlante con acceso web y móvil contribuirá a la gestión de factores de riesgo en el Centro de Salud de Chambo utilizando datos geográficos obtenidos de las fichas familiares?	Desarrollo de una aplicación web y móvil mapa parlante en el Centro de Salud Chambo	<p>Objetivo General Desarrollar una aplicación web y móvil mapa parlante en el Centro de Salud Chambo.</p>	<p>Independiente: Aplicación con acceso web y móvil</p>	<p>Sistema para la gestión de EAIS, generación de rutas de los EAIS, generación de reportes.</p>	<p>Aplicación con acceso web y móvil</p>	<p>Independiente: Proyección de Información. Numero de Módulos del sistema. Capas de información.</p>
		<p>Objetivo Específicos</p> <ul style="list-style-type: none"> • Analizar metodologías y herramientas para la representación del mapa parlante en el Centro de Salud Chambo. • Implementar una aplicación con acceso web y móvil mapa parlante en el Centro de Salud Chambo utilizando la metodología Scrum. • Evaluar la usabilidad de la aplicación con acceso web y móvil mapa parlante en el Centro de Salud Chambo aplicando el modelo de calidad ISO/IEC 25010. 	<p>Dependiente: Usabilidad de la aplicación con acceso web y móvil</p>	<p>Calidad del uso de la aplicación con acceso web y móvil para la gestión de EAIS, generación de rutas de los EAIS, y generación de reportes.</p>	<p>Usabilidad</p>	<p>Dependiente: Aprendizabilidad. Operabilidad. Protección contra errores de usuario. Estética de la interfaz de usuario.</p>

3.3. Métodos de análisis y procesamiento de datos

3.3.1. Método estadístico

Esta investigación ha seguido un enfoque cuantitativo que aborda una población menor a cincuenta.

- **Análisis exploratorio de datos**

Posterior a la evaluación de usabilidad utilizando una encuesta cerrada se evaluó la métrica de calidad de la ISO/IEC 25010, se ha implementado este método para analizar los datos recolectados. Este análisis se ha realizado a través de gráficos estadísticos, permitiendo una interpretación detallada de los valores obtenidos en indicadores específicos como: Aprendizabilidad, operabilidad, protección contra errores de usuario y estética de la interfaz de usuario.

- **Estadística descriptiva**

Este enfoque se ha empleado para ofrecer una descripción y resumen de los datos, utilizando la medida de tendencia central conocida como media o promedio. La estadística descriptiva ha sido crucial para obtener una visión general de los datos y comprender mejor las características de la población en relación con los indicadores específicos de evaluación de la usabilidad.

3.4. Procedimiento

3.4.1. Investigación de metodologías y herramientas para la representación del Mapa Parlante y generación de una tabla de decisión para la posterior implementación de la aplicación con acceso web y móvil.

3.4.2. Implementación de una aplicación con acceso web y móvil Mapa Parlante en el Centro de Salud Chambo utilizando la metodología Scrum.

- Definición de los requisitos de software
- Instalación y configuración de sistemas operativos, servidores, herramientas y tecnologías a utilizar.
- Diseño e implementación de la base de datos la cual está compuesta por las siguientes tablas:
 - brigada_eai: Enfocada en la gestión de las brigadas de los equipos integrales de salud, permitiendo la creación y actualización de estado (activo, inactivo) de cada brigada.
 - profesional: Enfocada en el ingreso y actualización del estado (activo, inactivo) de los profesionales de salud de la institución.
 - profesion: Ingreso de tipo de profesiones que estén disponibles en la institución.
 - info_eai: Permite el ingreso de información de las zonas que subdivide el mapa correspondiente a Chambo.
- Implementación del escenario de desarrollo
Instalación y configuración del sistema operativo Windows 10 Pro en el cual se implementó lo siguiente:
 - Sistema gestor de base de datos PostgreSQL para almacenar la base de datos llamada dbMapaParlante.

- IDE de desarrollo Visual Code
- Servidor node.js, dentro del cual se ejecuta el framework de Nest con el lenguaje de programación TypeScript para el funcionamiento del backend.
- Angular para trabajar el frontend con el lenguaje de programación TypeScript, además incorporación de OpenLayers.
- Apache Tomcat dentro del cual se despliega Geoserver.
- Integración con otros servicios
 - Se integró el API de Bing Maps para la visualización de mapas.
 - Se implementó el API de Openrouteservice para la planificación de rutas.
- Pruebas y lanzamiento
 - Creación de la base de datos en PostgreSQL
 - Instalación y configuración de herramientas de servidores, Bitwise SSH Client
 - Construcción de la aplicación de Node.js
 - Subir a producción el server App Node.js
 - Construcción de la aplicación de Angular
 - Subir a producción la web app Angular
 - Configurar sistema operativo Windows Server 2016 en el cual se implementó y configuró el servicio de Apache Tomcat que va a desplegar Geoserver.
- Capacitación a los usuarios finales del centro de salud de Chambo
- Evaluar la usabilidad a través de la aplicación de una encuesta en el Centro de Salud Chambo.
- Análisis y resultados.

3.5. Desarrollo de la propuesta

La implementación de la aplicación sigue el marco metodológico ágil SCRUM. A continuación, se describen de manera detallada las distintas fases de su ejecución.

3.5.1. FASE I: Inicio

Definición del equipo

La esencia de la metodología Scrum radica en la colaboración eficiente de un equipo compacto. A continuación, en la tabla 2. se presenta una descripción de los roles y responsabilidades asignados al equipo Scrum en el proyecto Mapa Parlante:

Tabla 2. Equipo Scrum Mapa Parlante

Acrónimo	Rol	Descripción	Responsable
PO	Product Owner	Comunica los requerimientos empresariales	Pablo Álvarez
SCM	Scrum Master	Coordina el equipo y asigna las tareas.	Gonzalo Allauca
DevIT	Equipo de desarrollo	Encargado del desarrollo del sistema	Noralma Lucio

3.5.2. FASE II: Planificación y estimación

Requerimientos funcionales

La Tabla 3 presenta de manera detallada los requerimientos funcionales necesarios para el desarrollo de la aplicación.

Tabla 3. Requerimientos funcionales

Responsable:		Noralma Lucio
Código	Descripción del Requerimiento	
RF-01	Las aplicaciones contarán con seis tipos de usuarios: el administrador y 5 usuarios EAIS (ER13, ER15, ER45, ER51, EU03).	
RF-02	Las aplicaciones dispondrán de los siguientes roles: administrador, usuario EAIS, donde el administrador es el encargado del control del sistema con todos los permisos y acceso a la información, el usuario EAIS podrá visualizar los mapas geospaciales correspondientes a su desempeño con marcación de rutas origen destino hacia el paciente que debe ser atendido, este tiene limitaciones en la visualización y recuperación de datos y obtención de reportes.	
RF-03	La aplicación será responsiva, es decir tendrá la posibilidad de adaptarse a cualquier dispositivo conectado a internet.	
RF-04	Las aplicaciones contarán con una pantalla de inicio de sesión donde se ingresará el usuario y la contraseña, las cuales son de carácter obligatorio para todos los usuarios del sistema, sin importar el nivel.	
RF-05	Las aplicaciones permitirán el acceso a los módulos de acuerdo con los roles establecidos para cada tipo de usuario.	
RF-06	Las aplicaciones permitirán desplegar subcategorías de las categorías principales, mencionadas en el apartado de Perspectiva del producto.	
RF-07	Las aplicaciones mostraran al usuario una lista de categorías en las cuales podrá realizar filtros para una representación específica en el mapa.	
RF-08	Las aplicaciones permitirán hacer o deshacer el zoom al momento de la navegación en el Mapa Parlante.	
RF-09	Las aplicaciones contarán con la representación de un mapa base en el cual se va a mostrar las siguientes categorías: (año, edad (1-100), sexo, riesgos biológicos, riesgos ambientales, riesgos socioeconómicos, grupos prioritarios, grupos vulnerables.)	
RF-10	Las aplicaciones mostraran la información respectiva acorde al rol de usuario (administrador, EAIS) con el que ingrese al sistema.	
RF-11	Las aplicaciones son capaces de marcar rutas desde un origen A, hacia un destino B (EAIS - paciente).	
RF-12	Las aplicaciones permitirán generar reportes según las variables filtradas disponibles en la sección de categorías e indicadores.	
RF-13	Las aplicaciones contarán con un formulario en el cual se realizará la gestión de Usuario EAIS y sus integrantes. (Crear-editar-actualizar).	
RF-14	Las aplicaciones permitirán realizar la búsqueda de información de un paciente mediante su cedula de identidad.	

Requerimientos no funcionales

La Tabla 4 detalla los criterios y especificaciones de los requerimientos no funcionales esenciales para el desarrollo de la aplicación.

Tabla 4. Requerimientos no funcionales

Responsable:		Noralma Lucio
Código	Descripción del Requerimiento	
RNF-01	Las aplicaciones controlarán el acceso y se lo permitirá únicamente a usuarios registrados que ingresen con su respectivo usuario y contraseña.	
RNF-02	Se validará que el campo usuario sea único.	
RNF-03	La contraseña para acceder al sistema será encriptada.	
RNF-04	Las aplicaciones almacenarán la información en un servidor con el gestor de base de datos PostgreSQL.	
RNF-05	Las aplicaciones guardarán la información relacionada con el Mapa Parlante para la generación de reportes.	
RNF-06	Las aplicaciones estarán disponibles las veinticuatro horas del día, los siete días de la semana, pero el horario de atención dependerá de cada médico o del Centro de Salud.	
RNF-07	La aplicación web estará disponible en los navegadores Chrome, Edge, Opera, Safari, y la aplicación móvil para Android e IOS.	
RNF-08	Las aplicaciones no requieren adquirir alguna licencia de software	
RNF-09	Se recuerda que un producto de software no se destruye, pero se desgasta, por lo tanto, algunos apartados deberán requerir mantenimiento al cabo del algún tiempo.	

Product Backlog

El Product Backlog se compone de tareas esenciales para el desarrollo del proyecto, identificadas como Historias de Usuario (HU) e Historias Técnicas (HT). La Tabla 5 presenta estas tareas, clasificando el esfuerzo requerido en una escala del 1 al 5, donde 1 representa menor esfuerzo y 5 indica un esfuerzo más significativo.

Tabla 5. Product Backlog

Ítem	Tarea	Esfuerzo
HT-01	Planteamiento del problema	4
HT-02	Análisis de requerimientos funcionales y no funcionales	5
HT-03	Elaborar las historias de usuario	4
HT-04	Definir la arquitectura del sistema	5
HT-05	Diseñar la interfaz de usuario	5
HT-06	Diseñar de la base de datos	5
HT-07	Instalación y configuración de herramientas de desarrollo	4
HT-08	Levantar la base de datos	5
HT-09	Elaborar manual técnico	4
HT-10	Elaborar manual de usuario	4

HT-11	Realizar pruebas de funcionamiento	5
HT-12	Evaluar la aplicación web utilizando el modelo de calidad ISO/IEC 25010	5
HT-13	Despliegue de aplicaciones	5
HU-01	Como administrador necesito acceder al sistema	5
HU-02	Como administrador necesito registrar usuarios	5
HU-03	Como administrador necesito agregar brigadas EAIS	4
HU-04	Como administrador necesito gestionar brigada EAIS	4
HU-05	Como administrador necesito visualizar información de pacientes	5
HU-06	Como administrador necesito filtrar la información de los pacientes	5
HU-07	Como administrador necesito generar rutas	5
HU-08	Como administrador necesito generar reportes	5
HU-09	Como EAIS necesito acceder al sistema	5
HU-10	Como EAIS necesito visualizar información de pacientes	5
HU-11	Como EAIS necesito filtrar información de pacientes	5
HU-12	Como EAIS necesito generar reportes	5
HU-13	Como EAIS necesito generar rutas	5

Diagramas de Casos de Uso

Los diagramas de casos de uso presentan las funcionalidades y usos que los usuarios tendrán en la aplicación, especificados para los tipos de usuario: Administrador y EAIS. En la figura 9, se detalla el caso de uso del Administrador, destacando las acciones que puede llevar a cabo.

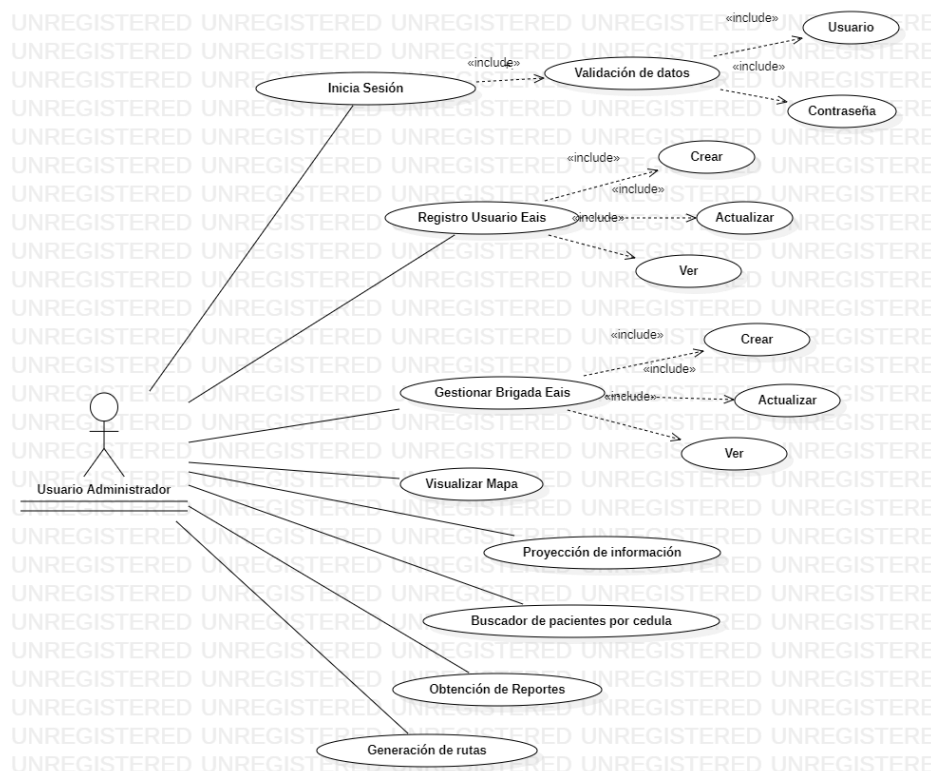


Figura 9. Diagrama de caso de uso para el administrador

En la figura 10, se detalla el caso de uso del EAIS, destacando las acciones que puede llevar a cabo.

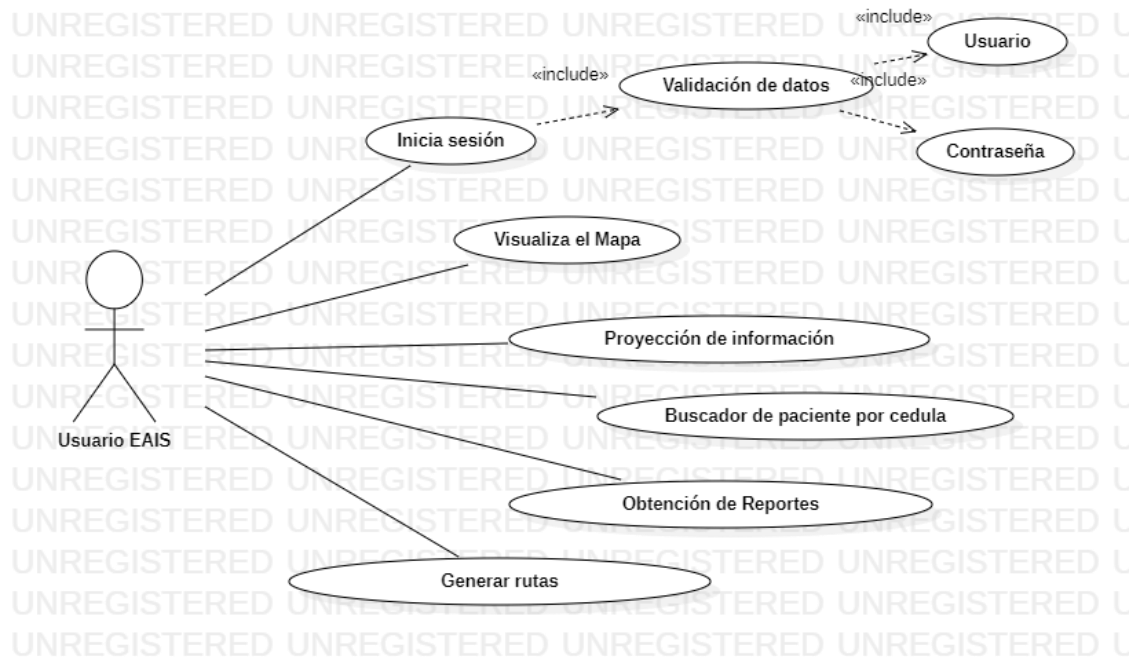


Figura 10. Diagrama de caso de uso para usuario EAIS

Sprint Backlog

El Sprint Backlog representa la planificación detallada de las tareas esenciales para la entrega del producto. La Tabla 6 presenta la programación específica de las tareas que el equipo abordará para cumplir con los objetivos dentro del tiempo establecido.

Tabla 6. Sprint Backlog

Nro.	Actividades	Semanas															
		Mes 1				Mes 2				Mes 3				Mes 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Sprint 1																	
HT-01	Planteamiento del problema	x															
HT-02	Análisis de requerimientos funcionales y no funcionales	x															
HT-03	Elaborar las historias de usuario		x														
HT-04	Definir la arquitectura del sistema		x														
Sprint 2																	
HT-05	Diseñar la interfaz de usuario			x													
HT-06	Diseñar de la base de datos			x													
HT-07	Instalación y configuración de herramientas de desarrollo		x	x													
HT-08	Levantar la base de datos				x												
Sprint 3																	
HU-01	Como administrador necesito acceder al sistema					x											
HU-02	Como administrador necesito registrar usuarios						x										
HU-03	Como administrador necesito agregar brigadas EAIS						x	x									
HU-04	Como administrador necesito gestionar brigada EAIS							x	x								
HU-05	Como administrador necesito visualizar información de pacientes								x								
HU-06	Como administrador necesito filtrar la información de los pacientes												x				
HU-07	Como administrador necesito generar rutas											x					
HU-08	Como administrador necesito generar reportes													x			
Sprint 4																	
HU-09	Como EAIS necesito acceder al sistema													x			

HU-10	Como EAIS necesito visualizar información de pacientes			x	
HU-11	Como EAIS necesito filtrar información de pacientes				x
HU-12	Como EAIS necesito generar rutas			x	x
HU-13	Como EAIS necesito generar reportes				x
Sprint 5					
HT-09	Elaborar manual técnico				x
HT-10	Elaborar manual de usuario				x
Sprint 6					
HT-11	Realizar pruebas de funcionamiento				x
HT-12	Evaluar la aplicación web utilizando el modelo de calidad ISO/IEC 25010				x
HT-13	Despliegue de aplicaciones				x

3.5.3. FASE III: Diseño

Arquitectura del sistema

En la arquitectura cliente/servidor con microservicios, visualizada en la figura 11, se destaca la colaboración dinámica entre el cliente, encargado de solicitar recursos, y el servidor, cuya función principal consiste en suministrar dichos recursos. Este enfoque incorpora el modularidad y la independencia de los microservicios, lo que posibilita una mayor flexibilidad y escalabilidad en el sistema. La figura 11 proporciona una representación gráfica que ilustra la interacción armoniosa entre estos componentes esenciales.

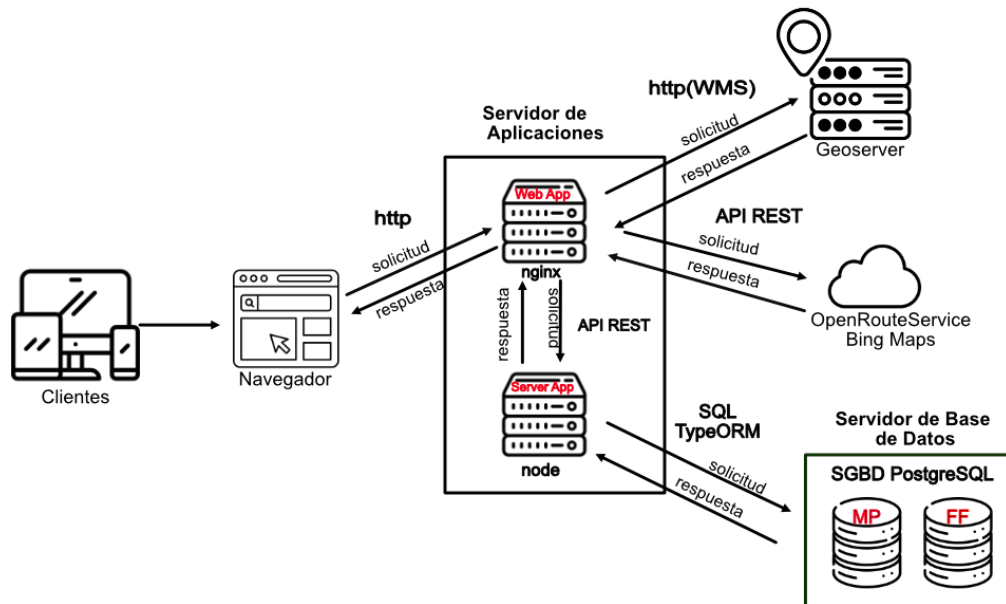


Figura 11. Arquitectura Cliente/Servidor con microservicios.

La arquitectura cliente/servidor con microservicios de Mapa Parlante se compone de diversos elementos que colaboran para brindar una experiencia integral. A continuación, se presenta una descripción detallada de cada componente:

- **Clientes:**

Son los usuarios finales que interactúan y acceden a la aplicación desde diversos dispositivos, ya sean web o móviles. Realizan peticiones y solicitan recursos al servidor para interactuar con las funciones del Mapa Parlante.

- **Navegador:**

Actúa como intermediario al alojar la aplicación mediante el enlace proporcionado por el servidor. Facilita las interacciones del usuario, permitiendo realizar peticiones al servidor de aplicaciones.

- **Servidor de Aplicaciones:**

Aloja el servidor Nginx que contiene la Web App desarrollado con Angular, presentando la interfaz del aplicativo, y el servidor de la aplicación Node.js. La interacción entre estos dos servicios se realiza mediante peticiones API REST.

- **Servidor Geoserver:**

Almacena las capas de mapas utilizadas en la aplicación, procesando peticiones HTTP conforme al estándar WMS para obtener y visualizar los mapas en la interfaz de Mapa Parlante.

- **API de Rutas y Mapas:**

Estas API permiten la comunicación eficiente con el servidor web, siguiendo la arquitectura REST. Su función principal es integrar y aprovechar los recursos de Bing Maps, además de facilitar el uso de las APIs de OpenRouteService para planificar y trazar rutas en la aplicación Mapa Parlante. Este componente es esencial para la visualización de mapas y la generación de rutas optimizadas en la aplicación.

- **Servidor de Base de Datos:**

Utiliza PostgreSQL para gestionar dos bases de datos esenciales: Mapa Parlante y Ficha Familiar. Se conecta al servidor Node App mediante solicitudes TypeORM y SQL, respectivamente.

Diseño de interfaz de Usuario

La concepción de la interfaz de usuario en este proyecto se realizó en concordancia con los requisitos establecidos por el Product Owner. Se articuló la necesidad de una interfaz que no solo fuese intuitiva sino también fácil de navegar, permitiendo a los usuarios identificar de manera rápida los elementos clave. La decisión se inclinó hacia un diseño que empleara colores agradables, buscando garantizar una experiencia visual cómoda durante periodos prolongados de uso.

En la figura 12, se presenta el prototipo de baja fidelidad o maqueta del diseño para la aplicación, en su versión con acceso web y móvil, representado en escala de grises. Este prototipo brinda una visión inicial de la disposición y estructura general de la interfaz, sirviendo como base para futuras iteraciones y refinamientos en el diseño final.

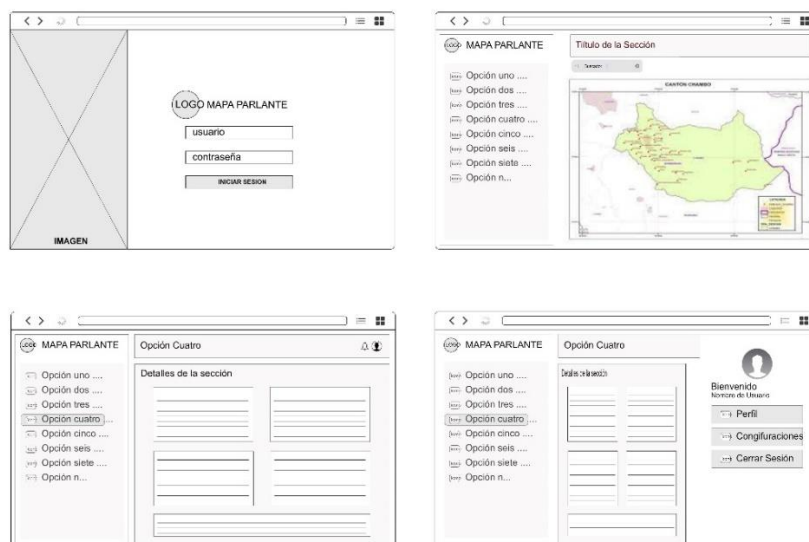


Figura 12. Wireframes de la aplicación

Diseño de la Base de Datos

En la figura 13, se muestra el modelo de la base de datos relacional que se utilizó para gestionar de forma eficiente los datos.

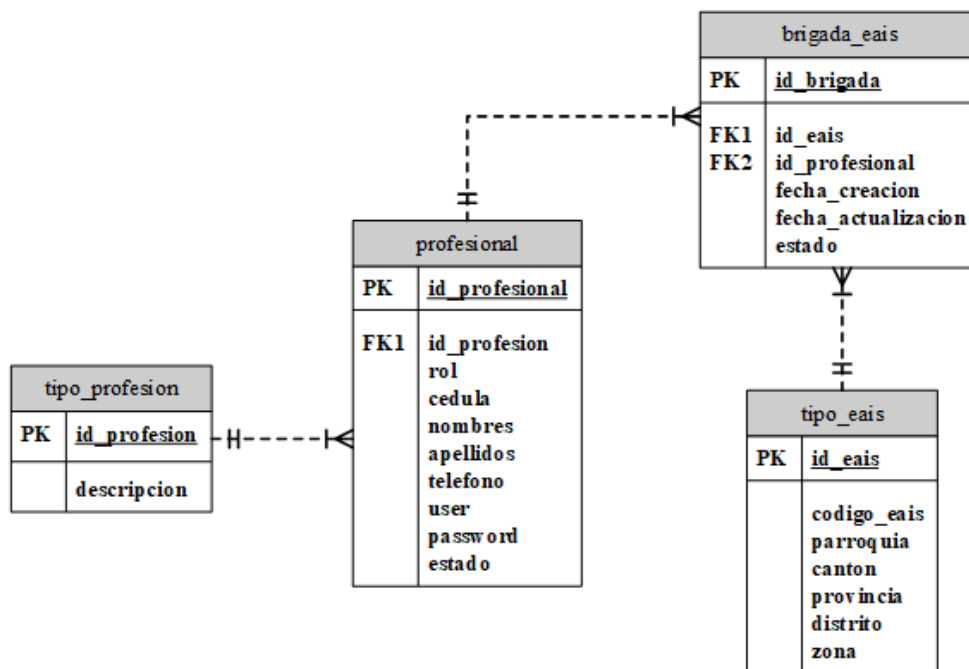


Figura 13. Modelo relacional de la base de datos

Herramientas de Desarrollo

En el desarrollo de la aplicación Mapa Parlante, se han seleccionado cuidadosamente una serie de herramientas y tecnologías que permiten un desarrollo eficiente y robusto. Cada elección de herramientas se hizo con el objetivo de garantizar la eficiencia, la escalabilidad y la facilidad de mantenimiento en cada componente de la aplicación, permitiendo una experiencia completa y fluida para los usuarios finales.

En la Tabla 7, se detallan las herramientas utilizadas durante el proceso de desarrollo de la aplicación Mapa Parlante y la evaluación de usabilidad.

Tabla 7. Herramientas utilizadas para el desarrollo de la aplicación Mapa Parlante

Categoría	Herramienta	Versión
Lenguaje de programación	TypeScript	NA
Framework	Angular	16
	Nest.js	10.3.0
Base de Datos	PostgreSQL	15.2.1
Servidor Web	Nginx	1.22.1
	Node.js	18.18.0
Servidor de Mapas	Geoserver	2.24.1
Infraestructura de Mapas en línea	OpenLayers	9.0.0
API para rutas	Openrouteservice	NA
API para mapas	Bing Maps	NA
IDE	Visual Studio Code	1.87.1
Norma para evaluar la usabilidad	ISO/IEC 25010	NA

PostgreSQL se utiliza para la gestión de la base de datos, en esta se implementó una base de datos dedicada al manejo completo del Mapa Parlante, para simplificar el trabajo con PostgreSQL en la aplicación del servidor desarrollada con Nest, se integró TypeORM, ofreciendo un mapeo de objetos más sencillo y facilitando las operaciones de la base de datos. Además, se integra la consulta a la base de datos externa denominada "Ficha Familiar", utilizando SQL únicamente para operaciones de consumo. Esta última base de datos pertenece a otro proyecto implementado en el Centro de Salud Chambo, siendo utilizada de manera independiente para obtener y buscar información, sin realizar modificaciones en sus datos originales.

Node.js compone la estructura del servidor, este se configuró utilizando el framework Nest.js, el cual permitió la creación de una aplicación escalable y modular. Esta aplicación opera como una API REST, proporcionando una interfaz eficiente para la interacción entre el cliente y el servidor de aplicaciones. Para construir la API, se adoptó la metodología modular de Nest.js, dividiendo la aplicación en módulos especializados para fomentar la cohesión y simplificar el mantenimiento. Se garantizó un funcionamiento robusto y confiable al utilizar tecnologías estándar, como el manejo de solicitudes HTTP, la validación de datos y el control de excepciones.

Angular se utiliza para el desarrollo de la aplicación con acceso web y móvil que opera en el entorno de desarrollo Visual Studio Code. Se optó por TypeScript debido a su capacidad para facilitar el mantenimiento del código y detectar errores en tiempo de compilación. En términos de diseño y estilos, se implementó el framework Tailwind CSS, ofreciendo un enfoque ágil y eficiente para el desarrollo de interfaces de usuario modernas. La estratégica utilización de módulos de Angular, como Core Module, Common Module, Forms Module, HTTP Module y Router Module, desempeña un papel crucial al simplificar y optimizar el rendimiento. Esta modularidad no solo contribuye a una arquitectura clara y organizada, sino que también mejora significativamente la eficiencia general del sistema.

Geoserver constituye la infraestructura del servidor de mapas en la aplicación, para la creación de servicios de mapas mediante el estándar abierto Web Map Service (WMS), que se empleó para consumir los mapas dentro de la aplicación de Mapa Parlante. Esta integración facilita una representación gráfica dinámica y precisa de la información geográfica proporcionada. La conexión entre GeoServer y Angular se establece a través de OpenLayers, pues la combinación de estas tecnologías garantiza una interfaz de usuario rica en funcionalidades y proporciona una experiencia de navegación geoespacial fluida.

API de Bing Maps amplía las opciones de visualización con mapas de alta calidad y diversas funciones geoespaciales. Esta integración proporciona una fuente adicional de información geográfica, enriqueciendo la diversidad de datos presentados en la aplicación.

Openrouteservice se utiliza en el ámbito de la planificación y navegación de rutas, este componente se integra de manera eficiente y precisa con la aplicación, permitiendo funcionalidades avanzadas para la planificación de rutas en Mapa Parlante.

3.5.4. FASE IV: Desarrollo

Desarrollo de la aplicación de lado del servidor con Nest.js

El desarrollo de la aplicación del lado del servidor se llevó a cabo empleando Nest.js, un framework basado en Node.js que sigue una arquitectura similar a la de Angular. En la figura 14, se presenta la estructura organizativa de carpetas y el módulo principal del proyecto en Nest.js.

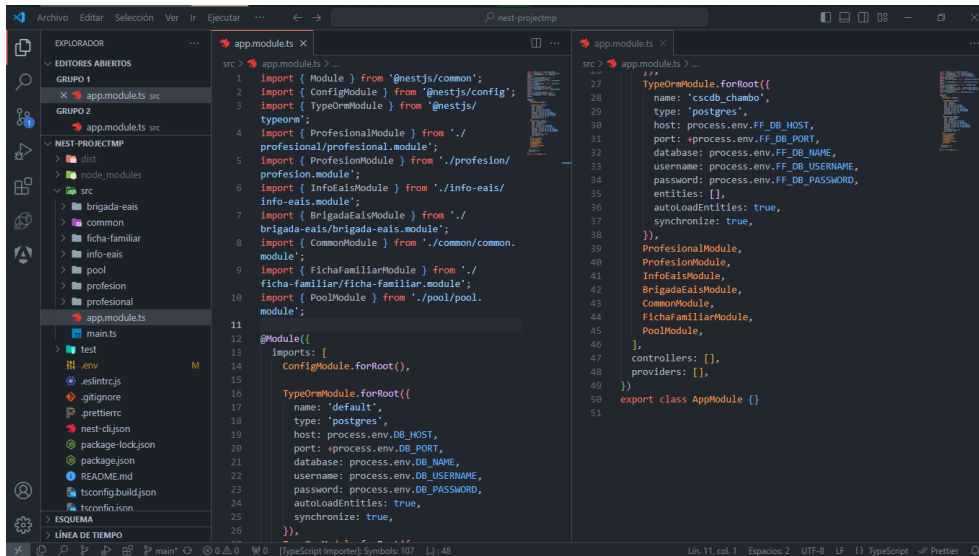


Figura 14. Estructura organizativa de carpetas en el módulo principal de Nest

A continuación, se presenta en la figura 15, la integración y configuración de PostgreSQL con la base de datos seleccionada para el proyecto.

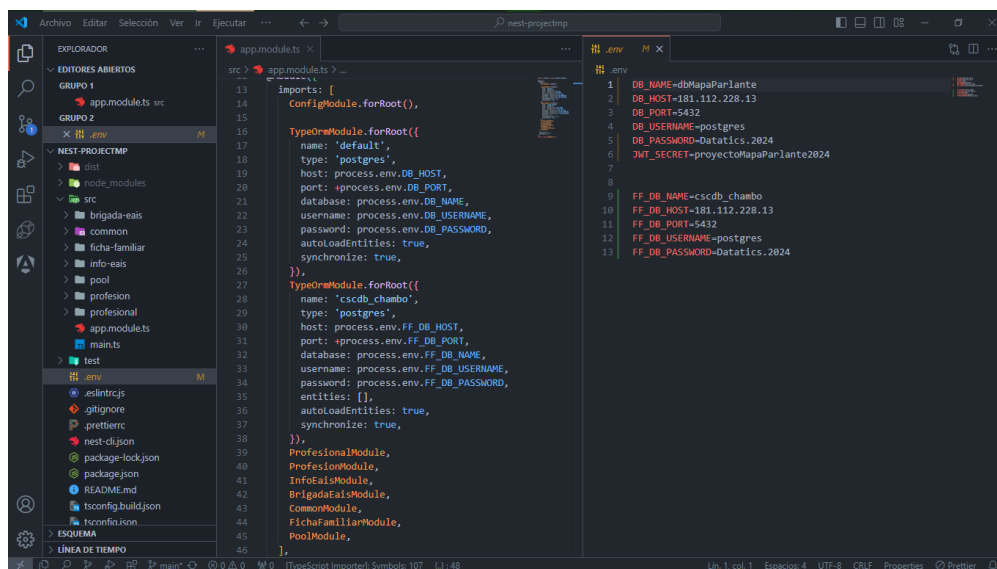


Figura 15. Configuración de PostgreSQL en el proyecto de Nest

En la figura 16, se visualiza la aplicación de la biblioteca TypeORM dentro de Nest.js, en el servicio denominado profesional, actuando como una capa de acceso a datos. Esta implementación posibilita la interacción con la base de datos mediante el uso de clases y objetos. Este enfoque resulta en una integración más intuitiva y orientada a objetos.


```

src > profesional > profesional.service.ts > ProfessionalService
export class ProfessionalService {
  async create(createProfessionalDto: CreateProfessionalDto) {
    try {
      const { password, ...usuarioData } = createProfessionalDto;
      const profesional = this.profesionalRepository.create({
        ...usuarioData,
        password: bcrypt.hashSync(password, 10),
      });
      await this.profesionalRepository.save(profesional);
      return profesional;
    } catch (error) {
      this.errorHandleDBException.errorHandleDBException(error);
    }
  }

  async findAll() {
    return this.profesionalRepository.find({
      relations: {
        profesion: true,
      },
      order: {
        apellidos: 'ASC',
      },
    });
  }

  async findOne(terms: string) {
    let user: Profesional;

    if (isUUID(terms)) {
      user = await this.profesionalRepository.findOne({
        where: {
          id_profesional: terms,
        },
      });
    }
  }
}

src > profesional > profesional.controllers.ts > ProfessionalController
export class ProfessionalController {
  @Post('login')
  authProfessional(@Body() authProfessionalDto: AuthProfessionalDto)
  return this.profesionalService.login(authProfessionalDto);
}

@Get('status-verify')
  @Auth()
  checkAuthStatus(@GetUser() user: Profesional) {
    return this.profesionalService.checkStatus(user);
  }

@Get('list')
  findAll() {
    return this.profesionalService.findAll();
  }

@Get('search/:term')
  findOne(@Param('term') term: string) {
    return this.profesionalService.findOne(term);
  }

@Get('searchEaisEnable/:term')
  findOneEnable(@Param('term', ParseUUIDPipe) term: string) {
    return this.profesionalService.findOneEnable(term);
  }

//METODO CONTROL PROFESIONALES EAIS
@Get('listProfesionalActive/')
  findActiveProfesional() {
    return this.profesionalService.findProfesionalActive();
  }

@Get('user-verify/:term')
}

```

Figura 16. Uso de TypeORM para conexión con la base de datos

En la figura 17, se observa la implementación de SQL puro, en el servicio de ficha-familiar, para la recuperación y búsqueda en la base de datos denominada cscdbfichafamiliar.

```

src > ficha-familiar > ficha-familiar.service.ts > FichaFamiliarService
export class FichaFamiliarService {
  async getLocation(
    year: string,
    age: string,
    gender: string,
    rbiological: string,
    renvironmental: string,
    rsocioeconomic: string,
    gpriority: string,
    gvulnerable: string,
  ) {
    try {
      let consult;

      /* const array = new Array(); */
      const array = [];
      let variable_number = 0;

      const sql_fichas = `select distinct(b.cscdbfamiliarid),
      longitudi, latitudi, 'n/a' as riesgo,
      nom_fam || ' ' || ape_fam as name, años as age, genero,
      cedula_fam as cedula , celular ,
      (case when genero = 'MASCULINO' then 'blue' else 'red'
      end ) as color ,
      (case when genero = 'MASCULINO' then 'man.png' else 'woman.
      png' end ) as icon
      from cscdbvivienda a, cscdbfamilia b
      where a.cscdbfamiliarid= b.cscdbfamiliarid and estado_fam
      = true
      `;

      const sql_base = ` select distinct(b.cscdbfamiliarid),b.
      id_jefta_hogar, 'n/a' as riesgo,
      nom_fam || ' ' || ape_fam as name, años as age, genero,
      cedula_fam as cedula , celular ,
      (case when genero = 'MASCULINO' then 'blue' else 'red'
      end ) as color ,
      (case when genero = 'MASCULINO' then 'man.png' else 'woman.
      png' end ) as icon
      from cscdbvivienda a, cscdbfamilia b
      where a.cscdbfamiliarid= b.cscdbfamiliarid and estado_fam
      = true
      `;

      consult = await this.dataSource.query(sql_fichas);

      return consult;
    } catch (error) {
      this.errorHandleDBException.errorHandleDBException(error);
    }
  }
}

src > ficha-familiar > ficha-familiar.controllers.ts > FichaFamiliarController
export class FichaFamiliarController {
  @Get('location')
  getLocation(
    @Query('year') year: string,
    @Query('age') age: string,
    @Query('gender') gender: string,
    @Query('rbiological') rbiological: string,
    @Query('renvironmental') renvironmental: string,
    @Query('rsocioeconomic') rsocioeconomic: string,
    @Query('gpriority') gpriority: string,
    @Query('gvulnerable') gvulnerable: string,
  ) {
    return this.fichaFamiliarService.getLocation(
      year,
      age,
      gender,
      rbiological,
      renvironmental,
      rsocioeconomic,
      gpriority,
      gvulnerable,
    );
  }

  @Get('categories')
  getCategories() {
    return this.fichaFamiliarService.getCategories();
  }

  @Get('searchId')
  getSearchId(@Query('cedula') cedula: string) {
    return this.fichaFamiliarService.getSearchId(cedula);
  }
}

```

Figura 17. Implementación de SQL para Interactuar con la BD Ficha Familiar

Desarrollo de la aplicación con acceso web y móvil utilizando Angular e integración con otros servicios

La construcción de la aplicación web se llevó a cabo mediante el uso del framework Angular, complementado por Tailwind CSS para una mayor eficiencia en el proceso. La figura 18 muestra el módulo raíz del proyecto Angular, junto con una vista previa de la aplicación.

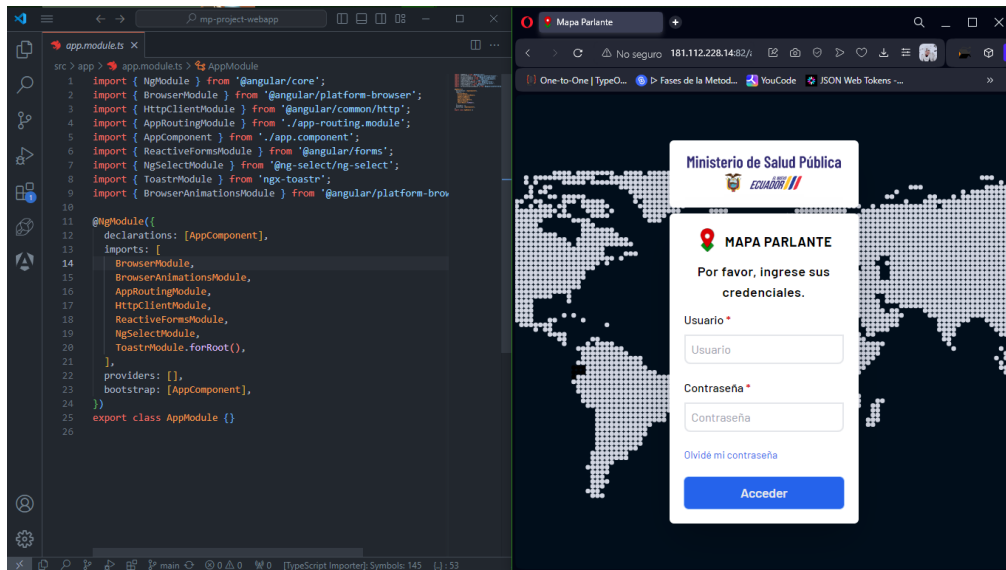
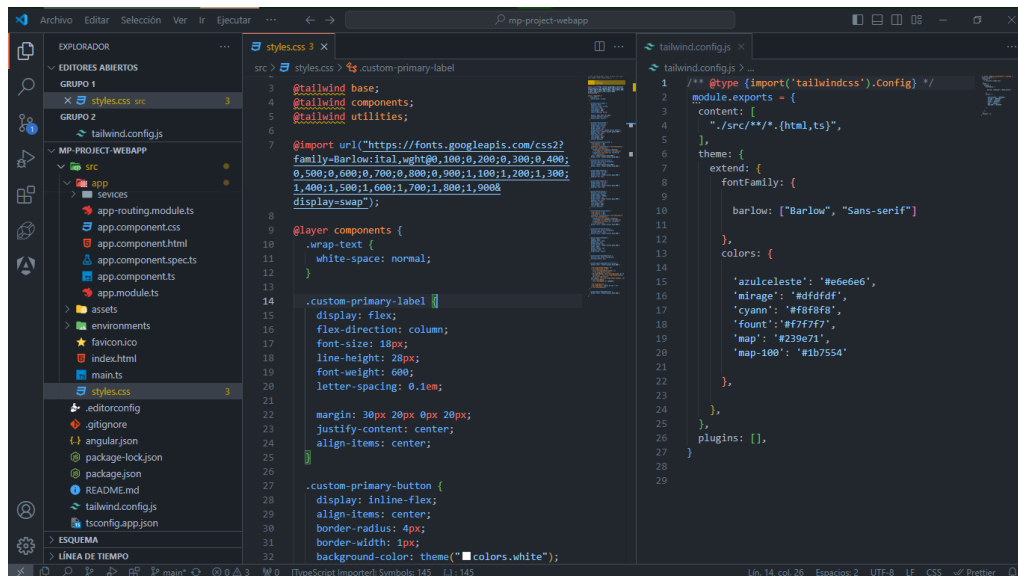


Figura 18. Vista previa del desarrollo de la aplicación Mapa Parlante

La figura 19 presenta el archivo de importación de Tailwind CSS y la declaración de los estilos globales, que permitieron diseñar el aplicativo para dispositivos que garantizan el acceso web y móvil.



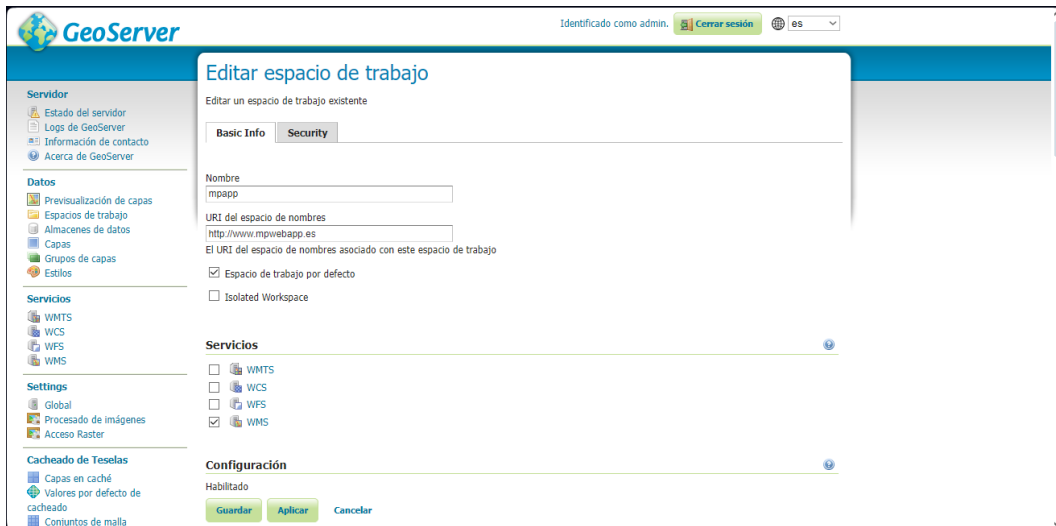


Figura 20. Espacio de trabajo en Geoserver

La figura 21 presenta los almacenes de datos denominados Cuadro_EAIS, EAIS, Geo_Loc, utilizados en la aplicación Mapa Parlante que sirven para configurar los tipos de datos que se van a mostrar a través del servicio WMS, en este caso serán archivos con formato shapefile(.shp).



Figura 21. Almacenes de datos en Geoserver

En la figura 22 se puede observar las capas denominadas CUADRO_EAIS, EAIS, GEO_LOC, que fueron creadas para ser representada en el mapa, además se muestra la configuración de una de las capas denominada CUADRO_EAIS.



Figura 22. Capas publicadas en Geoserver

La figura 23 muestra la configuración de los estilos denominados AREAN_EAIS, EAIS, Geo_Loc, aplicados para las capas añadidas que permitieron tener una mejor visualización de estos mapas, además se muestra la visualización de la capa CUADRO_EAIS aplicado el estilo correspondiente.



Figura 23. Configuración de estilos SLD en Geoserver

En la figura 24, se evidencia la conexión de Geoserver con el aplicativo frontend de Angular a través de OpenLayers en el componente denominado EAIS-information.

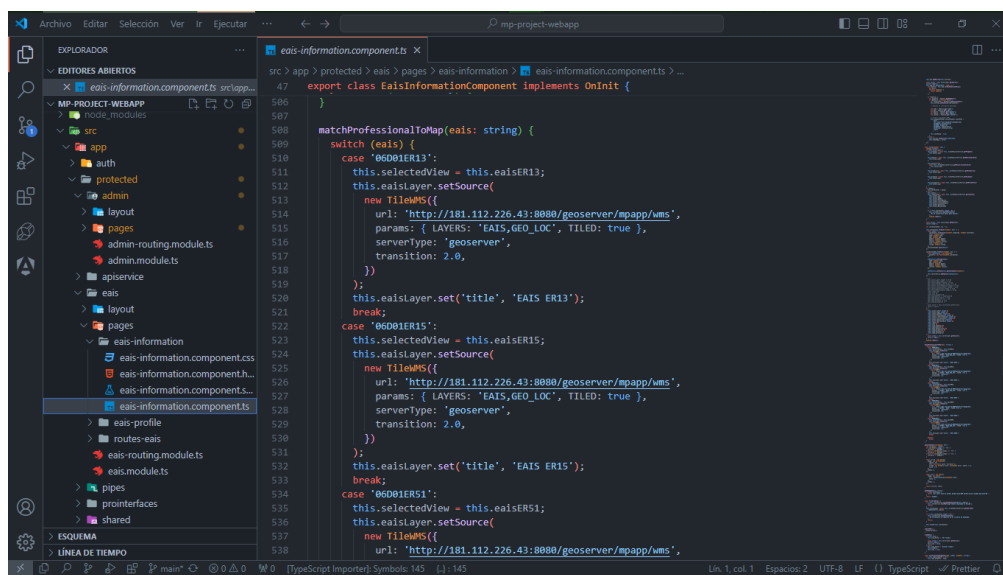


Figura 24. Conexión de Geoserver a Angular mediante OpenLayers

En la figura 25, se muestra el consumo del API de Bing Maps para la visualización del mapa en el aplicativo de Angular en el componente denominado EAIS-information.

```

export class EaisInformationComponent implements OnInit {
  overlay!: Overlay;
  showPopup: boolean = false;
  /*----- */

  relieveLayer = new TileLayer({
    title: 'Relieve',
    type: 'base',
    combine: true,
    visible: true,
    source: new OSM(),
  }) as BaseLayerOptions;

  satelliteLayer = new TileLayer({
    title: 'satélite',
    type: 'base',
    combine: true,
    visible: false,
    source: new BingMaps({
      key: 'AtairFX86wFzE8hSG9sRKO6d8vTf8TfuP8RH_0ozsvG1YwruakV6_B4LVtXjBe',
      imagerySet: 'AerialWithLabelsOnDemand',
    })),
    opacity: 0.85,
  }) as BaseLayerOptions;

  eaisLayer = new TileLayer({
    visible: true,
  }) as BaseLayerOptions;

  baseMaps = new LayerGroup({
    title: 'Capas',
    layers: [this.satelliteLayer, this.relieveLayer],
  }) as GroupLayerOptions;
}

```

Figura 25. Consumo del API de Bing Maps

La figura 26 presenta el uso del API de OpenRouteService para la planificación de rutas y cómo se integra en la aplicación Angular en el componente denominado routes-EAIS.

```

export class RoutesEaisComponent implements OnInit {
  //Open Route Service
  async getRoute(start: number[], end: number[]) {
    if (start[0] != 0 && end[0] != 0) {
      let url = `https://api.openrouteservice.org/v2/directions/driving-car?api_key=${this.onrApiKey}&start=${start[0]},${start[1]}&end=${end[0]},${end[1]}`;
      this.http.get(url).subscribe({
        next: (response: any) => {
          // Maneja la respuesta aquí

          const vectorRoute = new VectorSource();
          // Borra la ruta anterior
          this.routeLayer.getSource().clear();

          // Extrae la geometría de la ruta de la respuesta
          let route = response.features[0].geometry;
          let transformedCoordinates = route.coordinates.map((coord: any) =>
            transform(coord, 'EPSG:4326', 'EPSG:3857')
          );

          // Crea un objeto de la ruta para OpenLayers
          let routeFeature = new Feature({
            type: 'route',
            geometry: new LineString(transformedCoordinates),
          });

          // Añade la ruta al mapa
          vectorRoute.addFeature(routeFeature);

          this.routeLayer = new VectorLayer({
            vectorSource: vectorRoute,
          });
        }
      });
    }
  }
}

```

Figura 26. Integración de Openrouteservice en Angular

3.5.5. FASE V: Pruebas y Lanzamiento

Después de finalizar el desarrollo de la aplicación, se sometió a pruebas exhaustivas, comenzando con pruebas a nivel de código y posteriormente realizando pruebas de integración para verificar su correcto funcionamiento de acuerdo con los requisitos iniciales. Se llevó a cabo una evaluación minuciosa de la usabilidad de la aplicación Mapa Parlante, en conformidad con los estándares establecidos por la norma ISO/IEC 25010. Se diseñó una encuesta específica, la cual fue aplicada a profesionales de la salud en Chambo, con el propósito de recopilar comentarios valiosos y retroalimentación sobre la experiencia de uso de la aplicación.

Esta encuesta abordó aspectos clave de la usabilidad, como la facilidad de aprendizaje, operabilidad, manejo de errores y aspectos estéticos de la interfaz de usuario. Los profesionales de la salud desempeñaron un papel crucial al proporcionar percepciones valiosas, identificando áreas de mejora y validando la eficacia de las características implementadas. Los resultados de estas pruebas de usabilidad, junto con las correcciones y ajustes necesarios, se incorporaron en la fase final de desarrollo antes del lanzamiento.

En el proceso de lanzamiento de la aplicación Mapa Parlante, se configuró el servidor de aplicaciones Nginx, así como el servidor de mapas Geoserver implementado en un entorno Windows Server 2016. Estas configuraciones se llevaron a cabo con el objetivo de garantizar un despliegue exitoso y un rendimiento óptimo en un entorno de producción real. Este enfoque integral en pruebas y lanzamiento contribuyó a asegurar que la aplicación fuera robusta y cumpliera con las expectativas de los usuarios finales.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. Resultados

Análisis de herramientas para la representación del Mapa Parlante en el Centro de Salud Chambo.

Como resultado del primer objetivo del proyecto de investigación se presenta en la tabla 8, el análisis de la elección de las herramientas para el desarrollo de la aplicación.

Tabla 8. Análisis de herramientas para el Desarrollo de la Aplicación Mapa Parlante

Criterio	Herramienta Elegida	Otra Herramienta	Justificación
Framework de Desarrollo del Servidor	Nest	Spring Boot	Se selecciono Nest.js sobre Spring Boot por la integración natural con Angular esencial para mantener una arquitectura coherente en toda la aplicación. Nest.js ofrece esta integración de manera más fluida, compartiendo código y estructuras de proyecto directamente con Angular [15]. Además, el equipo de desarrollo ya está familiarizado con TypeScript, esto reduce la curva de aprendizaje y permite una transición más efectiva.
Framework de Desarrollo Web	Angular	React.js	La elección de Angular se basó en la integración natural con TypeScript, asegurando así que la tecnología seleccionada se alinee de manera óptima con las necesidades y preferencias del equipo de desarrollo en el contexto del proyecto Mapa Parlante [9].
Lenguaje de programación	TypeScript	JavaScript	La elección de TypeScript frente a JavaScript se basa en el tipado estático de TypeScript, además este es un superconjunto de JavaScript, que proporciona una mayor robustez al código y permite la detección temprana de errores durante el desarrollo [12]. Además, Angular y Nest.js, las herramientas seleccionadas, son compatibles con TypeScript, lo que facilita la integración y mejora la coherencia en el desarrollo [12].
Gestión de Base de Datos	PostgreSQL	MongoDB	La elección de PostgreSQL sobre MongoDB para la base de datos de Mapa Parlante se basó en la integración con TypeORM que facilitó el mapeo de objetos para la simplificación de las operaciones de la base de datos. La estructura relacional de PostgreSQL se adaptó óptimamente a la

Servidor de Mapas	Geoserver	ArcGIS	<p>complejidad de los datos del proyecto, respaldada por su soporte para transacciones ACID y su robustez en el manejo de datos estructurados [23]. Estas características específicas del proyecto se alinearon de manera coherente con las ventajas ofrecidas por un modelo de datos relacional.</p> <p>La elección de Geoserver sobre ArcGIS se basó en su eficiente integración con Angular mediante OpenLayers [5]. Desde una perspectiva económica, Geoserver resulta más rentable al evitar costosas licencias asociadas con ArcGIS. Además, la comunidad activa y el soporte continuo de Geoserver garantizan estabilidad y eficacia a largo plazo, respaldando así su elección como la opción más adecuada.</p>
Librería de Visualización de Mapas	OpenLayers	Google Maps JavaScript API	<p>La elección de OpenLayers sobre Google Maps JavaScript API para la librería de visualización de mapas se fundamenta en la preferencia por una solución de código abierto y la flexibilidad que ofrece. OpenLayers proporciona control sobre la implementación y personalización de la visualización de mapas, sin depender de servicios externos y evitando posibles costos asociados con licencias [33]. Esta decisión se alinea con la necesidad de integración específica y adaptación precisa a los requisitos del sistema, así como con la consideración de evitar restricciones y tarifas relacionadas con el uso de servicios como Google Maps.</p>
API para Rutas	Openrouteservice	Google Maps JavaScript API	<p>La elección de Openrouteservice frente a Google Maps JavaScript API para la generación de rutas se basa en la preferencia por soluciones de código abierto, la flexibilidad que ofrece, y la ausencia de limitaciones y tarifas asociadas [35]. Esta decisión se alinea con la filosofía de evitar dependencias de servicios propietarios y optimizar la adaptación a las necesidades específicas del proyecto.</p>
Proveedor de Mapas Adicional	Bing Maps	Google Maps	<p>La elección de Bing Maps como proveedor de mapas adicional se debe a su API robusta y bien documentada, que ofrece una amplia variedad de funcionalidades de mapeo [30]. Además, la flexibilidad de licencia y los costos más bajos en comparación con Google Maps hacen que Bing Maps sea una opción económicamente atractiva.</p>

Implementación una aplicación con acceso web y móvil Mapa Parlante en el Centro de Salud Chambo

Con el marco metodológico Scrum como guía, la implementación de la aplicación Mapa Parlante en el Centro de Salud Chambo avanzó de manera ágil y eficiente. A continuación, exploraremos las pantallas principales de la aplicación, destacando cómo logra cumplir con los requisitos y expectativas definidos en la fase de diseño, para más detalles ver anexo 3.

En la figura 27 se muestra las pantallas correspondientes al diseño web de la aplicación.

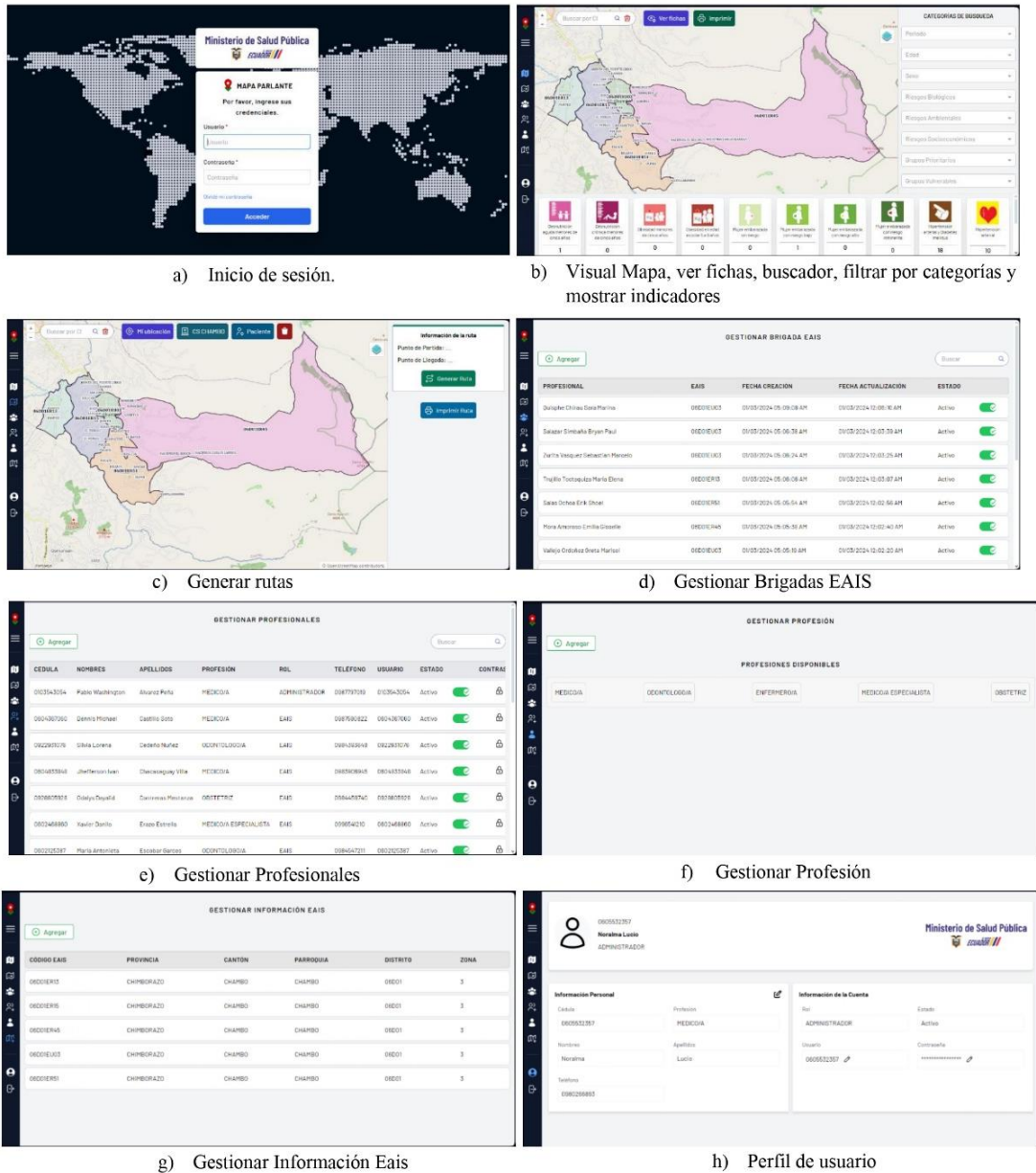


Figura 27. Diseño web de la aplicación

La figura 28 presenta las pantallas correspondientes al diseño móvil de la aplicación.

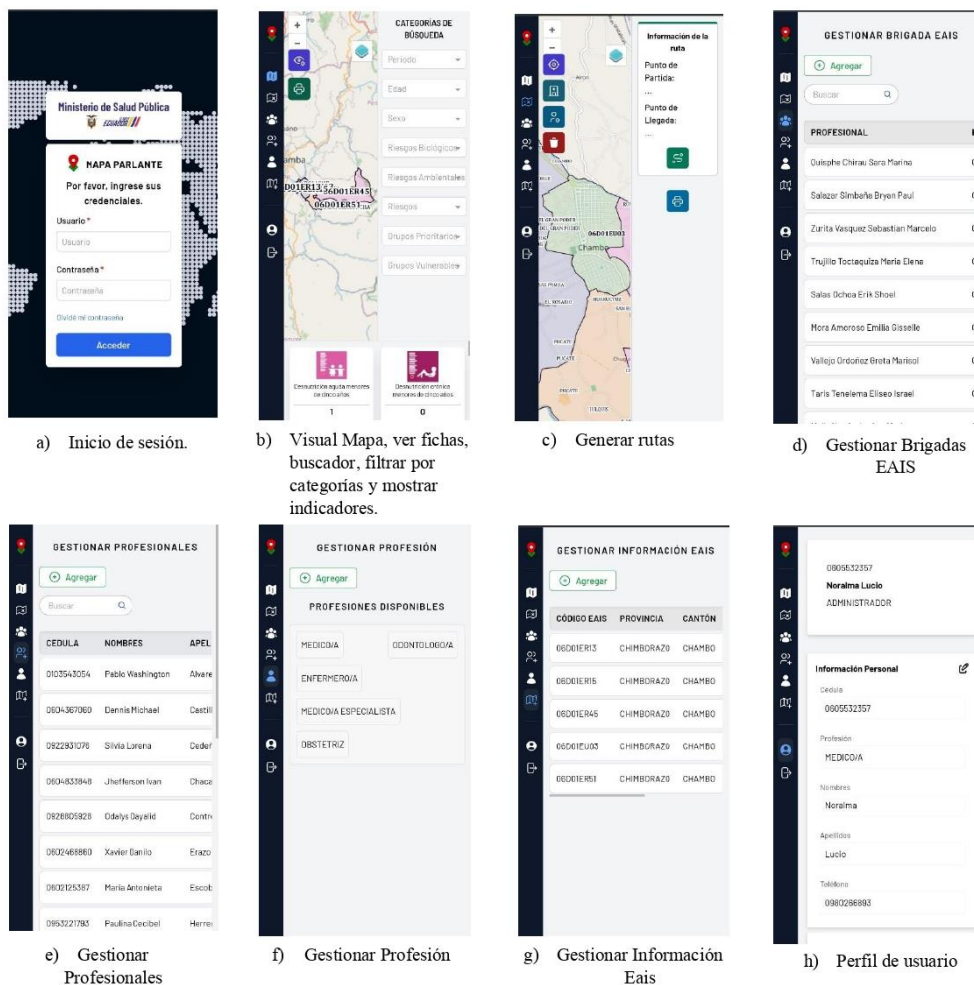


Figura 28. Diseño móvil de la aplicación

Evaluación de la usabilidad de la aplicación con acceso web y móvil Mapa Parlante en el Centro de Salud Chambo aplicando el modelo de calidad ISO/IEC 25010

En aras de garantizar una experiencia de usuario, la evaluación de la usabilidad de la aplicación Mapa Parlante se llevó a cabo aplicando rigurosos estándares del modelo de calidad ISO/IEC 25010. Esta fase crítica del proyecto busca analizar la satisfacción del usuario mediante el criterio de usabilidad, con las métricas: Aprendizabilidad, operabilidad, protección contra errores de usuario y estética de la interfaz de usuario.

Para ello se diseñó un instrumento específico, una encuesta cerrada (ver Anexo 4); para cuantificar la usabilidad de la aplicación. Este método de evaluación se basó en la escala de Likert, que proporciona una medida cuantitativa detallada de la experiencia del usuario, permitiendo una evaluación más precisa y objetiva de diversos aspectos de la usabilidad. A continuación, exploraremos los resultados detallados de esta evaluación, que ofrece una visión integral de cómo la aplicación cumple con los criterios de usabilidad establecidos.

La encuesta se realizó a 13 profesionales del área de la salud del Centro de salud de Chambo tipo B, beneficiarios de la aplicación Mapa Parlante con acceso web y móvil, detallados en la figura 29.

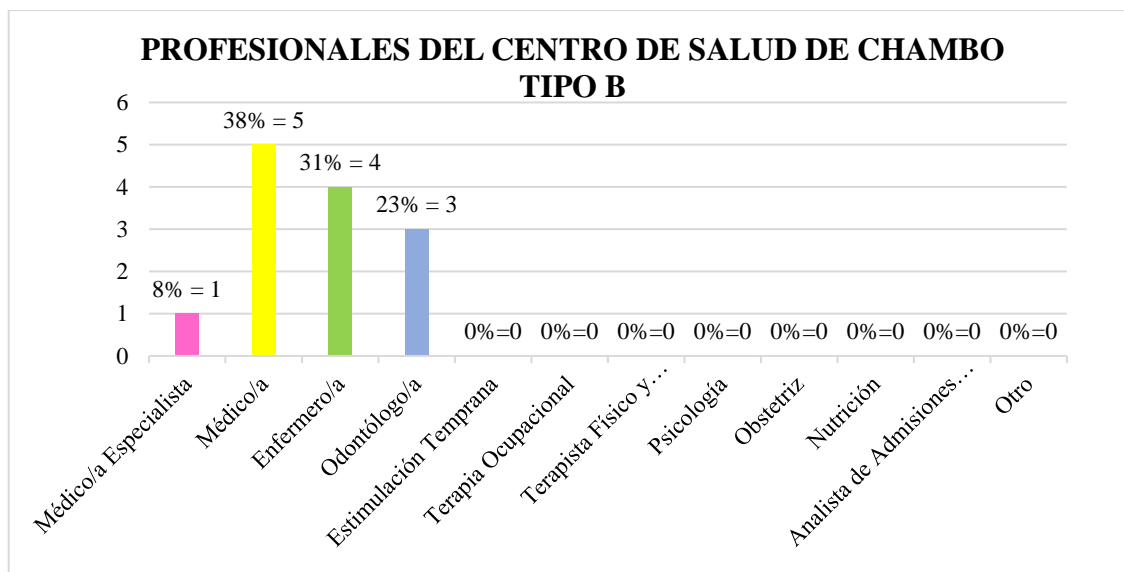


Figura 29. Población de estudio

Una vez realizada la encuesta se puede evidenciar que, de 13 profesionales del área de la salud que laboran en el Centro de salud de Chambo tipo B, el 38% equivalente a 5 profesionales son Médicos/as General, el 31% equivalente a 4 profesionales son Enfermeros/as, el 23% equivalente a 3 profesionales son Odontólogos/as, y el 8% equivalente a 1 profesional es Médico/a Especialista.

SECCIÓN DE EVALUACIÓN DE APRENDIZABILIDAD: Refiere a la capacidad del producto que permite al usuario aprender su aplicación.

Pregunta 1. ¿Qué tan fácil le resultó aprender a utilizar la aplicación desde el principio?

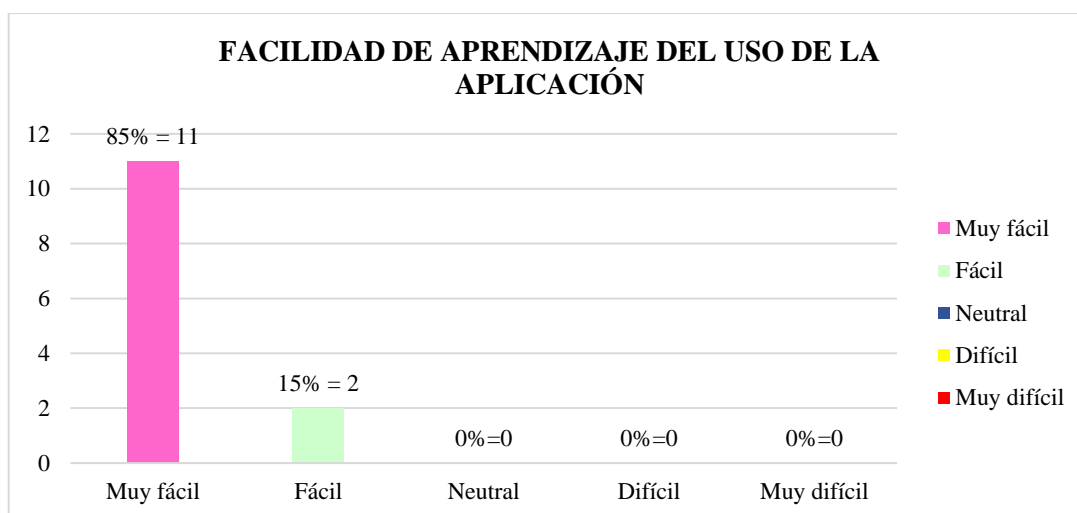


Figura 30. Aprendizabilidad - Facilidad de aprendizaje del uso de la aplicación

La figura 30 presenta el análisis mediante la gráfica de barras que indica que la mayoría de los encuestados (85%, equivalente a 11 de 13 participantes) encontraron que aprender a usar la aplicación Mapa Parlante fue Muy fácil. Adicionalmente, un 15% (2 de 13 participantes) indicó que fue Fácil aprender a utilizarla. No se registraron respuestas en las categorías Neutral, Difícil o Muy difícil.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta una curva de aprendizaje excepcionalmente accesible. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 92,08 sobre un máximo posible de 95. Este resultado indica que la mayoría de los usuarios encontraron la aplicación intuitiva y fácil de aprender desde el principio. La figura 31 ilustra estos hallazgos de manera gráfica.

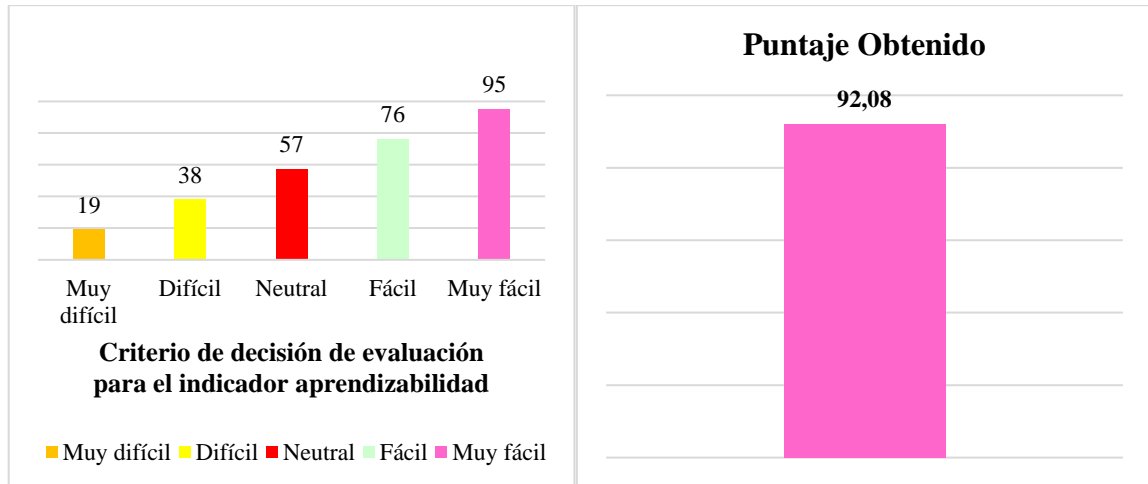


Figura 31. Resultado del indicador aprendizabilidad – Pregunta 1

Pregunta 2. ¿Las funciones de la aplicación son intuitivas y fácil de entender?

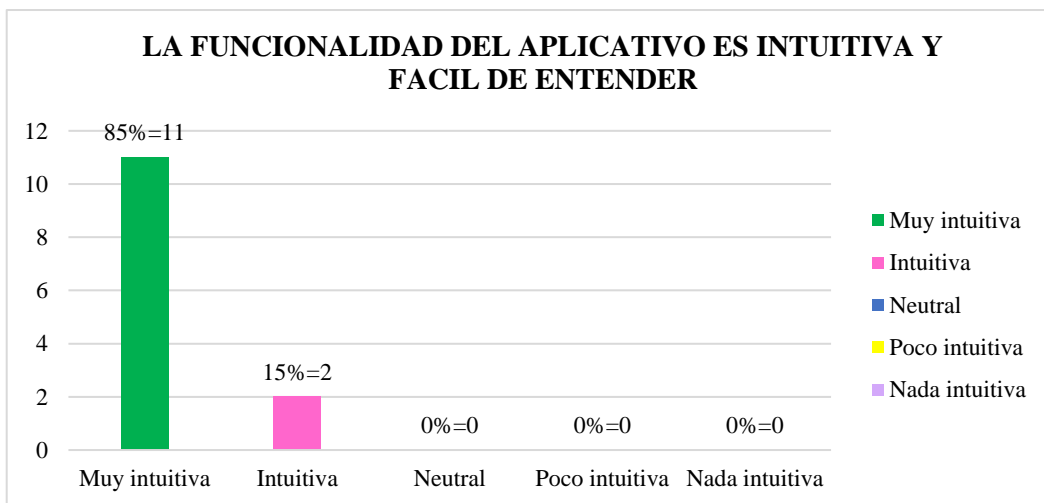


Figura 32. Aprendizabilidad - Intuición del aplicativo

Mediante el gráfico de barras (véase la figura 32), se puede observar que, la mayoría de los encuestados (85%, equivalente a 11 de 13 participantes) considera que las funciones de la aplicación Mapa Parlante son Muy intuitivas y fáciles de entender. Adicionalmente, un 15% (2 de 13 participantes) indicó que las funciones son Intuitivas. No se registraron respuestas en las categorías Neutral, Poco intuitiva o Nada intuitiva.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta funciones intuitivas y fáciles de entender. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 92.08 sobre un máximo posible de 95. La figura 33 ilustra estos hallazgos de manera gráfica.

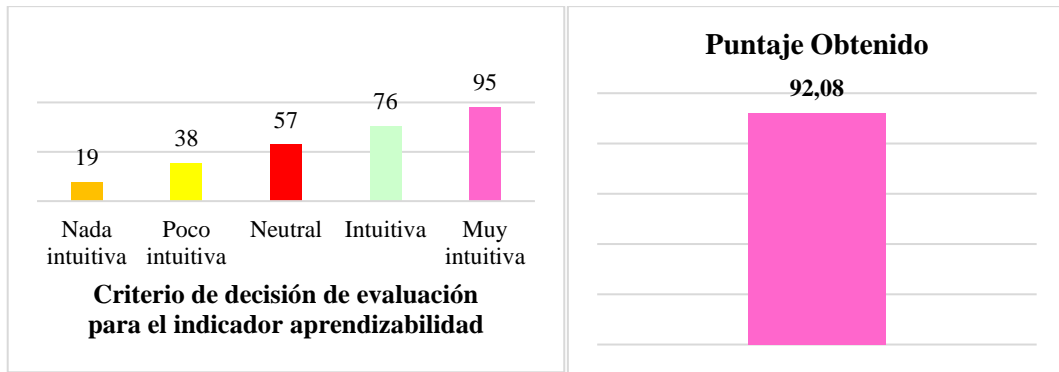


Figura 33. Resultado del indicador aprendizabilidad – Pregunta 2

Pregunta 3. ¿Cómo evalúa la claridad, organización y navegabilidad de la aplicación para acceder a las diversas funciones y acciones que ofrece?

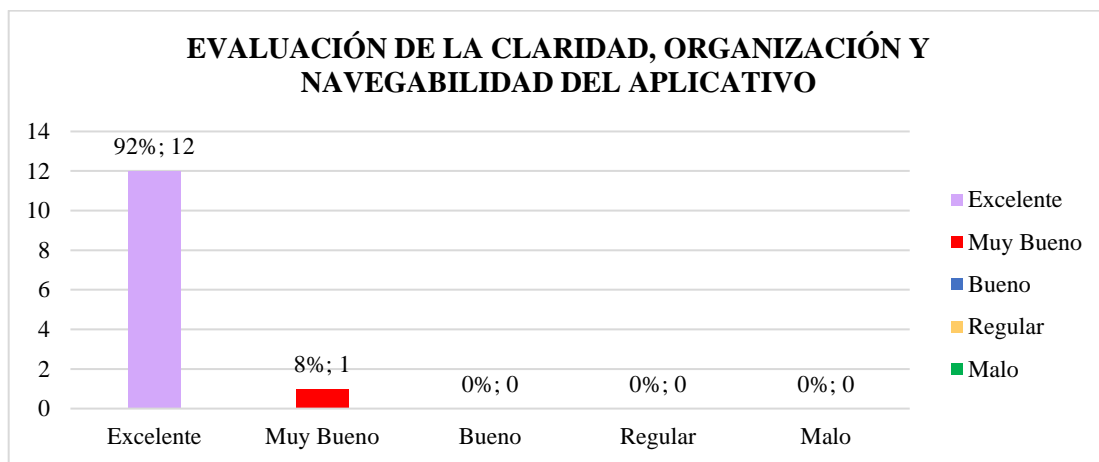


Figura 34. Aprendizabilidad - Navegabilidad entre las funciones

Mediante el gráfico de barras (véase la figura 34), se puede observar que, la mayoría de los encuestados (92%, equivalente a 12 de 13 participantes) calificó la claridad, organización y navegabilidad de la aplicación Mapa Parlante como Excelente. Adicionalmente, un 8% (1 de 13 participantes) indicó que estas características son Muy Buenas. No se registraron respuestas en las categorías Bueno, Regular o Malo.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta claridad, organización y navegabilidad. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 93.05 sobre un máximo posible de 95. La figura 35 ilustra estos hallazgos de manera gráfica.

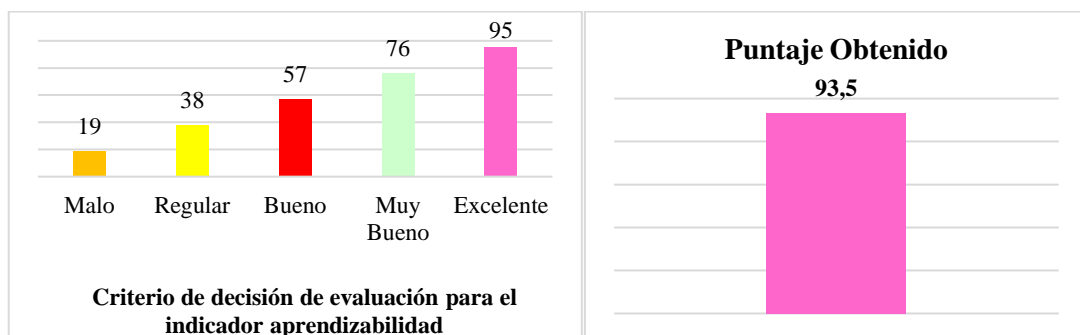


Figura 35. Resultado del indicador aprendizabilidad – Pregunta 3

La tabla 9 muestra el promedio de los puntajes obtenidos en el criterio de aprendizabilidad de la aplicación Mapa Parlante. Los puntajes para las preguntas P1, P2 y P3 son 92.08, 92.08 y 93.54 respectivamente. El promedio general de estos puntajes es 92.56%. Esto indica que, en promedio; los profesionales de la salud calificaron la aprendizabilidad de la aplicación como alta.

Tabla 9. Promedio obtenido del criterio de aprendizabilidad

PREGUNTAS	PUNTAJE OBTENIDO
P1	92.08
P2	92.08
P3	93.54
PROMEDIO	92.56

SECCIÓN DE EVALUACIÓN DE OPERABILIDAD: Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

Pregunta 4. ¿Cómo calificaría la eficiencia de la aplicación en términos de velocidad y respuesta?

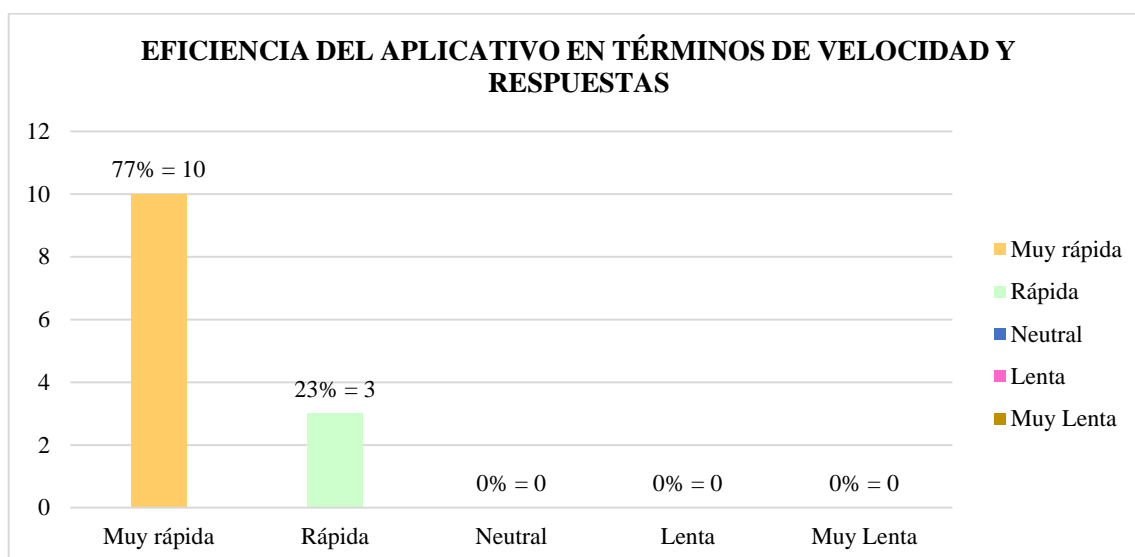


Figura 36. Operabilidad - Eficiencia de la aplicación

Mediante el grafico de barras (véase la figura 36), se puede observar que, la mayoría de los encuestados (77%, equivalente a 10 de 13 participantes) calificó la eficiencia de la aplicación en términos de velocidad y respuesta como Muy rápida. Adicionalmente, un 23% (3 de 13 participantes) indicó que estas características son Rápidas. Es importante destacar que no se registraron respuestas en las categorías Neutral, Lenta o Muy Lenta.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta eficiencia en términos de velocidad y respuesta. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 90.06 sobre un máximo posible de 95. La figura 37 ilustra estos hallazgos de manera gráfica.

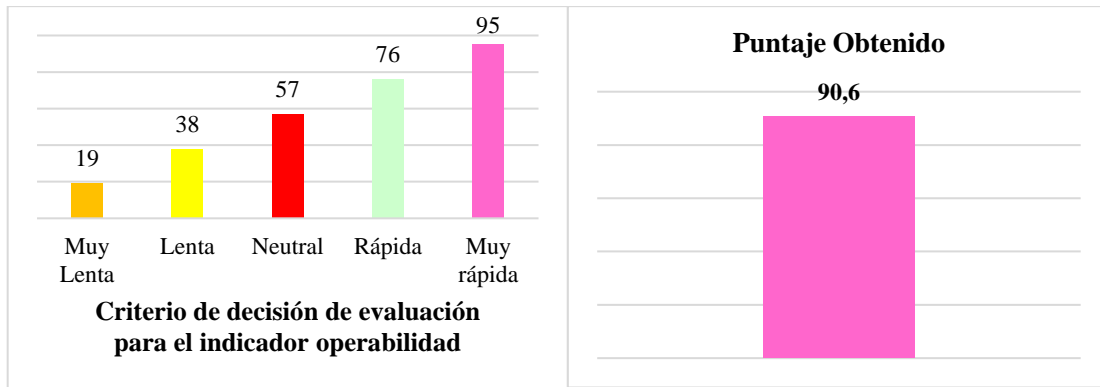


Figura 37. Resultado del indicador operabilidad- Pregunta 4

Pregunta 5. ¿Las funciones principales de la aplicación son fáciles de encontrar y acceder?

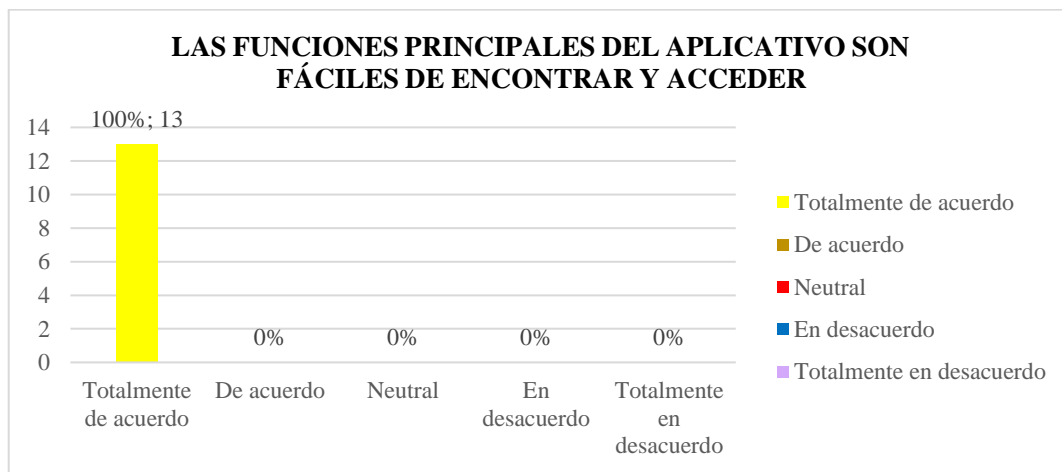


Figura 38. Operabilidad - Acceso a funciones

Mediante el gráfico de barras (véase la figura 38), se puede observar que, todos los encuestados (100%, equivalente a 13 de 13 participantes) calificaron que están Totalmente de acuerdo con que las funciones principales son fáciles de encontrar y acceder. No se registraron respuestas en las categorías De acuerdo, Neutral, En desacuerdo, Totalmente en desacuerdo.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta funciones principales fáciles de encontrar y acceder. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 95 sobre el máximo posible de 95. La figura 39 ilustra estos hallazgos de manera gráfica.

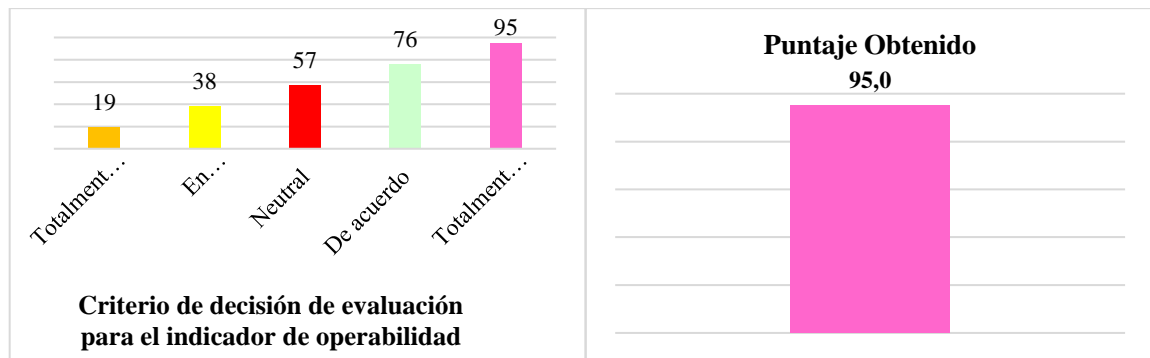


Figura 39. Resultado del indicador operabilidad- Pregunta 5

Pregunta 6. ¿Cómo calificaría la consistencia en la disposición y etiquetado de los elementos de la interfaz? (Por ejemplo, botones, menús, opciones).

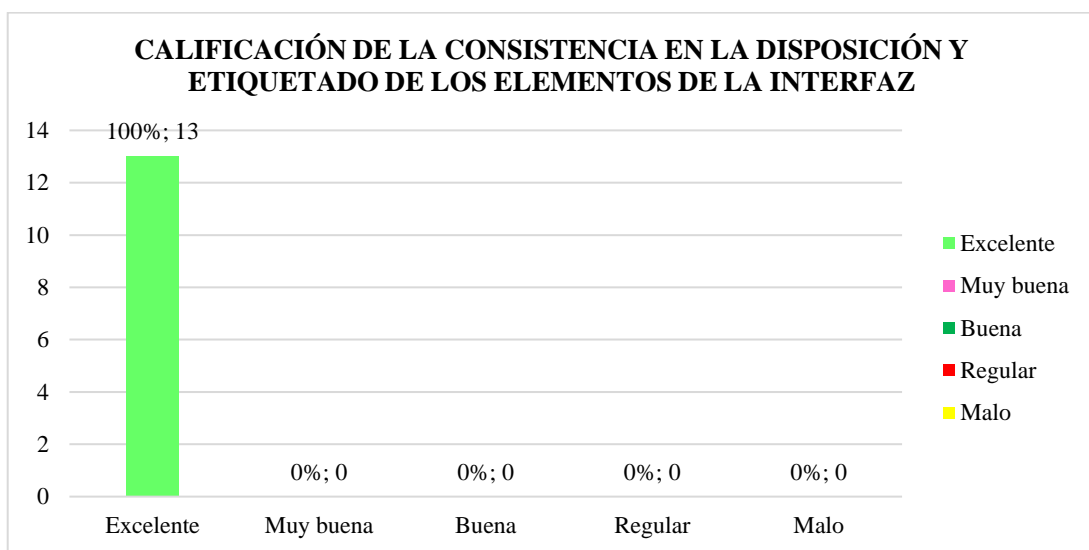


Figura 40. Operabilidad - Consistencia de elementos

Mediante el gráfico de barras (véase la figura 40), se puede observar que, todos los encuestados (100%, equivalente a 13 de 13 participantes) calificaron la consistencia en la disposición y etiquetado de los elementos de la interfaz como Excelentes. No se registraron respuestas en las categorías Muy buena, Buena, Regular, Malo.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta consistencia en la disposición y etiquetado de los elementos de la interfaz. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 95 sobre el máximo posible de 95. La figura 41 ilustra estos hallazgos de manera gráfica.

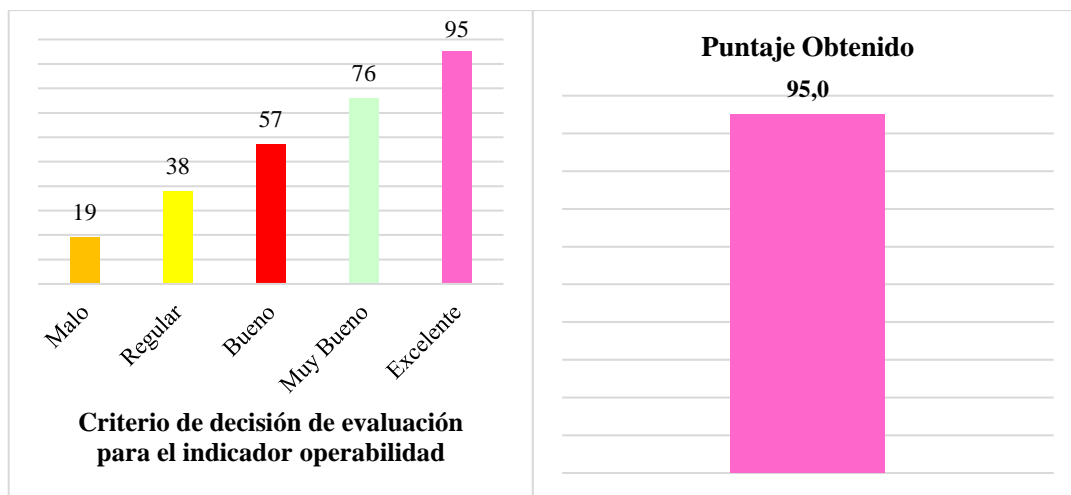


Figura 41. Resultado del indicador operabilidad– Pregunta 6

La tabla 10 presenta el promedio de los puntajes obtenidos en el criterio de operabilidad de la aplicación. Los puntajes para las preguntas P1, P2 y P3 son 90.62, 95.00 y 95.00 respectivamente. El promedio general de estos puntajes es 93.54%. Esto indica que, en promedio, los profesionales de la salud calificaron la operabilidad de la aplicación como alta.

Tabla 10. Promedio obtenido del criterio de operabilidad

PREGUNTAS	PUNTAJE OBTENIDO
P4	90.62
P5	95.00
P6	95.00
PROMEDIO	93.54

SECCIÓN DE EVALUACIÓN DE PROTECCIÓN CONTRA ERRORES DEL USUARIO:
Capacidad del sistema para proteger a los usuarios de hacer errores.

Pregunta 7. ¿Los mensajes de error le ayudan a identificar que salió mal y como puede solucionarlo?

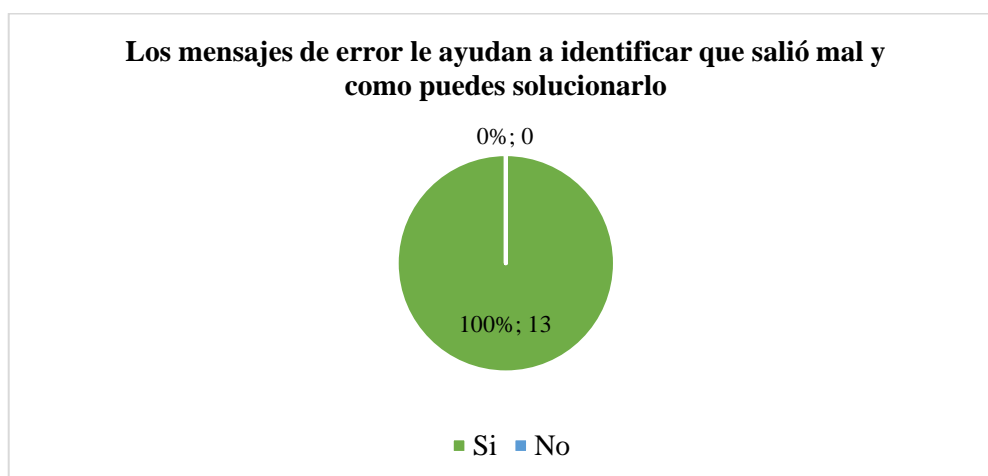


Figura 42. Protección contra errores de usuario - Mensajes de error

Mediante el gráfico circular (véase la figura 42), se puede observar que, todos los encuestados (100%, equivalente a 13 de 13 participantes) calificaron que los mensajes de error le ayudan a identificar que salió mal y como puede solucionarlo.

Pregunta 8. ¿La aplicación presenta mensajes de confirmación antes de realizar acciones que sean irreversibles o que puedan tener un impacto significativo?

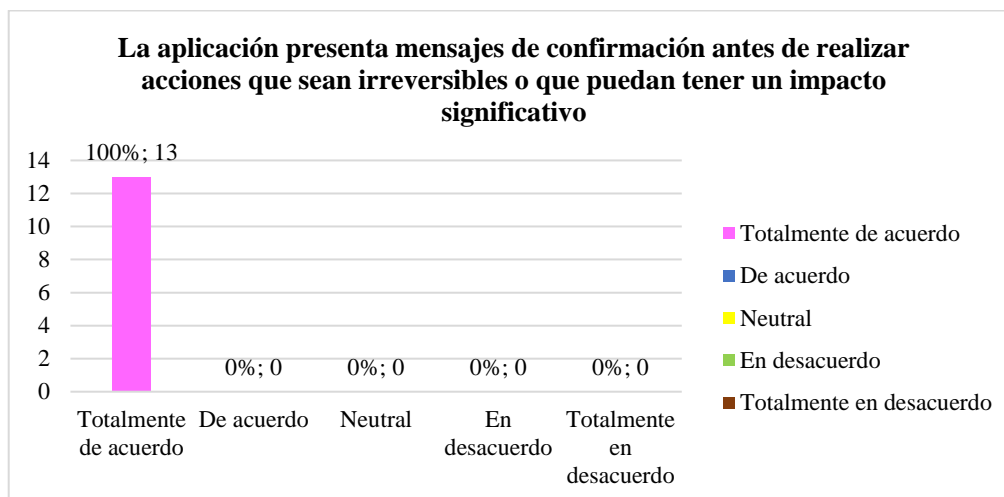


Figura 43. Protección contra errores de usuario - Mensajes de confirmación

Mediante el gráfico de barras (véase la figura 43), se puede observar que, todos los encuestados (100%, equivalente a 13 de 13 participantes) calificaron que están Totalmente de acuerdo que la aplicación presenta mensajes de confirmación antes de realizar acciones que sean irreversibles o que puedan tener un impacto significativo. No se registraron respuestas en las categorías De acuerdo, Neutral, En desacuerdo, Totalmente en desacuerdo.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta mensajes de confirmación antes de realizar acciones que sean irreversibles o que puedan tener un impacto significativo. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 95 sobre el máximo posible de 95. La figura 44 ilustra estos hallazgos de manera gráfica.

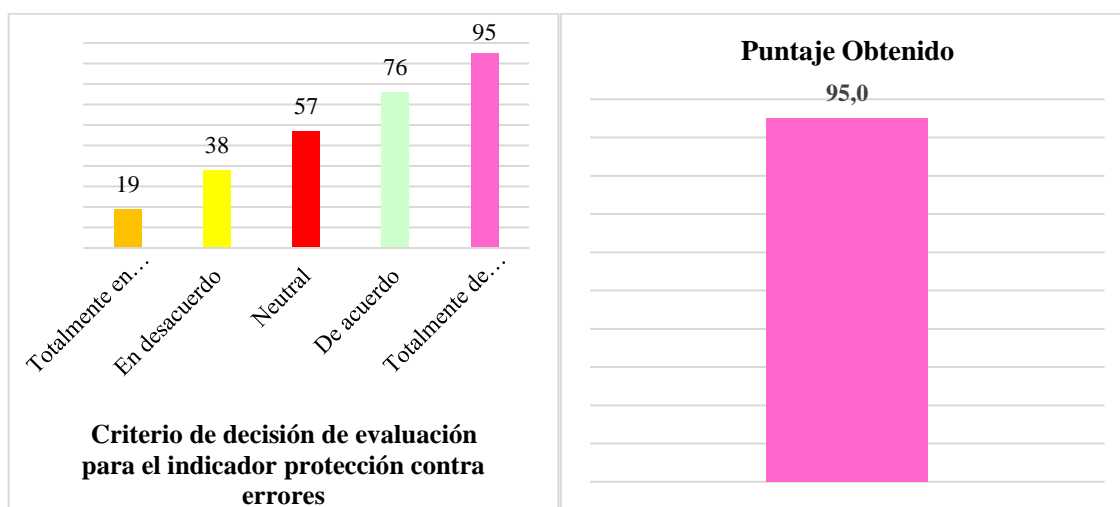


Figura 44. Resultado del indicador protección contra errores - Pregunta 8

Pregunta 9. ¿Considera que las notificaciones se presentan con claridad y son fácilmente comprensibles para el usuario?

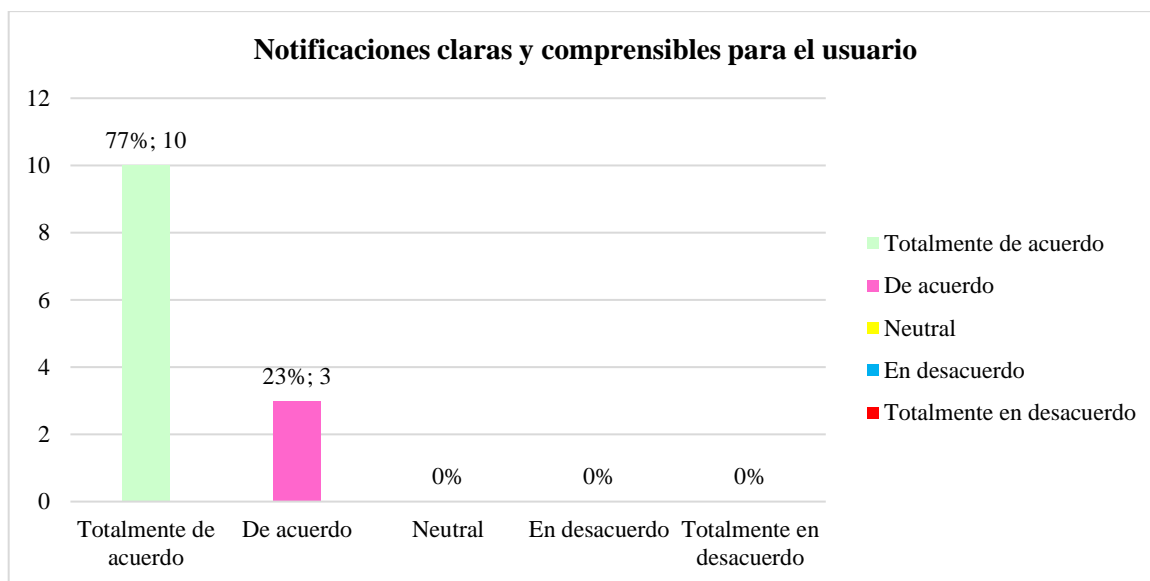


Figura 45. Protección contra errores de usuario - Notificaciones

Mediante el grafico de barras (véase la figura 45), se puede observar que, los encuestados (77%, equivalente a 10 de 13 participantes) consideran que están Totalmente de acuerdo que las notificaciones se presentan con claridad y son fácilmente comprensibles para el usuario. Adicionalmente, un 23% (3 de 13 participantes) indicó que están De acuerdo con estas características. No se registraron respuestas en las categorías Neutral, En desacuerdo, Totalmente en desacuerdo.

Basándonos en el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación contiene notificaciones que se presentan con claridad y son fácilmente comprensibles para el usuario. En el test de usuario, la aplicación obtuvo un puntaje sobresaliente de 90.60 sobre el máximo posible de 95. La figura 46 ilustra estos hallazgos de manera gráfica.

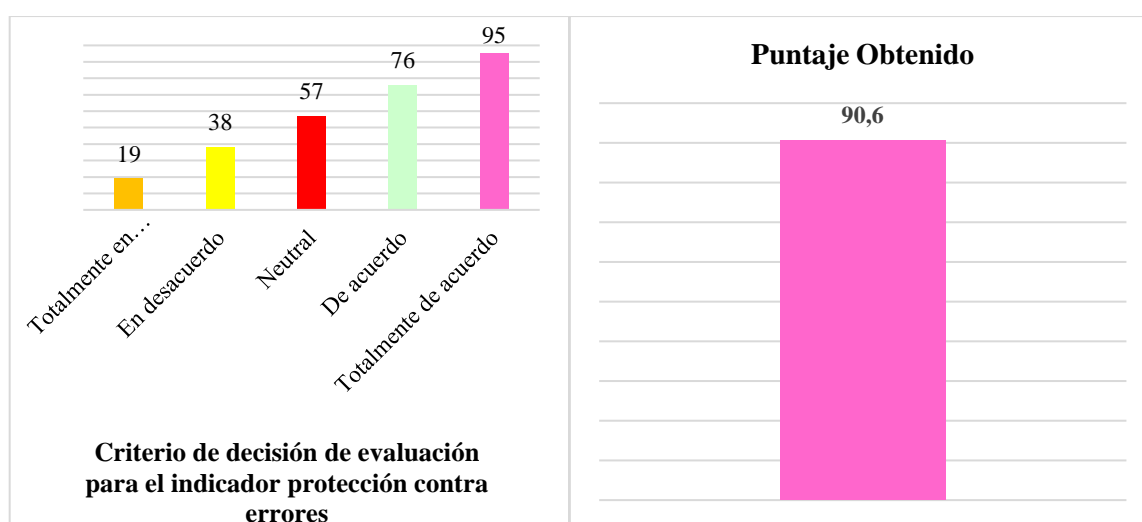


Figura 46. Resultado del indicador protección contra errores - Pregunta 9

La tabla 11 presenta el promedio de los puntajes obtenidos en el criterio de protección contra errores de usuario en la aplicación. Los puntajes para las preguntas P1, P2 y P3 son 95.00, 95.00 y 90.62 respectivamente. El promedio general de estos puntajes es 93.54%. Esto indica que, en promedio, los profesionales de la salud calificaron la protección contra errores de usuario de la aplicación como alta.

Tabla 11. Promedio obtenido del criterio de protección contra errores de usuario

PREGUNTAS	PUNTAJE OBTENIDO
P7	95.00
P8	95.00
P9	90.62
PROMEDIO	93.54

SECCION DE EVALUACION DE ESTÉTICA DE LA INTERFAZ DE USUARIO:
Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

Pregunta 10. ¿Encuentra visualmente atractiva la interfaz de usuario de la aplicación?

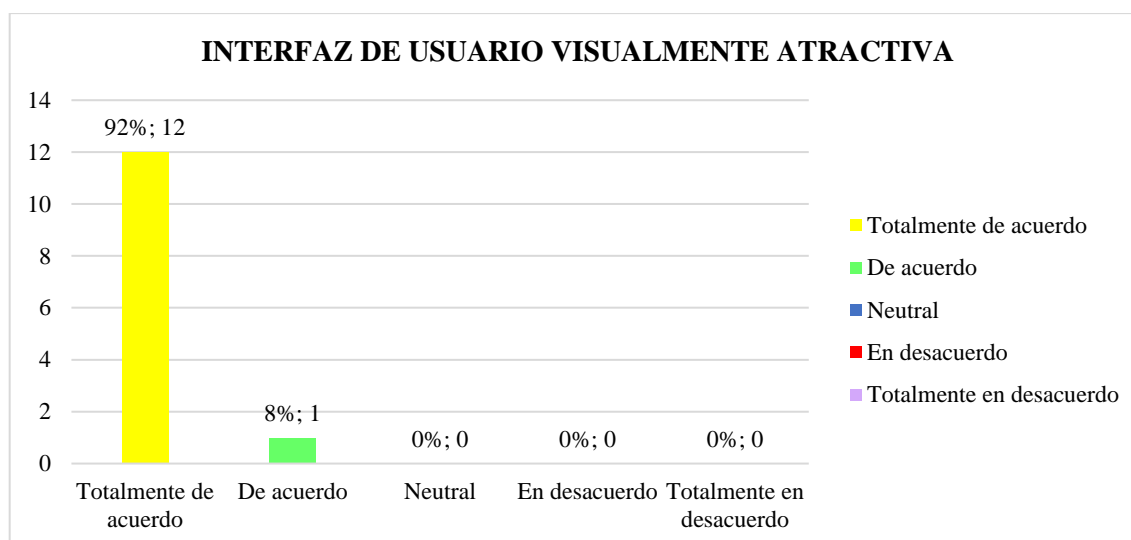


Figura 47. Estética de la interfaz de usuario - Visualmente atractiva

Mediante el grafico de barras (véase la figura 47), se puede observar que, los encuestados (92%, equivalente a 12 de 13 participantes) calificaron que están Totalmente de acuerdo con que encuentran visualmente atractiva la interfaz de usuario de la aplicación. Adicionalmente, un 8% (1 de 13 participantes) indicó que está De acuerdo con estas características. No se registraron respuestas en las categorías Neutral, En desacuerdo, Totalmente en desacuerdo.

Mediante el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación es visualmente atractiva en la interfaz de usuario de la aplicación. En el test de usuario, la aplicación obtuvo un puntaje de 93.50 sobre el máximo posible de 95. La figura 48 ilustra estos hallazgos de manera gráfica.

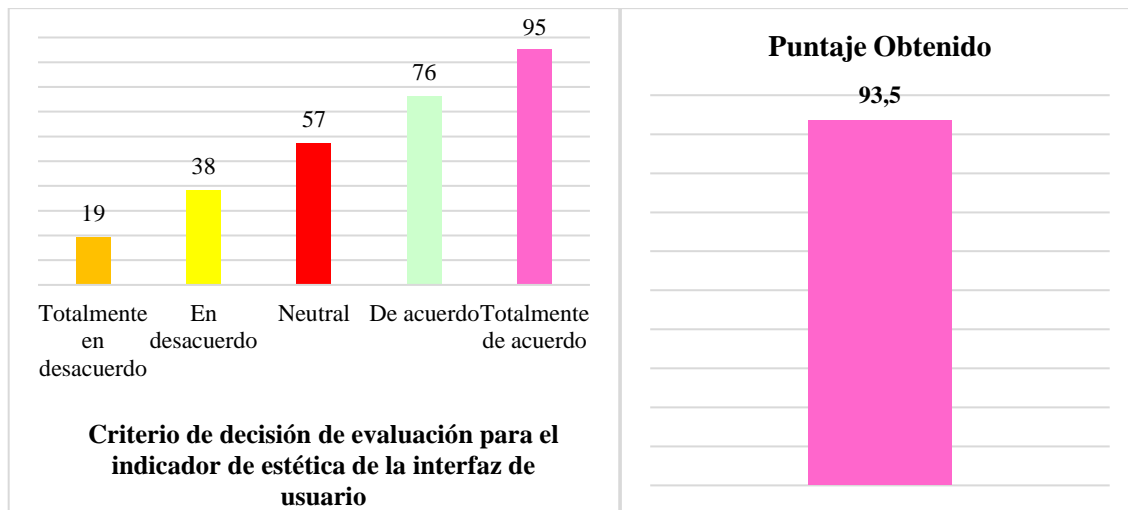


Figura 48. Resultado del indicador estética de la interfaz de usuario – Pregunta 10

Pregunta 11. ¿La distribución de espacio en la aplicación le parece eficiente y bien organizada?

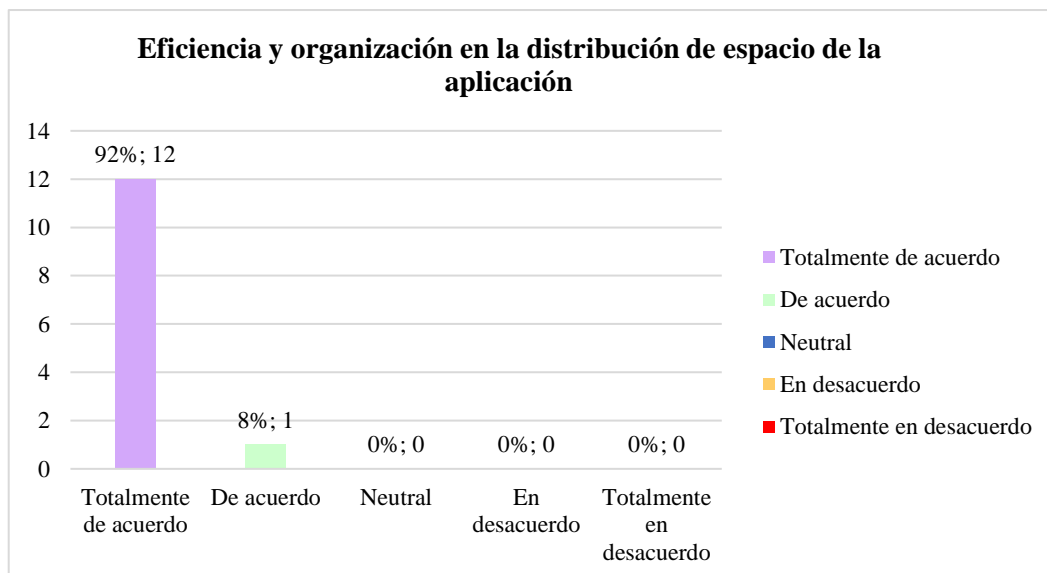


Figura 49. Estética de la interfaz de usuario - Distribución

Mediante el gráfico de barras (véase la figura 49), se puede observar que, los encuestados (92%, equivalente a 12 de 13 participantes) están Totalmente de acuerdo que la distribución de espacio en la aplicación le parece eficiente y bien organizada. Adicionalmente, un 8% (1 de 13 participantes) indicó que está De acuerdo con estas características. No se registraron respuestas en las categorías Neutral, En desacuerdo, Totalmente en desacuerdo.

Mediante el análisis de los criterios de decisión establecidos, se puede concluir que la distribución de espacio en la aplicación le parece eficiente y bien organizada. En el test de usuario, la aplicación obtuvo un puntaje de 93.50 sobre el máximo posible de 95. La figura 50 ilustra estos hallazgos de manera gráfica.

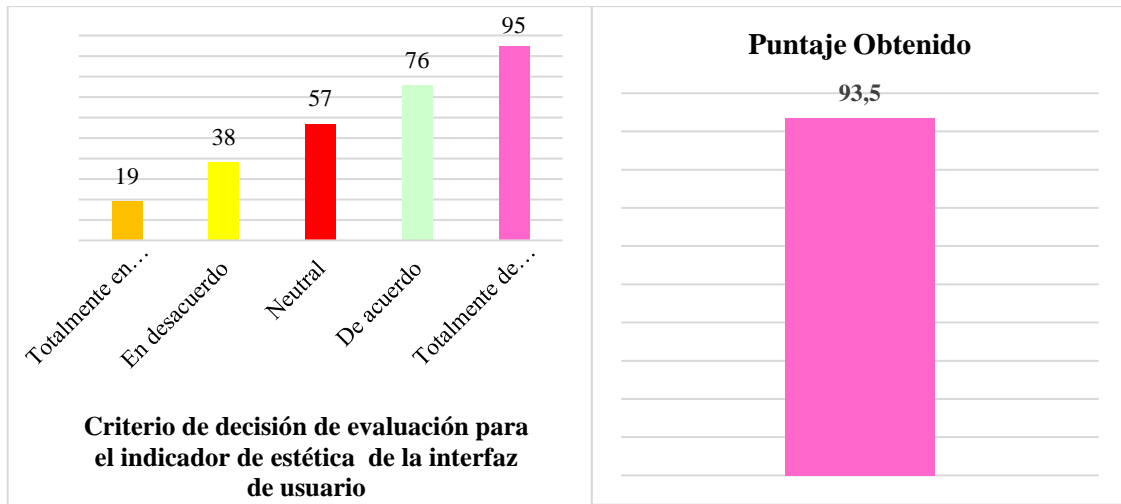


Figura 50. Resultado del indicador estética de la interfaz de usuario – Pregunta 11

Pregunta 12. ¿Cómo evaluaría la utilización de colores, fuentes y elementos visuales en la interfaz de la aplicación en términos de su atractivo y coherencia?

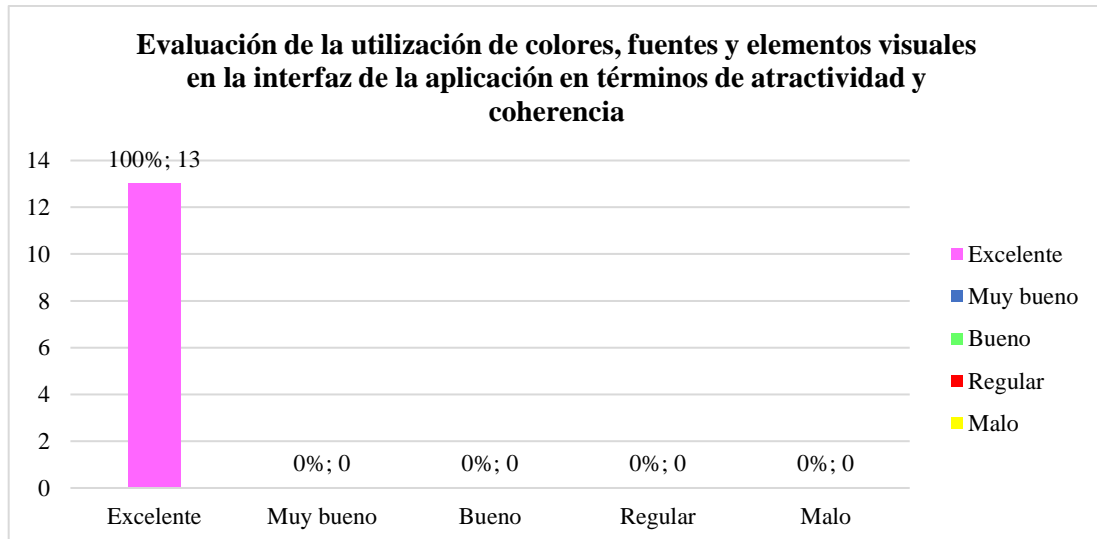


Figura 51. Estética de la interfaz de usuario - Utilización de elementos

Mediante el gráfico de barras (véase la figura 51), se puede observar que, todos los encuestados (100%, equivalente a 13 de 13 participantes) evaluaron la utilización de colores, fuentes y elementos visuales en la interfaz de la aplicación en términos de su atractivo y coherencia como Excelente. No se registraron respuestas en las categorías Muy bueno, Bueno, Regular o Malo.

Mediante el análisis de los criterios de decisión establecidos, se puede concluir que la aplicación presenta un excelente manejo en la utilización de colores, fuentes y elementos visuales en la interfaz. En el test de usuario, la aplicación obtuvo un puntaje de 95 sobre un máximo posible de 95. La figura 52 ilustra estos hallazgos de manera gráfica.

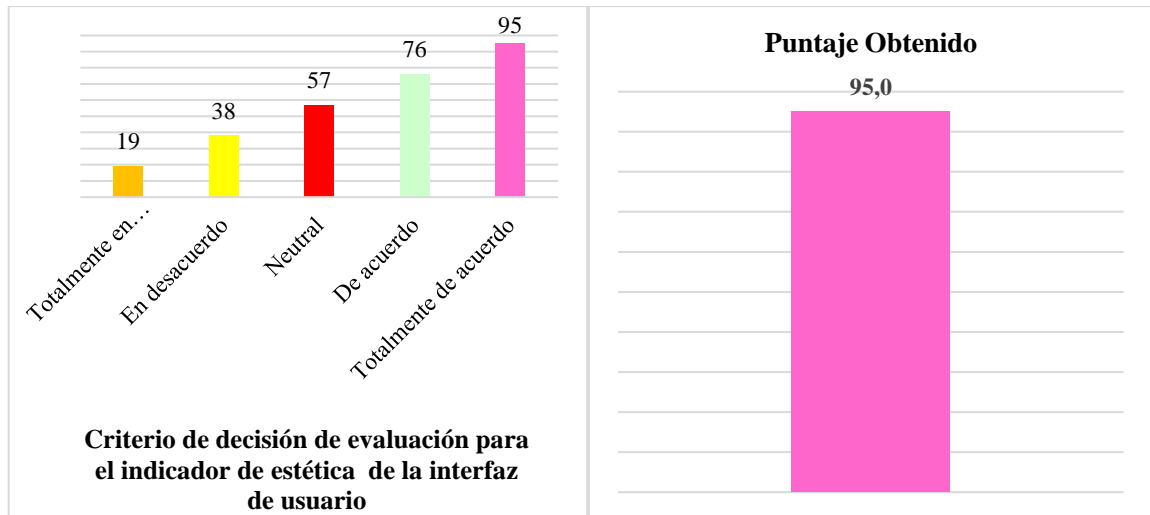


Figura 52. Resultado del indicador estética de la interfaz de usuario – Pregunta 12

La tabla 12 presenta el promedio de los puntajes obtenidos en el criterio de estética de la interfaz de usuario. Los puntajes para las preguntas P1, P2 y P3 son 93.50, 93.50 y 95.00 respectivamente. El promedio general de estos puntajes es 94.00%.

Tabla 12. Promedio obtenido del criterio de la estética de la interfaz de usuario

PREGUNTAS	PUNTAJE OBTENIDO
P10	93.50
P11	93.50
P12	95.00
PROMEDIO	94.00

4.2. Discusión

En la fase inicial del proyecto, se llevó a cabo un análisis exhaustivo de las herramientas disponibles para la representación del Mapa Parlante en el Centro de Salud Chambo. La elección de TypeScript, Angular, Nest.js, PostgreSQL, Geoserver, OpenLayers, Openrouteservice y Bing Maps se justifica a través de una investigación bibliográfica. Estos incluyeron aspectos como la integración con Angular, la familiaridad del equipo de desarrollo con TypeScript, y consideraciones económicas y de licencias de software. La coherencia entre estas herramientas fue esencial para mantener una arquitectura consistente y facilitar el desarrollo y la integración de la aplicación.

La metodología Scrum demostró ser una elección acertada para la implementación de la aplicación en el Centro de Salud Chambo. El enfoque iterativo y colaborativo facilitó una respuesta ágil a los cambios y permitió una entrega continua.

La aplicación con acceso web y móvil se construyó con éxito gracias a la implementación de estas herramientas y metodologías, asegurando su funcionalidad integral durante la cual la aplicación fue sometida a una evaluación detallada de usabilidad utilizando el modelo de calidad ISO/IEC 25010. Criterios específicos, como aprendizabilidad, operabilidad, protección contra errores de usuario y estética de la interfaz de usuario, se evaluaron mediante una encuesta a profesionales de la salud del Centro de Salud Chambo.

Los resultados obtenidos en estas áreas permitieron identificar que la aplicación permite al usuario familiarizarse con sus funciones en un 92.56 % de aprendizabilidad, su operabilidad se destaca con un 93.54 %, la evaluación de protección contra errores de usuario fue de 93.54 % y en termino de estética de interfaz de usuario se obtuvo un promedio de 94.00 %. Estos hallazgos se conectan directamente con las elecciones de herramientas y metodologías, destacando la importancia de una integración coherente.

La investigación y desarrollo de la aplicación Mapa Parlante han contribuido significativamente al ámbito de la salud y la tecnología, puesto que proporciona funciones de visualización de distintos tipos de mapas, consumo y visualización de datos obtenidos de Ficha Familiar, filtrar la visualización mediante categorías de búsqueda, búsqueda de paciente mediante la cedula, generación de rutas, generación de reportes, permitiendo una mejor de gestión de EAIS y toma de decisiones.

Los resultados obtenidos proporcionan valiosas lecciones aprendidas para proyectos similares. La aplicación no solo cumple con los objetivos establecidos, sino que también establece un precedente para futuras implementaciones de tecnologías similares en entornos de salud a nivel parroquial, cantonal, provincial, zonal y nacional.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El análisis de metodologías y herramientas para el desarrollo de la aplicación Mapa Parlante en el Centro de Salud Chambo ha permitido una selección acertada de tecnologías modernas opensource como Node.js, Angular, Tailwind CSS, Geoserver, OpenLayers, Bing Maps y Openrouteservice; obteniendo como resultado una aplicación funcional, dinámica y altamente interactiva que cumple con los estándares establecidos por la norma ISO/IEC 25010.
- La integración de OpenLayers con Angular permitió enriquecer tanto la interactividad del mapa, como la inclusión de APIs externas de Bing Maps y Openrouteservice proporcionando acceso a mapas detallados y servicios de enrutamiento sin costo alguno.
- El uso de la metodología Scrum en la implementación de aplicaciones Mapa Parlante y la arquitectura cliente-servidor con microservicios permitieron un despliegue exitoso, eficiente, flexible y escalable; proporcionando una solución tecnológica robusta y alineada con los objetivos del proyecto.
- La evaluación de la usabilidad de la aplicación Mapa Parlante en el Centro de Salud Chambo, aplicando el modelo de calidad ISO/IEC 25010, ha arrojado resultados altamente positivos. Los indicadores revelan un alto grado de aprendizabilidad, operabilidad, protección contra errores de usuario y estética de la interfaz de usuario, con puntuaciones significativas, como el 92.56% en aprendizabilidad, 93.54% en operabilidad y protección contra errores, así como 94% en la estética de la interfaz, la aplicación respalda el diseño intuitivo y la eficacia de la aplicación, consolidando su posición como una herramienta tecnológica accesible y eficiente para los profesionales de la salud en el Centro de Salud Chambo.

5.2.Recomendaciones

- En el desarrollo de la aplicación Mapa Parlante basada en una arquitectura cliente/servidor con microservicios es esencial que se adopte un enfoque proactivo para mantenerse al día con las tendencias y actualizaciones de las herramientas tecnológicas utilizadas incluyendo Node.js, Angular, OpenLayers, Geoserver, OpenRouteService entre otras tecnologías y servicios relacionados que permitan reducir el tiempo y los costos asociados al proceso para garantizar una entrega más rápida y confiable de las nuevas funcionalidades y mejoras a los usuarios finales.
- Realizar un análisis a detalle de la especificación de requerimientos de software y necesidades del cliente antes de iniciar el desarrollo de la aplicación en base a las directrices dadas por el estándar IEEE, adoptándose como guía para las Especificaciones de Requisitos de Software (IEEE Std. 830-1998). Esta práctica garantizará una comprensión clara de las expectativas del proyecto y facilitará la toma de decisiones informadas a lo largo de todo el proceso de desarrollo
- Implementar prácticas de seguridad robustas en el desarrollo de software. Esto incluye la utilización de técnicas de cifrado, como el cifrado de extremo a extremo, para proteger la confidencialidad de los datos. Además, es fundamental gestionar cuidadosamente las contraseñas, mediante el uso de técnicas como el almacenamiento seguro y el uso de algoritmos de hash para protegerlas de posibles ataques. Asimismo, se sugiere realizar auditorías periódicas del código y de la infraestructura para identificar posibles vulnerabilidades y asegurar el cumplimiento de los estándares de seguridad. Estas medidas garantizarán la seguridad y confiabilidad de los datos, proporcionando una capa adicional de protección para la aplicación y mitigando los riesgos de seguridad.

BIBLIOGRAFÍA

- [1] World Health Organization, «Proyecto de estrategia mundial sobre salud digital 2020-2025,» 2020. [En línea]. Available: https://cdn.who.int/media/docs/default-source/documents/200067-1b-full-draft-digital-health-strategy-with-annex-cf-6jan20-cf-rev-10-1-clean-sp_1c8b2b9c-4c25-4efb-8553-9f466028b583.pdf?sfvrsn=4b848c08_4. [Último acceso: 04 03 2024].
- [2] E. López Lara, C. Posada, J. Moreno y U. d. Sevilla, «LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA,» [En línea]. Available: <http://imsturex.unex.es/MUIETSIG/TEMA1.pdf>. [Último acceso: 04 03 2024].
- [3] Universidad Laica Eloy Alfaro de Manabí, Atención primaria de salud, Manta: Departamento de Edición y Publicación Universitaria (DEPU), 2018.
- [4] M. A. Moreno Castillo, «GEOINFORMÁTICA Y SALUD: REVISIÓN, CARACTERIZACIÓN Y ANÁLISIS DE PLATAFORMAS SIG BASADAS EN WEB PARA VIGILANCIA EN SALUD PÚBLICA,» *Revista Internacional de Ciencia y Tecnología de la Información Geográfica*, n° 831, pp. 79-80, 2023.
- [5] Geoserver, «Geoserver,» [En línea]. Available: <https://geoserver.org/about/>. [Último acceso: 05 03 2024].
- [6] Ministerio de Salud Pública, MANUAL DEL MODELO DE ATENCION INTEGRAL DEL SISTEMA NACIONAL DE SALUD, Quito: MSP Ecuador, 2018.
- [7] A. M. Paola B. Vidal, «Experiencia de Usuario + Web Responsivo: Un Estudio desde la Perspectiva de un Enfoque Integrado,» Semantic Scholar, Patagonia, 2020.
- [8] L. Puciarelli, Angular, Buenos Aires: Six Ediciones, 2020.
- [9] T. M.-A. A. Q.-M. Mauricio Rea-Peñañiel, «Integración de componentes Enterprise Java Beans y Front End Angular para analizar el desempeño de acceso a datos,» VOL. 5, Ibarra, 2023.
- [10] F. O. Mayorga y D. G. Iza, Aplicación de página única (spa) utilizando el framework angular para el reclutamiento y selección del personal en la Empresa Eléctrica Ambato Regional Centro Norte S.A., Ambato: Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 2023.
- [11] Angular, «Introducción a los conceptos de Angular,» Angular Lat, [En línea]. Available: <https://docs.angular.lat/guide/architecture>. [Último acceso: 05 03 2024].
- [12] L. Puciarelli, Angular: TypeScript–Arquitectura–Instalación–Directivas y Bindings–Forms–Ruteo y más., Buenos Aires: Six Edicion, 2020.

- [13] Á. I. S. Sisa y M. E. N. Vilema, «Aplicación web progresiva para automatizar la gestión de inventario del hospital universitario andino utilizando REACT,» 2022. [En línea]. Available: <http://dspace.unach.edu.ec/handle/51000/9017>. [Último acceso: 05 03 2024].
- [14] L. Pucciarelli, Node.js, Buenos Aires: Plandos, 2020.
- [15] E. Haro y T. Guarda, «Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot,» *Revista Ibérica de Sistemas y Tecnologías de la Información*, vol. I, pp. 309-321, 2019.
- [16] Nest.js, «Nest.js Documentation,» A Progressive Node.Js Framework, [En línea]. Available: <https://docs.nestjs.com>. [Último acceso: 05 03 2024].
- [17] M. Sabo, «NestJS,» *Josip Juraj Strossmayer University of Osijek. Department of Mathematics. Chair of Applied Mathematics. Computer Science Research Group*, 2020.
- [18] A. D. Phan, «Developing back-end of a web application with NestJS framework: Case: Integrify Oy's student management system,» 2020.
- [19] M. Marqués, Base de Datos, Col·lecció Sapientia.
- [20] F. L. O. Rivera, Base de datos relacionales, Medellin: ITM.
- [21] Á. P. Gómez y J. J. R. Jalca, Fundamentos sobre la gestion de Base de Datos, Alicante: 3Ciencias.
- [22] M. P. Z. Ordóñez y J. R. M. Ríos, Administracion de base de datos con postgresql, Alicante: 3Cielo, 2017.
- [23] J. D. Chavez, Cliente PQSL de POSTGRESQL, Venezuela: IEASS, 2020.
- [24] The PostgreSQL Global Development Group, «PostgreSQL,» Official documentation, [En línea]. Available: <https://www.postgresql.org/docs/>. [Último acceso: 05 03 2024].
- [25] TypeORM, «TypeORM,» TypeORM, [En línea]. Available: <https://typeorm.io>. [Último acceso: 05 03 2024].
- [26] C. Castillo y K. Cancino, «Diseño de un Sistema web para el control de Curriculum Vitae Electrónico de personal docente basado en una arquitectura orientada a servicios (API REST),» *Revista de Investigación en Tecnologías de la Información: RITI*, pp. 28-42, 2022.
- [27] IBM, «REST APIs Design,» [En línea]. Available: <https://www.ibm.com/topics/rest-apis>. [Último acceso: 05 03 2024].

- [28] J. M. Hernández, «Mapa Parlante,» JARDÍN BOTÁNICO DE BOGOTÁ JOSÉ CELESTINO MUTIS, Bogota, 2017.
- [29] dokumen.tips, «Mapa Parlante,» 2019. [En línea]. Available: <https://dokumen.tips/documents/mapa-parlante-salud.html>. [Último acceso: 05 03 2024].
- [30] ArcGIS Resources, «Introducción a SIG,» esri, [En línea]. Available: <https://resources.arcgis.com/es/help/getting-started/articles/026n0000000t000000.htm>. [Último acceso: 05 03 2024].
- [31] A. Morales, «GeoServer,» mappingGis, [En línea]. Available: <https://mappinggis.com/2022/06/geoserver-para-novatos/>. [Último acceso: 05 03 2024].
- [32] Open Geospatial Consortium, «Web Map Service,» [En línea]. Available: <https://www.ogc.org/standard/wms/>. [Último acceso: 04 03 2024].
- [33] MappingGIS, «OpenLayers vs Leaflet ¿cuál es mejor?,» MappingGIS, [En línea]. Available: <https://mappinggis.com/2016/11/openlayers-vs-leaflet-mejor/>. [Último acceso: 04 03 2024].
- [34] C. P. J. B. A. K. N. F. S. Z. M. & Z. A. Ludwig, «TRAFFIC SPEED MODELLING TO IMPROVE TRAVEL TIME ESTIMATION IN,» *Volumen XLVIII The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. Volumen XLVIII, pp. 109-116, 2023.
- [35] J. Psotta, «openrouteservice,» github, [En línea]. Available: <https://github.com/GIScience/openrouteservice>. [Último acceso: 05 03 2024].
- [36] ISO25000, «ISO/IEC 25010,» [En línea]. Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3>. [Último acceso: 05 03 2024].
- [37] N. B. J. M. & T. M. C. Tymkiw, «Scrum como Herramienta Metodológica para el Aprendizaje de la Programación,» *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, n° 26, pp. 81-89, 2020.

ANEXOS

Anexo 1. Bosquejo para la aplicación responsiva

En la figura 53, se presenta el prototipo de baja fidelidad o maqueta del diseño para la aplicación en su versión con acceso móvil, representado en escala de grises.



Figura 53. Diseño responsivo de la aplicación

Anexo 2. Guía rápida para la implementación del servidor de Geoserver

A continuación, en la tabla 13 se presenta las herramientas necesarias para la implementación del servidor de Geoserver.

Tabla 13. Herramientas necesarias para la implementación de Geoserver.

Herramienta	Versión
Java	Compatible con GeoServer. Jdk V17
Apache Tomcat	Ver ultimo estable y compatibilidad con GeoServer V9
GeoServer	2.24.1

- **Primer paso**

Verificar si está instalado Java (JRE) en el dispositivo a utilizar.

- **Segundo paso**

Instalación de Apache Tomcat 9 y configuraciones necesarias.

En la figura 54 se muestra el proceso de instalación de Apache Tomcat 9.

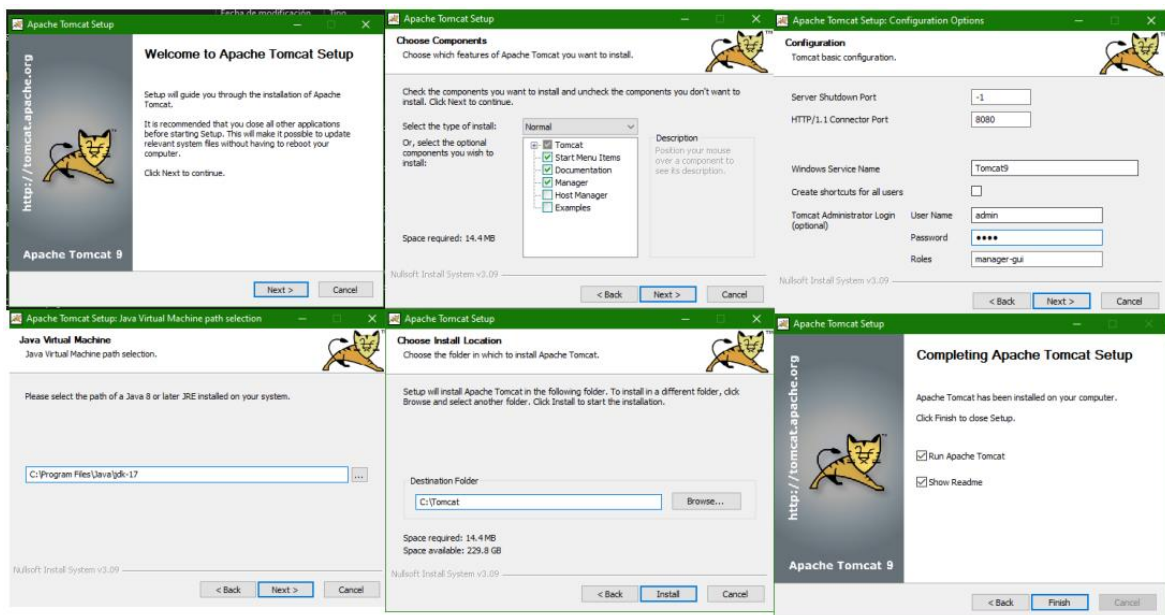


Figura 54. Proceso de instalación de Apache Tomcat

Una vez instalado probar en el navegador su funcionamiento mediante la url correspondiente: <http://localhost:8080>

- **Tercer paso**

Instalación de Geoserver

Para instalar GeoServer accedemos a la página de inicio de Apache Tomcat, dentro de esta página presionamos el botón “Manager App”. Ingresamos las credenciales.

Dentro del gestor de aplicaciones web de Apache Tomcat en la sección Desplegar / Archivo WAR a desplegar, examinamos y seleccionamos el fichero de geoserver.war previamente descargado, véase la figura 55. Presionamos en desplegar y esperamos varios minutos hasta que finalice la acción.



Figura 55. Implementar Geoserver dentro del gestor de aplicaciones de Apache

Tras esta acción, aparecerá la carpeta de Geoserver en el directorio y accedemos a ella mediante las credenciales por defecto que nos da Geoserver. Una vez ingresado a este servidor podemos cargar las capas necesarias para consumirlas en nuestra aplicación.

En la figura 56, se muestra el servidor de Geoserver funcional.



Figura 56. Servidor de Geoserver activo

Anexo 3. Pantallas de la aplicación con acceso web y móvil Mapa Parlante y la descripción de sus funciones.

La aplicación Mapa Parlante cuenta con dos tipos de usuarios: administradores y EAIS, cada uno con funciones específicas para utilizar las distintas características de la aplicación.

El usuario administrador y EAIS comparten las siguientes funciones:

- **Inicio de sesión**

Pueden acceder utilizando las credenciales específicas de administrador o EAIS a través de la interfaz de usuario representada en la figura 57.



Figura 57. Inicio de sesión

- **Módulo de visualización del mapa**

En este módulo, los usuarios pueden llevar a cabo diversas funciones, como se muestra en la figura 58, que incluyen:

- Visualizar el mapa correspondiente a Chambo, cargado desde Geoserver. Además, tienen la capacidad de interactuar con las diversas capas de mapas disponibles, tales como relieve, mapa topográfico, satélite y el mapa específico de Chambo.
- Acceder a la información total de las fichas registradas en el mapa, la cual se carga directamente desde la base de datos de Ficha Familiar.
- Realizar búsquedas específicas utilizando el número de cédula para encontrar a pacientes registrados en la Ficha Familiar.
- Generar reportes personalizados según los criterios de búsqueda, filtrado o visualización que el usuario prefiera.
- Realizar búsquedas detalladas utilizando las categorías presentes en el mapa.
- Visualizar un resumen de los indicadores presentes en el mapa, mostrando la ubicación exacta de cada uno.

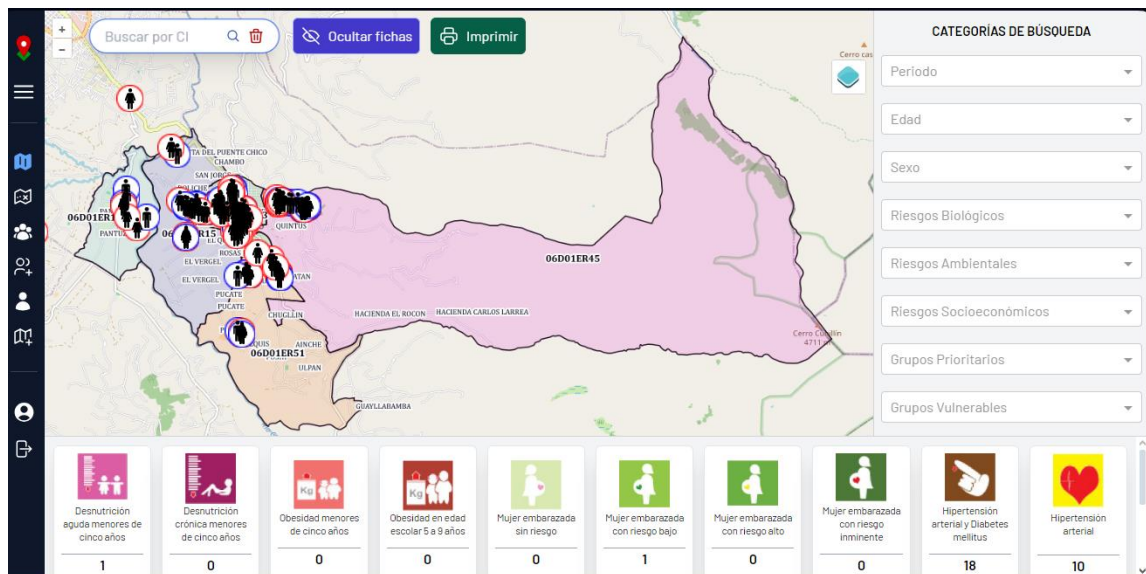


Figura 58. Módulo de visualización del mapa

- **Módulo para generación de rutas**

En este módulo, según se visualiza en la figura 59, los usuarios tienen la capacidad de generar rutas desde la ubicación de un paciente hasta el centro de salud de Chambo. Al realizar esta acción, podrán obtener detalles precisos sobre la distancia a recorrer, el tiempo estimado del trayecto y la posibilidad de redirigirse a Google Maps para obtener información adicional. Asimismo, cuentan con la opción de generar un reporte detallado de la ruta mediante el botón correspondiente en la aplicación.

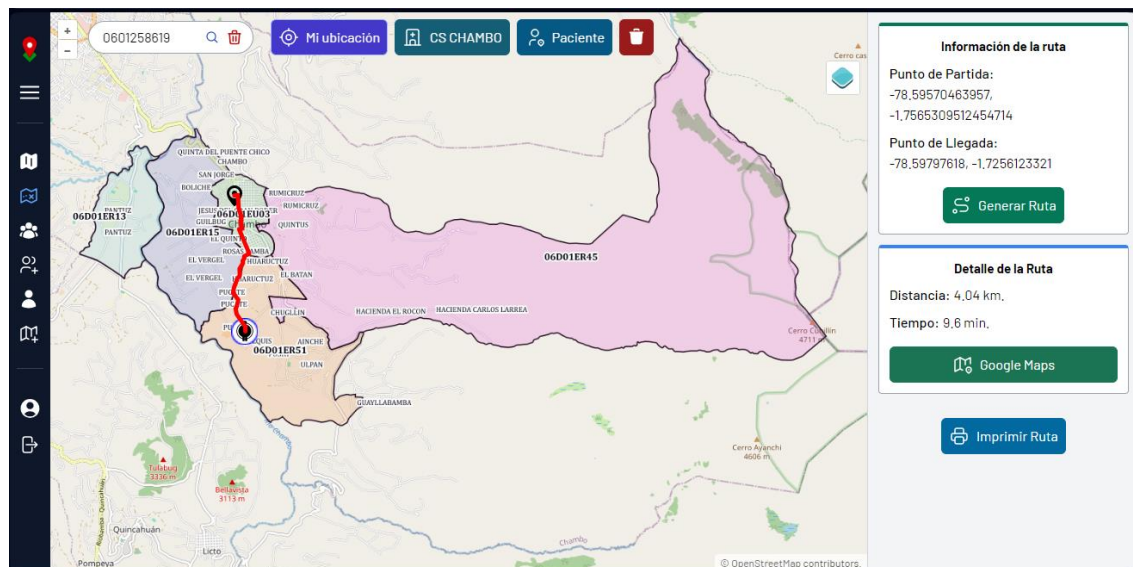


Figura 59. Módulo para generación de rutas

- **Módulo de gestión del perfil del usuario**

La funcionalidad de Gestión de Perfil de Usuario permite a todos los usuarios, independientemente de su rol en el sistema, acceder y administrar su información personal, véase la figura 60.

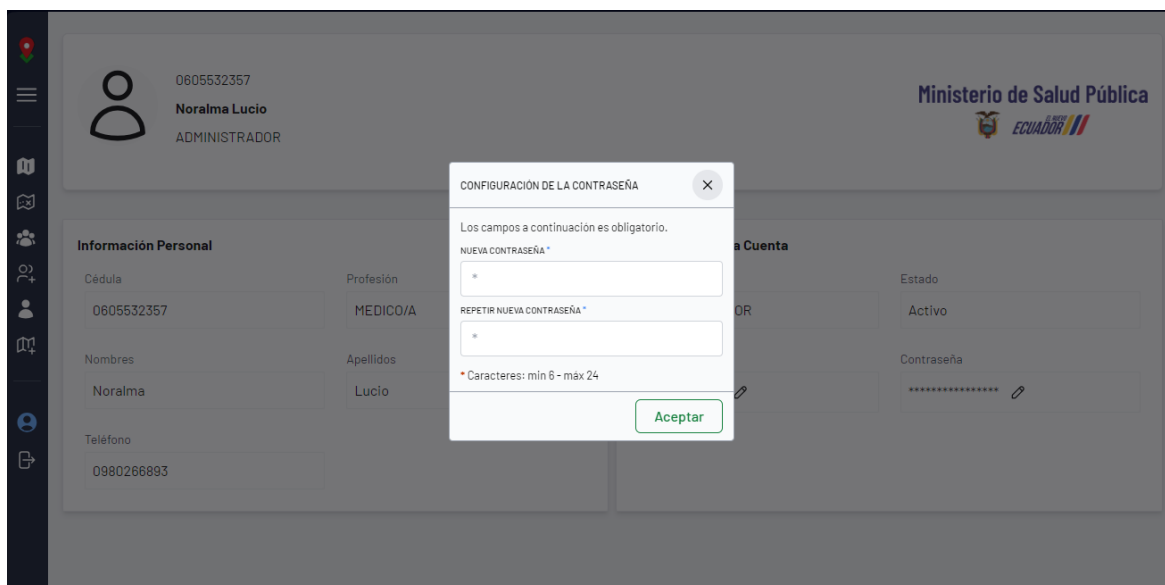


Figura 60. Módulo de gestión del perfil del usuario

Funciones permitidas para el administrador

- **Módulo de gestión de Brigada EAIS.**

El administrador tiene la capacidad de administrar las brigadas EAIS destinadas a las visitas domiciliarias. Podrá asignar múltiples profesionales de salud a sectores específicos del mapa identificados por el código EAIS. Además, cuenta con la posibilidad de visualizar las brigadas creadas y actualizar su estado, incluyendo la opción de desactivar a un profesional de una brigada EAIS, como se ilustra en la figura 61.

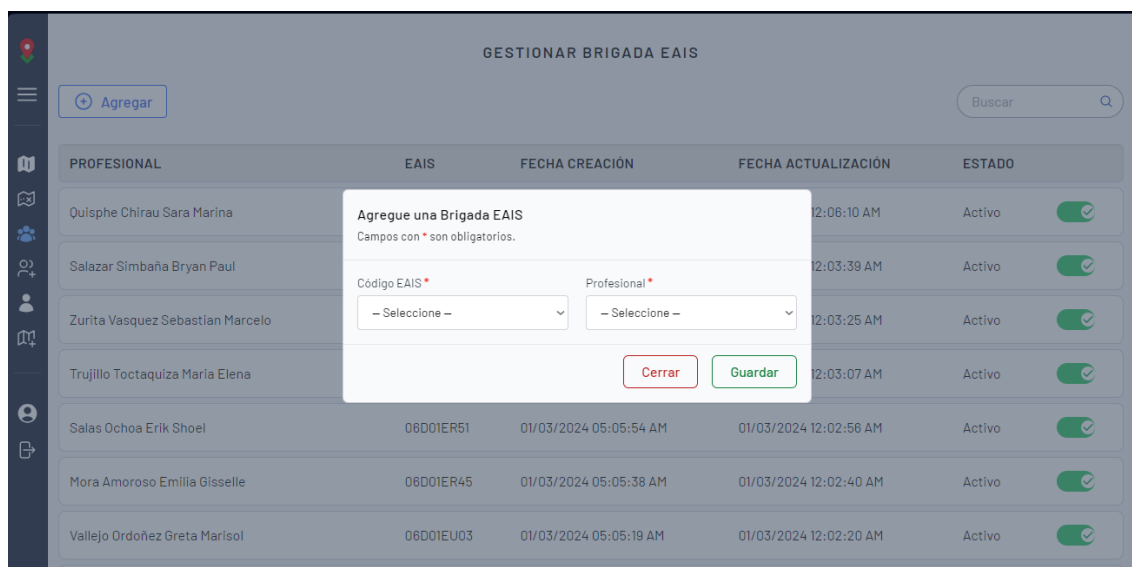


Figura 61. Módulo de gestión de brigada EAIS

- **Módulo de gestión de Profesionales**

Este módulo permite la incorporación de nuevos usuarios al sistema de acuerdo con los requisitos y necesidades del Centro de Salud de Chambo. Ofrece la funcionalidad de actualizar el estado del usuario, permitiendo su activación o desactivación en la aplicación. Además, el administrador tiene la capacidad de restablecer la contraseña de aquellos usuarios

que requieran esta acción, la cual se configurará con el número de cédula correspondiente a cada usuario, como se ilustra en la figura 62.

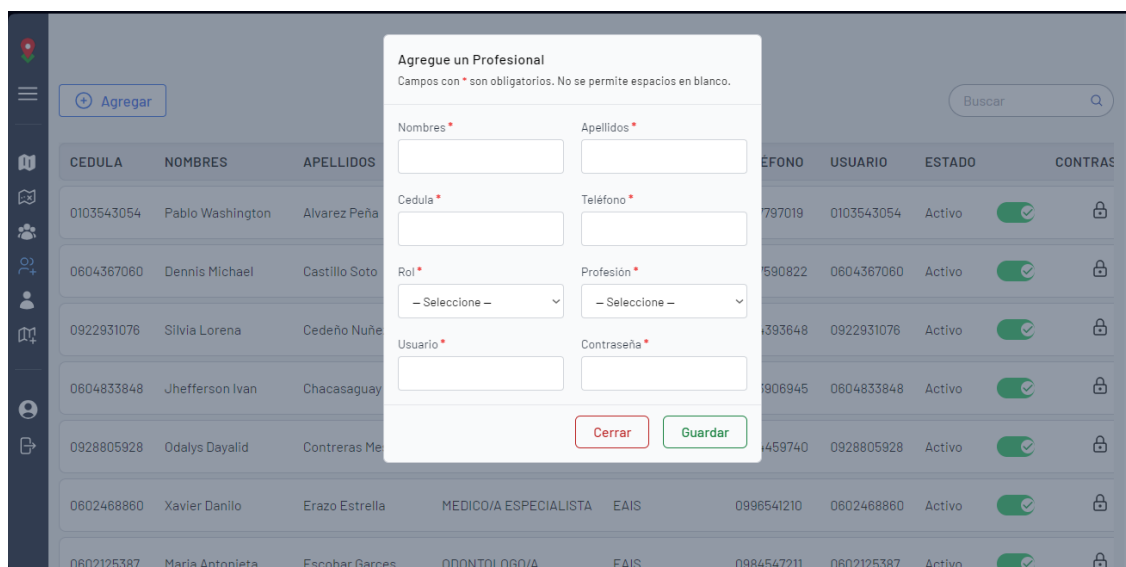


Figura 62. Módulo de gestión de profesionales

- **Módulo de gestión de Profesión**

Se ingresará las profesiones de los profesionales de la salud disponibles en Chambo, esto necesario para crear a nuevos usuarios dentro de la aplicación, véase la figura 63.

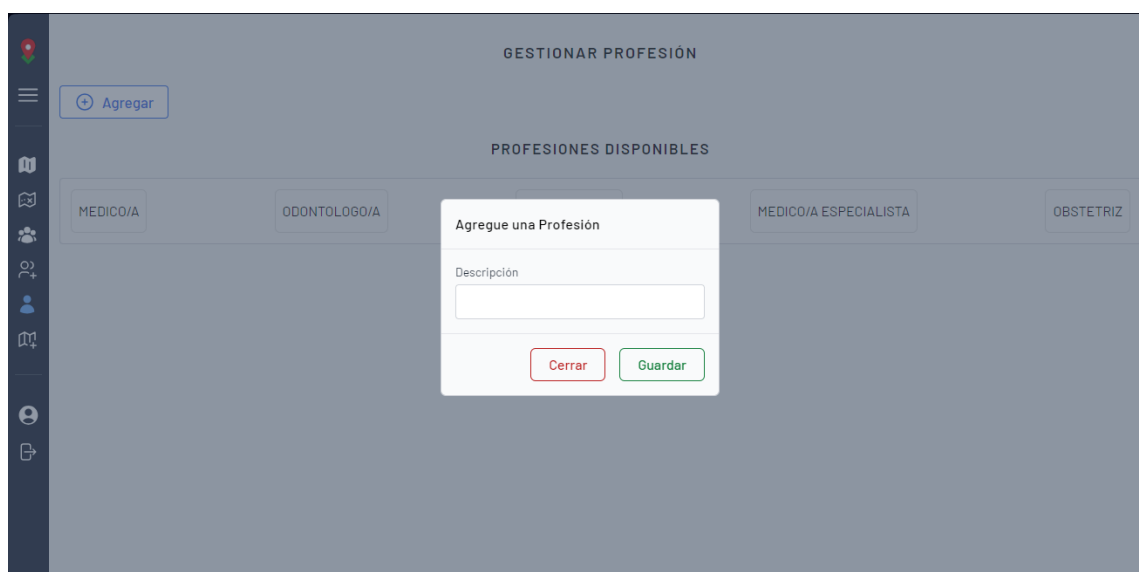


Figura 63. Módulo de gestión de profesión

Módulo de gestión de Información EAIS

Este módulo, tal como se muestra en la figura 64, ingresa la información relativa a la distribución del mapa de Chambo, representado mediante el código EAIS, es fundamental para la gestión de las brigadas EAIS y la orientación de los usuarios con brigadas activas hacia el mapa correspondiente.

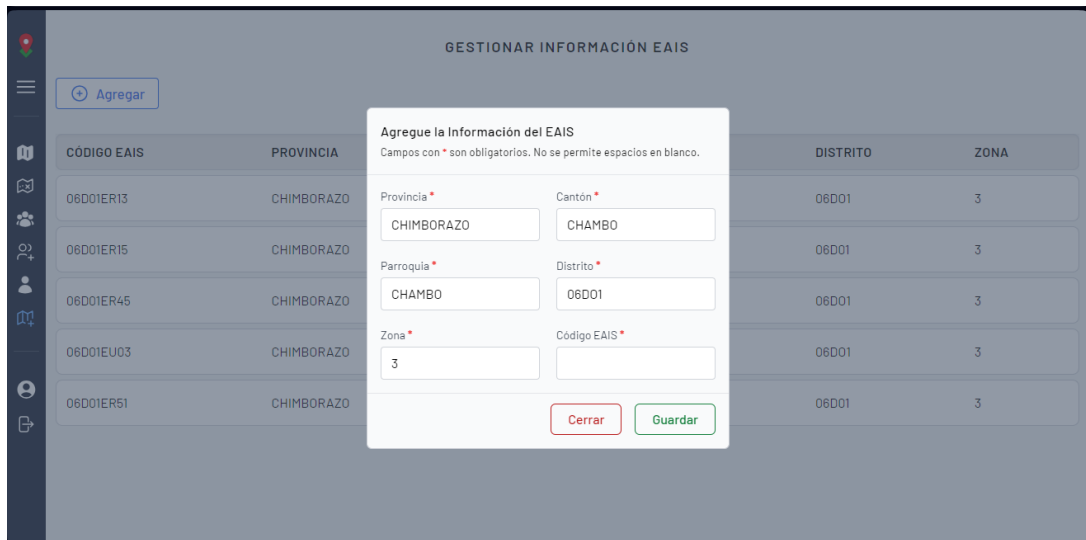


Figura 64. Módulo de gestión de información EAIS

Anexo 4. Cuestionario aplicado a los profesionales de la salud de Chambo para evaluar la usabilidad de la aplicación.

10/3/24, 22:34

Mapa Parlante

Mapa Parlante

El presente cuestionario tiene como objetivo medir la aceptación, satisfacción y usabilidad de la aplicación web mapa parlante implementada en el Subcentro de Salud de Chambo mediante cuatro criterios de usabilidad.

* Indica que la pregunta es obligatoria

Información del encuestado

Estimado/a participante,

Agradecemos sinceramente tu disposición para completar esta encuesta. La siguiente información es necesaria para validar el cuestionario y será tratada con absoluta confidencialidad.

1. Nombres y Apellidos *

2. Cédula *

3. Profesión o Cargo *

Marca solo un óvalo.

- Médico/a Especialista
- Médico/a
- Enfermero/a
- Odontólogo/a
- Estimulación Temprana
- Terapia Ocupacional
- Terapeuta Físico y Rehabilitador
- Psicología
- Obstetrix
- Nutrición
- Analista de Admisiones y Atención al Usuario
- Otro: _____

APRENDIZABILIDAD

Capacidad del producto que permite al usuario aprender su aplicación.

4. **¿Qué tan fácil le resultó aprender a utilizar la aplicación desde el principio? ***

Marca solo un óvalo.

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil

5. **¿Las funciones de la aplicación son intuitivas y fácil de entender ? ***

Marca solo un óvalo.

- Muy intuitiva
- Intuitiva
- Neutral
- Poco intuitiva
- Nada intuitiva

6. **¿Cómo evalúa la claridad, organización y navegabilidad de la aplicación para acceder a las diversas funciones y acciones que ofrece? ***

Marca solo un óvalo.

- Excelente
- Muy Bueno
- Bueno
- Regular
- Malo

OPERABILIDAD

Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

7. **¿Cómo calificaría la eficiencia de la aplicación en términos de velocidad y respuesta?** *

Marca solo un óvalo.

- Muy rápida
 Rápida
 Neutral
 Lenta
 Muy lenta

8. **¿Las funciones principales de la aplicación son fáciles de encontrar y acceder?** *

Marca solo un óvalo.

- Totalmente de acuerdo
 De acuerdo
 Neutral
 En desacuerdo
 Totalmente en desacuerdo

9. **¿Cómo calificaría la consistencia en la disposición y etiquetado de los elementos de la interfaz?** (Por ejemplo, botones, menús, opciones) *

Marca solo un óvalo.

- Excelente
 Muy buena
 Buena
 Regular
 Malo

PROTECCIÓN CONTRA ERRORES DE USUARIO

Capacidad del sistema para proteger a los usuarios de hacer errores.

10. **¿Los mensajes de error le ayudan a identificar que salió mal y como puedes solucionarlo?** *

Marca solo un óvalo.

- Si
 No

11. **¿La aplicación presenta mensajes de confirmación antes de realizar acciones que sean irreversibles o que puedan tener un impacto significativo?** *

Marca solo un óvalo.

- Totalmente de acuerdo
 De acuerdo
 Neutral
 En desacuerdo
 Totalmente en desacuerdo

12. **¿Considera que las notificaciones se presentan con claridad y son fácilmente comprensibles para el usuario?** *

Marca solo un óvalo.

- Totalmente de acuerdo
 De acuerdo
 Neutral
 En desacuerdo
 Totalmente en desacuerdo

ESTÉTICA DE LA INTERFAZ DE USUARIO

Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

13. **¿Encuentra visualmente atractiva la interfaz de usuario de la aplicación? ***

Marca solo un óvalo.

- Totalmente de acuerdo
 De acuerdo
 Neutral
 En desacuerdo
 Totalmente en desacuerdo

14. **¿La distribución de espacio en la aplicación le parece eficiente y bien organizada? ***

Marca solo un óvalo.

- Totalmente de acuerdo
 De acuerdo
 Neutral
 En desacuerdo
 Totalmente en desacuerdo

15. **¿Cómo evaluaría la utilización de colores, fuentes y elementos visuales en la interfaz de la aplicación en términos de su atractivo y coherencia? ***

Marca solo un óvalo.

- Excelente
 Muy Bueno
 Bueno
 Regular
 Malo

Comentarios adicionales

16. **En general, ¿Cómo calificaría su experiencia con la aplicación? ***

(1: Muy insatisfecho, 5: Muy satisfecho)

Marca solo un óvalo.

- 1 2 3 4 5
-

17. Recomendaría la implementación de esta aplicación en contextos similares *

Marca solo un óvalo.

- Definitivamente sí
- Probablemente sí
- Neutral
- Probablemente no
- Definitivamente no

18. ¿Existe algún comentario que te gustaría compartir sobre tu experiencia con la aplicación mapa parlante?

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios