



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

SISTEMA PARA LA DIGITALIZACIÓN DE PARQUES TURÍSTICOS POR MEDIO DE INFOMACIÓN GEOREFERENCIADA Y CÓDIGOS QR A TRAVÉS DE UNA APLICACIÓN WEB PROGRESIVA – GEOPARQUES

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:
LUIS VEAS CASTILLO

ANALISTA EN CIENCIAS COMPUTACIÓN CIENTÍFICA,
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN,
MAGISTER EN INGENIERÍA EN INFORMATICA,
MAGISTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN INFORMÁTICA
DOCTOR EN CIENCIAS DE LA INGENIERÍA, MENCIÓN INFORMÁTICA

PROFESOR CO-PATROCINANTE:
VALERIA HENRÍQUEZ NORAMBUENA
INGENIERO CIVIL EN INFORMÁTICA,
MAGISTER EN DIRECCIÓN DE MARKETING
DIRECTO Y DIGITAL,
DOCTORA EN SOFTWARE, SISTEMAS Y
COMPUTACIÓN

PROFESOR INFORMANTE:
MARIANA VILLARROEL MANFREDI
INGENIERO CIVIL EN INFORMÁTICA

BASTIÁN IGNACIO VILLANUEVA MEZA

VALDIVIA – CHILE
2024

AGRADECIMIENTOS

Nunca pensé que llegaría tan rápido el momento de escribir esto, ni yo me lo creo.

Primero que todo, quiero agradecer a mi familia, porque fueron ellos los que siempre creyeron en mí y estuvieron apoyándome en todos los momentos de mi vida y en todas mis metas e ideas locas. Agradecer a mi hermana, la más apañadora; a mi papá, el que siempre sabía que podía; a mi mamá, que siempre me dijo que yo podía, que me decía que estudiara más y que confiara en mis capacidades; y a la Normi, que siempre ha estado preocupada de mí desde que era muy pequeño y que desde siempre me ha visto delante de una pantalla de mi computador.

¿Cómo no agradecer a mis amigos del colegio? Fueron un pilar fundamental en este logro. Cómo no recordar las tardes y noches en Skype jugando y LOL, cuando salíamos a compartir algo para comer, una chela o lo que fuera, o las veces que nos fuimos a vacacionar a la casa de Lican Ray de Fonsi. Pucha que lo pasamos bien, esas ocasiones fueron una desconexión para mí y me ayudaron a recargar energía para seguir con mis estudios. Es por eso que agradezco esos momentos. Nada más que agradecer a Fonsi (Matías), al Negro (Cristian), al Tostado (Meza).

Por otra parte, mis amigos que hice en la universidad, uuuf, un pilar fundamental. Mi grupo de trabajo, mis compañeros de salidas a compartir unas chelitas o piscolas y esas idas a Curiñanco para sacarnos el semestre de encima. Quiero agradecer a Franco, a Diego, a McSebe (Seba), a Challa (Alonzo), el alma de la fiesta. Gracias chicos por todos los momentos y por ser un excelente grupo de trabajo. Se les quiere caleta.

Al final, agradecer al profe Luis Veas y a la profe Valeria Henríquez, que confiaron en mí para la realización de este pequeño proyecto y que siempre estuvieron ahí, dándome nuevas ideas, retroalimentando mi trabajo y corrigiéndome. En serio, lo valoré mucho. Además, agradecer a todos los profes que siempre creyeron en mí.

¡Gracias a Todos!

ÍNDICE

ÍNDICE.....	I
ÍNDICE DE TABLAS	V
ÍNDICE DE FIGURAS	VI
RESUMEN	IX
ABSTRACT	X
1. INTRODUCCIÓN.....	1
1.1 INNOVACIÓN.....	2
1.2 IMPACTO.....	2
1.3 MOTIVACIÓN	3
1.4 OBJETIVOS	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos	4
2. SITUACIÓN ACTUAL	5
2.1 IMPLEMENTACIÓN DE TECNOLOGÍAS EN MUSEOS Y PARQUES TURÍSTICOS.....	5
2.2 APLICACIONES TECNOLÓGICAS SIMILARES EN PARQUES TURÍSTICOS O JARDINES BOTÁNICOS.....	6
2.2.1 Museo Thyssen	6
2.2.2 Museo del Louvre	6
2.2.3 Jardín Botánico de Carlos Thays	6
2.2.4 El Jardín Botánico José Celestino Matius	7
2.2.5 App Guía Turística en Cantón la Maná	7
2.2.6 App Rutas Turísticas para la Comunidad Vereda San Pablo en Fusagasugá Colombia	7
2.2.7 Jardín Botánico UMCE	7
2.2.8 Jardín Botánico de la Universidad de Talca	8
2.2.9 Arbotag	8
2.3 TECNOLOGÍAS IMPLEMENTADAS.....	9
2.3.1 Aplicación Web Progresiva	9
2.3.1.1 Archivo de Configuración Manifiesto	10
2.3.1.2 Service Worker	10
2.3.2 Geolocalización Basada en Tecnología GPS.....	12
2.3.3 Códigos QR	13
3 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN.....	16
3.1 EL ORIGEN DE GEOPARQUES	16
3.2 REQUERIMIENTOS	18
3.2.7 Requisitos Funcionales Visitantes	19
3.2.2 Requisitos Funcionales Administrador.....	22
3.2.3 Requisitos No Funcionales	26

3.3 TECNOLOGÍAS PARA EL DESARROLLO.....	27
3.3.1 JavaScript.....	27
3.3.2 React.js	27
3.3.3 Node.js	27
3.3.4 Express.js	28
3.3.5 Tailwind CSS.....	28
3.3.6 MongoDB	28
3.4 ARQUITECTURA DE LA SOLUCIÓN	29
3.4.1 Diagrama De Contexto	30
3.4.2 Sistema GeoApi	31
3.4.4 Aplicación Geoparques Visitante	37
3.2.5 Aplicación GeoParques Administrador	42
3.2.6 Sistema Geoparques en su Totalidad	45
4. PLANIFICACIÓN E IMPLEMENTACIÓN DE LA SOLUCIÓN	47
4.1 PLANIFICACIÓN	47
4.2 METODOLOGÍA DE TRABAJO.....	47
4.2.1 Scrum.....	47
4.2.2 Tablero Kanban	50
4.2.3 Bitácora de Trabajo	50
4.3 IMPLEMENTACIÓN DE LA SOLUCIÓN	51
4.3.1 Desarrollo de la GeoApi	51
4.3.2 Desarrollo de la Aplicación del Administrador	51
4.3.2.1 Capa de presentación	51
4.3.2.1.1 Inicio de Sesión	52
4.3.2.1.2 Vista Mapa.....	53
4.3.2.1.3 Vista Visualizar y Agregar Árboles.....	53
4.3.2.1.4 Vista Ver y Administrar Ficha Especie	55
4.3.2.1.5 Vista Visualizar y Agregar Ficha Informativa	56
4.3.2.1.6 Panel de Administración de la Aplicación del Visitante	57
4.3.2.2 Capa Controladora.....	58
4.3.3 Desarrollo de la Aplicación del Visitante.....	59
4.3.3.1 Capa de Presentación	59
4.3.3.1.1 Vista de Bienvenida.....	59
4.3.3.1.2 Vista Mapa.....	60
4.3.3.1.3 Vista especie	62
4.3.3.1.4 Vista QR	63
4.3.3.2 Desarrollo del Service Worker para la Habilitación de la Aplicación como PWA	64
4.3.3.3 Desarrollo del Manifiesto para la Aplicación PWA	65
4.3.3.4 Capa Controladora.....	66
4.3.4 Desarrollo del Módulo Códigos QR	67
4.3.4.1 Implementación	67
4.3.4.2 Implementación en la Aplicación del Administrador.....	69
4.3.4.3 Implementación en la Aplicación del Visitante.....	71
4.3.5 Desarrollo del Módulo Geolocalización	74

4.3.5.1 Implementación de la Geolocalización a través del GPS de un Dispositivo Móvil	75
4.3.5.2 Función de Geoproximidad Implementada en la Aplicación	77
4.3.6 Desarrollo de la Internacionalización de la Aplicación	79
4.3.7 Despliegue	81
5. VERIFICACIÓN Y VALIDACIÓN.....	83
5.1 CIBERSEGURIDAD	83
5.1.1 Pruebas de Vulnerabilidad	83
5.1.1.1 Explicación Vulnerabilidad de Estado Crítico	85
5.1.1.2 Explicación Vulnerabilidad de Estado Alto	86
5.1.2 Recomendaciones para Vulnerabilidades no Críticas	86
5.2 PRUEBA DE RENDIMIENTO	86
5.2.1 Realización de las Pruebas	87
5.2.2 Ejecución de las Pruebas	88
5.2.3 Resultados.....	90
5.3 METODOLOGÍA DE VALIDACIÓN USABILIDAD	93
5.3.1 Descripción del Formulario	93
5.3.2 Pruebas Usabilidad de Pasillo.....	94
5.3.3 Pruebas de Usabilidad en Taller	94
5.3.4 Resultado de las Pruebas	95
5.3.4.1 Resultado Pregunta Sobre Conocimiento de una Aplicación PWA	95
5.3.4.2 Resultado Sobre las Actividades de la Vista Principal.....	96
5.3.4.3 Resultado sobre las Actividades de la Vista del Mapa	98
5.3.4.4 Resultado Sobre la Actividad de Activar Georreferenciación del Mapa....	101
5.3.4.5 Resultado Sobre la Utilización de los Código QR	103
5.3.4.6 Resultado Sobre la Utilización de Vista Especie.....	106
5.3.4.7 Retroalimentación Final de los Encuestados	108
5.3.4.8 Resultados de Usabilidad Utilizando System Usability Scale.....	108
6. CONCLUSIONES Y TRABAJO FUTURO	112
6.1 ESTADO ACTUAL	112
6.2 CUMPLIMIENTO DE LOS OBJETIVOS.....	112
6.2.1 Diseñar y Desarrollar una Aplicación Utilizando la Metodología de Desarrollo Aplicación Web Progresiva (PWA)	112
6.2.2 Integrar un Sistema de Geolocalización que Permita Rastrear la Ubicación en Tiempo Real del Visitante y Proporcionar Información Mediante su Posición.	113
6.2.3 Establecer un Sistema de Código QR: Diseñar y Desplegar Códigos QR en Ubicados Estratégicas Dentro del Parque, Vinculadas a las Fichas Especies Determinadas por los Administradores.	114
6.2.4 Llevar el Software a un Ambiente de Producción con el Propósito de Ponerlo a Disposición de los Usuarios Finales.	115
6.2.5 Llevar a Cabo Prueba de Validación de Funcionalidades de la Aplicación (PWA), Asegurando que se Cumpla con los Requisitos y Expectativas Planificadas	115
6.3 TRABAJO FUTURO.....	116
7. REFERENCIAS.....	118

ANEXOS.....	122
ANEXO A: BASE DE DATOS QUE SE REALIZÓ EN CONJUNTO CON LOS INTEGRANTES DE LOS DISTINTOS SUBPROYECTOS	122
ANEXO B: DESCRIPCIÓN DE LA COLECCIÓN TREECARDS (FICHA ESPECIE)	122
ANEXO C: DESCRIPCIÓN DE LA COLECCIÓN INGRESO PERSONA.....	125
ANEXO D: PRIMERA CARTA GANTT	127
ANEXO E: USER HISTORY MAPING DE LA APLICACIÓN ADMINISTRADOR.....	128
ANEXO F: TABLERO KANBAN.....	129
ANEXO G: BITÁCORA	131
ANEXO H: RUTAS (ENDPOINT) DE INFORMACIÓN DE LA GEOAPI.....	131
ANEXO I: RUTAS DE LA CAPA CONTROLADORA DE LA APLICACIÓN DEL ADMINISTRADOR	133
ANEXO J: MANUAL DE USO APLICACIÓN GEOPARQUES VISITANTES	134
ANEXO K: JSON REAL DE LA CONFIGURACIÓN DEL MANIFIESTO	135
ANEXO L: RUTAS (ENDPOINT) CAPA CONTROLADORA DE LA APLICACIÓN VISITANTE.....	136
ANEXO M: JSON CON LAS VARIABLES EN IDIOMA INGLÉS	139
ANEXO N: JSON REAL DE LA CONFIGURACIÓN DEL MANIFIESTO	140
ANEXO O: CARTA GANTT ACTUALIZADA	141
ANEXO P: DOCUMENTACIÓN DE DESPLIEGUE GEOAPI	141
ANEXO Q: DOCUMENTACIÓN DE DESPLIEGUE APLICACIÓN GEOPARQUES -VISITANTE...	141
ANEXO R: DOCUMENTACIÓN DE DESPLIEGUE APLICACIÓN GEOPARQUES – ADMINISTRADOR	142
ANEXO S: TABLA DE LAS DISTINTAS ALERTAS DE SEGURIDAD	142
ANEXO T: INFORME DE CIBERSEGURIDAD GENERADO POR ZAP PARA GEOPARQUES.	142
ANEXO U: FORMULARIO PARA VALIDACIÓN DE USABILIDAD	142

ÍNDICE DE TABLAS

Tabla	Página
Tabla 1 : Comparativa de tecnologías por proyecto	9
Tabla 2 : las necesidades de los usuarios que quieren para interactuar con el sistema	18
Tabla 3 : Requisitos, vista de Bienvenida.....	19
Tabla 4 : Requisitos, vista del mapa	20
Tabla 5 : Requisitos, vista todas las especies	20
Tabla 6 : Requisitos, vista código QR	21
Tabla 7 : Requisitos, vista de Participación de los visitantes y mejora continua en la experiencia de GeoParques.....	21
Tabla 8 : Requisitos, vista mapa general	22
Tabla 9 : Requisitos, vista Árbol	22
Tabla 10 : Requisitos, vista de fichas de árboles	23
Tabla 11 : Requisitos, vista de Fichas informativa.....	24
Tabla 12 : Requisito, vista de panel de configuración para administrar la aplicación de los visitantes del parques	25
Tabla 13 : Requisitos no funcionales.....	26
Tabla 14 : Componentes y buenas prácticas de metodología de trabajo Scrum.....	48
Tabla 15 : Extracto rutas de la capa controladora GeoApi.....	51
Tabla 16 : Extracto de las rutas de la capa controladora aplicación administradores	58
Tabla 17 : Variables del manifiesto que fueron configuradas con su descripción	66
Tabla 18 : Extracto de las rutas de la capa controladora aplicación visitantes	67
Tabla 19 : Parámetros de los grupos de hilo.....	89
Tabla 20 : Grados para puntaje SUS	111
Tabla 21 : descripción de los atributos de colección treeCards	123
Tabla 22 : Descripción de los atributos de la colección ingresoPersona.....	125
Tabla 23 : Rutas de información de la GeoApi	131
Tabla 24 : Rutas de información de la capa controladora administrador	133
Tabla 25 : Rutas de información de la capa controladora visitante	136
Tabla 26 : tipos de alertas de vulnerabilidades	142

ÍNDICE DE FIGURAS

Figura	Página
Figura 1 : Forma en que interactúa el Servicio Worker con la aplicación y servidor.....	11
Figura 2 : Trilateración técnica para captar la posición de un usuario	13
Figura 3 : Código QR posicionado en letrero dentro de un parque	14
Figura 4 : Elementos básicos de un código QR.....	15
Figura 5 : Plataforma GeoParques y sus distintas funcionalidades	16
Figura 6 : Diagrama de contexto del proyecto en general	17
Figura 7 : Actores interactuando con aplicaciones de GeoParques	18
Figura 8 : Ejemplo de dato georreferenciado utilizando la estructura geometry	29
Figura 9 : Forma de representar los componentes en un diagrama C4.....	30
Figura 10 : Diagrama de contexto del sistema de Geoparques.....	31
Figura 11 : Diagrama de contenedores arquitectura GeoApi	32
Figura 12 : Componentes de la capa controladora	34
Figura 13 : Modulo de Árboles y su persistencia	35
Figura 14 : Base de datos Calafate perteneciente del sistema GeoApi	36
Figura 15 : Arquitectura de la aplicación Geoparques – Visitantes	38
Figura 16 : Capa controladora de la aplicación Geoparques-Visitante	39
Figura 17 : Base de datos Torobayo perteneciente al sistema Geoparques – Visitante	40
Figura 18 : Arquitectura de la Aplicación Geoparques–Administrador	42
Figura 19 : Capa controladoras de Geoparques-Administrador y sus distintos componentes	44
Figura 20 : Colección de la base de datos Geoparques – Administrador	45
Figura 21 : Sistema GeoParques en su totalidad	46
Figura 22 : User story mapping del usuario visitante	49
Figura 23 : Vista de Inicio de sesión	52
Figura 24 : Vista del Mapa	53
Figura 25 : Vista añadir y actualizar árboles	55
Figura 26 : Vista para administrar ficha especie	56
Figura 27 : Vista para administrar fichas informativas	57
Figura 28 : Configuración app visitante, panel de bienvenida	58
Figura 29 : Vista de inicio en la aplicación del visitante	60
Figura 30 : Vista mapa.....	61
Figura 31 : Drawer con opciones para interactuar con el mapa	62
Figura 32 : Vista de todas las especies en el parque.....	63
Figura 33 : Vista QR.....	64
Figura 34 : Script de la configuración Service Worker	65
Figura 35 : QR almacenando la ID de la ficha especie	68
Figura 36 : QR, almacenando la URL de la aplicación además de la ID de la ficha especie	69
Figura 37 : Código de generar QR.....	70
Figura 38 : Modal para la visualización del código QR	71
Figura 39 : Código del ModalQR	72

Figura 40 : Fragmento de código del componente PaginaMapa	73
Figura 41 : Apariencia de la interfaz al momento de escanear un código QR.	73
Figura 42 : Diagrama de secuencia escaneo de un código QR.....	74
Figura 43 : Activación de la localización del visitante	75
Figura 44 : Componente para captar la posición del visitante y visualizarlo en el mapa.....	76
Figura 45 : Componentes para la función de Geoproximidad.....	77
Figura 46 : Componente encargado de filtrar los árboles mediante un radio establecido	78
Figura 47 : Diagrama de secuencia de la función de Geoproximidad	79
Figura 48 : Colección de ejemplo.....	80
Figura 49 : Ejemplo de endpoint para obtener información mediante el idioma seleccionado	80
Figura 50 : Diagrama de despliegue del sistema GeoParques	81
Figura 51 : Configuración para escaneo automático de vulnerabilidades en ZAP	84
Figura 52 : Resultado del escáner realizado, en donde se muestra las vulnerabilidades encontradas	85
Figura 53 : Evidencia de vulnerabilidad critica parchada	85
Figura 54 : Evidencia de vulnerabilidad alta parchada.....	86
Figura 55 : Prueba del endpoint para obtener todo los árboles	88
Figura 56 : Configuración de la petición HTTPS	89
Figura 57 : Gráfico de hilos activo a lo largo del tiempo	90
Figura 58 : Gráfico de transacciones a lo largo del tiempo	91
Figura 59 : Gráfico de tiempo de respuesta a lo largo del tiempo.....	92
Figura 60 : Extracto del monitoreo del servidor de la GeoApi	92
Figura 61 : Formato de la escala de Likert utilizado en la preguntas SUS	94
Figura 62 : Gráfico de barra de las respuestas sobre el conocimiento de las PWA	96
Figura 63 : Respuesta de la pregunta de tarjeta de información del parque	97
Figura 64 : Respuestas sobre la actividad de cambiar el idioma	98
Figura 65 : Respuesta de la pregunta como ver la información de un árbol	99
Figura 66 : Respuesta a la pregunta si le resulta útil la información de la ficha especie.	99
Figura 67 : Respuesta a la pregunta sobre la facilidad o dificultad de reportar una especie	100
Figura 68 : Respuestas sobre la pregunta de activar georreferenciación	101
Figura 69 : Respuesta sobre la activación de la Geoproximidad.....	102
Figura 70 : Resultado de las actividades para activar localización y Geoproximidad	103
Figura 71 : Resultado de la actividad de escanear un código QR	104
Figura 72 : Resultado de la pregunta de por qué medio se escanea el código QR	105
Figura 73 : Resultado de las actividades para escanear códigos QR	105
Figura 74 : Resultado sobre encontrar la opción de explorar las especies	107
Figura 75 : Resultado sobre visualizar en el mapa una especie.....	107
Figura 76 : Resultados de preguntas SUS en prueba de pasillo	109
Figura 77 : Resultados de preguntas SUS en pruebas del taller	110
Figura 78 : Código QR desplegado en el jardín botánico.....	114
Figura 79 : Base de datos construida por todo el equipo.....	122
Figura 80 : Primera carta Gantt	127
Figura 81 : User history maping de la aplicación administrador.....	128
Figura 82 : Tablero Kanban	129

Figura 83 : Configuración manifiesto.....	135
Figura 84 : JSON con las variables en ingles	139
Figura 85 : JSON del Manifiesto.....	140
Figura 86 : Carta Gantt actualizada	141

RESUMEN

En la actualidad, al visitar parques turísticos o jardines botánicos, es común encontrar información sobre las especies mediante señaléticas como tótems, letreros físicos y fichas. Sin embargo, estos elementos pueden deteriorarse, dañarse o ser vandalizados. Además, resulta inviable colocar fichas informativas para todas las especies presentes en un parque. Geoparques surge de la necesidad de digitalizar la información de los parques turísticos y ofrecerla a través de una aplicación multiplataforma que no requiere descarga para su uso.

La información se presenta mediante hitos informativos georreferenciados, visibles en un mapa. Un hito destacado es la ficha de especie, que contiene información detallada de las especies ubicadas en el parque. Para acceder a la información, el usuario puede activar la localización de su dispositivo móvil, lo que le permite ver su posición en el parque y descubrir las especies cercanas mediante la función de GeoProximidad. Además, se puede acceder a la información mediante códigos QR estratégicamente ubicados, lo cual sirve tanto para informar al visitante como para promover la aplicación.

GeoParques también cuenta con una aplicación diseñada exclusivamente para los administradores de parques turísticos. En esta, los administradores pueden gestionar la información que ven los visitantes, leer errores reportados y sugerencias de mejoras propuesta por los visitante.

Para la construcción de esta plataforma se planificó y tomaron requisitos, se desarrolló una arquitectura de software adaptada a las demandas y necesidades de ambas aplicaciones. Las aplicaciones se desarrollaron desde cero y se llevaron a producción. Este proceso quedó registrado en este documento.

Finalmente, se realizaron pruebas de usabilidad con usuarios finales para validar la efectividad del sistema y asegurar que cumpliera con los estándares mínimos para operar en producción. Estas pruebas reflejaron cómo los usuarios interactuaba con la aplicación y ayudaron a identificar errores en la aplicación del visitante que fueron corregidos. Además, demostraron que la aplicación tiene una excelente usabilidad según las respuestas *al System Usability Scale*.

ABSTRACT

Currently, when visiting tourist parks or botanical gardens, it is common to find information about species through signage such as totems, physical signs, and plaques. However, these elements can deteriorate, become damaged, or be vandalized. Additionally, it is unfeasible to provide informational plaques for all the species present in a park.

GeoParques arises from the need to digitize the information of tourist parks and offer it through a multiplatform application that does not require downloading for use. The information is presented through georeferenced informational milestones, visible on a map. A notable milestone is the species card, which contains detailed information about the species located in the park. To access the information, the user can enable their mobile device's location, allowing them to see their position in the park and discover nearby species through the GeoProximidad, function. Additionally, information can be accessed via strategically placed QR codes, serving both to inform the visitor and promote the application.

GeoParques also features an application designed exclusively for park administrators. In this application, administrators can manage the information that visitors see, read reported errors, and review suggestions for improvements proposed by visitors.

For the construction of this platform, planning and requirement gathering were conducted, and a software architecture was developed to meet the demands and needs of both applications. The applications were developed from scratch and deployed to production. This process is documented in this report.

Finally, usability tests were conducted with end-users to validate the system's effectiveness and ensure it met the minimum standards for operation in production. These tests showed how users interacted with the application and helped identify errors in the visitor application that were corrected. Furthermore, they demonstrated that the application has excellent usability according to the System Usability Scale responses.

1. INTRODUCCIÓN

Valdivia se caracteriza por ser una ciudad turística. En enero del año 2020, se destacó por tener una alta ocupación en alojamiento turístico a nivel nacional, con un porcentaje de ocupación del 91,1% (Sernatur, s. f.). Esto podría deberse a la gran cantidad de destinos y panoramas con naturaleza que esta ciudad puede ofrecer. Además, Valdivia cuenta con una amplia variedad de parques para visitar, adentrarse en la naturaleza, disfrutarla y aprender de ella. Sin embargo, cuando se visitan parques turísticos, aún se utilizan métodos tradicionales para entregar información y guiar al turista, como mapas, trípticos, tótems con información, etc. A veces, esta información no está actualizada por diversos motivos, tales como que los carteles en el parque se encuentran dañados, la situación climática ha causado estragos, o simplemente el parque no cuenta con los recursos necesarios para mantener esta información al día. Esta situación genera problemáticas para los visitantes, como la posibilidad de perderse dentro del parque o quedar con dudas sobre las distintas especies de flora presentes.

Actualmente, existen aplicaciones nativas para dispositivos móviles que, mediante la geolocalización, permiten posicionarnos en un mapa y orientarnos para evitar perdernos. Sin embargo, estas aplicaciones no se centran en el ámbito turístico ni en el aprendizaje sobre la naturaleza. Ejemplos de estas aplicaciones incluyen Google Maps, Strava, Nike Run Club. Aunque estas aplicaciones cumplen con su función de orientación mediante tecnología GPS, no proporcionan el valor adicional que sí ofrecería una aplicación diseñada específicamente para la presentación de información al turista y su posicionamiento dentro parque.

Otro aspecto relevante es que muchas de estas aplicaciones necesitan ser descargadas e instaladas, lo que implica el consumo de recursos computacionales, incluyendo capacidad de procesamiento y espacio de almacenamiento. Además, las personas pueden no estar interesadas en descargar una aplicación en sus dispositivos móviles si solo la van a usar en contadas ocasiones. Por lo tanto, desarrollar una aplicación web que no requiera descarga en ningún dispositivo se vuelve de gran importancia para su acceso rápido. Para lograr esto, se puede desarrollar una aplicación multiplataforma que se pueda ejecutar en todos los dispositivos que cuenten con un navegador web. Esto también presenta desafíos, como asegurar una compatibilidad total de los dispositivos actuales, incluyendo PCs, tablets y Smartphones.

Para dar solución a la problemática de acceso a la información en los parques turísticos, el profesor patrocinante de este proyecto de título generó un proyecto denominado **GeoParques**, que buscaba abordar las problemáticas previamente mencionadas y otras más. Además, se aspira a que los administradores del parque pudieran gestionar la información que visualizarán los visitantes en los destinos turísticos. El proyecto se presentó al equipo del Jardín Botánico de la Universidad Austral de Chile, con el objetivo de digitalizar la información que actualmente se presenta en el jardín mediante diversos letreros en los árboles y tótems físicos, haciéndola accesible en una plataforma web que

ambos tipos de usuarios, visitantes y administradores del parque, pudieran visualizar y gestionar.

El Jardín Botánico accedió a la propuesta, ya que está en búsqueda de implementar nuevas tecnologías para ofrecer una experiencia satisfactoria a sus visitantes. Este interés se debe a un incremento en el número de visitantes al jardín botánico, donde en 2016 se registró una cifra de 23.529, que se incrementó a 58.248 en 2019, según datos de la comunidad de la Universidad Austral de Chile (Comunicaciones UACH, 2020). Además, en 2020, se emprendió un proyecto denominado 'Rescate de la historia del Jardín Botánico de la Universidad Austral de Chile, desde una perspectiva histórica y contemporánea', según una nota de prensa publicada por la universidad. La iniciativa busca recuperar el legado del Jardín Botánico, evidenciando su relevancia cultural y simbolismo como reflejo de la belleza y naturaleza de Valdivia.

1.1 Innovación

Para dar origen a este proyecto, la misión es ofrecer un valor agregado tanto a los visitantes del parque como a sus administradores. Por ello, se planeó el desarrollo de una aplicación web progresiva, la cual se puede utilizar como multiplataforma. Esta aplicación está diseñada para brindar información detallada a los visitantes sobre la diversa flora presente en el parque. Para lograrlo, se mostrarán fichas de árboles que contendrán información detallada sobre cada especie perteneciente al parque. Esta información estará accesible a través de códigos QR ubicados en distintos puntos estratégicos del parque y en los propios árboles.

Además, se integrará un sistema de proximidad basado en la tecnología de geolocalización que aprovecha el sensor GPS de los dispositivos móviles. Esta implementación permite al visitante recibir información contextual en función de su ubicación, incluyendo detalles sobre su posición actual. Si se acerca a una especie importante dentro del parque, podrá visualizarla en la aplicación, ya que todas las especies están georreferenciadas.

Para aprovechar el uso de la geolocalización, se introducirán distintos tipos de marcadores, a los cuales se ha denominado “hitos de información”. Entre estos, destacan los tótems digitales, que ofrecen datos sobre la historia del lugar. Otro hito informativo son los miradores, los cuales brindan información sobre diversos lugares de interés dentro del parque, además de mostrar imágenes de diferentes épocas de dicho mirador. Esta estrategia sumergirá al visitante en una experiencia tanto educativa como entretenida durante su recorrido y estancia en el parque.

1.2 Impacto

Se espera que este trabajo de título mejore la experiencia de cómo los visitantes de parques turísticos acceden a información dentro de estos, a través de la digitalización de la información. Además, se busca que los visitantes aprendan sobre las distintas especies presentes en el jardín botánico de la UACH, incrementando así su conocimiento sobre la

flora existente en la región. Por otro lado, la aplicación (PWA) podrá dar acceso a esta información de forma rápida sin la necesidad de descargar una aplicación en el dispositivo móvil del visitante así mejorando la accesibilidad de esta, lo que ayudará a que el usuario no rechace el uso de la aplicación.

Por parte de los administradores de parques, se espera mejorar la forma en que gestionan la información dentro del parque, permitiéndoles actualizar la información de la aplicación de manera fácil. Además, se anticipa una reducción de costos, ya que actualmente se utilizan tótems físicos donde se muestra información relevante de los parques, y la creación de estos es costosa. Sin embargo, gracias a la aplicación, se busca que cada vez se instalen menos de estos tótems hasta que se eliminen por completo, lo que podría resultar en una reducción de costos considerable.

Finalmente, se espera que esta aplicación genere suficiente datos para entregar información valiosa a los administradores del parque, como qué especies captan más la atención de los visitantes o cuáles son los horarios de mayor actividad en la aplicación. Esta información podría ser utilizada por los administradores para tomar decisiones encaminadas a mejorar la experiencia de los visitantes.

1.3 Motivación

Debido a la gran afluencia de visitantes que ha experimentado la ciudad de Valdivia, así como a la rica diversidad de especies de flora que alberga, es crucial disponer de información digitalizada de estas especies para facilitar su actualización de la manera más sencilla posible. Sin embargo, existe un problema significativo al momento de querer actualizar esta información, debido, principalmente a que los administradores de parques no disponen de sistemas para realizar actualizaciones de esta información. Por esta razón, cuando necesitan realizar alguna modificación en sus aplicaciones o páginas web, requieren la asistencia especializada de un informático, ya que algunas de estas modificaciones deben realizarse mediante código o a través de plataformas específicas, como lo por ejemplo un WordPress¹ o una aplicación construida solo con código que el administrador no entiende.

Presentar esta información de manera accesible a los visitantes es esencial, ya que les permite aprender y descubrir nuevas plantas, árboles, historias y explorar lugares culturales e históricos que los parques ofrecen. Esto enriquece la experiencia del visitante y fomenta una mayor apreciación por la biodiversidad y el patrimonio cultural de Valdivia. La solución a este desafío implica desarrollar un sistema que empodere a los administradores de parques con las herramientas necesarias para gestionar y actualizar la información de manera autónoma, garantizando así que los visitantes tengan acceso a datos enriquecedores de las distintas especies durante su exploración de los parques.

¹ es un sistema de gestión de contenido (CMS) de código abierto, utilizado principalmente para la creación de sitios web y blogs debido a su facilidad de uso y flexibilidad .

1.4 Objetivos

1.4.1 Objetivo General

Enriquecer con tecnología la experiencia de los visitantes de parques turísticos por medio de una aplicación multiplataforma (PWA), la cual entregue la información mediante códigos QR y un sistema de proximidad basado en la geolocalización.

1.4.2 Objetivos Específicos

- Diseñar y desarrollar una aplicación utilizando la metodología de desarrollo Aplicación Web Progresiva (PWA) que permita a los visitante acceder a información detallada sobre biodiversidad de las distintas especies de flora del parque sin requerir descargas adicionales.
- Integrar un sistema de geolocalización basado en dispositivos móviles que permita rastrear la ubicación de los visitantes en tiempo real dentro del parque y proporcionar información relevante basada en su posición.
- Establecer un sistema de códigos QR: Diseñar y desplegar códigos QR en ubicaciones estratégicas dentro del parque, vinculados a las fichas especies e información de relevancia determinada por los administradores.
- Llevar a cabo prueba de validación de funcionalidades de la aplicación (PWA), asegurando que se cumpla con los requisitos y expectativa planificadas.
- Llevar el software a un ambiente de producción con el propósito de ponerlo a disposición de los usuarios finales.

2. SITUACIÓN ACTUAL

Para proporcionar una explicación clara de este trabajo de título, este capítulo se divide en las subsecciones 2.1 a 2.3. Se explorarán los sistemas existentes que enriquecen a los visitantes con información relevante. Además, se analizará si estas funciones mejoran la experiencia de los visitantes y se determinará cuántas de estas opciones emplean metodologías similares a las propuestas en este trabajo para proporcionar la información a los visitantes de parques turísticos.

2.1 Implementación de Tecnologías en Museos y Parques Turísticos.

Las personas están acostumbradas a integrar la tecnología en todos los aspectos de su vida cotidiana. Lo acompaña constantemente su dispositivo móvil, el cual utiliza para realizar tareas que anteriormente no se asociaban con un teléfono, como, por ejemplo, hacer las compras del supermercado con solo un clic (Puromarketing, s.f.). Debido a esto, actualmente están acostumbrados a que incluso las tareas más básicas se puedan ejecutar mediante el teléfono. Las personas que poseen un dispositivo móvil, suelen mirarlo un promedio de hasta 150 veces al día y un 73% admite que siente pánico si no siente el móvil cerca o cree que lo ha perdido (Universal, s.f.).

Por esta razón, organizaciones, empresas e instituciones han tenido que adaptar sus servicios, productos o experiencias para que sean accesibles a través de tecnología móvil (ABAMobile, s.f.). Esta adaptación responde a la necesidad de seguir integrando tecnologías en lo que se ofrece, con el fin de que las organizaciones no se queden atrás y puedan satisfacer las expectativas del usuario al que están dirigidas.

Los museos y parques turísticos se han preocupado en que sus visitantes se sientan lo más cómodos posible durante sus visitas y que puedan aprender sobre las diferentes obras o especies que se encuentran en exposición (Euemuseografia, 2021). Es por esto que los administradores buscan constantemente nuevas formas innovadoras de brindar una experiencia única a sus visitantes. Debido a esto, no quisieron quedarse atrás con la implementación de nuevas tecnologías, en donde puedan ofrecer maneras comprensivas y atractivas de brindar información rápida y accesible, con el fin de poder entender el patrimonio que esos lugares ofrecen.

A partir de comienzos del S.XIX los museos comenzaron a utilizar internet y a crear sus propias páginas webs para informar sobre sus exposiciones, horarios, tarifas, forma de llegar y localización (Fernández & González, 2019). No pasaron muchos años hasta que llegó la nueva generación de la tecnología en donde se centra todo en el teléfono móvil, las redes sociales, las páginas multimedia y las apps. Además de tecnologías como pantallas táctiles, reconocimiento fácil, códigos QR, sensores (internet de las cosas) y hasta robots, cada día tiene más peso y grado en nuestra sociedad.

Chile no se ha quedado atrás con la implementación de tecnologías en sus museos y parques turísticos. Debido a la pandemia de COVID-19 que afectó al mundo entero, los

museos de Chile tuvieron que acelerar la implementación de tecnología y de nuevas formas de presentar la información a sus visitantes, quienes ya no podían realizar visitas presenciales (patrimoniocultural.gob, s. f.), es por ello que, en ese instante, los directores de museos de Chile se reunieron con el fin de conversar sobre la implementación de nuevas tecnologías en el patrimonio cultural. **Sergio Muños**, quien es el director del **Museo de Historia Natural**, destacó la experiencia que han tenido con la tecnología desde que se encomendó a realizar una nueva museografía con colecciones 3D. por otro lado **Ximena Pezoa** directora del **Museo de Artes Decorativas** se sumó a la idea que la tecnología virtual es una herramienta que viene a complementar y no reemplazar lo presencial.

2.2 Aplicaciones Tecnológicas Similares en Parques Turísticos o Jardines Botánicos

A continuación, se presentan distintas soluciones propuestas por entidades en las cuales se implementaron diversas tecnologías. Primero se examinarán las propuestas internacionales y, posteriormente, las nacionales.

2.2.1 Museo Thyssen

El Museo Thyssen, es un museo español que ofrece a sus usuarios un sistema avanzado de audioguías conocido como smartguide, que es una especie de archivo multimedia que permite hacer la visita mucho más completa. Se adquiere en los mostradores del museo para alquilarla durante la visita. Esta smartguide acopla las obras con música teniendo en cuenta su contexto, ofrece audios, videos, también la opinión de expertos, información sobre los autores, geolocalización planos de todas las salas y complementado con códigos QR posee reconocimiento visual, siendo solo necesario apuntar la guía al cuadro para que esta reconozca y visualice toda la información sobre ella (Fernández & González, 2019).

2.2.2 Museo del Louvre

El museo del Louvre fue unos de los primeros en introducir la tecnología en sus galerías, Siendo en la actualidad el museo más visitado a nivel internacional y el primero en permitir ver sus obras desde cualquier parte del mundo con solo poseer un dispositivo móvil, Tablet, u computador. El año 2019, se inició un proyecto el cual buscaba renovar su estrategia digital, con el objetivo enriquecer la experiencia de los visitantes antes, durante y después de la visita al Louvre. Además de desarrollar herramientas para ayudar a los empleados a interactuar con los visitantes y a realizar su trabajo de forma más eficiente (Díaz, s. f.).

2.2.3 Jardín Botánico de Carlos Thays

El ministerio de ambiente y espacios públicos de Argentina ha implementado un proyecto de señaléticas en el jardín botánico de Carlos Thays de la ciudad de Buenos Aires. Así mejorando la experiencia de los visitantes mediante la inclusión de planos y direcciones,

información sobre los lugares del parque. Este proyecto se enriquece con la adopción de códigos QR, una tecnología que ofrece información detallada a través de dispositivos móviles, los QR se utilizan para expandir información sobre las especies y áreas temáticas en el jardín, vinculando el entorno digital con el físico (buenosaires.gob, s. f.).

2.2.4 El Jardín Botánico José Celestino Matius

El jardín Botánico de Bogota Jose Celestino Matius es una institución dedicada a la investigación y conservación de la flora, especialmente de la región andina. En donde ha mejorado significativamente un novedoso sistema de sistematización, georreferenciación y señalización de plantas. Esto incluye una base de datos avanzada y una aplicación, resultando en un inventario actualizado que cuenta con 19.546 acciones georreferenciadas. Este inventario comprende 184 familias, 723 géneros, y 1186 especies incluyendo 89 especies en categoría de amenaza a nivel nacional (Cadena, Sánchez, Velásquez, 2021).

2.2.5 App Guía Turística en Cantón la Maná

En Cantón la Maná en Ecuador, una tesista presentó un proyecto para obtener su título como Ingeniera Civil en informática. En donde el objetivo era poder mejorar la experiencia de los turistas tanto nacionales como internacionales, ya que existía una ineficacia de las guías y mapas tradicionales que a menudo estaban desactualizados. La propuesta consiste en el desarrollo e implementación de un sistema de geolocalización y una aplicación móvil específicas para los turistas del Cantón La Maná. Esta Aplicación permitirá a los usuarios identificar y acceder fácilmente a las diversas rutas turísticas mediante los dispositivos móviles (Lisbeth, 2020).

2.2.6 App Rutas Turísticas para la Comunidad Vereda San Pablo en Fusagasugá Colombia.

Juan Ladino, en su proyecto para obtener el título de Ingeniería de Sistema, presentó un proyecto con el objetivo de desarrollar una aplicación digital adaptada a las necesidades de la comunidad de la vereda San Pablo en Fusagasugá, Colombia. Facilitando el acceso a diversos puntos ecológicos a lo largo de rutas turísticas. La aplicación se centra en la geolocalización en tiempo real de los turistas, proporcionando contenido multimedia e información sobre cada punto de interés (Ladino, 2022).

2.2.7 Jardín Botánico UMCE

La Universidad Metropolitana de Ciencias de la Educación (UMCE), como parte de un proyecto nombrado “Plataforma Interactiva GPS/QR/Web para el jardín botánico UMCE” buscaba introducir una tecnología de enseñanza innovadora que utiliza códigos QR para acceder a información sobre especies nativas “en terreno”. Cristian Villagra, quien es el coordinador del proyecto, destacó el valor de las herramientas tecnologías como complemento a la experiencia educativa directa, convirtiendo los teléfonos móviles, a menudo vistos como distractores, en herramientas pedagógicas valiosas (Acuña, s. f.).

2.2.8 Jardín Botánico de la Universidad de Talca

Por otra parte, el Jardín Botánico de la Universidad de Talca ha introducido una aplicación de realidad aumentada, llamada, Jardín Botánico Tours & Walks, con el fin de educar a escolares de la región del Maule sobre su diversa flora y fauna. Esta aplicación, disponible gratuitamente en Google Play y APP Store, ofrece información interactiva sobre las especies al enfocar con sus teléfonos los códigos QR dispersos por el jardín (Honour, 2016).

Por último, se revisa una solución que no necesariamente entrega valor a los visitantes de parques o museos, pero sí a los administradores de los distintos parques municipales. A continuación, se presenta la siguiente solución para saber el estado de salud de los árboles que cuenta los parques

2.2.9 Arbotag

Arbotag es una solución de la empresa Sercotal, donde se busca aportar eficiencia en la gestión y administración del arbolado de forma remota y en línea. ArboTag entrega información en tiempo real sobre la condición de un árbol y cuenta con su aplicación Arbotag Vecino, donde se puede participar en el bienestar de los árboles escaneando el código QR dispuesto en cada uno, y solicitar actividades como poda, mantenimiento, riego y seguridad. Esta tecnología ya está implementada por varios municipios, como, por ejemplo: Las Condes, Santiago, La Reina, Municipalidad de Chillán, entre otros (Arbotag, s. f.).

Es evidente que existe un conjunto mínimo de tecnologías que se repiten en los distintos proyectos. Esto se puede apreciar en la Tabla 1, donde las columnas representan las tecnologías utilizadas y las filas corresponden a los proyectos analizados. En esta tabla, se marca con una "x" los proyectos que emplean cada tecnología. Como se puede observar, todas las iniciativas utilizan tecnología móvil y, al menos, una de las dos tecnologías mencionadas para proporcionar información. Con esta información, podemos validar que estas tecnologías son ampliamente usadas y efectivas al momento de entregar información al usuario.

Tabla 1 : Comparativa de tecnologías por proyecto

	Tecnología Móvil	Códigos QR	Geolocalización
2.2.1	X	X	X
2.2.2	X		
2.2.3	X	X	
2.2.4	X		X
2.2.5	X		X
2.2.6	X		X
2.2.7	X	X	X
2.2.8	X	X	
2.2.9	X	X	

2.3 Tecnologías Implementadas

Como se mencionó y valido anteriormente, se implementarán tecnologías para poder obtener la información que luego podrá visualizar el visitante mediante su dispositivo móvil. Estas tecnologías son: una aplicación móvil que se implementara utilizando una metodología de aplicación web progresiva (PWA) que no necesita descargarse, georreferenciación a partir del GPS de los dispositivos móviles y la lectura de códigos QR que están desplegados por el parque.

Para poder comprender cómo se implementará una aplicación web que no requiere descargas adicionales en los dispositivos móviles, es esencial entender cómo funciona esta tecnología que se verá a continuación.

2.3.1 Aplicación Web Progresiva

Una aplicación web progresiva (PWA), es una aplicación web que se ve y se comporta como una aplicación móvil. Las PWA están diseñadas tomando como referencia las funciones nativas de los dispositivos móviles, y se accede a ellas a través de los navegadores web de dichos dispositivos. Esto evita que los usuarios finales tengan que realizar la descarga de una aplicación desde una tienda de aplicaciones (Quiroz, 2022).

El objetivo principal de las aplicaciones web progresivas es reducir la distinción entre las apps nativas y la web móviles, para esto usan tecnologías basadas en estándares y se ejecutan en contenedores seguros. Se puede enviar notificaciones, trabajar sin conexión y ser accesible desde la pantalla de inicio de los dispositivos móviles, tal como si fueran una aplicación nativa.

A continuación, se describirán el conjunto de metodologías que se utilizan para obtener como resultado una PWA.

2.3.1.1 Archivo de Configuración Manifiesto

Para añadir una PWA al menú de aplicaciones (menú de inicio) de nuestro dispositivo, es necesario especificar ciertas características en un archivo conocido como manifest.json. Este archivo en formato JSON² es crucial para el funcionamiento de la PWA en dispositivos móviles e incluye atributos como:

- El nombre de la aplicación
- El icono que se mostrará en el inicio
- Color de la aplicación
- Si la aplicación puede utilizarse de forma horizontal u vertical.

Entre otros elementos que se pueden configurar en este archivo. Mas adelante, en el Capítulo 4 implementación de la solución, se indicará como fue configurado este archivo.

2.3.1.2 Service Worker

Un archivo de configuración indispensable para que una aplicación PWA funcione correctamente es el *Service Worker*, que actúa como una API³ entre el servidor donde está alojada la aplicación y la propia aplicación contenida en el dispositivo. *El Service Worker* almacena información en la caché del dispositivo móvil y se actualiza en respuesta a eventos específicos, tal como se puede apreciar en la Figura 1. Además, este componente ofrece funcionalidades clave, como:

- Utilizar la aplicación sin conexión a internet.
- Recibir notificaciones.
- Acceder a los sensores del dispositivo, como la cámara y el GPS.

² JSON, es un formato ligero de intercambio de datos, fácil de leer y escribir para las personas, y fácil de parsear y generar para máquinas.

³ API (Interfaz de programación de Aplicaciones) es un conjunto de reglas que permite que diferentes programas informáticos se comuniquen entre sí.

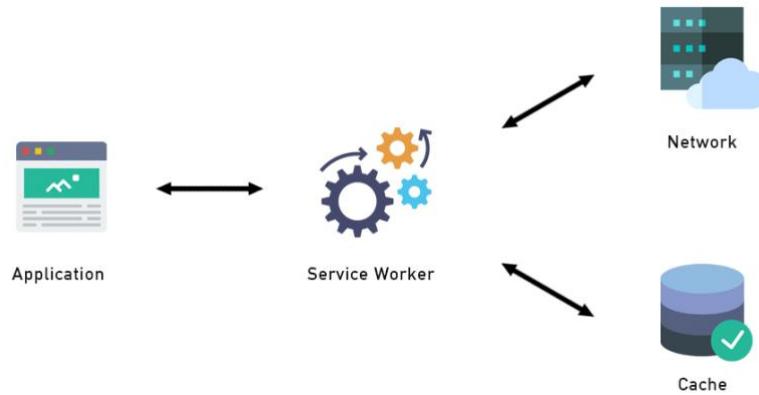


Figura 1 : Forma en que interactúa el *Servicie Worker* con la aplicación y servidor⁴

2.3.1.3 Diseño Responsivo

El diseño responsivo es un formato de programación que permite ajustar una aplicación web automáticamente al tamaño y disposición de las pantallas de los dispositivos de los usuarios. Las aplicaciones web responsivas cambian para ofrecer la mejor experiencia a los visitantes desde sus teléfonos inteligentes, tabletas o computadoras de escritorio (Hubspot, 2023).

2.3.1.4 Por qué no Elegir una Aplicación Nativa

Las aplicaciones nativas son desarrolladas específicamente para operar en un sistema operativo particular, como Android, iOS o Windows, entre otros. Son aquellas que se descargan desde una tienda de aplicaciones, como Play Store (Android) o App Store (iOS).

Para cada sistema operativo en el que se desee instalar estas apps, se debe desarrollar una versión específica. Por ejemplo, si se crea una aplicación para dispositivos móviles y se desea que esté disponible para el público, es necesario desarrollar la aplicación dos veces: una para Android y otra para iOS (ABAmobile, s.f.). Esto se debe a que cada sistema operativo tiene su propia forma de funcionar y sus reglas específicas para ejecutar las aplicaciones programadas.

Las aplicaciones nativas ofrecen varias ventajas, como un alto nivel de personalización, la capacidad de operar sin conexión a internet y mayor seguridad. No obstante, también presentan desventajas significativas, como el costo elevado de desarrollo, el tiempo y los recursos generales necesarios (ABAmobile, s. f.).

⁴ <https://medium.com/@resatDev/the-power-of-service-workers-3ff974ddf3de>

Además, surge otra desventaja: a las personas a menudo les resulta tedioso instalar aplicaciones nativas para propósitos generales. Por ejemplo, una persona va a un supermercado, y al ingresar una promotora le ofrece un descuento por descargar una aplicación en el teléfono, es probable que ella y otras personas opten por no hacerlo por diversos motivos, como la falta de interés o que el dispositivo móvil no cuente con espacio de almacenamiento disponible. Esto podría significar que el supermercado no logre recabar la información deseada debido a la baja adopción de la aplicación. En cambio, si la promotora sugiere escanear un código QR que abre el navegador en el dispositivo para completar un formulario breve, es más probable que las personas prefieran esta opción debido a la simplicidad de los pasos a seguir.

Por lo tanto, no siempre una aplicación nativa es la mejor opción; todo depende del propósito para el cual se desarrolla y sus objetivos generales.

Por otro lado, es común que surja la duda sobre por qué no optar simplemente por una página web donde el visitante pueda consumir la información. Para abordar esta pregunta, es crucial entender las diferencias entre una página web, un sitio web, y una aplicación web.

- **Página web**, es un documento escrito con HTML⁵ y puede incluir CSS⁶, JavaScript⁷, imágenes, videos. Esta puede mostrar contenido estático o podría estar conectada a una base de datos para mostrar contenido dinámico.
- **Sitio web**, es un conjunto de páginas web estructuradas en un dominio. Es muy común que las personas se refieran a páginas web cuando en verdad se trata de un sitio web.
- **Aplicación web**, también es un sitio web porque igual es un conjunto de páginas web dentro de un dominio. Sin embargo, una aplicación web es un software creado con tecnología web. Con el fin de albergar toda la lógica de negocio de una identidad (EDteam, 2021).

2.3.2 Geolocalización Basada en Tecnología GPS

Para entender bien cómo entregar información basada en la localización de una persona debemos de entender varios conceptos que se aplican en esta tecnología.

La geolocalización es un término que se utiliza para describir la capacidad de detectar y registrar los lugares donde un usuario u objeto en donde se encuentran (Arimetrics, 2022). Por otra parte, geo posicionar, significa situar una persona en un punto u objetivo en un plano cartográfico.

El sistema de posicionamiento global (GPS) es un sistema de navegación que utiliza satélites, un receptor y algoritmos para sincronizar datos de localización, velocidad y

⁵ HTML, es el estándar que permite crear y estructurar páginas, funciona a través de etiquetas y atributos.

⁶ CSS, es un lenguaje que define el estilo de los documentos HTML, como el diseño, colores y fuentes.

⁷ JavaScript, es lenguaje de programación que permite crear contenido interactivo en páginas web.

tiempo, en donde se utiliza mayormente para viajes aéreos, marítimos y terrestres. Esta tecnología funciona a través de una técnica llamada trilateración (Geotab, 2020), utilizada para calcular la ubicación, la velocidad y la elevación. La trilateración recopila señales de los satélites para enviar información de la ubicación al dispositivo que cuente con esta tecnología. En la Figura 2, podemos apreciar visualmente cómo se aplica esta técnica.



Figura 2 : Trilateración técnica para captar la posición de un usuario⁸

La posición se obtiene triangulando señales de al menos tres satélites. Para calcular las distancias, el GPS mide el tiempo que tarda en llegar la señal desde el satélite hasta la Tierra. Para ello, es crucial que los satélites estén equipados con relojes extremadamente precisos. Además, para determinar la ubicación precisa, es necesario conocer tanto el tiempo que tarda la señal en llegar como la posición exacta del satélite en el espacio.

2.2.3 Códigos QR

Los códigos QR, es un tipo de código que almacenan información en dos dimensiones y la hacen accesible. QR son las iniciales Quick Response (respuesta rápida), su nombre tiene sentido, ya que un escáner procesa datos y ejecuta órdenes al momento. Cualquier cámara, ya sea un smartphone o Tablet, puede leer estos códigos con la aplicación

⁸ <http://www.aviacionglobal.com/articulos-tecnicos-de-aviacion/el-gps-y-sus-aplicaciones-i/attachment/trilateracion/>

apropiada (ionos, s. f). Estos son de forma cuadrada y los podemos encontrar en papel, letreros o en algún tipo de posters como lo muestra la Figura 3.



Figura 3 : Código QR posicionado en letrero dentro de un parque

Fueron creados en 1994 por una empresa japonesa en el sector del automóvil para el seguimiento de piezas, el código QR se diferenció de los códigos de barras tradicionales por su capacidad para almacenar gran cantidad de datos.

Funcionamiento de un Código QR

El código QR contiene un patrón en un gráfico cuadrado en el cual se ha incrustado información en formas de puntos y líneas negras y blancas que la aplicación lee (actualmente podemos encontrar QR diferentes formas y colores). Un código QR puede llegar a contener hasta 177 x 77 elementos y un texto de hasta media página DIN A4. Expresados en cifras, un código QR tiene la capacidad de hasta 4296 caracteres alfanuméricos o 7089 dígitos decimales en el nivel más bajo de corrección de errores. Cuando más alto sea el nivel de corrección de errores, menos capacidad tendrá el código QR (ionos, s. f.). En la Figura 4 podemos apreciar los elementos básicos del QR.



Figura 4 : Elementos básicos de un código QR⁹

El número uno, corresponden a los tres cuadrados en la esquina del código, que sirven de orientación al escáner. Por otro lado, donde se muestra el número dos, es el modo patrón en donde se contiene la información del código QR.

⁹ <https://www.ionos.es/digitalguide/online-marketing/vender-en-internet/que-es-un-codigo-qr/>

3 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

En este capítulo se describe el proceso de diseño del sistema GeoParques. Este capítulo se dividió en subsecciones que explicarán el origen del sistema, detallarán la identificación de requisitos tanto funcionales como no funcionales, el diseño de la arquitectura implementada y las diferentes herramientas que se utilizaron en el desarrollo y administración del proyecto.

3.1 El Origen de GeoParques

Tal como se mencionó en la introducción, el profesor patrocinante generó un proyecto que se denominó GeoParques, con el objetivo de abordar las diversas problemáticas que se mencionaron anteriormente, tanto para los visitantes como para los administradores de los parques. Estas dificultades podrían solucionarse ofreciendo distintas maneras de acceder a información de interés de los atractivos del parque, tal como se podrá apreciar en la siguiente Figura 5.



Figura 5 : Plataforma GeoParques y sus distintas funcionalidades

Existen distintas formas de consumir información, lo que llevó a la decisión de dividir el proyecto en cinco subproyectos. Estos se concibieron para funcionar de manera totalmente independiente, permitiendo su desarrollo por separado. Sin embargo, también se diseñaron de tal manera que pudieran integrarse en una única plataforma, ofreciendo valor tanto a los visitantes como a los administradores de los parques. Estos cinco subproyectos, como se puede apreciar en la Figura 6, representan un enfoque modular y flexible hacia la resolución del problema y la mejora de la experiencia en los parques.

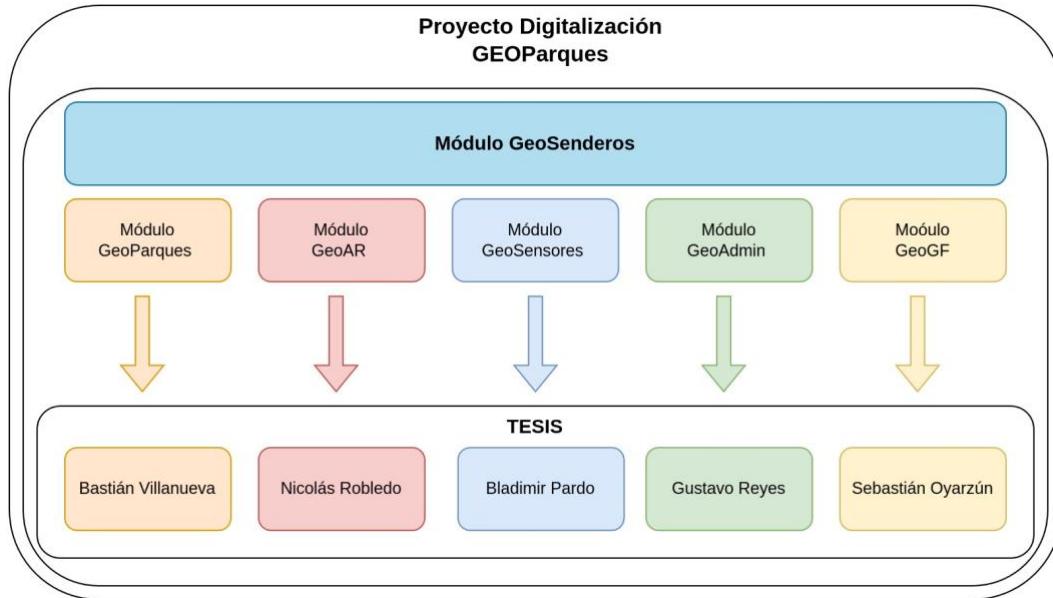


Figura 6 : Diagrama de contexto del proyecto en general

En este contexto, los contenedores son componentes del sistema y subproyectos aparte. GeoParques, en particular, es el subproyecto orquestador o punto de partida para los otros módulos, incorporando las características mencionadas en la introducción. A continuación, se mencionarán los distintos subproyecto y sus características:

- **GeoParques:** se enfoca en brindar información mediante códigos QR y localización por GPS a través de una aplicación PWA para los visitantes y otra aplicación diseñada para los administradores de parques.
- **GeoAR:** se enfoca a brindar una experiencia con realidad aumentada a los visitantes de parques.
- **GeoSensores:** este módulo utiliza sensores desplegados por el parque y entrega información mediante la activación de los sensores cuando el visitante se acerca a esté.
- **GeoGF:** este se caracteriza por brindar juegos interactivos a los visitantes de parques.
- **GeoAdmin:** este módulo se enfoca en proporcionar un dashboard diseñado para la toma de decisiones mediante información recopilada por los otros módulos. Es importante no confundir con GeoParques – Administrador que se detallara más adelante.

Al inicio del proyecto se trabajó en equipo con los cinco integrantes de los distintos subproyectos, con el objetivo de alinear las ideas y los requisitos del proyecto en general. Por esta razón, se decidió desarrollar una base de datos central que todos los subproyectos pudieran utilizar. Más adelante se explicará con mayor detalle esta base de datos.

Para la creación de GeoParques, era necesario disponer de dos aplicaciones independientes: una donde los visitantes pudieran visualizar la información, denominada “GeoParques – Visitantes” o simplemente GeoParques, y otra aplicación para que los administradores de parques pudieran gestionar la información de manera autónoma, a la cual se le denominó “GeoParques – Administrador”. En la Figura 7, se aprecia las dos distintas aplicaciones y los dos actores que interactúan con el sistema.

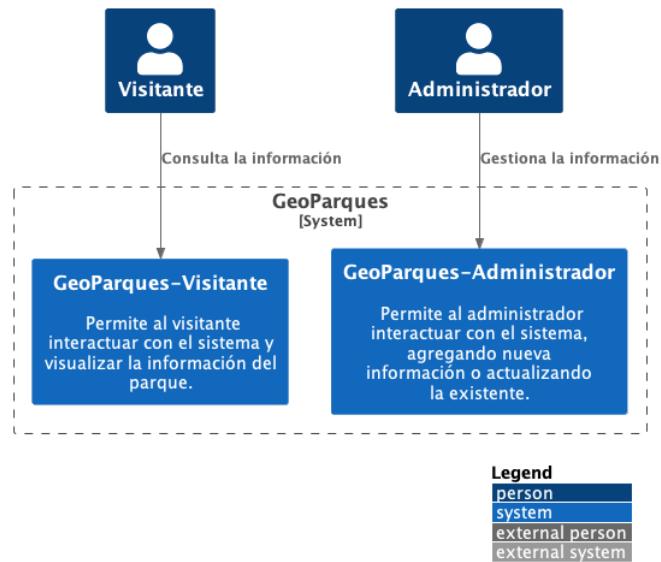


Figura 7 : Actores interactuando con aplicaciones de GeoParques

3.2 Requerimientos

Con el fin de poder brindar las funcionalidades necesarias a los dos tipos de usuarios contemplados por este sistema, se realizaron reuniones semanales con el equipo de trabajo. Además, se llevaron a cabo reuniones con el equipo del jardín botánico, que inicialmente fueron dirigidas por el profesor patrocinante, pero finalmente se tuvo una con el equipo donde se pudieron detectar las necesidades que tenían tanto los visitantes de parques turísticos como los administradores del parque. En la Tabla 2, se muestra las necesidades que los usuarios requieren al interactuar con la aplicación.

Tabla 2 : las necesidades de los usuarios que quieren para interactuar con el sistema

Usuario	Descripción
Visitante	El visitante busca desconexión y disfrutar de la naturaleza, donde pueda aprender sobre las distintas especies de flora existentes en el lugar. Además, si desea encontrar una especie que le gusta y conoce su nombre, quiere poder buscarla y que el sistema le entregue la posición exacta de la especie. Desea conocer los distintos miradores que

	ofrece el parque y poder encontrarlos con el mapa. Quiere estar informado sobre los eventos en el jardín botánico y poder leer información importante sobre este.
Administrador	El administrador desea poder gestionar toda la información accesible al visitante, desde los textos de bienvenida hasta los hitos informativos que se podrán visualizar en el mapa. Además, quiere agregar nuevos árboles al sistema, pudiendo asociar una ficha de especie existente. También desea poder activar o desactivar la visualización de un árbol en la aplicación del visitante, con el fin de mostrarlo o no. Quiere acceder a los reportes de sugerencias dejados por los visitantes, así como a los reportes que puedan indicar si alguna especie está en la posición incorrecta o si hay algún error en la ficha. Además, quiere poder agregar eventos para que los visitantes puedan visualizarlos y recordar fechas importantes.

Dado que los dos tipos de usuarios presentan distintas funcionalidades, se optó por separar los requisitos específicos para visitantes y administradores.

3.2.7 Requisitos Funcionales Visitantes.

En esta sección se presentarán, por medio de historias de usuario, los requisitos captados para los visitantes de parques. Se optó por separar la información en diferentes vistas con el objetivo de distribuir la manera en que se presenta la información a los visitantes. Esto se hizo para evitar sobrecargar una sola vista con demasiado contenido. Estas se definieron en las Tablas de 3 a 7.

Tabla 3 : Requisitos, vista de Bienvenida

N	Enunciado de la Historia	Criterios de Aceptación
1	Como Visitante, quiero recibir una bienvenida y comprender el concepto del sistema geoparques al iniciar la aplicación.	<p>La cabecera principal debe mostrar claramente el nombre del parque y valores de importancia.</p> <p>Debe haber un segmento de portada que proporcione una introducción a geoparques y al jardín botánico.</p> <p>La portada debe incluir un botón de acceso rápido que lleve al mapa.</p>
2	Como visitante, quiero poder visualizar una imagen del jardín botánico.	<p>La imagen debe ser de alta calidad y cargar rápidamente.</p>
3	Como visitante, necesito conocer más del parque por medio de tarjetas que me brinden un resumen de información relevante.	<p>Debe haber al menos dos tarjetas de información en la página de bienvenida.</p> <p>Una de las dos tarjetas debe contener información sobre el parque y un botón que enlace a más detalles.</p> <p>La segunda Tarjeta, debe presentar información del sistema de Geoparques y un botón que enlace a más detalles.</p>

Tabla 4 : Requisitos, vista del mapa

N	Enunciado de la Historia	Criterios de Aceptación
1	Como Visitante, quiero poder interactuar con un mapa del parque, en donde pueda encontrar marcadores con información detallada de los distintos hitos informáticos.	El mapa debe cargar centrado en el parque con todos los hitos informativos habilitados visibles.
		Al hacer clic en los marcadores, debe aparecer una ventana modal con el hito de información.
		La interacción con el mapa y los marcadores debe ser fluida y sin demoras.
2	Como visitante, necesito acercar o alejar la vista del mapa, con el fin de explorar el parque con detalle o tener una visión general.	El zoom no debe permitir alejarse tanto para que no se pierda la vista del parque.
		Los controles de zoom deben ser fáciles de usar en dispositivos móviles y de escritorio.
3	Como visitante, quiero poder filtrar los hitos informativos en el mapa para ver sólo aquellos que me interesan.	Debe haber un botón en el mapa para abrir un panel de filtrado.
		Los visitantes deben poder activar o desactivar categoría de hitos (como árboles, fichas informativas, etc.)
		El mapa debe actualizar los marcadores visibles inmediatamente después de cambiar los filtros.
4	Como visitante, deseo utilizar la geolocalización para identificar mi posición actual en el parque y descubrir hitos informativos cercanos.	Un botón debe permitir a los visitantes activar la geolocalización en su dispositivo móvil.
		La posición actual del visitante debe indicarse claramente en el mapa.
		La aplicación debe mostrar los hitos cercanos a la posición actual del visitante.

Tabla 5 : Requisitos, vista todas las especies

N	Enunciado de la Historia	Criterios de Aceptación
1	Como visitante, quiero ver una lista de todas las especies del parque, para poder aprender sobre ellas.	Debe haber una tarjeta para cada especie del parque mostrando una imagen, el nombre conocido, nombre científico.
		Al hacer clic en una tarjeta, debe abrirse una ventana modal con información detallada de la especie.
		La interfaz debe ser visualmente atractiva y facilitar la lectura de la información.

2	Como visitante, quiero poder buscar especies por su nombre para encontrar rápidamente las que me interesan.	Debe haber una función de búsqueda que permita a los visitantes encontrar las especies por su nombre.
		Las tarjetas de especies que no correspondan a la búsqueda deben ocultarse para mostrar solo los resultados relevantes.
		La búsqueda debe ser rápida y mostrar los resultados a medida que el usuario escribe.

Tabla 6 : Requisitos, vista código QR

N	Enunciado de la Historia	Criterios de Aceptación
1	Como visitante, quiero entender cómo usar los códigos QR dentro del parque para acceder a la información de manera rápida y eficiente.	La página debe ofrecer una explicación clara y concisa sobre el propósito y el uso del código QR en el parque.
		Debe haber ejemplos visuales o instrucciones para guiar al visitante del proceso de escaneo del QR.
		La información debe ser fácil de entender para personas de todas las edades y niveles de habilidad tecnológica.
2	Como visitante, necesito poder activar la cámara de mi dispositivo móvil a través de la aplicación para escanear códigos QR en el parque.	Debe haber un botón destacado que, al ser presionado, active la cámara del dispositivo móvil.
		La función de la cámara debe ser compatible con la variedad de dispositivos móviles y sistemas operativos.
		Después de escanear un código QR, la aplicación debe redirigir al visitante a la información específica vinculada a ese código.

Tabla 7 : Requisitos, vista de Participación de los visitantes y mejora continua en la experiencia de GeoParques

N	Enunciado de la Historia	Criterios de Aceptación
1	Como visitante, deseo compartir mis impresiones y sugerencias sobre GeoParques, con el objetivo de que el sistema mejore con el tiempo.	Debe haber un formulario accesible y fácil de usar para enviar recomendaciones y opiniones.
		Los campos del formulario deben incluir correo electrónico, evaluación de la experiencia, disfrute de la visita y opiniones generales.
		El formulario debe ser intuitivo y garantizar la privacidad de los datos personales proporcionados.

2	Como visitante, necesito una confirmación de que mi formulario de recomendaciones ha sido enviado correctamente.	Al enviar el formulario, debe haber una indicación visual o mensaje de que ha sido recibido correctamente.
		La fecha y hora del envío deben ser registradas automáticamente para contextualizar la retroalimentación.

3.2.2 Requisitos Funcionales Administrador

A continuación, se presentarán los Requisitos Funcionales por medio de vistas de usuario para las necesidades de los administradores de parques. Estas se definieron en las Tablas 8 a 12.

Tabla 8 : Requisitos, vista mapa general

N	Enunciado de la Historia	Criterios de Aceptación
1	Como administrador, quiero filtrar los hitos de información en el mapa para poder visualizar solo aquellos que son de interés.	El mapa debe estar centrado en la ubicación del parque
		Todos los hitos de información deben visualizarse en el mapa.
2	Como administrador, quiero filtrar los hitos de información en el mapa para poder visualizar solo aquellos que son de interés.	Debe haber un panel de opciones con checkboxes para cada tipo de información.
		Al marcar o desmarcar un checkbox, los hitos correspondientes deben aparecer o desaparecer del mapa en tiempo real.
		Los cambios en el filtro deben ser intuitivos y sin necesidad de recargar la página o realizar acciones adicionales.

Tabla 9 : Requisitos, vista Árbol

N	Enunciado de la Historia	Criterios de Aceptación
1	Como administrador, quiero poder agregar un nuevo árbol al sistema a través de un formulario.	Debe haber un formulario para añadir un nuevo árbol con campos para nombre científico, latitud, longitud y ficha de árbol asociada.
		El formulario debe validar la entrada de datos para asegurar la precisión de la ubicación y la información.
		Después de la inserción, el árbol debe aparecer en el mapa y la lista con la información correspondiente.

2	Como administrador, necesito visualizar los árboles en el mapa, con el fin de gestionar los árboles de forma eficiente.	<p>Todos los árboles deben estar representados en el mapa con un marcador personalizado.</p> <p>Cuando el Árbol esté desactivado el marcador se tiene que mostrar de otro color.</p> <p>Al seleccionar un marcador de árbol, debe mostrarse información breve del mismo y el panel de configuración.</p>
3	Como administrador, quiero tener un buscador de árboles por nombre científico para encontrar rápidamente información específica.	<p>La lista de árboles debe permitir la selección individual o múltiple de entradas.</p> <p>Los árboles seleccionados deben ser los únicos en mostrarse en el mapa tras la selección.</p> <p>Debe ser posible alternar la visualización de un árbol en el mapa desde la lista sin tener que abrir el mapa.</p>
4	Como administrador, necesito un panel de configuración de árbol para editar, activar/desactivar, ver la ficha y eliminar árbol	<p>Al hacer click en un marcador de árbol debe mostrar un menú configurable con opciones de edición, activación/desactivación, visualización de ficha y eliminación.</p> <p>Los resultados de la búsqueda deben mostrar todos los árboles que coincidan con el nombre científico.</p> <p>El botón de ver ficha debe mostrar toda la información relacionada con el árbol seleccionado.</p> <p>Eliminar un árbol debe retirarlo de la vista activa, pero mantenerlo en la base de datos por la persistencia de los datos.</p>

Tabla 10 : Requisitos, vista de fichas de árboles

N	Enunciado de la Historia	Criterios de Aceptación
1	Como administrador, necesito visualizar un listado de todas las fichas de árbol para gestionarlas de manera efectiva.	<p>Debe haber una vista de lista que muestre todas las fichas de árbol con su nombre binomial e ID.</p> <p>La lista debe incluir un panel de configuración para cada ficha con opciones de edición y eliminación.</p> <p>La interacción con cualquier ficha en la lista debe ser intuitiva y no requiere recargar la página.</p>
2	Como administrador, quiero buscar fichas de árbol por nombre binomial para encontrar rápidamente información específica.	<p>Debe haber un buscador que permita introducir el nombre binomial de una ficha de árbol.</p> <p>Los resultados de la búsqueda deben mostrar todas las fichas que coincidan con el término de búsqueda en la lista.</p> <p>La función de búsqueda debe ser rápida y actualizarse dinámicamente con los términos de búsqueda.</p>

	3 Como administrador, necesito un panel de configuración para cada ficha de árbol que me permita editar, visualizar y eliminar la información de las fichas.	Debe haber botones de configuración para cada ficha que permite editar, visualizar, generar/ver QR y eliminar.
		El botón de editar debe cargar el formulario con los datos actuales de la ficha para su modificación.
		El botón de generar/ver QR debe crear un código QR para poder descargarlo.
		El botón de eliminar debe retirar la ficha de la vista activa, pero mantenerla en la base de datos.

Tabla 11 : Requisitos, vista de Fichas informativa

N	Enunciado de la Historia	Criterios de Aceptación
1 Como administrador, quiero poder agregar nuevas fichas informativas al mapa para proporcionar información útil a los visitantes durante su recorrido por el parque.		Debe haber un botón interactivo que al ser presionado permite agregar un marcador en el mapa.
		Al colocar el marcador, se debe desplegar una ventana modal con un formulario para ingresar título, imágenes, descripción, longitud y latitud.
		Las coordenadas deben llenarse automáticamente con la ubicación del marcador en el mapa.
2 Como administrador, necesito poder visualizar todas las fichas informativas en el mapa para gestionarlas eficientemente.		Cada ficha informativa debe ser representada por un marcador en el mapa.
		Los marcadores deben diferenciarse por color para indicar si están activados (visibles para los visitantes) o desactivados.
		Al seleccionar un marcador, debe mostrarse información breve del mismo y el panel de configuración.
3 Como administrador, quiero poder visualizar y gestionar todas las fichas informativas en forma de lista para tener un control detallado de la información disponible en el parque.		Debe haber una lista de todas las fichas informativas con su nombre, ID y opciones de configuración.
		La lista debe permitir al administrador realizar las mismas acciones que en el mapa, incluyendo editar, activar/desactivar y eliminar.
		La interfaz de la lista debe ser clara y fácil de usar para gestionar rápidamente las fichas informativas.
4	Como administrador, quiero tener un panel de configuración para cada ficha	Al hacer clic en un marcador, debe aparecer un menú configurable con opciones para editar, activar/desactivar, visualizar y eliminar la ficha.

	<p>informativa que me permita editar, activar/desactivar, visualizar y eliminar la información de las fichas</p>	<p>Debe ser posible modificar todos los campos de una ficha informativa a través de un formulario accesible desde el botón de editar.</p>
		<p>La opción de visualización debe mostrar cómo se verá la ficha informativa para el visitante.</p>
		<p>La eliminación de una ficha informativa debe ser permanente tanto en la aplicación del administrador como en la del visitante, pero manteniendo la información en la base de datos para fines de persistencia.</p>

Tabla 12 : Requisito, vista de panel de configuración para administrar la aplicación de los visitantes del parque

N	Enunciado de la Historia	Criterios de Aceptación
1	Como administrador, quiero editar el contenido de la portada inicial para dar una bienvenida personalizada a los visitantes de la aplicación.	Debe existir una opción para editar el texto de bienvenida en la portada inicial.
		La interfaz de edición debe permitir la inserción de texto y su formateo básico.
		Los cambios deben reflejarse en tiempo real en la aplicación del visitante después de guardar.
2	Como administrador, necesito cambiar o editar el contenido de las tarjetas en la portada para actualizar la información destacada.	Debe haber una opción para editar el contenido de las tarjetas A y B.
		El administrador debe ser capaz de cambiar textos de la tarjeta.
		Los cambios deben ser guardados y actualizados inmediatamente en la aplicación del visitante.
3	Como administrador, deseo gestionar las noticias y eventos que aparecen en la página de inicio de la aplicación del visitante.	Debe haber una función para crear nuevas noticias o eventos a través de un formulario con campos para nombre, descripción, y fechas de inicio y término.
		El administrador debe ser capaz de ver una lista de todas las noticias con opciones para editar, desactivar/activar y eliminar cada una.
		El panel de configuración de noticias debe ser fácil de usar y permitir cambios inmediatos en la aplicación del visitante.
4	Como administrador, necesito un panel de configuración de noticias eficiente para editar, visualizar, habilitar/deshabilitar y eliminar noticias y eventos.	Cada noticia debe tener un botón de configuración para realizar acciones de edición, visualización, habilitación/deshabilitación y eliminación.
		El botón de editar debe utilizar el mismo formulario que se usa para crear una noticia.

	<p>La opción de habilitar/deshabilitar debe reflejar su estado actual y cambiar la visibilidad de la noticia en la aplicación del visitante.</p>
	<p>El botón de eliminar debe quitar la noticia de la vista activa y de la lista de noticias, pero mantenerla en la base de datos.</p>

3.2.3 Requisitos No Funcionales

Tabla 13 : Requisitos no funcionales

N	Requisito	Descripción
1	Seguimiento de especies que llaman la atención de los visitantes	Cuando una persona clic en una especie mediante un marcador o escanee un código QR, el sistema debe ser capaz de guardar esa información, con el objetivo de identificar cuáles son las especies más visitadas.
2	La aplicación del visitante tiene que ser responsive	La aplicación debe adaptarse a los distintos tamaños de pantalla, debido a la naturaleza de aplicación web progresiva.
3	Internacionalización.	La aplicación del visitante debe tener la capacidad de cambiar el idioma si el usuario lo considera pertinente.
5	Modularidad	El sistema debe estar diseñado en componentes o módulos independientes. Que pueden ser utilizados de manera independiente.
6	Rendimiento	La aplicación del visitante debe ser capaz de manejar un volumen de peticiones que se asemeje al número promedio de personas que visitan el parque en un día.

3.3 Tecnologías para el Desarrollo

En esta subsección, se mencionan las principales tecnologías utilizadas en la realización de este trabajo de título, las cuales son fundamentales para la arquitectura del sistema, desarrollo y funcionamiento de los distintos módulos de la plataforma GeoParques.

3.3.1 JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, utilizado principalmente para el desarrollo web, está pensado para agregar potencial de interacción y dinamismo a las páginas web (Hubspot, 2023). Este lenguaje permite crear contenido de forma interactiva, controlar multimedia, animar imágenes y prácticamente todo lo que puedes apreciar en las distintas aplicaciones web de hoy en día.

Actualmente, este lenguaje de programación no solo se utiliza para el desarrollo de interfaces de usuario, sino también para el desarrollo de aplicaciones de escritorio, móviles y del lado del servidor de la aplicación.

3.3.2 React.js

React es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web (Hostinger, 2023). Fue desarrollada por Facebook en el 2013 con la finalidad de facilitar la creación de las páginas y aplicaciones web, por medio del desarrollo de componentes reutilizables e interactivos que conforman parte de la interfaz de usuario.

La reutilización de componentes es muy práctica, ya que permite crear, por ejemplo, un botón y usarlo en distintas partes de la interfaz simplemente llamándolo en el script¹⁰ correspondiente.

3.3.3 Node.js

Node.js es un entorno de ejecución para JavaScript de código abierto y multiplataforma, enfocado en el desarrollo de aplicaciones del lado del servidor. Facilita a los desarrolladores la creación de aplicaciones de red, de manera rápida y escalables, mediante un código fácil de comprender (Kinsta, 2023).

Su propósito es permitir la ejecución de código JavaScript en el sistema operativo sin necesidad de un navegador web, utilizando para ello el motor V8 de Google Chrome. Además, como ya se ha mencionado, es compatible con diversas plataformas, lo que permite su ejecución en sistemas operativos como Windows, Mac OS, Linux, entre otros, haciendo que las plataformas sean fácilmente portables y multientornos.

¹⁰ Es un archivo de código que se ejecuta para realizar tareas específicas automáticas.

3.3.4 Express.js

Express es el framework¹¹ de Node.js más utilizado para el desarrollo de servidores web backend¹² con JavaScript (Kinsta, 2023). Este framework proporciona las herramientas necesarias para el desarrollo de backend, simplificando el enrutamiento y la gestión de las respuestas a las solicitudes web.

3.3.5 Tailwind CSS

Es un framework CSS que permite un desarrollo ágil, basado en clases de utilidad que se pueden aplicar con facilidad en los códigos HTML, además cuenta con un flujo de desarrollo que permiten optimizar la escritura de código CSS (Knowmadmood, 2021). En otras palabras, tailwind permite simplificar el estilar y dar forma a las aplicaciones web con simples paso.

3.3.6 MongoDB

MongoDB, es una base de datos NoSQL orientada a documentos. Se diferencia de las bases de datos relacionales por su flexibilidad y rendimiento (Datascientest, 2022). A diferencia de una base de datos relacional SQL, MongoDB no se basa en tablas y columnas, si no que los datos se almacenan como colecciones y documentos.

Los documentos en MongoDB son pares clave/valor que sirven como la unidad básica de datos. Cada base de datos de MongoDB contiene colecciones, las cuales a su vez contienen documentos. Cada documento puede tener una estructura diferente y puede contener un número variable de campos. Además, el tamaño y el contenido de cada documento también pueden variar, con un tamaño máximo por documento de hasta 16 MB (Mongodb, s.f.-b).

¿Por qué se ha decidido utilizar una base de datos no relacional en vez de una relacional?

Las bases de datos NoSQL son esquema-libres o poseen esquemas flexibles. Esto permite que, al necesitar agregar un nuevo dato a un documento, no sea necesario realizar una modificación significativa en la estructura de la base de datos. Por ejemplo, si se tiene una colección de datos de árboles que sigue una estructura de datos específica, pero se desea agregar una característica particular a un árbol en la base de datos, es posible añadir ese atributo únicamente a dicho árbol, sin afectar las estructuras de datos de los otros árboles.

Cuando se planeó el proyecto, se decidió optar por una base de datos NoSQL debido a los requisitos propuestos por el cliente, ya que existe una gran cantidad de árboles que cuentan con características únicas que se podrían agregar a la base de datos sin afectar la estructura

¹¹ Es un conjunto estructurado de herramientas, librería y convenciones que simplifica el desarrollo de software.

¹² Se refiere al desarrollo del lado del servidor, en donde se maneja la lógica del negocio, base de datos, gestionando las peticiones del clientes y enviando peticiones adecuadas.

de los otros árboles. Esto permite una rápida escalabilidad y flexibilidad al modelo de datos. Cabe mencionar que existe una gran cantidad de datos que podrían agregarse más adelante a la base de datos para seguir escalando el proyecto.

Por último, se utilizó MongoDB debido a su tolerancia y facilidad para trabajar con datos georreferenciados. El operador para esto se denomina "geometry" y se utiliza en consultas geoespaciales de MongoDB para especificar una geometría GEOJSON (Mongodb, s.f.). En la Figura 8 se muestra la estructura que se sigue para georreferenciar datos en MongoDB.

```
{  
  "geometry": {  
    "type": "Point",  
    "coordinates": [-73.24897, -39.80393]  
  }  
}
```

Figura 8 : Ejemplo de dato georreferenciado utilizando la estructura geometry

3.4 Arquitectura de la Solución

En esta subsección del capítulo, se describe la arquitectura de software implementada para el correcto funcionamiento del flujo de información. Además, se explican las razones que sustentan esta forma de describir el sistema.

Descripción General del Sistema GeoParques

Para organizar un sistema que está compuesto de diversos componentes que interactúan entre sí, los ingenieros de software se especializan en un campo conocido como arquitectura de software. La cual se basa en la toma de decisiones de diseño cruciales para organizar el software y asegurar que cumpla con los atributos de calidad deseados (Ken, 2023). Una buena arquitectura es fundamental para alcanzar el éxito del proyecto, ya que afecta a la calidad, el rendimiento y la escalabilidad del software.

Para una explicación más detallada de la arquitectura, los diferentes diagramas se explicarán empezando con un enfoque general y, gradualmente, profundizando en los distintos componentes para examinar su arquitectura interna. Esto se logra mediante el uso de los diagramas C4, que son ideales para este tipo de enfoque escalonado, en donde se empieza con los diagramas de contexto hasta llegar a diagramas de código. (C4mode, s.f.).

En la Figura 9, se detalla cómo estos diagramas se clasifican en componentes y en distintos colores para proporcionar un entendimiento correcto.

Legend
person
system
container
external person
external system
external container

Figura 9 : Forma de representar los componentes en un diagrama C4

- **Person:** este color y componente representan a la persona que interactúa con otros componentes.
- **System:** este color y componente representan un sistema en general.
- **Container:** este color y componente representan los distintos componentes que tiene un sistema.
- **External System:** este componente representa un sistema externo.

A continuación, se presenta el primer diagrama en la Figura 10, el cual muestra el enfoque más general del sistema. Posteriormente, se presentarán los diagramas más detallados de cómo están compuestos internamente los distintos componentes.

3.4.1 Diagrama De Contexto

Este diagrama ofrece una vista general de los distintos componentes que forman parte del proyecto de título GeoParques. Aquí se visualizan el sistema de la aplicación del visitante y la aplicación del administrador de parques, ambas conectadas a la GeoApi, que les provee de la información necesaria para su visualización y gestión.

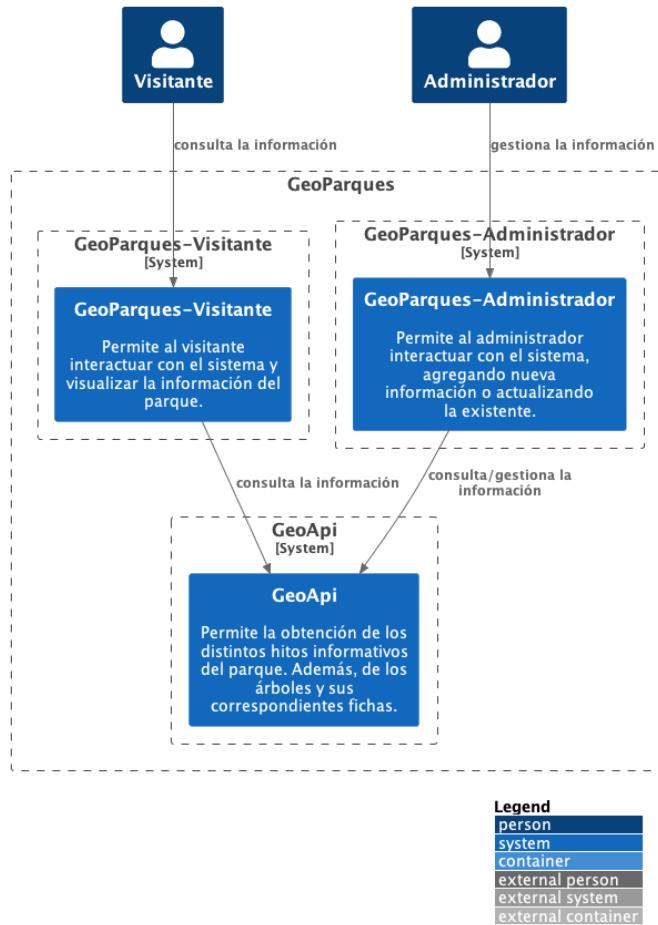


Figura 10 : Diagrama de contexto del sistema de Geoparques

3.4.2 Sistema GeoApi

Cuando se planeó el sistema general, se tuvo la idea de tener una API REST¹³ central que funcionara como núcleo del sistema. Esta tenía que entregar y gestionar toda la información del negocio, es decir, toda la información de importancia para los parques. Se realizó de esta forma para que los distintos subproyectos que forman parte del proyecto general del profesor patrocinante pudieran alimentarse de esta API. En la Figura 11 se puede apreciar la arquitectura de este sistema.

¹³ REST, es un conjunto de principios de arquitectura que utilizan métodos HTTP estándar para permitir la comunicación e interacción con servicios web.

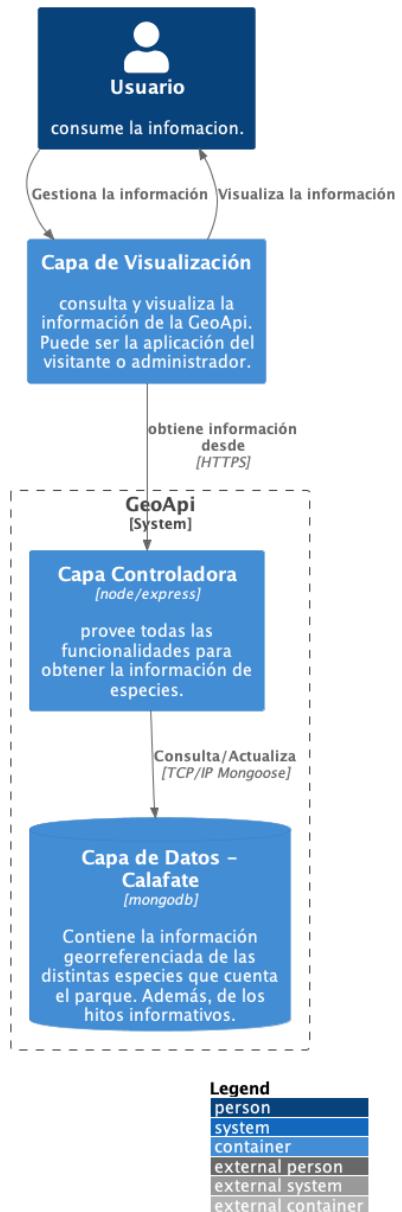


Figura 11 : Diagrama de contenedores arquitectura GeoApi

Para poder construir este sistema y los otros existentes, se decidió utilizar una arquitectura Modelo-Vista-Controlador (Mozilla s. f.), la cual es un patrón de diseño de arquitectura de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Este enfoque enfatiza una separación entre la lógica de negocios y su visualización. Esta "separación de responsabilidades" proporciona una mejor división del trabajo y una mejora en el rendimiento, ya que hace más fácil realizar una escalabilidad en caso de necesitarlo. A continuación, se darán a conocer las capas de la GeoApi.

- **La capa de visualización** se encarga de presentar la información al usuario. En el caso de este sistema, está diseñado para que esta capa sea representada por las distintas aplicaciones existentes en el proyecto en general, como la aplicación del visitante y la del administrador. Aquí, el usuario podrá interactuar con la información entregada por la capa controladora.
- **La capa controladora** se encarga de la lógica del negocio y el procesamiento de datos. Esta capa maneja las solicitudes de las distintas aplicaciones, procesa la información y realiza las operaciones necesarias para poder entregar la información que la capa de visualización solicita.
- **La capa de datos** corresponde a la base de datos, que se encarga de almacenar la diversa información esencial para el negocio. En este caso, la base de datos de la GeoApi, denominada "Calafate", almacena los árboles del parque, las fichas de especies, fichas informativas, senderos, entre otros elementos relevantes.

A continuación, se describe con mayor detalle la capa controladora y la capa de datos, con el objetivo de entender como están organizadas internamente.

Capa Controladora

La capa controladora del sistema de la GeoApi es un servicio REST. Está construida con Node.js y Express para funcionar como tal. Internamente, sigue una arquitectura que permite organizar los distintos módulos que contiene. En la Figura 12, se presentan los distintos módulos de este servicio, los cuales se separan de la siguiente manera:

- **App:** se encarga de inicializar el servicio y establecer conexión con los distintos módulos.
- **Módulos:** se encarga de agrupar funcionalidades relacionadas. En la Figura 12 se aprecia que existe uno por cada hito informático (ficha informativa, tótem, etc.). Internamente, este módulo se descompone en varios componentes, que son module, router y controller.
- **Módulo Sistema de Imágenes:** este módulo está encargado de recibir, entregar y gestionar las imágenes del sistema. Este servicio funciona gracias a Multer¹⁴.
- **Módulo de Persistencia:** encargado de gestionar los datos, este módulo se descompone internamente en model, el cual es el modelo de la colección de datos correspondiente. Repository, el cual se encarga de procesar los datos y de obtenerlos o insertarlos en la base de datos. Esta se comunica con la base de datos gracias a Mongoose¹⁵.

¹⁴ Es un middleware para Node.js que facilita la carga de archivo en aplicaciones.

¹⁵ Es una biblioteca para Node.js, la cual facilita la interacción con la base de datos.

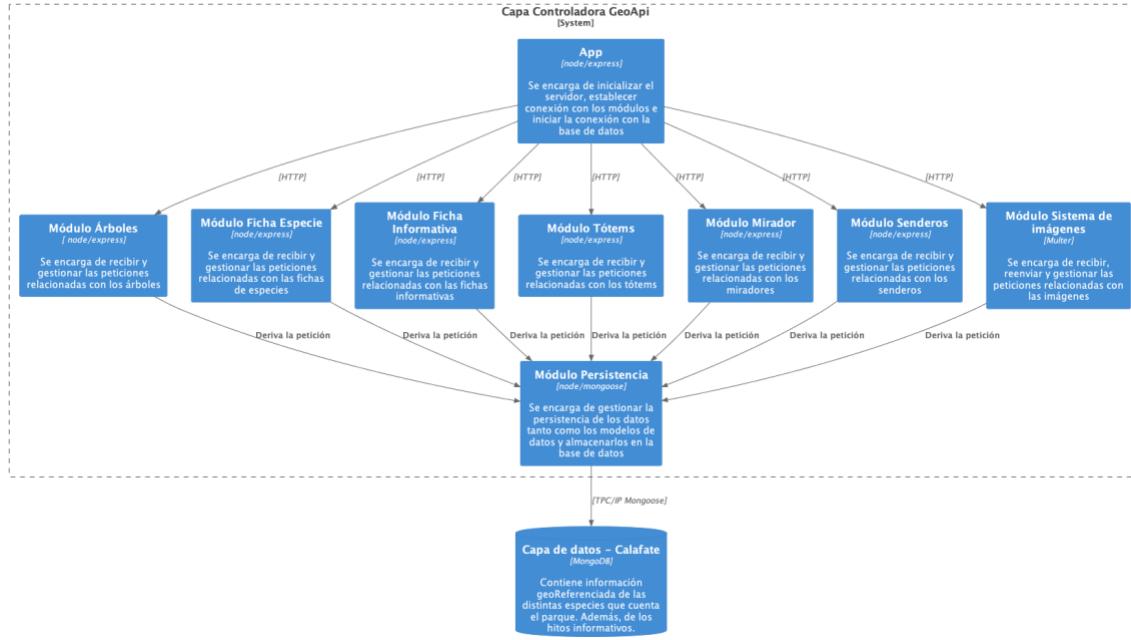


Figura 12 : Componentes de la capa controladora¹⁶

Para explicar con mayor detalle las funcionalidades de un módulo en particular de la capa controladora que se muestran en la Figura 12, así como el módulo de persistencia, tomaremos como ejemplo el módulo de árboles. En la Figura 13 se examinan sus componentes internos.

Esta práctica de seguir separando por componentes la API se le conoce generalmente como **arquitectura en capas** o **diseño de software Modular**.

- **Module:** se encarga de recibir las peticiones de un endpoint ¹⁷ correspondiente y lo derivarla al router.
- **Router:** gestiona las rutas y los endpoints. Actúa como un intermediario que interpreta las rutas solicitadas por el usuario, determinando qué Controller debe ejecutarse.
- **Controller:** recibe las entradas proporcionadas por el Router y maneja la lógica de negocios para cada operación recibida.
- **Repository:** se encarga de manejar y procesar los datos para posteriormente solicitar u almacenar a la base datos.
- **Model:** representa la estructura de datos y la lógica de negocio asociada a esos datos. Además, establece conexión con la base de datos.

¹⁶ https://drive.google.com/file/d/1cfuPwKSmw-KJuLSEOD9vgcR0l3qLkclK/view?usp=drive_link

¹⁷ Es una URL específica donde se pueden acceder a los recursos o servicios ofrecidos por la API.

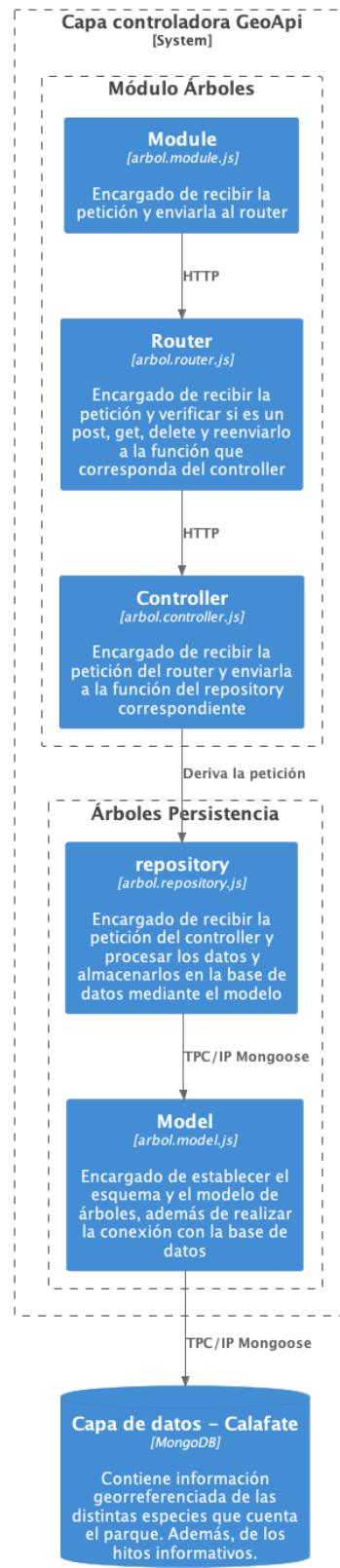


Figura 13 : Modulo de Árboles y su persistencia

Capa de Datos

La capa de datos del sistema GeoApi está construida específicamente para guardar todos los datos de relevancia de un parque o jardín botánico. Estos datos incluyen árboles, fichas de especies de los árboles, fichas informativas, miradores, tótems, senderos, imágenes, entre otros. Por esta razón, cuando se diseñó esta base de datos, se realizó de forma grupal con todo el equipo que tiene un proyecto relacionado con geoparques.

Para la realización del modelamiento de esta base de datos, se llevaron a cabo diversas reuniones con el equipo para analizar la data entregada por el jardín botánico y empezar a conformar esta base de datos. Primero se comenzó por la colección de los árboles y posteriormente se realizaron las fichas de especies que estarían asociadas a un árbol. Para realizar este trabajo, el equipo tuvo que realizar una búsqueda exhaustiva de las distintas especies que cuenta el parque y empezar a recopilar información importante de cada especie.

Cabe mencionar que esta base de datos es de carácter comunitario, es decir, todos los participantes de los distintos proyectos de geoparques pueden utilizarla en sus trabajos de título. En la Figura 14 se presenta la base de datos denominada Calafate con las colecciones utilizadas en este trabajo de título. Para ver la base de datos completa se puede revisar en el Anexo A: Base de datos que se realizó en conjunto con los integrantes de los distintos subproyectos.

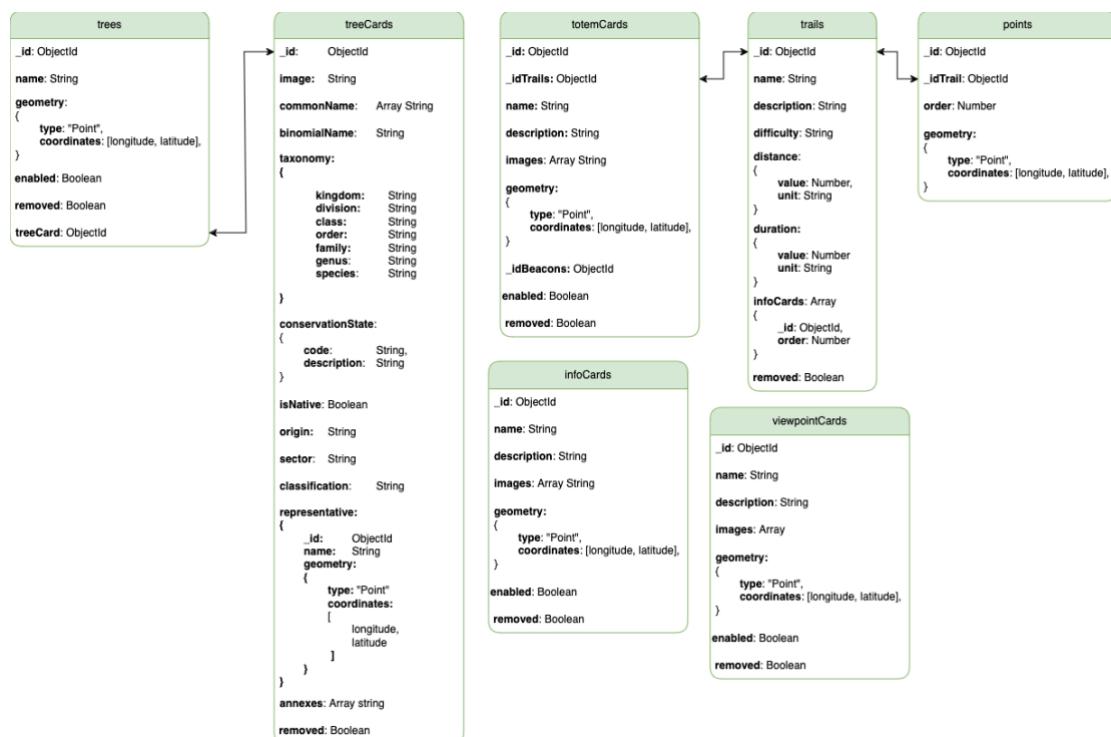


Figura 14 : Base de datos Calafate perteneciente del sistema GeoApi

A continuación, se explican las colecciones más importante para este proyecto de título

- La colección **tres**: almacena documentos de los distintos árboles que cuenta el parque. Además, está relacionada con **treeCards**. Se espera que esta colección, con el pasar del tiempo, contenga información única de cada árbol, ya que, como se mencionó anteriormente, existen árboles con características únicas.
- La colección **treeCards**: almacena la información de las distintas especies pertenecientes al parque, como por ejemplo su nombre científico, nombres comunes, entre otros detalles. En el Anexo B: Descripción de la colección treeCards (Ficha especie), se describe qué significa cada atributo que tiene esta colección.
- La colección **Infocards**: almacena información sobre un hito significativos en geoparques, los cuales informan a los visitantes sobre acontecimientos históricos o eventos temporales relevantes que sucedieron en esos lugares.
- La colección **ViewPointCards** o **miradores**: almacena información sobre otro hito informativo en el que se da a conocer al visitante un mirador importante dentro del parque, mostrando una descripción del mirador y varias imágenes.

Finalmente, la GeoApi es esencial para que la plataforma pueda funcionar y entregar información tanto a los administradores como a los visitantes. Sin embargo, por sí sola no entrega valor para el usuario final, ya que, si se realiza una petición a nuestra capa controladora, esta devolverá un archivo JSON con la información solicitada. Es por esto que a continuación se detalla la arquitectura de las aplicaciones desarrolladas en este trabajo de título.

La arquitectura de las otras partes del sistema, que corresponde a las aplicaciones. Para mantener la explicación concisa y evitar monotonía, se describirán los componentes de manera breve y sin entrar en demasiado detalle cómo se realizó con la GeoApi.

3.4.4 Aplicación Geoparques Visitante

Desde un principio se planeó que cada actor que utiliza el sistema pudiera tener su aplicación por separado. Esto otorga un grado de seguridad a nuestro sistema en general y permite mantener un sistema funcionando en varios componentes de manera independiente.

Dado lo anterior, la aplicación del visitante cuenta con una arquitectura de componentes y sigue la misma lógica que la empleada en la GeoApi, que es un modelo de vista controlador. En la Figura 15, se puede apreciar la arquitectura que sigue esta aplicación.

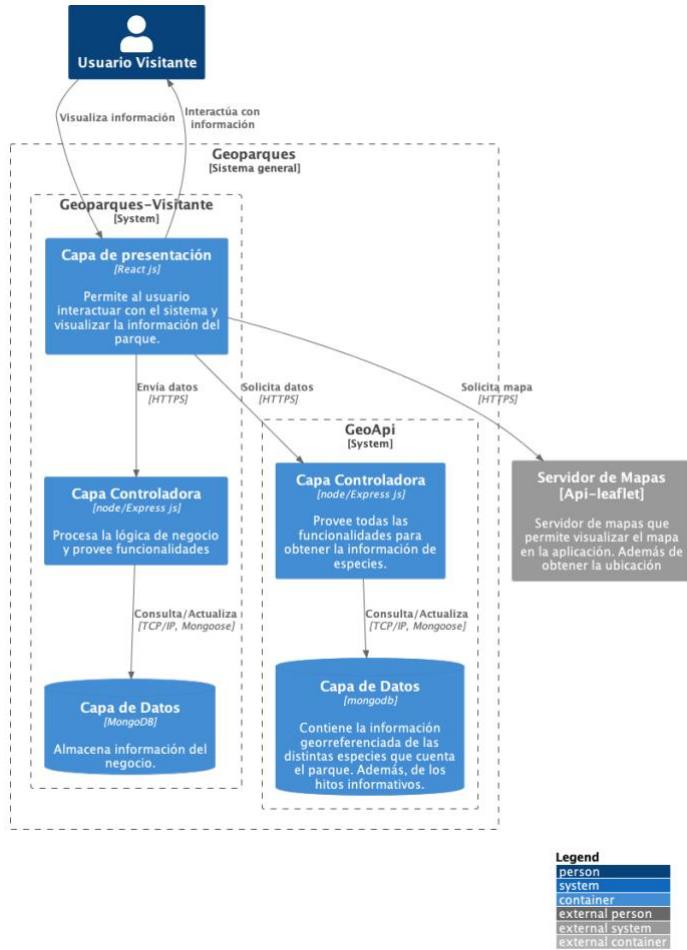


Figura 15 : Arquitectura de la aplicación Geoparques – Visitantes

- **La capa de presentación:** es la aplicación visual con la que el visitante puede interactuar con la información. Como se mencionó anteriormente, se ha desarrollado utilizando una metodología de aplicación web progresiva (PWA), y está construida con React. Incluye componentes esenciales como el escaneo de códigos QR y la obtención de la georreferenciación del visitante. Esta capa se conecta con su propia capa controladora que pertenece al sistema y, además, se enlaza con el sistema de GeoApi mencionado anteriormente para la obtención de información importante de los parques. Finalmente, se conecta al servidor de mapas de Leaflet, que proporciona el mapa para interactuar con él y visualizar la información a través del mapa del parque.
- **La capa controladora:** como se mencionó en la sesión anterior, es la capa encargada de la lógica de negocios y del procesamiento de datos. En este caso, la capa controladora de la aplicación para visitantes de geoparques proporciona datos como las portadas de la aplicación y la información de los posts y eventos que se pueden publicar en la aplicación. Además, se encarga de almacenar los

seguimientos de ingresos de las personas a la aplicación y registrar cuáles son las especies que más les llamaron la atención.

- **La capa de datos:** en el sistema de la aplicación del visitante, proporciona todos los datos necesarios para el funcionamiento del sistema y, además, se encarga de guardar el registro de las personas.

Capa Controladora

Esta está construida siguiendo la misma arquitectura interna que la GeoApi y está desarrollada con Node.js y Express.js, para continuar con el stack de tecnologías que se mencionaron anteriormente.

En la Figura 16, se pueden apreciar los distintos componentes con los que cuenta la controladora de la aplicación del visitante. A continuación, se presentan los más relevantes:

- **Módulo vistaInicial:** se encarga de gestionar la información del texto que el visitante ve al ingresar a la aplicación.
- **Módulo reporteFalloVisitante:** administra y gestiona los reportes que el visitante puede dejar en la aplicación sobre posibles errores en la información proporcionada.
- **Módulo eventos:** gestiona los eventos que el administrador de parques puede incluir en la aplicación del visitante.
- **Módulo trackSistema:** se encarga de administrar el seguimiento de las actividades que realiza el usuario en la aplicación.

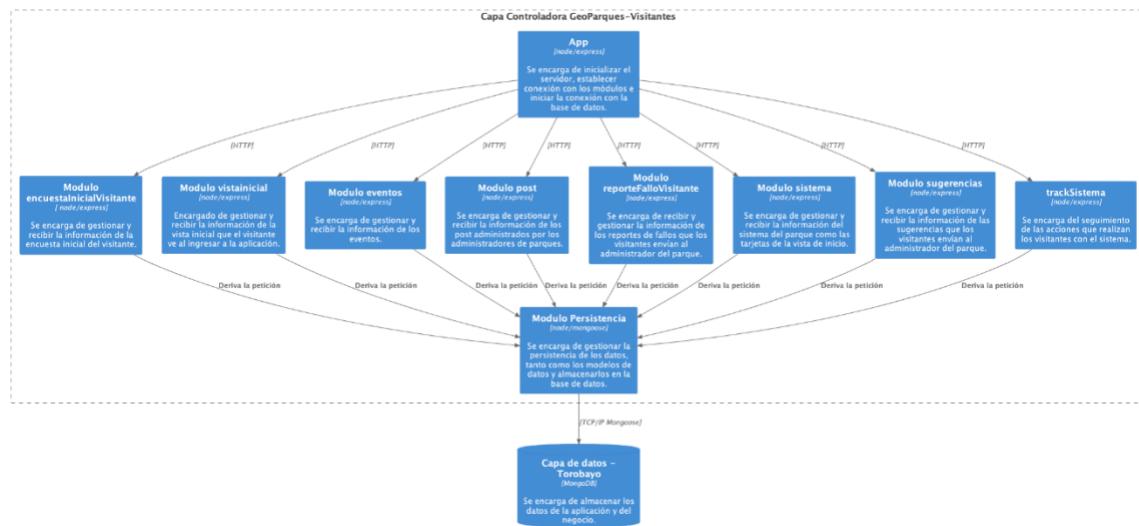


Figura 16 : Capa controladora de la aplicación Geoparques-Visitante

Capa de Datos

La capa de datos del sistema Geoparques–Visitante está diseñada para almacenar información de la aplicación del visitante. El objetivo es que pueda guardar la información que el visitante ve en la aplicación, como por ejemplo, la pantalla de bienvenida. Esta información está almacenada en la base de datos, ya que puede ser modificada por los administradores del parque. Además, guarda información del seguimiento que se realiza al visitante, por ejemplo, cuando ingresa a la aplicación o cuando selecciona una especie o escanea el código QR de una especie; esta información queda almacenada en la base de datos con el objetivo de analizarla posteriormente.

En la Figura 17, se muestran las distintas colecciones que posee la base de datos, denominada Torobayo. Cabe mencionar que, al ser una base de datos no relacional, existen colecciones que contienen documentos que no cuentan con todos los atributos. Por ejemplo, cuando una persona quiere ver una especie y no ha activado la geolocalización de su dispositivo, la posición no se guarda en el documento.

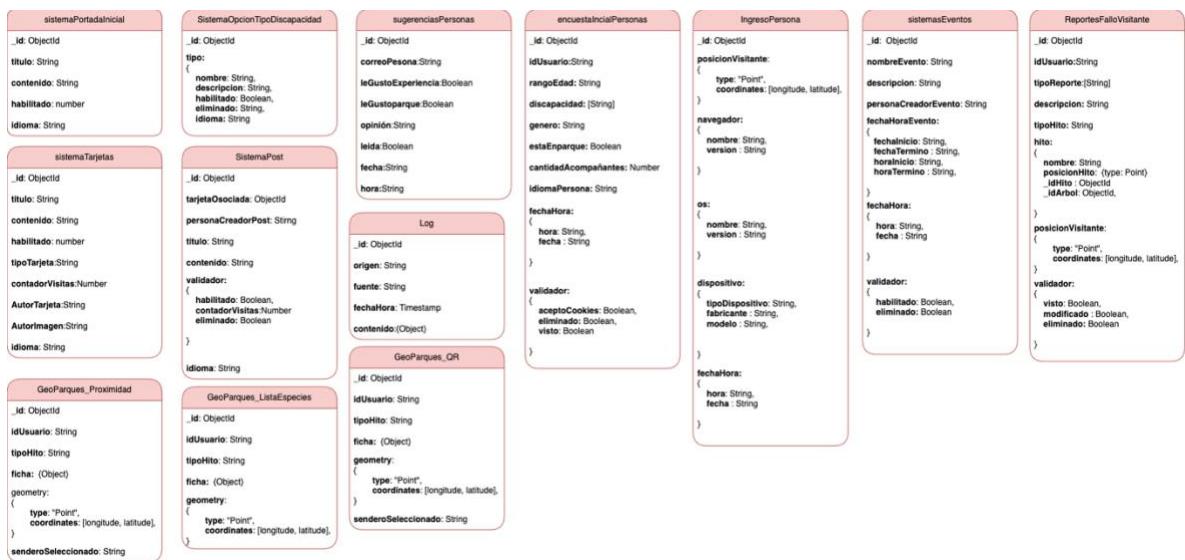


Figura 17 : Base de datos Torobayo perteneciente al sistema Geoparques – Visitante

A continuación. se explican las colecciones más relevantes de la base de datos Torobayo.

Primero se explican las colecciones que empiezan con la palabra 'sistema'. Como se mencionó anteriormente, estas sirven para alimentar a la aplicación con información que el visitante puede observar y leer:

- **SistemaPortadaInicial**: esta colección almacena la información que da la bienvenida al visitante en la aplicación y tiene atributos como título, contenido, habilitado e idioma.

- **SistemaTarjetas:** la colección tarjetas almacena información de dos tarjetas que se pueden visualizar en la vista de inicio de la aplicación. Una tarjeta contiene información del jardín botánico o de cualquier otro parque que utilice el sistema de Geoparques y la otra tarjeta permite visualizar información sobre qué es Geoparques. Cuenta con atributos como título, contenido, tipo de tarjeta, habilitado, entre otros.
- **SistemaEventos:** colección que almacena información de distintos eventos que se van a realizar en el parque. Esta información se puede visualizar en la vista inicial de la aplicación y contiene atributos como nombre del evento, fecha y hora del evento, y validador que sirve para habilitar o eliminar el evento.

Como se mencionó anteriormente, existen colecciones que están orientadas a almacenar información sobre la actividad del visitante en el parque o cualquier apreciación que este tenga:

- **IngresoPersona:** esta colección está destinada a guardar la información de cuando el usuario se conecta a la aplicación. Cuenta con atributos como posición del visitante, navegador, dispositivo, sistema operativo y fecha y hora. En el Anexo C: Descripción de la colección ingreso Persona, se describirá más en detalle cada atributo de la colección.
- **EncuestaInicialPersona:** esta colección guarda información de una encuesta que se realiza al visitante cuando ingresa por primera vez a la aplicación, con la finalidad de obtener información que posteriormente puede ser analizada. Cuenta con atributos como rango etario, género, discapacidad, idioma del visitante, entre otros.
- **ReporteFalloVisitante:** colección que guarda información de reportes que los visitantes puedan hacer sobre alguna especie. Por ejemplo, si se está visualizando una especie y existe información incorrecta, el visitante puede reportarla. Cuenta con atributos como tipo de reporte, descripción, tipo de hitos, hito, entre otros.

Por último, las colecciones que sirven para realizar un seguimiento al visitante. Estas colecciones tienen la finalidad de guardar datos que posteriormente podrán ser consumidos por el módulo GeoAdmin, el cual está siendo desarrollado por un integrante del grupo dedicado a realizar análisis de los datos obtenidos por los distintos proyectos de título que involucran a Geoparques:

- **GeoparquesProximidad:** esta colección guarda información cuando un visitante está navegando por el mapa y selecciona una especie. La especie seleccionada queda almacenada en esta colección, donde se guarda la ID del usuario, la ficha y la geometría, que es la posición en donde el visitante accedió a esa ficha.
- **GeoparquesListaEspecie:** esta colección guarda información de cuando el visitante está en la vista de todas las especies y selecciona una para visualizarla. Cuenta con los mismos atributos que GeoparquesProximidad.
- **Geoparques_QR:** esta colección guarda información cuando el visitante escanea un código QR en el parque. Cuenta con atributos como ID del usuario (si es la

primera vez que ingresa a la aplicación por haber escaneado el QR, queda como usuario anónimo), geometría, ficha, entre otros.

- **Log:** esta colección guarda la información de todas estas colecciones con el fin de que, a la hora de que el integrante del equipo realice la extracción de datos, se puedan hacer de la manera más fácil posible. Cuenta con atributos como origen, fuente, fecha y hora, contenido. Esta colección es parte del trabajo de título de Gustavo Reyes.

3.2.5 Aplicación GeoParques Administrador

La aplicación del administrador del sistema GeoParques es de suma importancia, ya que son ellos quienes administran la información para que posteriormente los visitantes puedan visualizarla. Por ello, se diseñó y desarrolló una arquitectura en la que la aplicación del administrador se pudiera conectar con todos los demás componentes del sistema, con el objetivo de permitir la administración de estos componentes. En la Figura 18, se presenta la arquitectura propuesta.

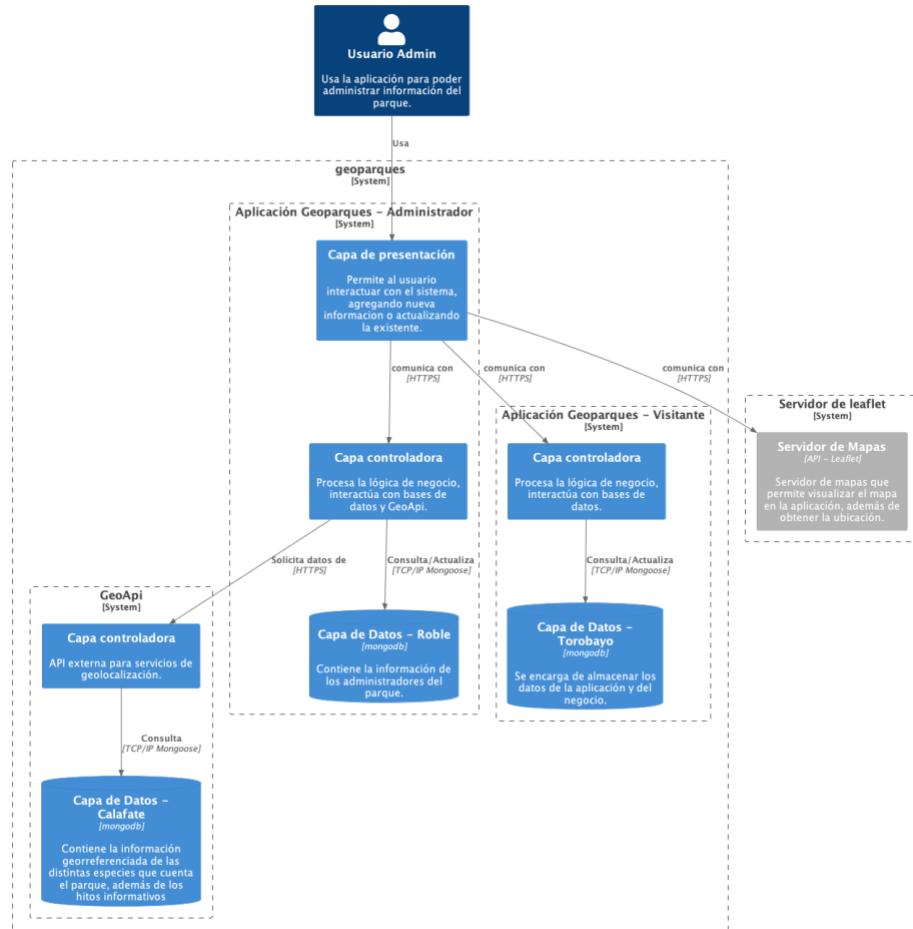


Figura 18 : Arquitectura de la Aplicación Geoparques-Administrador

Se presentan las distintas capas que cuenta la aplicación del administrador de parques.

- **Capa de presentación:** es la capa visual de la aplicación, en donde el usuario puede interactuar con una interfaz de usuario y visualizar agregar/modificar distinta información. Esta capa está conectada a diferentes componentes como a la capa controladora de GeoParques-visitantes, esto con el fin de poder administrar la información que visualiza el visitante, además de poder leer las sugerencias que deja. Además se conecta al servidor de mapa de Leaflet en donde puede recuperar el mapa para visualizarlo en la aplicación.
- **Capa controladora:** esta capa tiene la finalidad de poder procesar los datos del usuario para poder autenticarlo en la aplicación. Además, se conecta con la GeoApi con el objetivo de poder recuperar la información de las especies y poder modificarla correspondientemente. Es importante detallar que el usuario administrador tiene que estar autenticado para poder realizar cambios en la GeoApi.
- **Capa de datos:** esta capa almacena los datos de los distintos administradores con que cuenta el parque.

Capa controladora

La capa controladora de la aplicación del administrador del parque la podemos apreciar en la Figura 19, y cuenta con los siguientes componentes:

- **Módulo Usuario:** Se encarga de agrupar y gestionar las funcionalidades relacionadas con los usuarios. Dentro de este componente se encuentran el module, el router y el controller.
- **Middleware y Proxy:** Es el encargado de recibir las peticiones que van hacia la GeoApi y verifica primero que el usuario que realizó la petición esté autenticado, para luego reenviar la petición.
- **Generar Token:** Este componente asigna un token a cada usuario cuando se autentica en la aplicación, lo cual sirve como medida de seguridad. Está implementado con la biblioteca JWT¹⁸.
- **Envío de Email:** Este módulo se encarga de enviar correos electrónicos a los administradores de parques, por ejemplo, para recuperar contraseña.

¹⁸ JSON web Token, es un estándar para crear tokens de acceso que permiten la transferencia segura de información entre partes como un objeto JSON, utilizado en la autenticación y autorización.

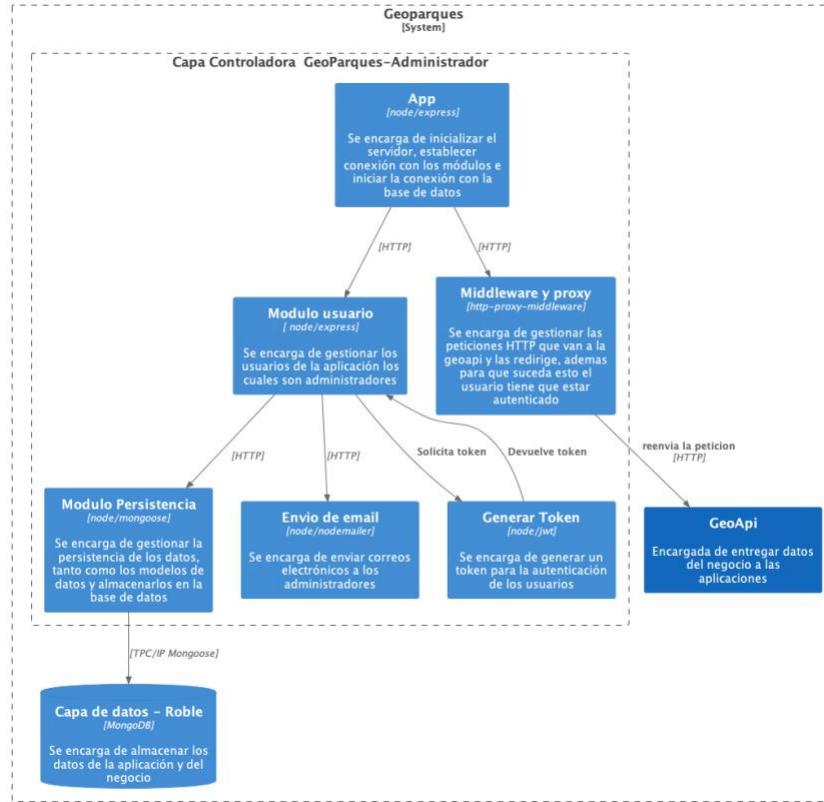


Figura 19 : Capa controladoras de Geoparques-Administrador y sus distintos componentes

Capa de datos

La capa de datos de la aplicación del administrador del parque es una de las más pequeñas, debido a que casi toda la información se almacena en las bases de datos mencionadas anteriormente. En la Figura 20 se puede apreciar la colección con la que cuenta esta base de datos, a la que se ha denominado Roble.



Figura 20 : Colección de la base de datos Geoparques – Administrador

Esta base de datos contiene una colección hasta el momento, y se espera que pueda tener más colecciones con el tiempo.

- **Usuarios:** esta colección contiene los datos de los distintos administradores que puede tener el parque. Con esta información, pueden iniciar sesión en su aplicación. Esta cuenta con atributos como nombre, apellido, email, estado de habilitación, token, entre otros. Por otra parte, la contraseña del usuario se almacena de forma encriptada utilizando el algoritmo bcrypt, el cual genera hashes robustos que protegen eficazmente las contraseñas.

3.2.6 Sistema Geoparques en su Totalidad

En la Figura 21, se presenta el sistema en su totalidad con sus componentes interactuando entre sí. El objetivo de este diagrama es poder ver el sistema funcionando y operando como un todo.

Se puede apreciar cómo la aplicación del visitante interactúa con la GeoApi y con su propia capa controladora. También se observa que la aplicación del administrador consume datos de la GeoApi y de su propia capa controladora, además de interactuar con la capa controladora del visitante para obtener la información esencial para su administración. Por último, se destaca que ambas capas de presentación están conectadas al servidor de mapas de Leaflet para obtener visualmente el mapa y permitir la interacción con él.

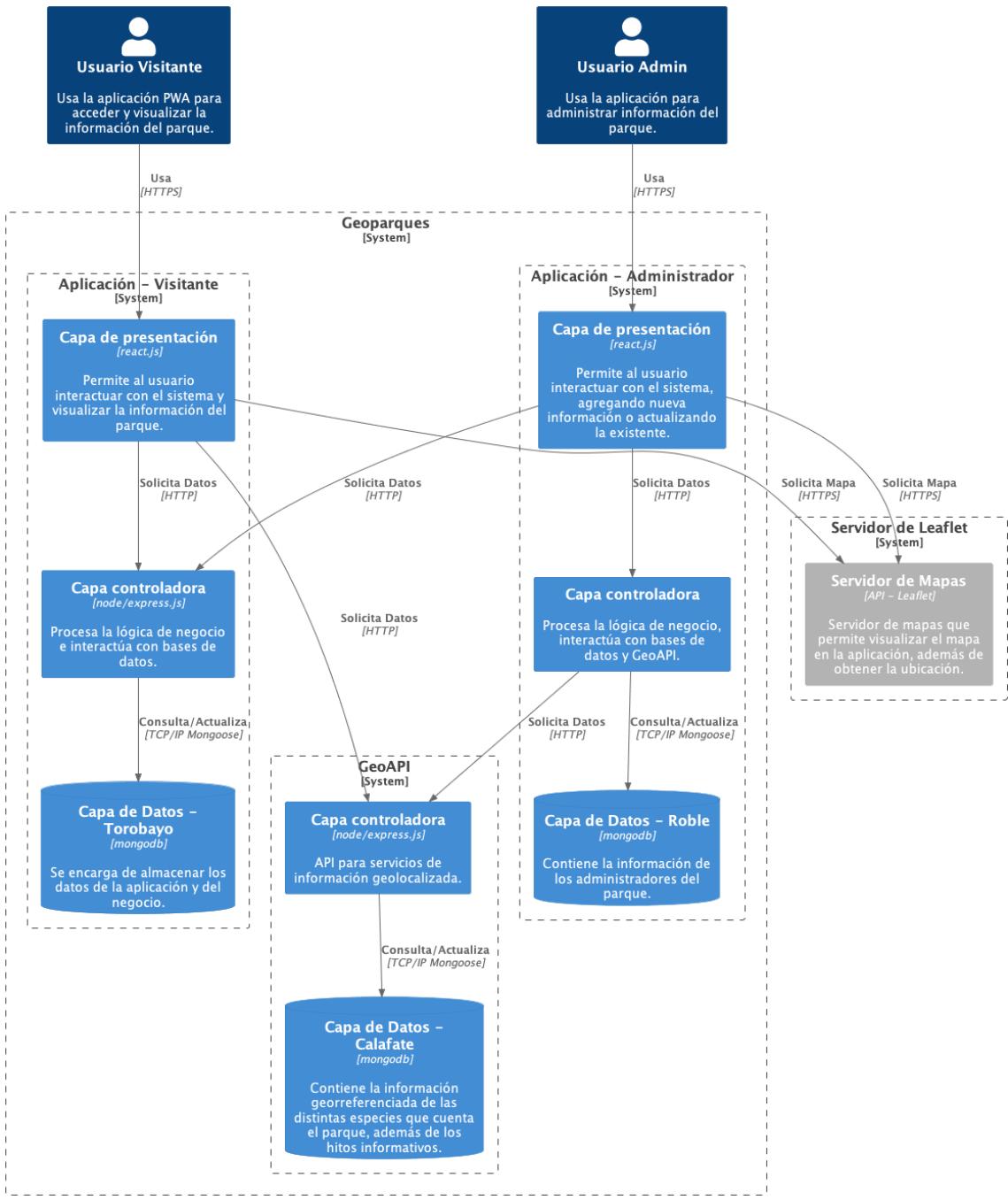


Figura 21 : Sistema GeoParques en su totalidad

4. PLANIFICACIÓN E IMPLEMENTACIÓN DE LA SOLUCIÓN

En esta sección se aborda la planificación efectuada para ejecutar el diseño referido en la sección previa. Se describe la metodología de trabajo y las diversas técnicas empleadas para mantener la organización y el orden durante el desarrollo del proyecto. Además, se detalla cómo se implementó la solución, el proceso de construcción y la manera en que se ejecutó.

4.1 Planificación

Como se mencionó, este proyecto de título fue planificado inicialmente por un equipo de trabajo con el objetivo de definir el alcance de cada subproyecto. Se organizaron reuniones semanales con el fin de detallar el alcance de cada subproyecto. Posteriormente, una vez establecido lo que cada integrante del equipo desarrollaría, se procedió a recabar requisitos y a interactuar con el personal del jardín botánico.

Después, se prosiguió con el trabajo para dar inicio al desarrollo de la aplicación, el cual se llevó a cabo empleando técnicas ágiles detalladas en la siguiente sección. El propósito de este enfoque era organizar el proceso de desarrollo tiene como fin aumentar la eficiencia y definir de forma clara el alcance de las distintas funcionalidades que el sistema necesita para satisfacer los requerimientos del usuario final. Además, se creó una carta Gantt, que se puede consultar en el Anexo D: Primera Carta Gantt, con el objetivo de tener una planificación semanal de las tareas a desarrollar.

Sin embargo, esta carta Gantt no se siguió al pie de la letra, ya que el proceso de desarrollo de las distintas funcionalidades del sistema se aceleró, resultando en tiempo adicional que permitió la incorporación de nuevas funcionalidades.

4.2 Metodología de Trabajo

4.2.1 Scrum

Como se mencionó en la sección anterior, el objetivo era llevar a cabo un desarrollo lo más ágil posible, con el fin de que este fuera adaptable a cualquier cambio de circunstancia, ya sea un cambio en los requisitos o en la tecnología utilizada. Es por esto que se implementaron varias técnicas ágiles, siendo una de las principales **Scrum**. El cuál es el marco de trabajo e ágil de desarrollo de software más utilizado del mundo (Deloitte, s. f.). Fue presentado por Jeff Sutherland y Ken Schwab en 1995. Hoy en día, un gran número de equipos de desarrollo de software en el mundo lo usan o están en vía de usarlo.

Esta metodología de trabajo contempla un conjunto de componentes claves y de buenas prácticas como las que se pueden apreciar en la Tabla 14

Tabla 14 : Componentes y buenas prácticas de metodología de trabajo Scrum

Componentes claves	Buenas prácticas
Definir roles en los equipos de trabajo como el producto Owner, Scrum Master	Fomentar un entorno donde la comunicación fluya abiertamente entre todos los miembros del equipo.
Definir Periodos de desarrollo de igual duración, generalmente de dos a cuatro semanas, durante los cuales se realiza una porción significativa del trabajo	Flexibilidad y adaptación al cambio, poder adaptarse a cambios en los requisitos o en el entorno del proyecto
Reuniones diarias de scrum (daily), en donde el equipo discute el progreso y aborda problemáticas durante el desarrollo.	Entrega continua de valor, en donde se realiza entregas parciales y regulares del producto en donde permite obtener retroalimentación temprana y asegura que el producto final cumpla con las expectativas del cliente
Reuniones de planificación, revisión o retrospectiva de sprint, en donde se planifica el próximo sprint.	Fomentar la transparencia y la mejora continua, permitiendo al equipo reflexionar sobre su desempeño, identificar obstáculos y aplicar cambios para optimizar su eficiencia y calidad del trabajo.

Sería ideal que los equipos de trabajo pudieran utilizar esta metodología tal y como la describen sus creadores. Sin embargo, cada equipo de trabajo adapta Scrum a sus necesidades y conveniencia, ya que a veces resulta difícil seguirlo al pie de la letra. En este trabajo de título, se utilizaron algunos de los componentes clave y buenas prácticas de Scrum que se detallan a continuación:

- **Reuniones semanales:** Se implementaron reuniones semanales para revisar los estados de avance en la programación de los módulos de visitante y administrador. En los últimos 20 minutos de cada reunión, se planificaban los objetivos para la semana siguiente.
- **Flexibilidad:** Durante el desarrollo, algunos de los requisitos cambiaron, lo que requirió adaptaciones en el modelo de datos. Por esta razón, si se habían realizado avances en la programación y algo cambiaba, era necesario efectuar las modificaciones correspondientes.
- **Entrega continua de valor:** En todas las reuniones, tanto con el equipo como con el cliente, se mostraban los avances de programación. Esto permitía a las partes interesadas visualizar cómo funciona el sistema, además de obtener retroalimentación sobre lo construido hasta el momento. Así, era posible captar nuevos requisitos que podían surgir a partir de estas entregas tempranas.
- **Sprint:** Se intentó seguir el esquema de sprints para el desarrollo de la aplicación. Sin embargo, debido a un proceso acelerado en la construcción del software, se desarrollaron funcionalidades antes del tiempo planificado. Por lo tanto, se continuó programando con el resto de las funcionalidades que no correspondían al sprint planificado.

4.2.1 User Story Mapping

Otra de las técnicas que se utilizó para la construcción del sistema fue la utilización de un *User Story Mapping*, el cual es un ejercicio visual que ayuda a los gerentes de productos y sus equipos de desarrollo a definir el trabajo que creará la experiencia de usuario más placentera (Scrum.org, 2022). Esta técnica emplea el concepto de las historias de usuario, que comunican los requisitos desde la perspectiva del valor del usuario.

La idea general es posicionar las historias de usuario según su importancia y, mediante iteraciones que se realizan con el equipo de trabajo, posicionar las más importantes en la parte superior de la pizarra y las que son características secundarias en la parte inferior. Después de esto, se planifica hasta dónde va a llegar el primer *reléase* (para tener un producto mínimo viable). Esto se realiza posicionando en la pizarra una línea horizontal que divide las historias de usuario, donde las que quedan en la parte superior entran en el *reléase* y las de la parte inferior quedan para el siguiente. Esto se hace para poder tener las funcionalidades mínimas y más importantes para tener un sistema funcionando y poder mostrarlo al cliente con el fin de recibir retroalimentación continua.

Para la ejecución de este proyecto se realizó con el equipo de trabajo y el profesor patrocinante, se decidido realizar 3 *reléase*. A continuación, se presentará en la Figura 22 *User Story Mapping* de usuario para la aplicación de los visitantes. Además, en el Anexo E: *User history mapping* de la aplicación administrador, se encontrará el correspondiente a los administradores de parques.

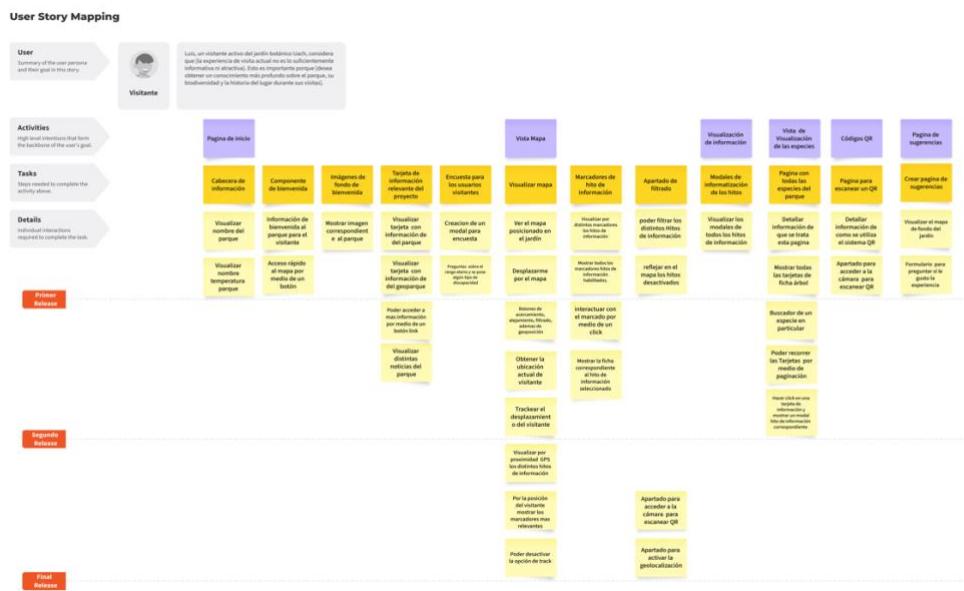


Figura 22 : *User story mapping* del usuario visitante¹⁹

¹⁹ https://drive.google.com/file/d/1me_xW35C_4oERyrTmI5PQMZP8WI-Yw8p/view

4.2.2 Tablero Kanban

Una técnica para planificar un desarrollo ágil fue la utilización de un tablero Kanban. El cual es una herramienta visual de gestión de proyectos que sirve para mapear y visualizar el flujo de trabajo, limitar el trabajo en curso y maximizar la eficiencia (DoneTonic, s. f.). Unos de sus objetivos principal es definir y delimitar el trabajo de una persona o grupo de trabajo.

El tableo se puede visualizar en el Anexo F: Tablero Kanban. Esté cuenta con cuatro columnas que detallaban el trabajo no empezado, el que tenía que empezar, en progreso y el terminado. Durante la mayor parte del tiempo, una gran cantidad de tareas estuvieron en la etapa de progreso. Este tablero sirvió para dar cuenta de la gran cantidad de trabajo que conllevaba la creación de distintos módulos de programación.

4.2.3 Bitácora de Trabajo

Como último recurso, se utilizó una bitácora simple de trabajo, que es un registro detallado de las actividades realizadas durante el día. En el Anexo G: Bitácora, se presenta la bitácora que se llevó a cabo, donde se anotaba la fecha y el trabajo realizado ese día. Lamentablemente, la bitácora no contempla todos los días utilizados para el desarrollo, debido a que hubo días en los cuales no se realizaron anotaciones por olvido.

4.3 Implementación de la Solución

En esta subsección, se detalla cómo se construyeron las distintas aplicaciones y se describe el desarrollo necesario para que la aplicación del visitante funcione como una aplicación web progresiva (PWA). Además, se explica cómo se implementaron el sistema de Código QR y el de Geoproximidad.

4.3.1 Desarrollo de la GeoApi

Lo primero que se desarrolló para poder gestionar la información fue la GeoApi. Una vez lista la base de datos Calafate, se procedió al desarrollo de la capa controladora. En la Tabla 15, se muestra un extracto de los endpoints creados para consultar la información de los árboles del parque. Para una revisión completa de todos los endpoints desarrollados, estos se encuentran detallados en el Anexo H: Rutas (endpoint) de información de la GeoApi.

Tabla 15 : Extracto rutas de la capa controladora GeoApi

	Tipo	Ruta	Descripción
A r b o l e s	POST	/api/arboles	Permite crear un árbol.
	GET	/api/arboles	Obtiene todos los árboles.
	GET	/api/arboles-habilitado	Obtiene los árboles que están habilitados en sistema.
	GET	/api/arboles-filter-ficha/:id	Obtiene los árboles según una ficha árbol asociada (obtiene los árboles de una especie correspondiente).
	GET	/api/arboles/:id	Obtiene un árbol según su id.
	PUT	/api/arboles/:id	Actualiza la información de un árbol según su id.
	PUT	/api/arboles-deac/:id	Desactiva un árbol según su id.
	PUT	/api/arboles-acti/:id	Activa un árbol según su id.
	DELETE	api/arboles/:id	Elimina el árbol del sistema según su id.

4.3.2 Desarrollo de la Aplicación del Administrador

Como se ha mencionado anteriormente, es de suma importancia el desarrollo de este módulo, ya que se buscaba que el administrador de parques pudiera revisar y actualizar la información de la aplicación que tiene el visitante.

4.3.2.1 Capa de presentación

Para la construcción de la aplicación se utilizó el *User Story Mapping* que se menciona anteriormente, en donde, se establecieron prioridades para determinar cuáles

funcionalidades se construirían con mayor prioridad. La siguiente lista muestra el orden de construcción:

- Iniciar sesión.
- Visualizar el mapa con los distintos hitos informativos que se están visualizando en la aplicación del visitante.
- Visualizar y agregar árboles.
- Ver y administrar ficha especie.
- Visualizar y agregar fichas informativas.
- Panel para la administración de la aplicación del visitante.

Como se mencionó en el Capítulo 3, esta aplicación se realizó con React, facilitando la construcción mediante el uso y reutilización de componentes. Para dar estilo a estos componentes, se utilizó Tailwind.

Además, en esta sección se detalla la construcción de los puntos mencionados. Es importante destacar que todos los componentes visuales que se muestran a continuación se realizaron desde cero.

4.3.2.1.1 Inicio de Sesión

La construcción de este componente de la aplicación no fue compleja, y el desarrollo no requirió demasiado tiempo. Además, esta parte de la aplicación, como se pudo notar, no estaba contemplada en los requisitos funcionales de la aplicación, esto debido a que se asume que una aplicación para la administración debe tener un inicio de sesión. En la siguiente Figura 23, se puede apreciar cómo se ve esta vista.

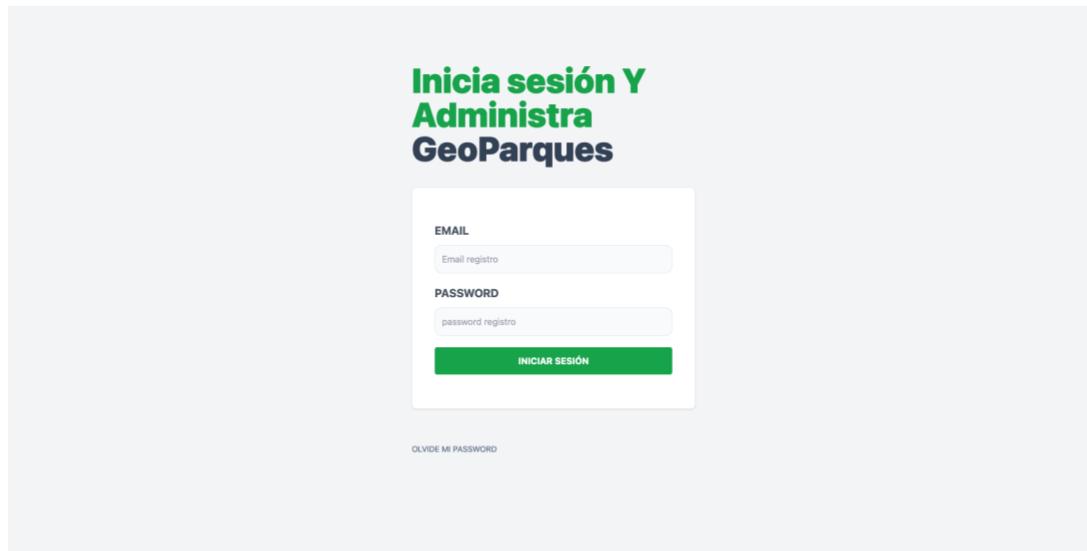


Figura 23 : Vista de Inicio de sesión

Cabe mencionar que cuando un parque decide contratar el sistema de GeoParques, se le asignarán credenciales con las cuales podrán ingresar al sistema para administrar la información. Además, si desean crear una nueva cuenta para otro administrador, pueden realizarlo dentro de la aplicación.

4.3.2.1.2 Vista Mapa

La vista del mapa está diseñada y desarrollada con el objetivo de que los administradores de parques puedan visualizar, en tiempo real, cómo se está viendo el mapa en la aplicación de los visitantes, sin que necesariamente tengan que ingresar a esta aplicación.

El diseño de esta vista, como se puede apreciar en la Figura 24, consiste en una pantalla dividida donde en la parte izquierda se pueden ver los marcadores (hitos informativos) y el administrador puede seleccionar los que deseé visualizar. En la parte derecha, se muestra el mapa del parque con los marcadores posicionados correspondientemente. Es importante mencionar que este patrón de diseño visual se mantiene en el resto de las vistas.

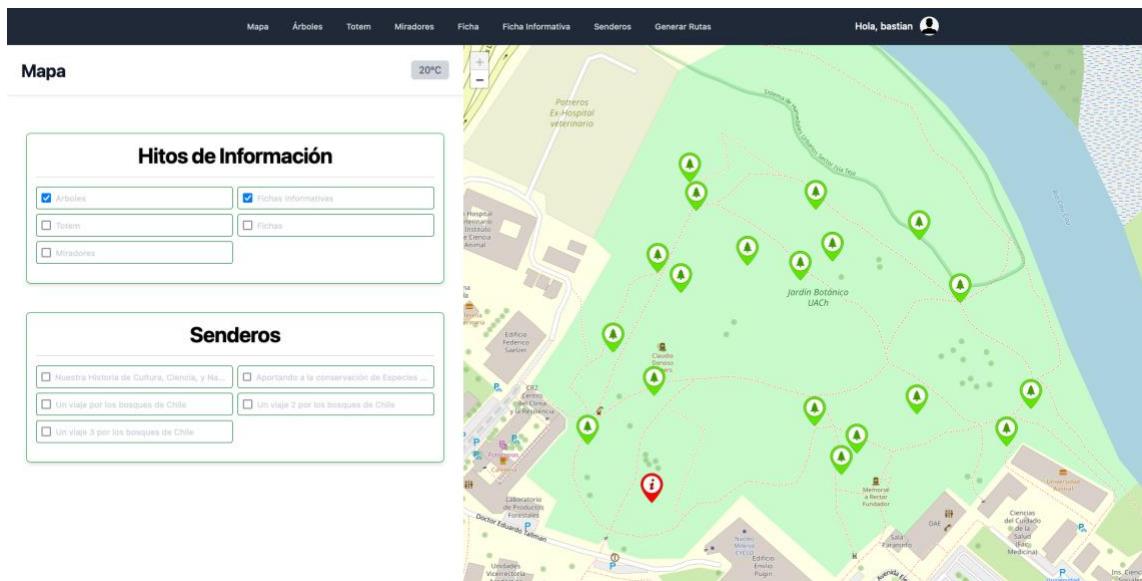


Figura 24 : Vista del Mapa

4.3.2.1.3 Vista Visualizar y Agregar Árboles

En esta vista, los administradores de parques podrán administrar la información de todos los árboles que cuenta el parque y los que se podrán visualizar en el mapa de la aplicación de los visitantes, como se puede apreciar en la Figura 25. En la parte izquierda, se puede ver el formulario para añadir o editar un árbol. Este formulario incluye los siguientes campos:

- Nombre Científico
- Selector para elegir a qué Ficha Especie corresponde.
- Latitud y longitud, donde el administrador puede escribir los datos manualmente o puede presionar el botón de localización, que es útil si está utilizando la aplicación en su dispositivo móvil y se encuentra en terreno.

En el lado izquierdo, se pueden apreciar todos los árboles registrados en el sistema, los cuales aparecen en el mapa representados por marcadores. Los marcadores de color verde indican árboles que están activados y, por lo tanto, son visibles en el mapa del visitante. Por su parte, los marcadores de color gris señalan a aquellos árboles que están desactivados.

Al hacer clic en un marcador, aparecerá un popup²⁰ que muestra información general del árbol y botones de configuración, donde se pueden realizar las siguientes acciones:

- Editar
- Ver ficha asociada
- Activar o desactivar el árbol
- Eliminar

En la Figura 25, también se visualiza el botón “abrir lista”, al ser presionado esconderá el mapa y mostrará los árboles en forma de lista, donde se podrán realizar las mismas acciones ya mencionadas. Este tipo de diseño y acciones se repiten en los otros hitos de información, como miradores y tótems.

²⁰ Es una ventana emergente que aparece sobre el contenido de una aplicación, utilizada para mostrar información adicional, formulario o para captar la atención del usuario.

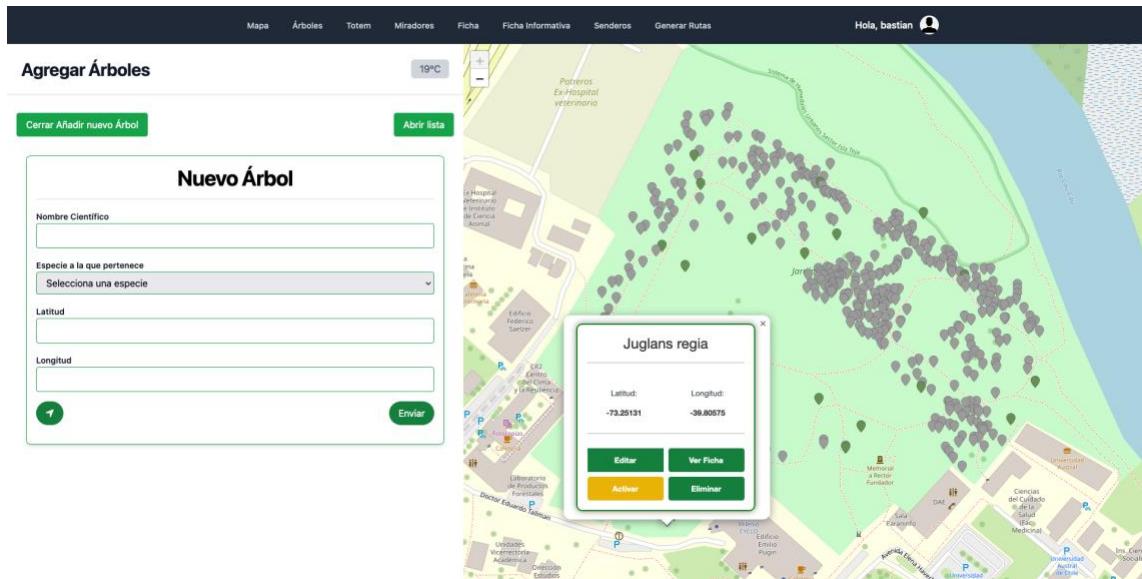


Figura 25 : Vista añadir y actualizar árboles

4.3.2.1.4 Vista Ver y Administrar Ficha Especie

En esta vista, los administradores del parque podrán gestionar las fichas de las especies, como se puede apreciar en la Figura 26. En la parte derecha, se visualizan en forma de lista todas las especies, que se pueden recorrer por medio de una paginación²¹ o buscándolas a través de un buscador ubicado en la parte superior derecha. Cada especie en la lista cuenta con botones de acciones, los cuales son los siguientes:

- **Editar:** permite editar la información de la ficha. Esta acción está fuera del desarrollo de este proyecto de título.
- **Ver Ficha:** permite visualizar la ficha de la especie.
- **QR:** al presionar el botón, se abrirá un modal con el código QR correspondiente a la ficha (Los detalles técnicos de la implementación se revisará en el Capítulo 4).
- **Eliminar:** permite eliminar la ficha informativa.

En la parte izquierda de la Figura 26, se puede apreciar el botón para crear la especie que, al igual que el botón de editar, están fuera del alcance de este proyecto de título.

²¹ Esta es una técnica utilizada en aplicaciones web para dividir grandes cantidades de datos en páginas más pequeñas y manejables, permitiendo a los usuarios navegar de manera incremental.

The screenshot shows a web-based application for managing tree species information. At the top, there's a navigation bar with links for 'Mapa', 'Árboles', 'Totem', 'Miradores', 'Ficha', 'Ficha Informativa', 'Senderos', and 'Generar Rutas'. On the right, it says 'Hola, bastian' with a user icon. Below the navigation is a search bar labeled 'Buscar Ficha' with a magnifying glass icon.

NOMBRE FICHA	ID	ACCIONES
Populus nigra	64ecbf2caf059ee883b59c76	Editar Ver Ficha QR Eliminar
Pseudotsuga menziesii	64ecbf2caf059ee883b59c77	Editar Ver Ficha QR Eliminar
Quercus nigra	64ecbf2caf059ee883b59c78	Editar Ver Ficha QR Eliminar
Quercus palustris	64ecbf2caf059ee883b59c79	Editar Ver Ficha QR Eliminar
Quercus robur	64ecbf2caf059ee883b59c7a	Editar Ver Ficha QR Eliminar
Quercus suber	64ecbf2caf059ee883b59c7b	Editar Ver Ficha QR Eliminar
Robinia pseudoacacia	64ecbf2caf059ee883b59c7c	Editar Ver Ficha QR Eliminar
Sequoia sempervirens	64ecbf2caf059ee883b59c7d	Editar Ver Ficha QR Eliminar
Taxus baccata	64ecbf2caf059ee883b59c7e	Editar Ver Ficha QR Eliminar
Ulmus carpinifolia	64ecbf2caf059ee883b59c7f	Editar Ver Ficha QR Eliminar

Below the table is a navigation bar with buttons for 'Prev', page numbers (1-9), and 'Next'.

Figura 26 : Vista para administrar ficha especie

4.3.2.1.5 Vista Visualizar y Agregar Ficha Informativa

Las fichas informativas sirven para avisar al visitante sobre algún tipo de suceso ocurrido en el parque. Esta vista se parece mucho a la de agregar árboles, pero tiene diferencias significativas, como se puede apreciar en la Figura 27. En la parte derecha, no se encuentra el formulario para agregar una nueva ficha informativa, ya que esta opción en esta vista se realizó en forma de modal. Esto debido a que se quiso probar y experimentar nuevas formas de agregar la información.

Además, en esta vista existe otra forma de agregar una ficha, la cual está marcada con un círculo rojo que se puede apreciar en la Figura 27. Al momento de presionar este botón, aparecerá un marcador que se puede desplazar por todo el mapa y el administrador podrá decidir dónde posicionarlo, esta acción se puede apreciar en el siguiente video²². Una vez definido el lugar para colocar el marcador, al presionar clic izquierdo aparecerá el modal para llenar la información, que cuenta con los siguientes campos.

- Título
- Imagen
- Descripción
- Latitud y longitud.

²² https://www.youtube.com/watch?v=EoLtR3vWeSI&ab_channel=bastidebasti

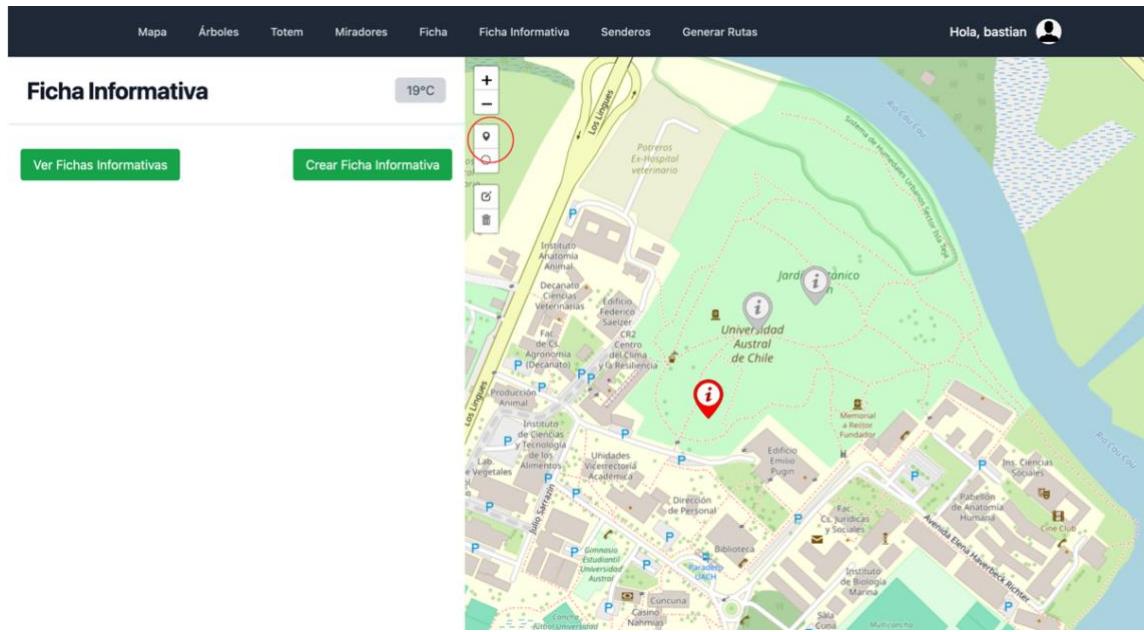


Figura 27 : Vista para administrar fichas informativas

4.3.2.1.6 Panel de Administración de la Aplicación del Visitante

En este apartado de la aplicación del administrador se puede gestionar parte del texto que visualiza el visitante, además de administrar sugerencias propuestas por los visitantes. En la Figura 28, se puede apreciar este panel en la vista Administrar bienvenida, donde, al igual que en las otras vistas, en la parte derecha se utiliza para administrar la información y en la parte izquierda está diseñada para escoger lo que se quiere gestionar, contando con las siguientes opciones:

- **Portada inicial:** permite que el administrador modifique el texto de bienvenida que los visitantes visualizan cuando ingresan por primera vez a la aplicación.
- **Tarjeta A:** el administrador puede actualizar la información que aparece en forma de tarjeta en la aplicación del visitante.
- **Tarjeta B:** esta tarjeta se trabaja de manera separada, ya que cuenta con información sobre qué es Geoparques; el administrador puede cambiar esta información por otra de su elección o dejar la misma.
- **Eventos:** se pueden administrar los eventos que sucederán en el parque para que los visitantes tengan conocimiento de estos.
- **Sugerencias:** el administrador puede visualizar todas las sugerencias que le dejan los visitantes en el parque.
- **Reportes:** se pueden visualizar y administrar los reportes que los visitantes dejan sobre algún error en la información de las especies.

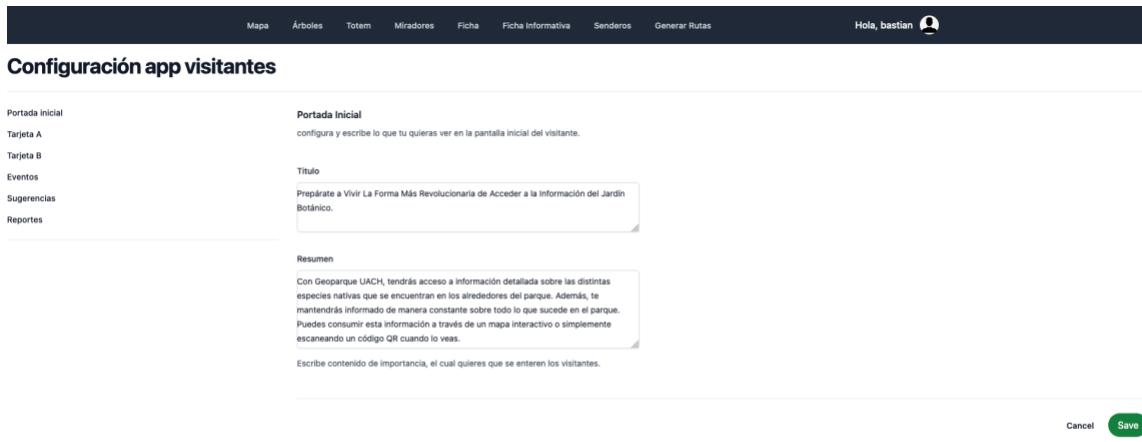


Figura 28 : Configuración app visitante, panel de bienvenida

4.3.2.2 Capa Controladora

La capa controladora de la aplicación del administrador está especialmente desarrollada para gestionar los datos de los usuarios (administradores del parque) y autenticarlos en el sistema. Además, permite conectarse con la GeoApi para obtener la información necesaria. En la Tabla 16 se muestra un extracto de los endpoints utilizados para obtener la información del usuario. Para una revisión más detallada de todos los endpoints disponibles en esta capa, se pueden consultar en el Anexo I: Rutas de la capa controladora de la aplicación del administrador.

Tabla 16 : Extracto de las rutas de la capa controladora aplicación administradores

	Tipo	Ruta	Descripción
U s u a r i o	POST	api/usuarios/login	Permite iniciar sesión en el sistema al administrador de parque.
	POST	api/usuarios/registrarAdmin	Permite crear un usuario administrador en la aplicación de geoparques-administrador (solo un administrador puede crear otra cuenta para otro administrador)
	GET	api/usuarios/confirmar_/:token	Permite confirmar el token de seguridad de una cuenta administradora.
	POST	api/usuarios/activar_cuenta/:token	Permite activar una cuenta al momento de registrar un nuevo administrador.

4.3.3 Desarrollo de la Aplicación del Visitante.

Como ya se ha mencionado anteriormente, esta aplicación está diseñada para que los visitantes de los parques turísticos puedan visualizar toda la información disponible en el sistema. Está desarrollada para ser multiplataforma, permitiendo su uso tanto en teléfonos celulares, tabletas como computadores. Esto es posible gracias a que está creada con un conjunto de tecnologías que permiten implementar una aplicación web progresiva (PWA), la cual nos permite utilizarla en todos los dispositivos que cuenten con un navegador de internet.

A continuación, se presenta la capa de presentación de esta aplicación, donde se examinarán las distintas vistas con las que cuenta esta aplicación y, posteriormente, se detallará cómo fue construido el sistema de geolocalización y del código QR.

4.3.3.1 Capa de Presentación

Este desarrollo se realizó siguiendo el *User Story Mapping* del visitante y se desarrolló de acuerdo con las prioridades establecidas en éste. Las cuales son las siguientes:

- Vista Inicial
- Vista Mapa
- Vista especies
- Vista QR

Además, se explica cómo se realizó la implementación de uno de los requisitos no funcionales, el cual es que la aplicación tenga la opción de cambiar su idioma a inglés.

Para la construcción de esta capa, se utilizaron las mismas tecnologías que en la aplicación del administrador, pero se agregaron algunas librerías para facilitar la construcción de los componentes. Entre estas se incluyen React, Tailwind y una biblioteca para este, que es DaisyUI²³.

Si se desea indagar más en todas las funcionalidades de la aplicación del visitante, esto estará disponible en el Anexo J: Manual de uso aplicación GeoParques Visitantes.

A continuación, se muestran las vistas en el mismo orden en que fueron desarrolladas.

4.3.3.1.1 Vista de Bienvenida

La Vista de Bienvenida o más conocida como pantalla de inicio, está diseñada y construida para proporcionar una bienvenida a los visitantes del parque cuando entran a la aplicación. Como se puede apreciar en la Figura 29, además del texto de bienvenida, se muestra un

²³ Es un plugin para Tailwind que proporciona componentes de interfaz de usuario preconstruidos y tematizables, facilitando el diseño de aplicaciones web sin la necesidad de escribir mucho CSS desde cero.

botón de acceso directo que, al presionarlo, dirige al mapa. Si se desliza hacia abajo, se muestran las tarjetas de información y los eventos que se realizarán en el parque.

El diseño para móviles presenta la barra de navegación en la parte inferior de la pantalla y tiene un botón al lado izquierdo donde los visitantes si lo presionan podrán visualizar y escoger a qué vista quieren navegar.



Figura 29 : Vista de inicio en la aplicación del visitante

4.3.3.1.2 Vista Mapa

Es considerada una de las vistas más importantes de la aplicación, ya que en esta se pueden utilizar casi todas las funcionalidades que contiene Geoparque - Visitante. Su diseño permite que el visitante pueda llegar a esta y utilizar todas las funcionalidades en un solo lugar sin tener que navegar a otras partes de la aplicación para obtener información.

En la Figura 30, se aprecia esta vista donde se visualiza el mapa del parque en toda la pantalla. En el mapa, se pueden ver posicionados los distintos marcadores que representan los hitos de información. Al hacer clic en alguno de estos marcadores, se despliega un modal con la información correspondiente al hito seleccionado. Además, en esta vista se puede observar que existe un botón en la parte superior derecha que, al presionarlo, la aplicación entregará la posición en la que se encuentra el visitante dentro del parque.

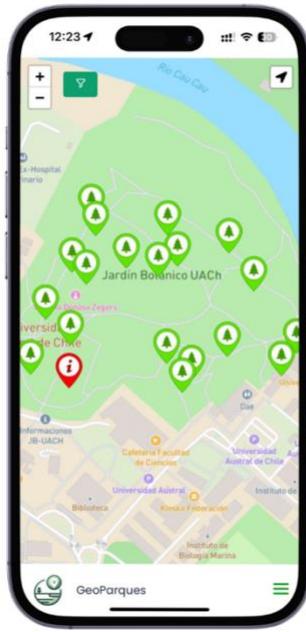


Figura 30 : Vista mapa

En la parte superior izquierda se encuentra un botón que se denomina “Panel de filtrado”. Al presionarlo, se abrirá un *Drawer*²⁴ como se puede apreciar en la Figura 31, el cual cuenta con las siguientes características:

- **Buscador:** esta opción permite buscar una especie por su nombre común. Al momento de escribir, se despliega un panel donde se visualizan las especies a partir de lo que se ha escrito hasta ese instante. Al seleccionar una especie, *Drawer* se cierra automáticamente y se muestra en el mapa mediante marcadores dónde se encuentra la especie seleccionada.
- **Menú de filtrado:** opción que permite al visitante filtrar por qué hito de información quiere visualizar. Simplemente tiene que deseleccionarlo en el checkbox y desaparecerá del mapa.
- **Botón de Geoproximidad:** este sirve para activar la función de Geoproximidad, en donde al activarla se esconden todos los árboles del mapa y se visualizan los que están más cerca de la geo posición del visitante.
- **Botón escanear QR:** al momento de presionar este botón se despliega un modal que muestra un marco con la cámara, en la cual se puede escanear un código QR.
-

²⁴ En el diseño de interfaces es un panel deslizante que puede aparecer desde un lado de la pantalla, se utiliza para alojar un menú de navegación o controles adicionales en aplicaciones móviles o web.

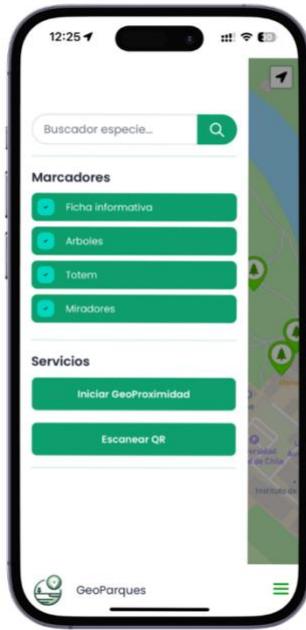


Figura 31 : *Drawer* con opciones para interactuar con el mapa

4.3.3.1.3 Vista especie

La vista está diseñada para que los visitantes del parque puedan conocer todas las especies por medio de unas tarjetas que cuenta con una imagen de la especie, su nombre común y su nombre científico, esta se puede apreciar en la Figura 32.

Además, al momento de presionar una tarjeta, se abre un modal con información de la especie. Esta también cuenta con un botón que tiene la icono de un marcador; al presionarlo, se dirige a la vista del mapa donde se mostrará dónde está posicionada esta especie en el parque. Por último, esta vista incluye un buscador que, al igual que el del *Drawer* del mapa, permite buscar las especies a partir de su nombre.

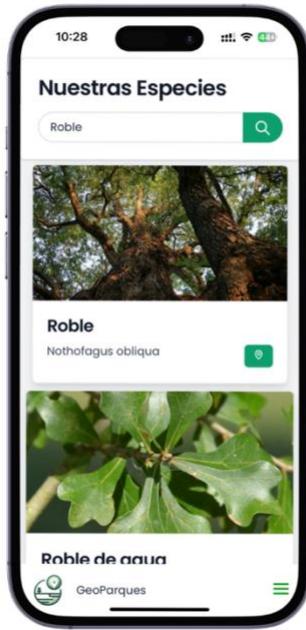


Figura 32 : Vista de todas las especies en el parque.

4.3.3.1.4 Vista QR

Esta vista está diseñada para que los visitantes o usuarios que utilicen la aplicación se enteren de que dentro del parque existen distintos códigos QR posicionados en los árboles con el fin de que sean escaneados si desean revisar más información de la especie correspondiente. En la Figura 33, se puede apreciar esta vista, que cuenta con un título, información y, finalmente, un botón tipo texto que se puede presionar para que aparezca el modal con la cámara y se pueda escanear el QR. Al momento de escanear un QR, el sistema dirige al usuario a la vista del mapa donde le mostrará el modal con la información correspondiente a la especie escaneada.



Figura 33 : Vista QR

4.3.3.2 Desarrollo del *Service Worker* para la Habilitación de la Aplicación como PWA

Una de las características principales de la aplicación del visitante es que se puede utilizar sin necesidad de instalarse desde una tienda de aplicaciones. Por este motivo, es esencial integrar en la aplicación del visitante, construida con React, un script llamado *Service Worker*. Este script permite que la aplicación web ofrezca características como la instalación en el dispositivo, capacidad de funcionar offline y recepción de notificaciones. Es importante mencionar que esta aplicación está diseñada para usarse siempre con internet; por lo tanto, la funcionalidad offline no es un objetivo de este trabajo de título.

En Figura 34, se muestra el script desarrollado para el *Service Worker* de la aplicación del visitante. Desde la línea 2 hasta la 14, se invoca el evento para que la aplicación instale el *Service Worker* en el dispositivo donde se está ejecutando, agregando caché y los archivos estáticos necesarios para su funcionamiento. En las líneas 16 hasta la 23, se muestra el evento '*activate*', que puede realizar tareas como la limpieza de cachés antiguas o en desuso. Por último, en las líneas 25 hasta la 43, se aprecia el evento '*fetch*', que interviene cada vez que se hace una petición a la red (servicios como la GeoApi), intentando primero responder con el recurso desde la caché y, si no es posible, realizando la petición a la red y almacenando la respuesta para futuras consultas a ese recurso.

```

1 // Service Worker
2 // Este evento se dispara cuando el Service Worker está siendo instalado.
3 self.addEventListener('install', function(event) {
4     // Se imprime un mensaje en la consola.
5     console.log('[Service Worker] Installing Service Worker ...', event);
6     // Se utiliza event.waitUntil para asegurarse de que la instalación no se considere completa hasta que se haya terminado de cachear los archivos.
7     event.waitUntil(
8         caches.open('static').then(function(cache) {
9             // Se añaden al caché los archivos estáticos que se quieren guardar para uso offline.
10            cache.addAll(['/index.html', '/mian.jxs', '/site.webmanifest']);
11        })
12    );
13 });
14
15 // Este evento se dispara cuando el Service Worker ha sido activado.
16 self.addEventListener('activate', function(event) {
17     console.log('[Service Worker] Activating Service Worker ....', event);
18     // Aquí se podrían hacer tareas de limpieza, como eliminar cachés antiguas.
19 });
20
21 // Este evento se dispara cada vez que se hace una petición desde nuestra página.
22 self.addEventListener('fetch', function(event) {
23     // Se intercepta la petición.
24     event.respondWith(
25         // Se busca en caché una respuesta para la petición,
26         caches.match(event.request).then(function(response) {
27             // Si se encuentra una respuesta en caché, se devuelve.
28             if (response) {
29                 return response;
30             } else {
31                 // Si no se encuentra una respuesta en caché, se realiza la petición a la red.
32                 return fetch(event.request).then(function(res) {
33                     // Se abre el caché dinámico.
34                     return caches.open('dynamic').then(function(cache) {
35                         // Se guarda la respuesta para futuras consultas.
36                         cache.put(event.request.url, res.clone());
37                         // Se devuelve la respuesta.
38                         return res;
39                     });
40                 });
41             }
42         });
43     });
44 });

```

Figura 34 : Script de la configuración Service Worker

4.3.3.3 Desarrollo del Manifiesto para la Aplicación PWA

El manifiesto de una aplicación web progresiva es un archivo JSON de configuración que proporciona información crucial para definir cómo se muestra la PWA. Este archivo controla aspectos visuales de la aplicación, desde el ícono de inicio hasta la pantalla de bienvenida que se muestra al usuario cuando hace clic en el ícono para abrir la aplicación.

En la Tabla 17 se muestran las variables con una descripción de las variables configuradas en el manifiesto. Además, en el Anexo K, se muestra el JSON con la configuración que está utilizando la aplicación. Con estos atributos se puede asegurar que la aplicación va a funcionar correctamente como PWA.

Tabla 17 : Variables del manifiesto que fueron configuradas con su descripción

Nombre	Descripción
<i>name</i>	El nombre completo de la aplicación en este caso es Geoparques. Este nombre se utiliza para mostrarlo en la pantalla de inicio del dispositivo móvil del usuario.
<i>short_name</i>	Es una versión más corta del nombre, que se utiliza en lugares en la interfaz gráfica del dispositivo móvil donde puede haber espacio limitado.
<i>start_url</i>	Es la URL que carga la aplicación al iniciar, es el punto de entrada de la PWA. En el caso de la aplicación del visitante, sería la vista inicial.
<i>display</i>	Define el modo de visualización de la aplicación, que puede ser en modo de pantalla completa (<i>fullscreen</i>) o independiente (<i>standalone</i>). Para la aplicación, se escogió la última opción, ya que esta permite que la aplicación se muestre como nativa.
<i>background_color</i>	Es el color que se utiliza durante la pantalla de inicio de la aplicación es importante para la experiencia visual, por ejemplo, cuando la aplicación está cargando.
<i>theme_color</i>	Este color se utiliza para los elementos de la interfaz, por ejemplo, la barra de estado de nuestro dispositivo móvil, donde se muestra la hora, la señal, etc. En el caso de la aplicación está definido en color blanco.
<i>icons</i>	Es una lista de iconos de diferentes tamaños para representar visualmente la aplicación, por ejemplo, en la pantalla de inicio o en la lista de aplicaciones instaladas.
<i>orientation</i>	Es la orientación predeterminada en la que se debe mostrar la aplicación, puede ser horizontal o vertical. En este caso, la aplicación se abre y se visualiza de forma vertical.
<i>scope</i>	Define el conjunto de URL a las que la PWA puede tener acceso. En el caso de la aplicación, se dejó liberado, es decir, puede tener acceso a todas las URL de Geoparques.

4.3.3.4 Capa Controladora

La capa controladora de la aplicación del visitante está desarrollada para brindar información esencial a la capa de presentación (PWA) de la aplicación del visitante. Además sirve para rastrear las acciones que realiza el visitante cuando interactúa con la aplicación. En la Tabla 18 se muestra un extracto de los endpoints correspondientes a esta capa. Para una revisión más detallada de todos los endpoints disponibles en esta capa, se

pueden consultar en el Anexo L: Rutas (endpoint) capa controladora de la aplicación visitante

Tabla 18 : Extracto de las rutas de la capa controladora aplicación visitantes

	Tipo	Ruta	Descripción
Encuesta inicial al visitante.	GET	/api-geo-info/encuesta-inicial	Obtiene todas las encuestas realizadas. Con el objetivo de analizarlas.
	GET	/api-geo-info/encuesta-inicial/:id	Obtiene una encuesta por su id.
	POST	/api-geo-info/encuesta-inicial	Sirve para crear encuesta al momento de que el usuario rellena todos los campos correspondientes a la encuesta inicial.
Módulo Vista Inicial	GET	/api-geo-info/portada-inicial/:idioma	Obtiene la portada inicial mediante el idioma.
	PUT	/api-geo-info/portada-inicial/:id	Modifica la portada inicial según su id.
	GET	/api-geo-info/tarjeta-inicial/:tipo	Obtiene una tarjeta de la vista inicial según su tipo (esta puede ser A o B).
	GET	/api-geo-info/tarjeta-inicial-id/:id	Obtiene una tarjeta mediante su id-

4.3.4 Desarrollo del Módulo Códigos QR

El desarrollo de este módulo es de suma importancia, ya que contempla uno de los objetivos específicos de este proyecto de título. Por esta razón, en esta sección del capítulo revisaremos cómo se implementó esta técnica en el sistema de Geoparques, comenzando por la planificación, el desarrollo en la aplicación del administrador y, posteriormente, cómo se implementó en la aplicación de los visitantes.

4.3.4.1 Implementación

Como se mencionó en el Capítulo 2, Situación actual, se revisó cómo funcionaban los códigos QR y su capacidad para guardar información. Por este motivo, era necesario tener una técnica para almacenar en el código la información de una ficha de especie para que, cuando sea escaneado, muestre la información correspondiente. Además, este código debía ser generado por el administrador mediante su aplicación.

Gracias al modelo de datos de la ficha especie, el cual cuenta con un identificador único, se consideró que el código QR podría almacenar este ID, como muestra la Figura 35. Cuando el código sea escaneado, la aplicación leerá el ID y buscará en la GeoApi la ficha especie correspondiente a esa ID para visualizarla. Este método tiene ventajas y desventajas.

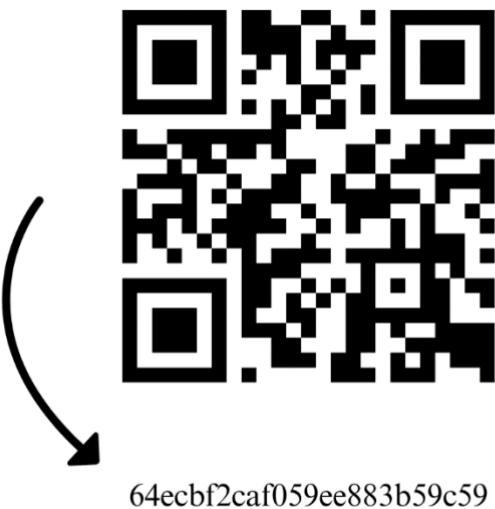


Figura 35 : QR almacenando la ID de la ficha especie

En términos de seguridad, este método es más seguro que incluir una URL completa en el código QR, como lo muestra la Figura 36 . Ya que no redirige a los usuarios finales a los navegadores de sus teléfonos móviles para utilizar la aplicación. En la actualidad, existen ataques de ciberseguridad utilizando códigos QR, conocidos como *quishing*²⁵, por lo que sería más recomendable utilizar.

Sin embargo, este método no permitiría una amplia difusión de la aplicación porque, al escanear un código QR con la cámara del dispositivo móvil o mediante una aplicación de terceros, no redirige a ninguna parte, solo se mostrará el ID de la ficha. Esto puede generar confusión al usuario que puede no tener idea de la existencia de Geoparques y solo vio un código QR posicionado en el parque y quiere obtener información. Por esta razón, la mejor opción para una aplicación que recién está saliendo al mercado es ofrecer la máxima difusión. Debido a esto, se decidió que el contenido que tendría el código QR sería la URL de la aplicación acompañada del ID de la ficha de especie.

²⁵ Son ataques en donde se utilizan los códigos QR para almacenar en este algún tipo de código malicioso, y al momento de escanearlo se redirige a un sitio en donde intentan robar información confidencial, como credenciales de inicio de sesión, o instalar programa maligno en el dispositivo.



Figura 36 : QR, almacenando la URL de la aplicación además de la ID de la ficha especie

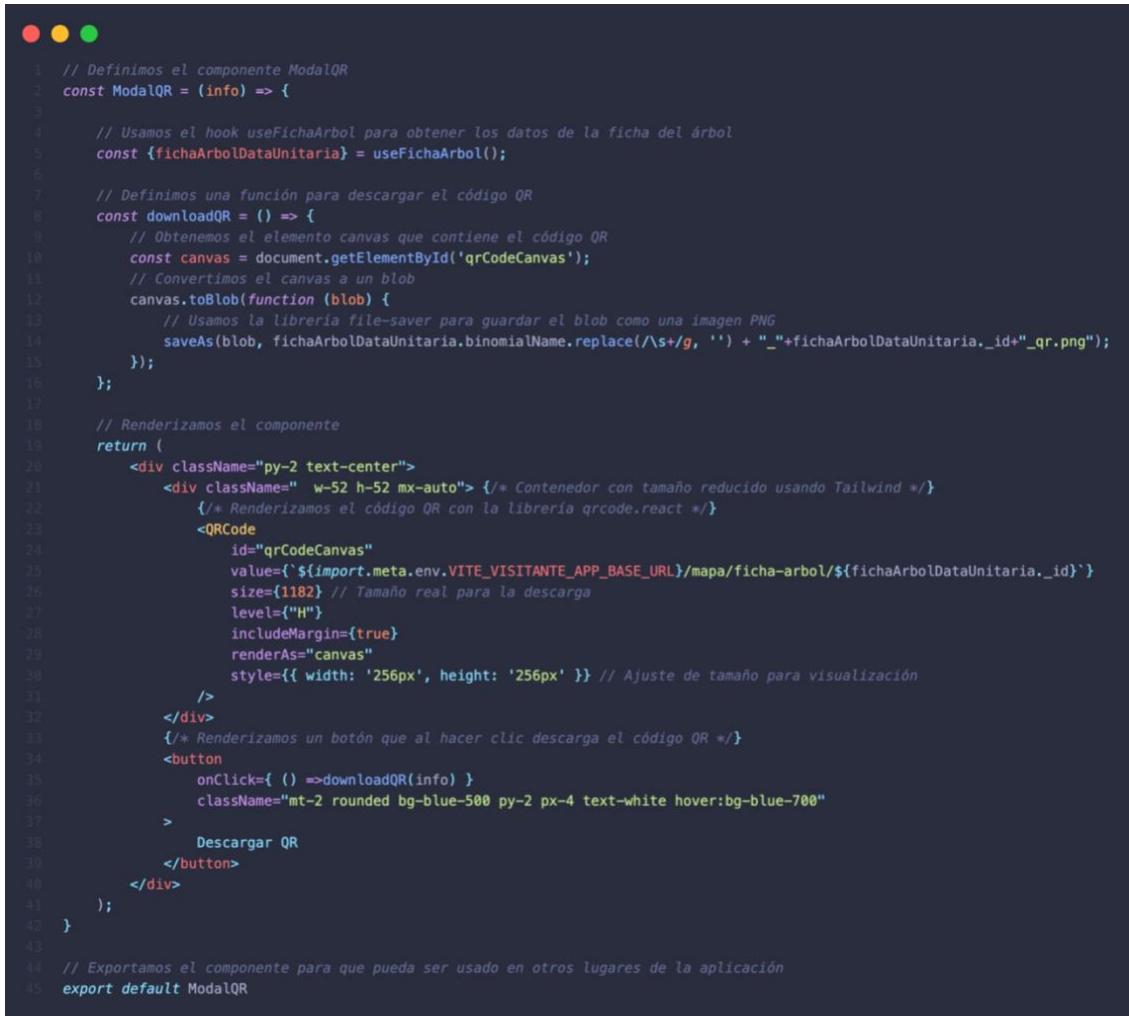
4.3.4.2 Implementación en la Aplicación del Administrador

Para explicar la implementación, se utilizará el código desarrollado. Esto se puede apreciar en la Figura 37, donde se muestra el componente en React denominado ModalQR, visible en la línea 2 del código. Posteriormente, en la línea 5, se llama a un hook²⁶ y se utiliza la variable “fichaArbolDataUnitaria”, que contiene la información de la ficha de especies seleccionada para generar el QR. Es importante mencionar que este componente se invoca cada vez que se presiona un botón para generar un código QR. En la línea 23, se puede observar cómo se realiza la llamada al componente QRCode, el cual tiene la tarea de generar el código QR. A este se le pasan variables como ‘value’, donde se introduce la URL que contendrá el QR; en este caso, se pasa una cadena concatenada que comienza con una variable de entorno que deriva a la URL de la aplicación del visitante, acompañada de una ruta que lleva al mapa y, finalmente, se concatena la ID de la ficha. Las otras variables son para configurar el QR, su tamaño original y el tamaño que se utilizará visualmente en la interfaz.

En la línea 34, hay una etiqueta HTML que crea un botón, el cual permite realizar la descarga del código QR. Se puede apreciar que este botón llama a la función que se

²⁶ Es una forma de compartir variables, métodos y otros tipos de datos entre componentes sin la necesidad de pasar props manualmente a través de cada nivel de árbol de componentes. Esto simplifica el código y mejora la gestión de datos a lo largo de la aplicación.

encuentra en la línea 8, llamada “downloadQR”, cuya misión es generar la imagen con el nombre binomial²⁷ de la especie concatenado con su ID y en formato PNG.



```

1 // Definimos el componente ModalQR
2 const ModalQR = (info) => {
3
4     // Usamos el hook useFichaArbol para obtener los datos de la ficha del árbol
5     const {fichaArbolDataUnitaria} = useFichaArbol();
6
7     // Definimos una función para descargar el código QR
8     const downloadQR = () => {
9         // Obtenemos el elemento canvas que contiene el código QR
10        const canvas = document.getElementById('qrCodeCanvas');
11        // Convertimos el canvas a un blob
12        canvas.toBlob(function (blob) {
13            // Usamos la librería file-saver para guardar el blob como una imagen PNG
14            saveAs(blob, fichaArbolDataUnitaria.binomialName.replace(/\s+/g, '') + "_" + fichaArbolDataUnitaria._id + ".png");
15        });
16    };
17
18    // Renderizamos el componente
19    return (
20        <div className="py-2 text-center">
21            <div className=" w-52 h-52 mx-auto"> /* Contenedor con tamaño reducido usando Tailwind */
22                /* Renderizamos el código QR con la librería qrCode.react */
23                <QRCode
24                    id="qrCodeCanvas"
25                    value={`${import.meta.env.VITE_VISITANTE_APP_BASE_URL}/mapa/ficha-arbol/${fichaArbolDataUnitaria._id}`}
26                    size={1182} // Tamaño real para la descarga
27                    level="H"
28                    includeMargin={true}
29                    renderAs="canvas"
30                    style={{ width: '256px', height: '256px' }} // Ajuste de tamaño para visualización
31                />
32            </div>
33            /* Renderizamos un botón que al hacer clic descarga el código QR */
34            <button
35                onClick={() => downloadQR(info)}
36                className="mt-2 rounded bg-blue-500 py-2 px-4 text-white hover:bg-blue-700"
37            >
38                Descargar QR
39            </button>
40        </div>
41    );
42 }
43
44 // Exportamos el componente para que pueda ser usado en otros lugares de la aplicación
45 export default ModalQR

```

Figura 37 : Código de generar QR

Obtención del Código QR en la interfaz gráfica

Para obtener un código QR de una ficha asociada, se debe ir a la vista de fichas de especies, mencionada anteriormente, y buscar la ficha de la cual se desea obtener el código QR. Posteriormente, se debe presionar el botón 'QR', el cual abrirá un modal, como se puede apreciar en la Figura 38, mostrando la imagen del QR para que pueda ser descargada posteriormente.

²⁷ Mas conocido como el nombre científico de la especie.

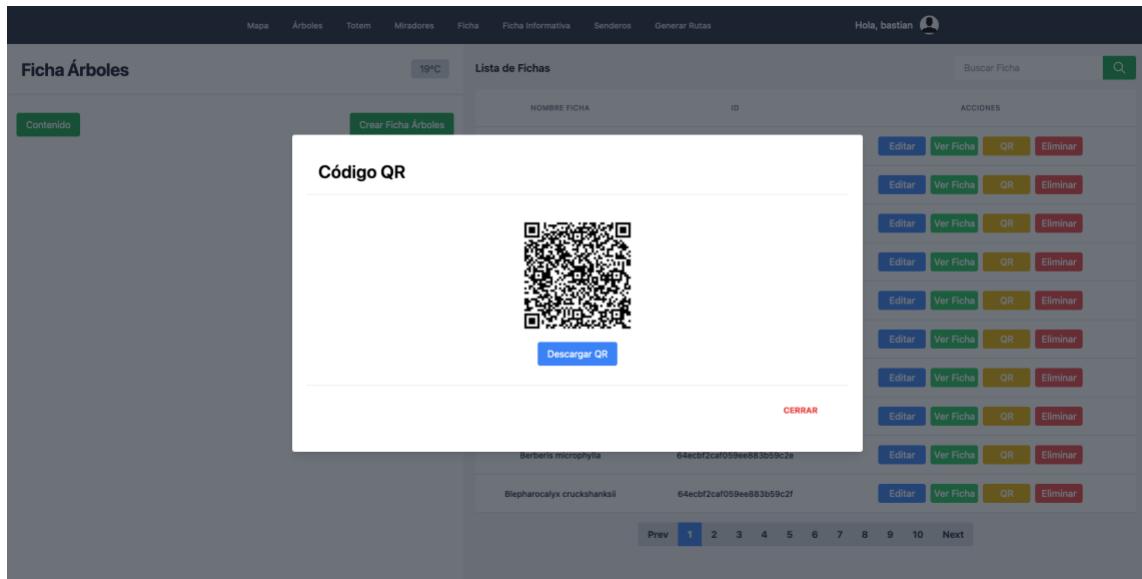


Figura 38 : Modal para la visualización del código QR

4.3.4.3 Implementación en la Aplicación del Visitante

Para la aplicación del visitante, es esencial contar con un componente que permita realizar la lectura del código QR. Posteriormente, el sistema debe poder decodificar la URL para que luego se muestre la información de la ficha especie al visitante.

Para esto, se diseñó un componente en React denominado `ModalQR`, el cual podemos ver en la Figura 39. En la línea 28, ocurre el retorno de este componente donde se inicializa el modal, se le dan los estilos y la configuración respectiva para que se pueda visualizar correctamente en la interfaz del visitante. En la línea 36, se realiza la llamada a un componente denominado `Scanner`²⁸, el cual facilita las funciones de abrir la cámara del dispositivo además de escanear el código. Dentro del `Scanner`, se agregan los parámetros de configuración, y uno de relevancia se aprecia en la línea 37, donde se entrega la función a la que se debe dirigir si el QR es escaneado de forma exitosa. Esta función se puede apreciar en la línea 8, donde se analiza la ruta y se verifica si cumple con ciertas condiciones, como que la URL contenga el endpoint “mapa/ficha-arbol”. En caso de que la URL escaneada por la aplicación del visitante no incluya este endpoint, el sistema automáticamente reconoce que el QR no corresponde a Geoparques y envía un aviso de error, lo cual se puede apreciar en las líneas 12 a 14. Si la URL contiene el endpoint, el sistema navegará hacia la vista del mapa, utilizando el endpoint que contenía el QR.

²⁸ Es un componente de la biblioteca `@yudiel/react-qr-scanner` para React, el cual permite escanear códigos QR utilizando la cámara de un dispositivo (npm.js, s. f.).



```

1 // Definimos el componente ModalQR
2 const ModalQR = ({avisoErrorCodigoQR}) => {
3   const navigate = useNavigate(); // Inicializamos el hook useNavigate
4   const {setQRActivate} = useSistemaTrack(); // Obtenemos la función setQRActivate del hook useSistemaTrack
5
6   // Función que se ejecuta cuando se obtiene un resultado del escaneo del código QR
7   const handleResult = (text, result) => {
8     const url = new URL(text); // Creamos un objeto URL a partir del texto obtenido
9     const ruta = url.pathname // Obtenemos la ruta de la URL
10
11    // Si la ruta no comienza con '/mapa/ficha-arbol/', mostramos un aviso de error y terminamos la ejecución de la función
12    if (!ruta.startsWith('/mapa/ficha-arbol/')) {
13      avisoErrorCodigoQR();
14      return;
15    }
16    setQRActivate(true); // Activamos el seguimiento del QR
17    // Scanner.stop(); // Detenemos el escaneo del QR
18    navigate(ruta); // Navegamos a la ruta obtenida
19    return;
20  };
21
22 // Función que se ejecuta cuando ocurre un error en el escaneo del código QR
23 const handleError = (error) => {
24   console.error(error.message); // Mostramos el mensaje de error en la consola
25 };
26
27 // Retornamos el JSX del componente
28 return (
29   <>
30     <dialog id="my_modal_3" ClassName="modal">
31       <div className="modal-box bg-white">
32         <form method="dialog">
33           <button className="btn btn-sm btn-circle btn-ghost absolute right-2 top-2"></button>
34         </form>
35         <div className="flex flex-col items-center justify-center">
36           <Scanner
37             onResult={handleResult} // Función que se ejecuta cuando se obtiene un resultado
38             onError={handleError} // Función que se ejecuta cuando ocurre un error
39             styles={{
40               container: { width: '90%', height: '90%', borderRadius: '20px' }, // Estilos del contenedor del escáner
41               video: { width: '100%', height: '100%' } // Estilos del video del escáner
42             }}
43             components={{
44               audio: false, // Desactivamos el audio
45             }}
46             stopDecoding={true} // Detenemos la decodificación
47           />
48         </div>
49       </div>
50     </dialog>
51   </>
52 );
53 );
54
55 export default ModalQR; // Exportamos el componente

```

Figura 39 : Código del ModalQR

Una vez que la aplicación se encuentra en el componente raíz del mapa, llamado **PaginaMapa**, como se puede apreciar en la Figura 40, se procede a obtener la ID que viene en el endpoint. Luego se realiza una llamada a una función, visible en la línea 6, que verifica la presencia de la ID en el endpoint con el objetivo de no provocar un error en el sistema en caso de que no exista. Después de validar la ID, se llama a la función mostrada en la línea 13, cuya misión es obtener la ficha de especie por medio de la GeoApi. Posteriormente, se verifica que la información haya sido recibida correctamente, como se puede ver en la validación de la línea 16. Si la información se obtiene correctamente, se invoca el modal correspondiente a la ficha de especie para que se visualice en pantalla. Finalmente, en la línea 20 se invoca la función postTrackQR, donde se registra en el sistema que el usuario ha visualizado una especie mediante el QR. Esta información, como se mencionó anteriormente, queda almacenada en la base de datos.

```

1 // useParams es un hook de react-router-dom que permite acceder a los parámetros de la ruta
2 const { id } = useParams();
3
4 // Esta función se encarga de obtener la información de la ficha del árbol si el id no es nulo ni indefinido
5 const llamarFichaArbol = () => {
6   if(id !== undefined && id !== null) {
7     obtenerInformacionFicha(id);
8   }
9 }
10
11 // Esta función recibe un id y hace una petición para obtener la información de la ficha del árbol con ese id
12 const obtenerInformacionFicha = (id_) => {
13   getFichaArbolById(id_).then( info ) => {
14     // Si la información obtenida no es falsa, se abre un modal con la información de la ficha del árbol
15     if(info!= false )
16       openModal(info, "TreeCard");
17     // Si QRActivate es verdadero, se hace un post para hacer seguimiento del QR
18     if (QRActivate){
19       postTrackQR("Ficha_Arbol",info);
20     }
21   });
22 };
23

```

Figura 40 : Fragmento de código del componente PaginaMapa

Escaneo del Código QR Mediante la Interfaz

En la Figura 41 se puede apreciar la apariencia de la interfaz, cuando se escanea un QR mediante la aplicación, visualizando el modal que se detalló recientemente.



Figura 41 : Apariencia de la interfaz al momento de escanear un código QR.

Para finalizar esta subsección, se explica mediante un diagrama de secuencia el funcionamiento del sistema cuando un visitante escanea un QR.

El diagrama de secuencia que se muestra en la Figura 42 detalla las interacciones entre las distintas componentes del sistema de Geoparques que permiten al usuario obtener la información del QR. En la primera parte del diagrama se observa cuando el visitante desea escanear un código QR; posteriormente, esa información se envía para ser obtenida por FrontEnd (aplicación del visitante) y luego mostrada al visitante. En la parte inferior del diagrama se muestran las interacciones que realiza la aplicación para almacenar el rastreo cuando el visitante escanea un QR.

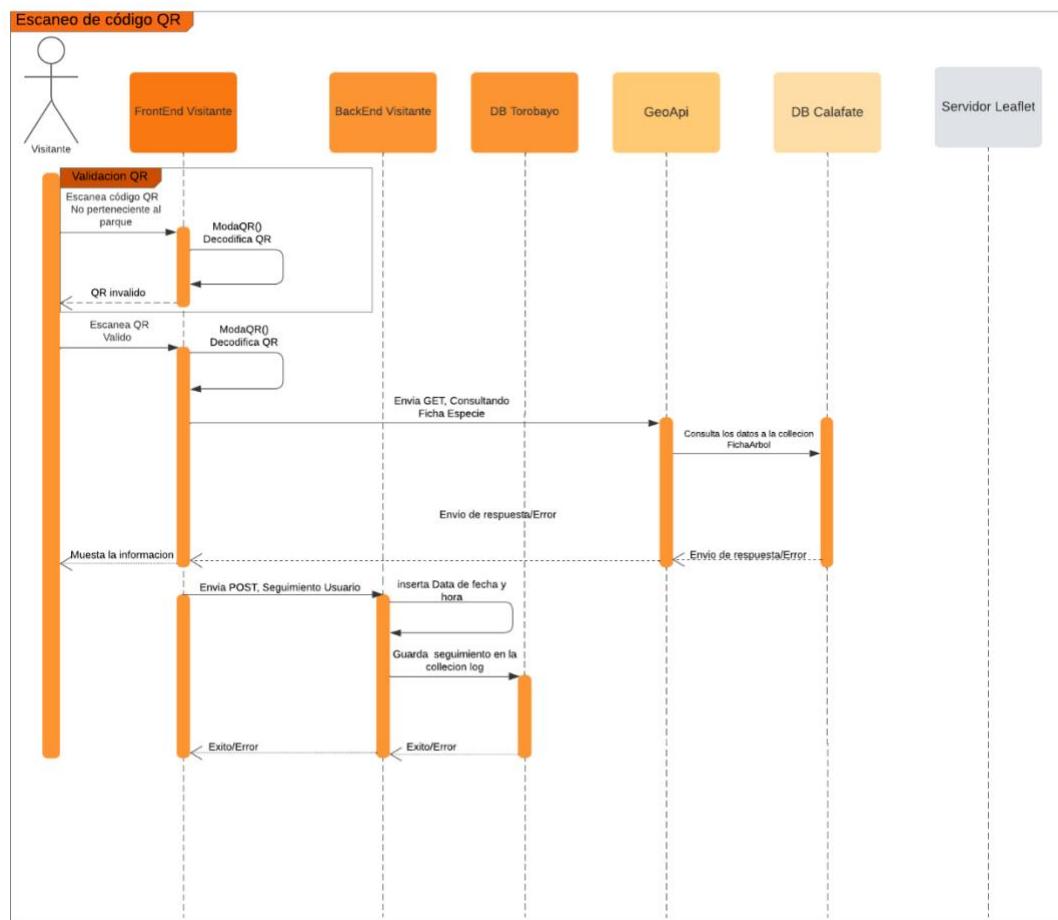


Figura 42 : Diagrama de secuencia escaneo de un código QR

4.3.5 Desarrollo del Módulo Geolocalización

Para el desarrollo de este proyecto de título, el módulo de Geolocalización es importante, ya que permite mostrar la localización geográfica de donde se encuentra el visitante en el

parque, además de mostrar información a este a través de su posición. Es por esto que en esta subsección del capítulo se detalla cómo se implementó dicho módulo en el sistema.

4.3.5.1 Implementación de la Geolocalización a través del GPS de un Dispositivo Móvil

En la planificación de esta parte del proyecto, se consideró esencial que, como mínimo, el visitante pudiera ver su localización en el mapa. Por ello, una de las primeras cosas que se implementaron en la vista del mapa, fue la capacidad de obtener y visualizar la posición en tiempo real del visitante en el parque, como se puede apreciar en la Figura 43. Esto fue posible gracias a un complemento existente para la biblioteca Leaflet, denominado leaflet.locatecontrol, que permite a los desarrolladores integrar funciones de localización geográfica en aplicaciones que utilizan Leaflet para mapas interactivos. Además, facilita el control sobre cómo y cuándo se muestra la ubicación del usuario en un mapa, mejorando la interacción y la funcionalidad de las aplicaciones con mapas interactivos.

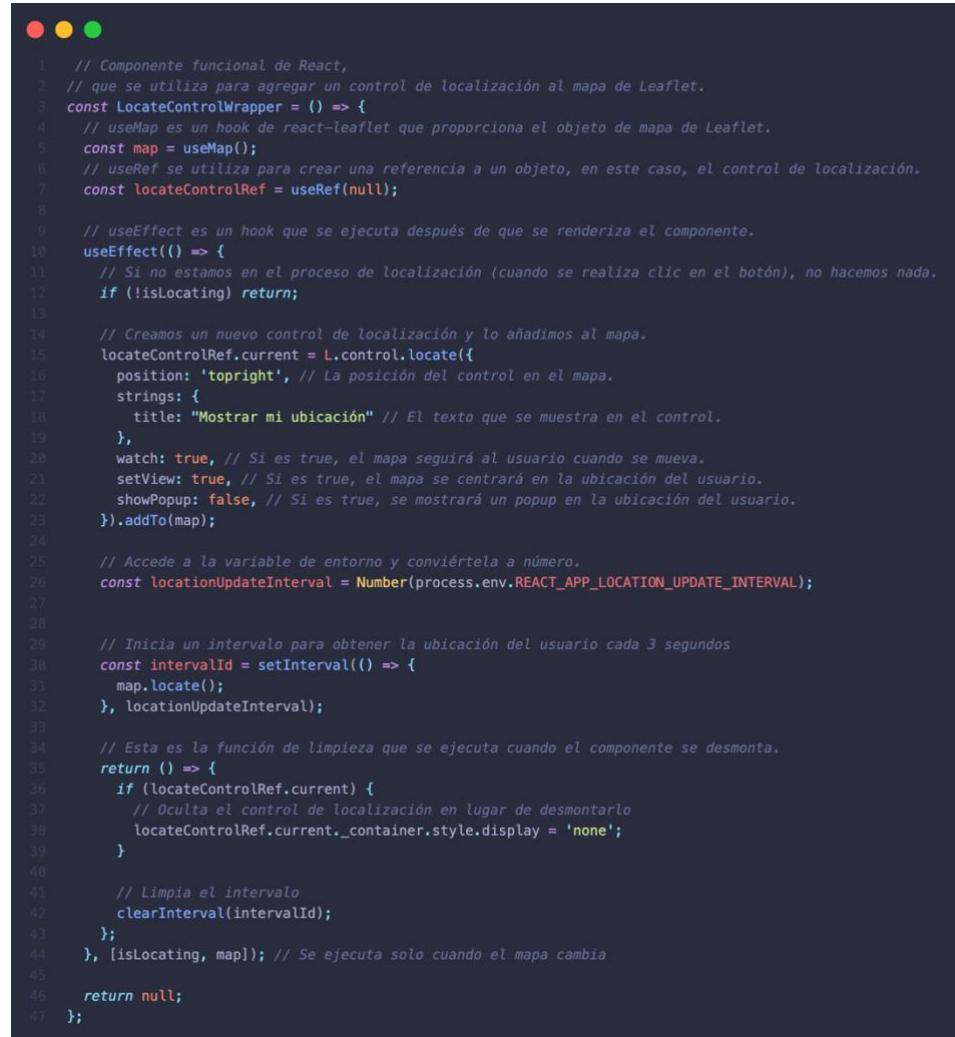


Figura 43 : Activación de la localización del visitante

Para integrar este complemento al mapa Leaflet, se desarrolló un componente llamado LocateControlWrapper, ilustrado en la Figura 44. En la línea 5, se invoca el hook del mapa de Leaflet para obtener el objeto completo y acceder a sus métodos. En la línea 15, se introduce un control que es el principal responsable de captar y mostrar la posición del visitante. Los parámetros configurables, visibles en las líneas 20 a 22, incluyen un parámetro para que el mapa siga al usuario mientras se mueve y otro para que se centre automáticamente al obtener la ubicación del usuario. Posteriormente, en la línea 23, se añade este control al mapa. En la Figura 43, se muestra este control como un botón en forma de flecha de brújula en la parte superior derecha, marcado con un círculo rojo; al

presionarlo, se activa el control de localización y se añade un círculo azul al mapa para indicar la posición del usuario.

Una vez activado el control, se utiliza la línea 30, en donde se establece una función que opera con un intervalo de tiempo, actualizando la posición del usuario en el mapa cada 3 segundos, la cual está contenida en una variable de entorno. Para finalizar el “Return”, que se detalla en la línea 35, desempeña la función de limpiar y desmontar el componente. Este procedimiento asegura que, en caso de cualquier error con este complemento, no se produzca una duplicación del mismo ni se muestre repetidamente al usuario.



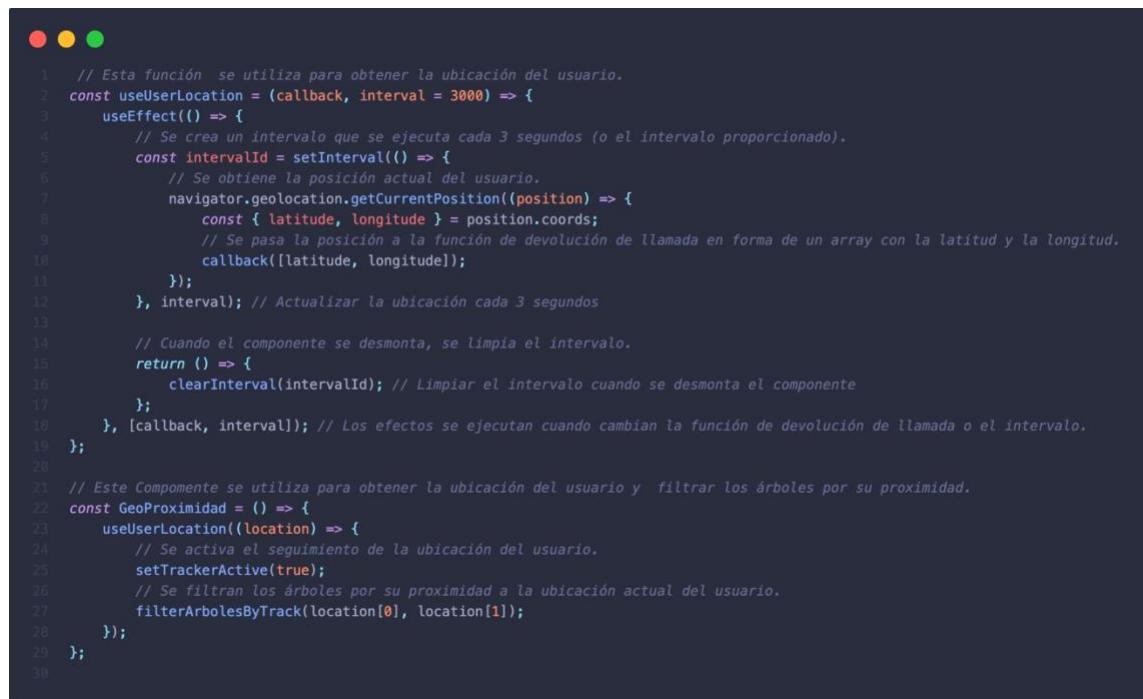
```
1 // Componente funcional de React,
2 // que se utiliza para agregar un control de localización al mapa de Leaflet.
3 const LocateControlWrapper = () => {
4   // useMap es un hook de react-leaflet que proporciona el objeto de mapa de Leaflet.
5   const map = useMap();
6   // useRef se utiliza para crear una referencia a un objeto, en este caso, el control de localización.
7   const locateControlRef = useRef(null);
8
9   // useEffect es un hook que se ejecuta después de que se renderiza el componente.
10  useEffect(() => {
11    // Si no estamos en el proceso de localización (cuando se realiza clic en el botón), no hacemos nada.
12    if (!isLocating) return;
13
14    // Creamos un nuevo control de localización y lo añadimos al mapa.
15    locateControlRef.current = L.control.locate({
16      position: 'topright', // La posición del control en el mapa.
17      strings: {
18        title: "Mostrar mi ubicación" // El texto que se muestra en el control.
19      },
20      watch: true, // Si es true, el mapa seguirá al usuario cuando se mueva.
21      setView: true, // Si es true, el mapa se centrará en la ubicación del usuario.
22      showPopup: false, // Si es true, se mostrará un popup en la ubicación del usuario.
23    }).addTo(map);
24
25    // Accede a la variable de entorno y conviértela a número.
26    const locationUpdateInterval = Number(process.env.REACT_APP_LOCATION_UPDATE_INTERVAL);
27
28
29    // Inicia un intervalo para obtener la ubicación del usuario cada 3 segundos
30    const intervalId = setInterval(() => {
31      map.locate();
32    }, locationUpdateInterval);
33
34    // Esta es la función de limpieza que se ejecuta cuando el componente se desmonta.
35    return () => {
36      if (locateControlRef.current) {
37        // Oculta el control de localización en lugar de desmontarlo
38        locateControlRef.current._container.style.display = 'none';
39      }
40
41      // Limpia el intervalo
42      clearInterval(intervalId);
43    };
44  }, [isLocating, map]); // Se ejecuta solo cuando el mapa cambia
45
46  return null;
47};
```

Figura 44 : Componente para captar la posición del visitante y visualizarlo en el mapa

4.3.5.2 Función de Geoproximidad Implementada en la Aplicación

La funcionalidad se nombró Geoproximidad y se atribuye a la combinación de 'Geo', que significa geoposicionamiento, y 'proximidad', reflejando cómo se muestra información relevante a medida que el visitante se acerca a una zona geográfica. Esta implementación está diseñada y desarrollada para permitir que los visitantes de los parques visualicen en el mapa las especies cercanas a medida que se aproximan a ellas. Por lo tanto, cuando esta opción esté activada, se ocultarán del mapa todas las especies excepto aquellas que estén más cerca del visitante. A continuación, se detallará cómo se implementó esta funcionalidad.

Cuando el visitante o usuario de la aplicación presiona el botón que se encuentra en el Drawer para activar la Geoproximidad, se invoca al componente que aparece en la línea 22, mostrado en la Figura 45. La responsabilidad de este componente es, en primer lugar, obtener la posición del visitante, para lo cual llama al componente mostrado en la línea 2 que realiza esta función. Es importante destacar que la aplicación siempre pregunta al usuario si desea compartir su ubicación. Una vez que el componente ha devuelto las coordenadas donde se posiciona el usuario, se procede a activar una variable que sirve para que el sistema sepa que la Geoproximidad está encendida, como se puede ver en la línea 25. Finalmente, las coordenadas se pasan a un método denominado 'filterArbolesByTrack', el cual se puede apreciar en la línea 27 y tiene la tarea de filtrar los árboles según nuestra posición.



```
1 // Esta función se utiliza para obtener la ubicación del usuario.
2 const useUserLocation = (callback, interval = 3000) => {
3   useEffect(() => {
4     // Se crea un intervalo que se ejecuta cada 3 segundos (o el intervalo proporcionado).
5     const intervalId = setInterval(() => {
6       // Se obtiene la posición actual del usuario.
7       navigator.geolocation.getCurrentPosition(position) => {
8         const { latitude, longitude } = position.coords;
9         // Se pasa la posición a la función de devolución de llamada en forma de un array con la latitud y la longitud.
10        callback([latitude, longitude]);
11      });
12    }, interval); // Actualizar la ubicación cada 3 segundos
13
14    // Cuando el componente se desmonta, se limpia el intervalo.
15    return () => {
16      clearInterval(intervalId); // Limpiar el intervalo cuando se desmonta el componente
17    };
18  }, [callback, interval]); // Los efectos se ejecutan cuando cambian la función de devolución de llamada o el intervalo.
19};
20
21 // Este Componente se utiliza para obtener la ubicación del usuario y filtrar los árboles por su proximidad.
22 const GeoProximidad = () => {
23   useUserLocation((location) => {
24     // Se activa el seguimiento de la ubicación del usuario.
25     setTrackerActive(true);
26     // Se filtran los árboles por su proximidad a la ubicación actual del usuario.
27     filterArbolesByTrack(location[0], location[1]);
28   });
29};


```

Figura 45 : Componentes para la función de Geoproximidad

En la función 'filterArbolesByTrack', los valores de latitud y longitud del visitante se reciben como se muestra en la Figura 46 en la línea 5. Luego, en la línea 7, se crea un objeto utilizando métodos de la biblioteca Leaflet para manejar las coordenadas de manera más eficiente. El proceso de filtrado comienza en la línea 13; inicialmente, se crea un objeto de coordenadas para cada árbol. Posteriormente, se utiliza el método 'DistanceTo' de Leaflet, que compara la distancia mediante un radio establecido de 50 metros. Si la posición actual del visitante está a menor o igual a 50 metros de un árbol, este se mantiene en el arreglo creado para los árboles filtrados, incluyendo todos los árboles que están dentro de los 50 metros de proximidad. Cabe destacar que este valor se obtiene de una variable de entorno por si en versiones futuras se desea modificar.



```

1 // Se importa la biblioteca leaflet para utilizar sus funciones.
2 import L from 'leaflet';
3
4 // Esta función filtra los árboles en base a su proximidad a la ubicación del usuario.
5 const filterArbolesByTrack = (latVisintante, lngVisitante) => {
6     // Se crea un objeto LatLng con la latitud y longitud del visitante.
7     const userLocation = L.latLng(latVisintante, lngVisitante);
8
9     // Accede a la variable de entorno y conviértela a número.
10    const maxDistance = Number(process.env.REACT_APP_MAX_DISTANCE);
11
12    // Se filtran los datos de los árboles. Solo se mantienen los árboles que están a maxDistance metros o menos del visitante.
13    const datafilter = treesData.filter(tree => {
14        // Para cada árbol, se crea un objeto LatLng con la latitud y longitud del árbol.
15        let marcadorPosicion = L.latLng(tree.geometry.coordinates[1], tree.geometry.coordinates[0]);
16
17        // Se calcula la distancia entre la ubicación del visitante y la del árbol.
18        // Si la distancia es menor o igual a maxDistance metros, el árbol se mantiene en el array.
19        return userLocation.distanceTo(marcadorPosicion) <= maxDistance;
20    });
21
22    // Se actualiza el estado con los árboles filtrados.
23    setTreeDataTracker(datafilter);
24 }

```

Figura 46 : Componente encargado de filtrar los árboles mediante un radio establecido

Para concluir, la funcionalidad de Geoproximidad se presenta en la Figura 47 un diagrama de secuencias que muestra cómo la aplicación del visitante identifica los árboles más cercanos a él. En este diagrama se destacan los componentes y métodos descritos anteriormente en esta subsección. Específicamente, se destaca un cuadrado titulado 'bucle', dentro del cual se encapsulan los componentes y métodos responsables de la secuencia de pasos y del flujo de información que se repite cada 3 segundos para obtener la información actualizada de los árboles más cercanos al visitante. Adicionalmente, se introdujo un método no discutido previamente, representado en la figura como la última parte de la secuencia y denominado 'getTreeDataTracker'. Este método es el encargado de realizar el cambio de datos; crea un arreglo temporal para almacenar los árboles filtrados, que posteriormente el mapa puede consumir para añadir los marcadores correspondientes a los árboles filtrados. Cuando el usuario decide desactivar la opción de Geoproximidad, se invoca al método 'getTreeDataTracker' para recompilar el arreglo con los árboles habilitados que están configurados para mostrarse siempre en el mapa.

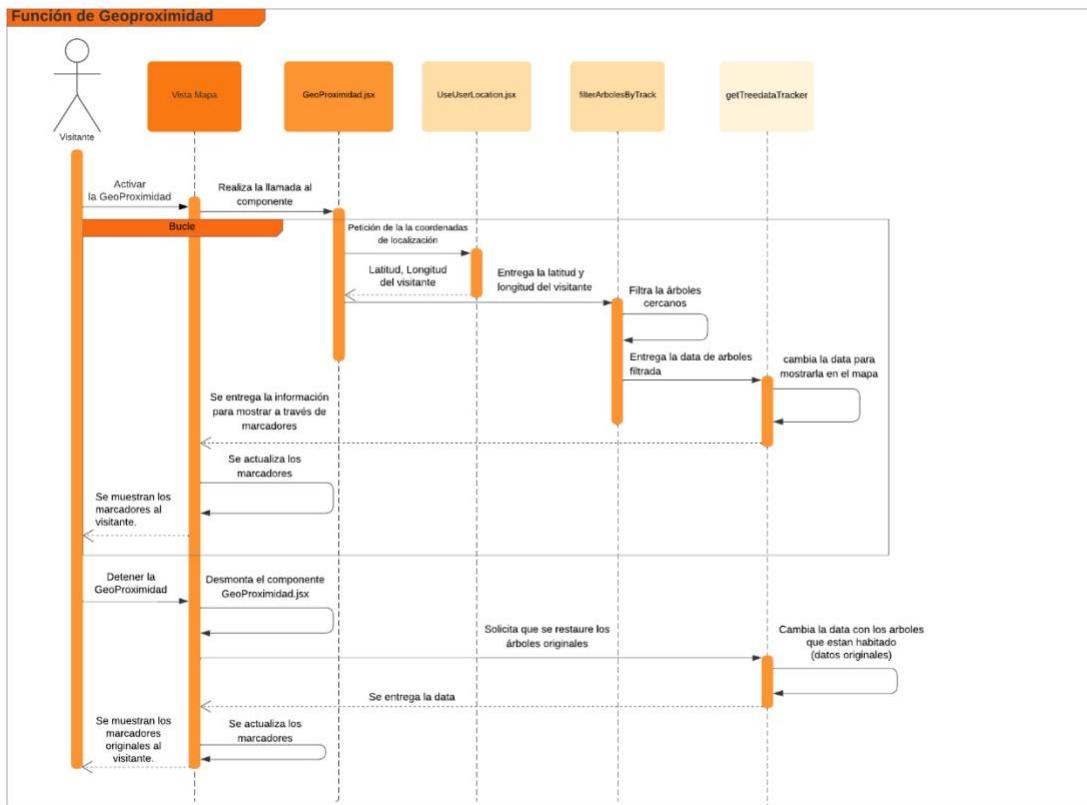


Figura 47 : Diagrama de secuencia de la función de Geoproximidad

4.3.6 Desarrollo de la Internacionalización de la Aplicación

La internacionalización se refiere al diseño y desarrollo de la aplicación para que sea compatible con diferentes culturas e idiomas, sin necesidad de realizar cambios adicionales en su estructura a nivel de código. Considerando que la aplicación desarrollada es para parques turísticos, debemos tener en cuenta que estos lugares son frecuentados por turistas que no siempre hablan español, y se reconoce al inglés como un idioma universal para la comunicación (Wallstreetenglish, 2022).

Métodos implementados para la Internacionalización

Se utilizaron dos métodos para ofrecer a los visitantes la opción de cambiar de idioma. El primero es que algunas colecciones de nuestra base de datos Torobayo incluyen un atributo de idioma, como se puede ver en la Figura 48. Estas colecciones contienen documentos en los dos idiomas mencionados. Por lo tanto, cuando se realiza un cambio de idioma en la aplicación del visitante, se modifica una variable de estado específica para el idioma; por ejemplo, si el idioma actual es español, la variable tiene el valor 'es', y si se cambia a inglés, esta cambia a 'en'.

Al modificar dicha variable, se realiza automáticamente una llamada a los endpoints que manejan el idioma, pasando el idioma seleccionado para obtener la información del documento en el idioma elegido. En la Figura 49 se muestra un ejemplo de cómo están construidos estos endpoints para facilitar la información en diferentes idiomas.

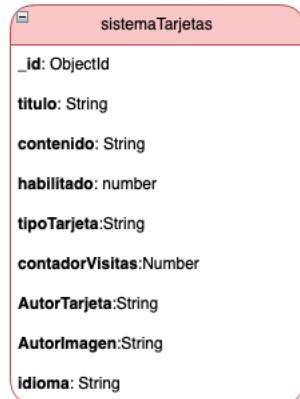


Figura 48 : Colección de ejemplo

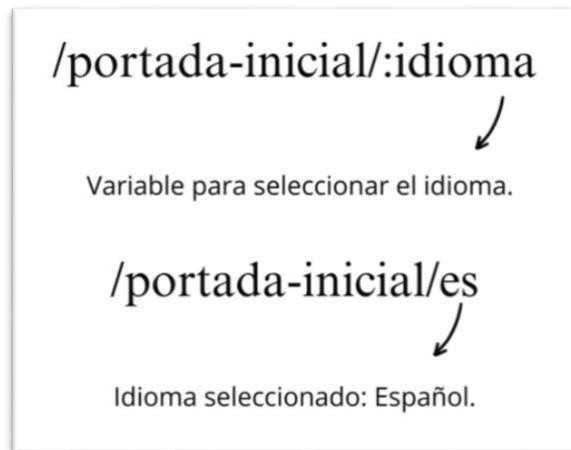


Figura 49 : Ejemplo de endpoint para obtener información mediante el idioma seleccionado

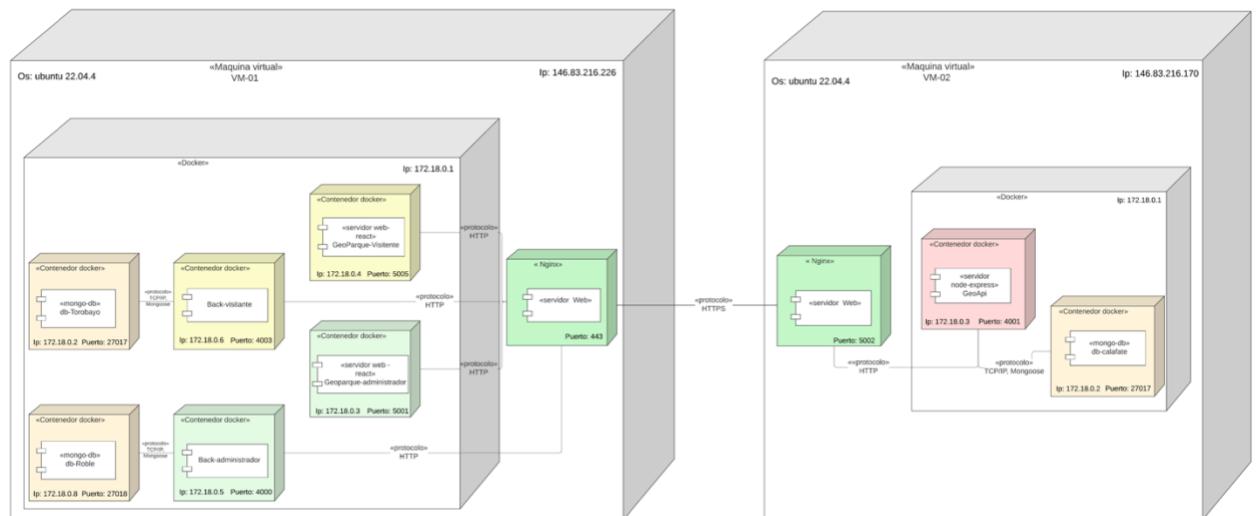
Como segundo método, se utilizó la librería i18n, que ofrece un conjunto de métodos para facilitar la internacionalización en aplicaciones web (npmjs.com, 2024). Esta librería opera utilizando archivos JSON para almacenar las traducciones en diferentes idiomas, un ejemplo de esto estarán disponibles para visualizarlos en el Anexo M: JSON con las variables en idioma inglés. Como se mencionó anteriormente, cuando el visitante cambia el idioma en la aplicación, se modifica la variable de estado. En el momento en que se realiza este cambio, se invoca un método de la librería que intercepta el texto necesario y busca la traducción correspondiente en el archivo JSON del idioma seleccionado.

4.3.7 Despliegue

En la Figura 50 se pueden apreciar los distintos componentes de sistema geoparques en producción, y cómo interactúan entre sí para poder brindar la información a los usuarios que utilizan el sistema.

Para desplegar en producción, se utilizaron dos máquinas virtuales proporcionadas por el Instituto de Informática de la Universidad Austral de Chile. Se planificó una arquitectura de despliegue en la que los componentes pudieran escalar ante un crecimiento o aumento en el número de personas que utilizarán la aplicación. Por esta razón, en la máquina VM-01 se desplegó la aplicación del visitante y la aplicación del administrador, además de sus bases de datos. En la máquina VM-02, se desplegó la GeoApi con su base de datos correspondiente, esto debido a que, como fue mencionado anteriormente, de esta API se conectarán varios sistemas y además se espera que este sistema reciba una gran cantidad de solicitudes, permitiendo que este componente pudiera escalar replicándolo u ocupando los máximos recursos de la máquina virtual. Es importante destacar que los componentes del sistema en ambas máquinas virtuales están encapsulados en contenedores Docker²⁹, facilitando el despliegue y el mantenimiento del sistema de GeoParques.

A Continuación se provee un breve explicación del proceso de despliegue.



²⁹ Docker es una plataforma de software que utiliza la virtualización a nivel de sistema operativo para entregar y gestionar aplicaciones en contenedores. Estos contenedores empaquetan el código de una aplicación junto con todas sus dependencias, permitiendo que la aplicación se ejecute de manera consistente en diferentes entornos computacionales (Docker, s. f.).

³⁰ https://drive.google.com/file/d/1wLj9_xYIOPRYcqWEu8II2HFdUXKWoNo1/view

Con la arquitectura de despliegue ya definida, se planificaron los días que se llevaría a producción el sistema, asignando dos semanas para esto, lo cual estaba planeado en la Carta Gantt modificada que se presenta en el Anexo O: Carta Gantt actualizada. Después, se planificó que la puesta en producción se llevaría a cabo en tres partes, las cuales se mencionan a continuación:

- **Desplegar a producción GeoApi:** como es uno de los componentes más importantes de la plataforma, se decidió que fuera el primero en desplegarse junto a su base de datos (calafate). Para esta operación se tardó un día.
- **Desplegar a producción la aplicación del visitante (Geoparques-Visitante):** se decidió que la segunda parte de la puesta en producción sería la aplicación del visitante, donde se tenía que desplegar la app PWA, su backend y la base de datos correspondiente. Esta tarea se llevó a cabo en 4 días, debido a que los componentes se fueron desplegando por partes: un día la base de datos, otro día el backend, y así sucesivamente.
- **Desplegar a producción la aplicación del administrador (Geoparques-Administrador):** por último, se decidió llevar a producción la aplicación del administrador, donde se desplegaron sus distintos componentes. Esta tarea tardó aproximadamente 3 días en tener la aplicación funcionando.

Si se desea profundizar en cómo se llevó a cabo este despliegue, se pueden revisar los tres Anexos, de la Anexo P a la Anexo R, que corresponden a los manuales de despliegue de la GeoApi, Geoparques-Visitante y Geoparques-Administrador.

Cabe destacar que, en la Figura 50, se puede apreciar que el componente Nginx está fuera de los contenedores de Docker y está presente en las dos máquinas virtuales. Este servicio Nginx actúa como un proxy invertido, que puede funcionar como servidor web. Un proxy invertido es un servidor encargado de distribuir el contenido de uno o más servidores, ocultando los otros componentes (servicios) de los usuarios finales (swhosting, s.f.). Este componente extra ofrece numerosos beneficios para el sistema, tanto en términos de seguridad como de escalabilidad. Como se mencionó, este componente recibe las peticiones que llegan de internet y las reenvía a otros componentes del sistema. Gracias a esto, los componentes del sistema no están expuestos directamente a internet, proporcionando una capa adicional de seguridad. Por otro lado, Nginx también puede proporcionar escalabilidad, ya que, si un componente del sistema está recibiendo muchas peticiones, se puede replicar este componente y Nginx puede actuar como un balanceador de carga, distribuyendo las peticiones tanto al componente original como al replicado. Es importante mencionar que los componentes del sistema están diseñados y programados para ser replicables.

5. VERIFICACIÓN Y VALIDACIÓN

En esta sección del trabajo de título, se busca validar que la solución propuesta cumple con las expectativas y estándares establecidos tanto en los requisitos no funcionales como con los objetivos específicos del proyecto. Para ello, se realizaron pruebas de ciberseguridad a la aplicación de GeoParques, así como pruebas de rendimiento y de usabilidad. Esto con el fin, de confirmar que la aplicación está en condiciones para estar en un ambiente productivo.

5.1 Ciberseguridad

En el mundo digital y conectado en el que vivimos hoy en día, es esencial proteger nuestros datos y aplicaciones, ya que esta información es valiosa para las organizaciones. Según Carlos Melo, “*existen dos tipos de empresas, las sufrieron un ciber ataque y las que aún no lo saben*” (Baseline, 2024). Para proteger la información digital, existe una rama de la informática que está en auge: la ciberseguridad, que es la práctica de defender las computadoras, servidores, dispositivos móviles, dispositivos electrónicos, redes y los datos de ataques maliciosos. Existen diversos métodos para proteger los sistemas informáticos, pero el primer paso esencial es realizar pruebas de vulnerabilidad. Estas pruebas permiten identificar los puntos débiles y las posibles entradas en nuestros sistemas que pueden estar expuestas sin nuestro conocimiento. Generalmente, estos puntos de entrada no se dejan abiertos intencionadamente por los desarrolladores, sino que quedan expuestos debido a errores de programación o configuración.

5.1.1 Pruebas de Vulnerabilidad

Para realizar pruebas de vulnerabilidad, se utilizan herramientas que están específicamente construidas para este fin. La herramienta que se empleó para realizar estas pruebas fue **ZAP**, que es un escáner de seguridad web de código abierto. Está diseñada para ser utilizada tanto como una aplicación de seguridad por usuarios comunes como una herramienta profesional para pruebas de penetración (zap proxy.org, s. f.)

El primer paso para realizar este tipo de pruebas es tener una planificación clara con el objetivo de no afectar a los usuarios activos de la aplicación. Por esta razón, la primera prueba de vulnerabilidad se llevó a cabo el 23 de abril de 2024, a las 9 de la noche. A esta hora, aunque el sistema ya estaba disponible para los usuarios finales, la cantidad de usuarios activos era baja. Se recomienda realizar estos tests en momentos en que la actividad de los usuarios sea mínima. Dado que el jardín botánico de la Universidad Austral de Chile está cerrado a las 9 de la noche, este horario resulta ideal para llevar a cabo las pruebas sin interrupciones.

ZAP ofrece la opción de realizar pruebas de manera automática, en la que solo es necesario proporcionar la URL de la API y el programa ejecuta el escaneo por sí mismo, como se muestra en la Figura 51. Debido a esta comodidad, se optó por esta alternativa para llevar a cabo las pruebas.

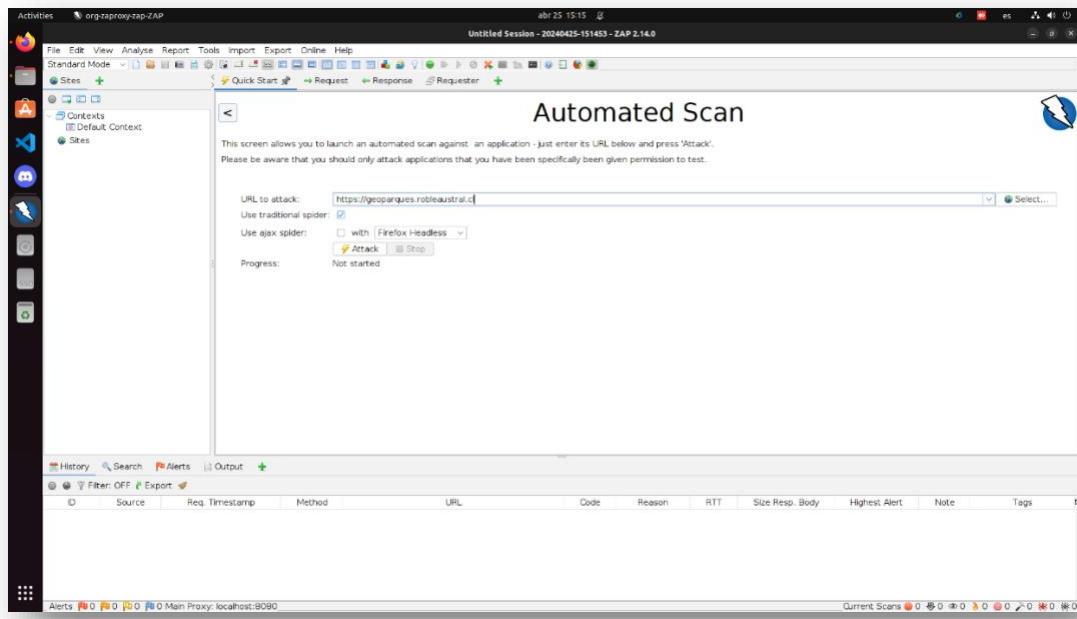


Figura 51 : Configuración para escaneo automático de vulnerabilidades en ZAP

Al presionar el botón "*Attack*", el escáner comienza a enviar peticiones a los servidores de Geoparques en busca de vulnerabilidades y errores de configuración en el servidor. Cabe mencionar que este proceso puede tardar entre 5 y 30 minutos, dependiendo del tamaño de la aplicación u sistema. Una vez terminado el escaneo, se entregan las vulnerabilidades encontradas, como se puede apreciar en la Figura 52. Se encontraron un total de 14 vulnerabilidades, lo cual puede sonar un poco alarmante; sin embargo, no todas son de carácter crítico. Estas alertas son una forma de clasificar las vulnerabilidades que utiliza ZAP y se pueden indagar en el Anexo S: Tabla de las distintas alertas de seguridad. A Continuación, se listan las vulnerabilidades encontradas según su clasificación.

- Una amenaza de estado Critica.
- Cuatro amenazas de estado Alta.
- Seis amenazas de estado Medio.
- Tres de amenaza de estado Bajo

Cabe mencionar que ZAP entrega un informe de estas vulnerabilidades en el que se explican y se dan recomendaciones sobre cómo solucionarlas. Este informe se encuentra en el Anexo T.

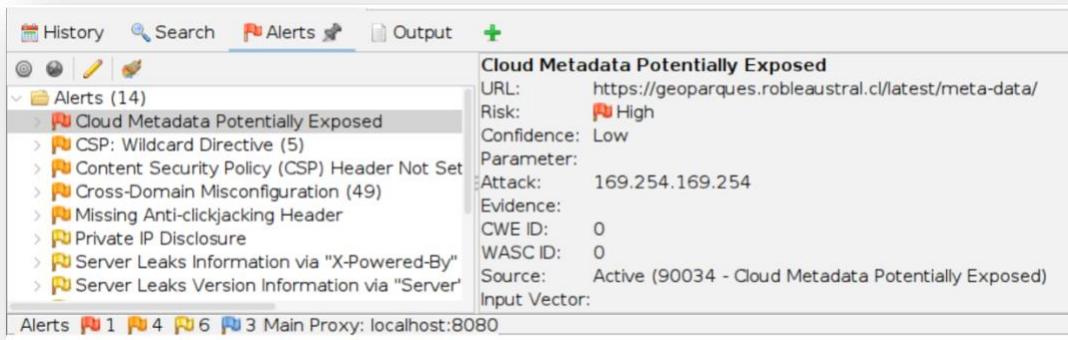


Figura 52 : Resultado del escáner realizado, en donde se muestra las vulnerabilidades encontradas

Para no extender demasiado esta sección, se indagarán las vulnerabilidades más críticas encontradas, que incluyen una de estado crítico y otra de estado alto.

5.1.1.1 Explicación Vulnerabilidad de Estado Crítico

Servicios en la nube como AWS, GCP y Azure proporcionan metadatos a través de una dirección IP interna no enrutable, "169.254.169.254". Si el servicio Nginx está mal configurado, podría estar exponiendo esta IP interna, permitiendo que sea accesible al usar esta dirección en el campo del encabezado 'Host'.

Solución

Para solucionar la vulnerabilidad de exposición de metadatos en servicio de Nginx, es esencial asegurarse de que la configuración del servidor impida el acceso externo a la dirección IP interna usada para los metadatos de la nube. Esto se puede lograr mediante la configuración adecuada de las reglas de firewall o ajustando las directivas del servidor en la configuración de Nginx para restringir el acceso a estas direcciones IP. En la Figura 53, se muestra evidencia de que la filtración de la IP ha sido efectivamente parcheada en la configuración de Nginx.

```
if ($host ~* ^169.254.169.254$) {
    return 444;
}
```

Figura 53 : Evidencia de vulnerabilidad critica parchada

5.1.1.2 Explicación Vulnerabilidad de Estado Alto

La Vulnerabilidad detectada corresponde a posibles ataques de tipo *Clickjacking*, los cuales son ataques basados en la interfaz que engaña a los usuarios de un sitio web para que sin saberlo hagan click en enlaces maliciosos (Mozilla.org, s. f.). Los atacantes incrustan sus enlaces maliciosos en botones o páginas legítimas en un sitio web.

Solución

Para prevenir este tipo de ataques, es recomendable utilizar encabezados de seguridad como 'X-Frame-Options' y 'Content-Security-Policy' en las respuestas HTTP. Estos encabezados ayudan a proteger la aplicación al restringir cómo y dónde se puede incrustar el contenido de la misma. En la Figura 54, se muestra cómo esta vulnerabilidad fue parcheada en el servidor Nginx, implementando estos encabezados para fortalecer la seguridad.

```
server {  
    server_name geoparques.robbleaustral.cl;  
  
    add_header X-Frame-Options "DENY";  
    add_header Content-Security-Policy "frame-ancestors 'self'";  
}
```

Figura 54 : Evidencia de vulnerabilidad alta parchada

5.1.2 Recomendaciones para Vulnerabilidades no Críticas

Como se observó, existían otras vulnerabilidades que no se detallaron anteriormente, debido a que eran de criticidad baja. Estas vulnerabilidades son fáciles de solucionar configurando adecuadamente los CORS en el servidor Nginx. Se espera que todas las vulnerabilidades se puedan solucionar con el tiempo. Además, se recomienda realizar este tipo de pruebas periódicamente para detectar posibles vulnerabilidades a nivel de configuración o código y parchearlas, evitando así sufrir ataques no deseados al sistema de GeoParques.

5.2 Prueba de Rendimiento

Actualmente, es crucial que los sistemas informáticos soporten adecuadamente un flujo de peticiones, dada la alta demanda en el uso que pueden llegar a tener. Por esta razón, realizar pruebas de rendimiento se ha convertido en una solución esencial para medir la eficacia de las aplicaciones. Estas pruebas, aplicadas a sistemas, aplicaciones o infraestructuras tecnológicas, evalúan su comportamiento bajo diversas condiciones de carga o estrés (sqz, 2023).

Al realizar este tipo de pruebas a los sistemas, se pueden identificar problemas de rendimiento, como cuellos de botella, tiempos de respuesta lentos, errores de funcionalidad y saturación de recursos.

Es importante diferenciar entre los distintos tipos de pruebas de rendimiento, ya que a menudo hay confusión entre las pruebas de rendimiento, estrés, carga y otras. A continuación, se describirán las más relevantes.

- **Pruebas de Spike**, evalúan cómo un sistema maneja incrementos repentinos y grandes de peticiones, simulando un aumento brusco en usuarios o solicitudes. Su objetivo es identificar puntos débiles y verificar que el sistema pueda recuperarse después de la sobrecarga. Esto ayuda a asegurar la estabilidad y el rendimiento del sistema ante demandas inesperadas (Hamilton, 2024).
- **Prueba de carga**, promedio evalúa la capacidad del sistema bajo una carga típica de producción. Se simula el comportamiento promedio mediante incrementos graduales de usuarios y solicitudes. Esta prueba mantiene dicha carga durante un período de tiempo determinado para observar cómo responde el sistema (Campos, 2023).
- **Pruebas de estrés**, evalúan el rendimiento del sistema bajo cargas significativamente más pesadas que las habituales. Aunque siguen un patrón similar a las pruebas de carga promedio, las pruebas de estrés imponen una carga mucho mayor. Estas pruebas proporcionan información valiosa sobre la capacidad y la estabilidad del sistema en condiciones de alta demanda (Campos, 2023).
- **Pruebas de volumen**, también conocidas como pruebas de inundación, son un tipo de prueba de software en las que el sistema se somete a un gran volumen de datos. Estas pruebas se realizan para analizar cómo el rendimiento del sistema se ve afectado al incrementar el volumen de datos en la base de datos (Hamilton, 2024) .

Para este trabajo de título, se ha decidido realizar una prueba de estrés, ya que, se sabe que el jardín botánico de la Universidad Austral de Chile recibe aproximadamente 1000 visitantes un día de verano. Con el objetivo de poner a prueba la capacidad del sistema bajo condiciones extremas, se optó por simular el acceso de aproximadamente 2000 usuarios a la aplicación, ingresando en grupos de 200 cada 10 segundos. Esto se hace con el propósito de determinar si el sistema puede manejar un número exacto de usuarios concurrentes sin experimentar fallos o colapsos.

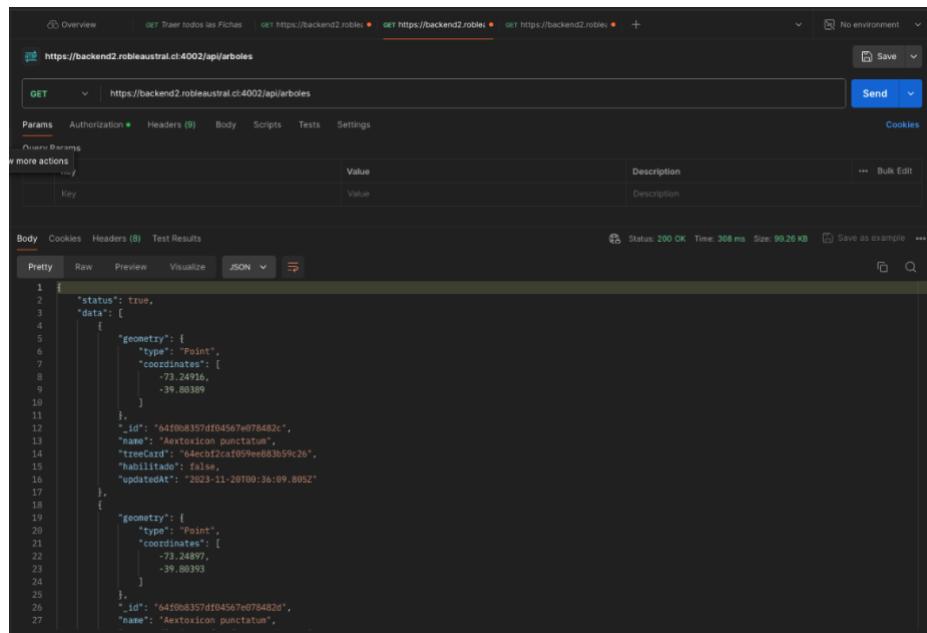
5.2.1 Realización de las Pruebas

Para realizar las pruebas, se utilizó un software diseñado específicamente para este propósito llamado JMeter, que es un proyecto de Apache utilizado como herramienta de pruebas de carga para analizar y medir el rendimiento de una variedad de servicios (Apache, s. f.).

5.2.2 Ejecución de las Pruebas

Para la ejecución de las pruebas, se realizaron el día 4 de junio del 2024 a las 14:43 pm. En donde, se decidió realizar las pruebas de carga a la GeoApi, ya que es uno de los módulos de nuestro sistema que recibe más peticiones hasta el momento.

Se decidió probar el endpoint donde se obtienen todos los árboles, ya que es uno de los más grandes en tamaño y uno de los más solicitados por varias aplicaciones del proyecto. En la Figura 55 se puede apreciar el endpoint siendo probado en Postman para confirmar que funciona correctamente.



```
1 {
2   "status": true,
3   "data": [
4     {
5       "geometry": {
6         "type": "Point",
7         "coordinates": [
8           -73.24916,
9           -39.80389
10        ]
11      },
12      "_id": "54fb0b357df04567e078482c",
13      "name": "Aextoxicum punctatum",
14      "treeCar": "4e4chfzca099ee883b59c26",
15      "habilitado": false,
16      "updatedat": "2023-11-20T00:36:09.805Z"
17    },
18    {
19      "geometry": {
20        "type": "Point",
21        "coordinates": [
22          -73.24897,
23          -39.80393
24        ]
25      },
26      "_id": "54fb0b357df04567e078482d",
27      "name": "Aextoxicum punctatum"
28    }
29  ]
30}
```

Figura 55 : Prueba del endpoint para obtener todo los árboles

Ya habiendo probado el endpoint y sabiendo que funciona correctamente, se procede a configurar un grupo en JMeter para realizar las pruebas. A este grupo de trabajo se le denominó "prueba de carga GeoApi", el cual cuenta con grupos de hilos basados en los parámetros que se pueden observar en la Tabla 19. Cada grupo cuenta con 200 hilos, aumentando hasta llegar a 2000 hilos concurrentes. Al ser 10 grupos de hilos, cada grupo tiene una duración de 10 segundos y se ejecutan uno después del otro. Según la estimación realizada por JMeter, las pruebas de carga con esta configuración tienen una duración total de 1 minuto y 40 segundos.

Tabla 19 : Parámetros de los grupos de hilo

Grupo de Hilos	Usuarios Concurrentes	Duración
1	200	10
2	400	10
3	600	10
4	800	10
5	1000	10
6	1200	10
7	1400	10
8	1600	10
9	1800	10
10	2000	10

Desde que se configuraron los grupos de hilos, a cada uno se le asignó una solicitud HTTPS dirigida a un endpoint específico para realizar consultas a la GeoApi, como se ilustra en la Figura 56.

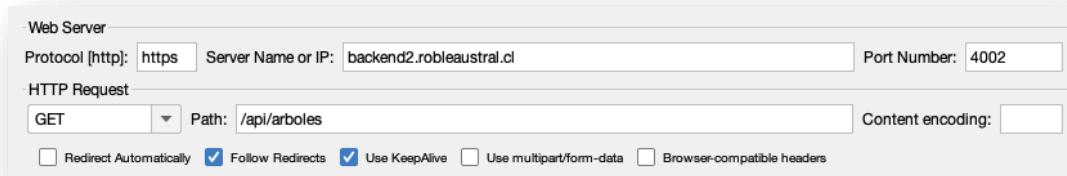


Figura 56 : Configuración de la petición HTTPS

Posteriormente, se configuraron los *listeners*³¹ que permiten obtener la información mediante datos y gráficos de las pruebas. Después de esta configuración, se accionó el botón para iniciar la prueba de carga, la cual duró un total de 1 minuto y 49 segundos, 9 segundos más de lo pronosticado por JMeter, lo cual indica que este pronóstico es bastante preciso.

³¹ Estos capturan y visualizan los resultados de las pruebas de rendimiento, mostrando datos en gráficos, tablas o archivos de texto.

5.2.3 Resultados

En esta sección, se mostrarán los gráficos obtenidos por los *listeners* configurados en JMeter y se presentará el monitoreo de la máquina virtual para ver cómo se comportó durante la prueba de carga. Además, se analizarán los resultados obtenidos de las pruebas, evaluando tanto los tiempos de respuesta y los errores como el rendimiento general del sistema bajo condiciones de alta carga.

La Figura 57 muestra el gráfico de hilos activos a lo largo del tiempo. Se observó cómo se incrementa la carga de usuarios concurrentes de manera lineal, empezando desde 0 hasta alcanzar los 2000 hilos al final de la prueba. Posteriormente, se simula que los usuarios comienzan a salir del sistema, por lo que se puede apreciar que al final la curva empieza a descender.

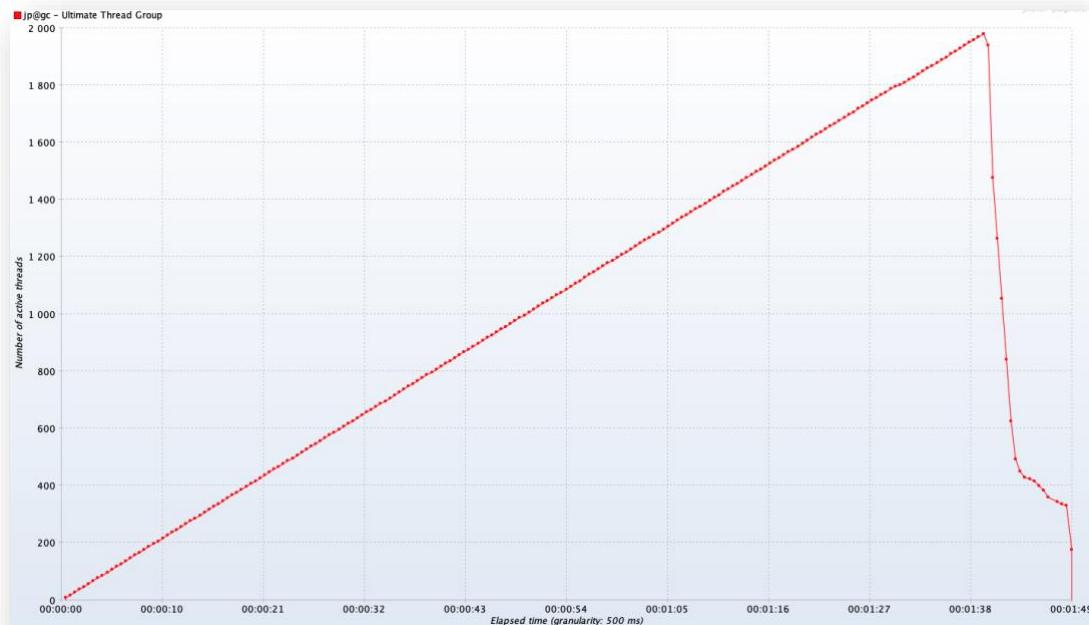


Figura 57 : Gráfico de hilos activo a lo largo del tiempo

La Figura 58 muestra que, inicialmente, las solicitudes exitosas se mantienen estables alrededor de 400 transacciones por segundo (TPS), mientras que las fallas son casi inexistentes. Sin embargo, al alcanzar los 900 hilos, las fallas comienzan a aparecer en el segundo 45 y aumentan rápidamente, alcanzando un máximo de cerca de 3600 TPS. Esto indica que el sistema maneja bien la carga inicial, pero no puede escalar adecuadamente bajo alta demanda. Por lo tanto, se puede concluir que el sistema empieza a fallar cuando

hay 900 usuarios realizando peticiones al mismo tiempo, lo cual es un número bastante alto y positivo en comparación con lo que se espera que soporte el sistema, por lo que es un resultado óptimo. Dado que la aplicación está en sus inicios, no se prevé que llegue a ese número de peticiones por segundo.

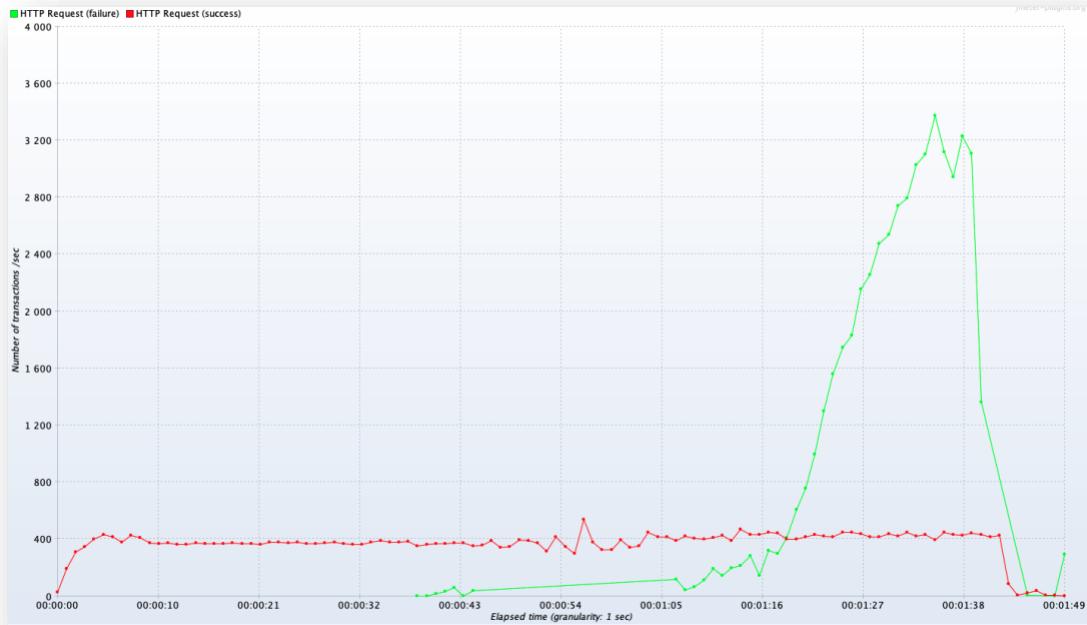


Figura 58 : Gráfico de transacciones a lo largo del tiempo

En la Figura 59, se observa el gráfico de tiempo de respuesta a lo largo del tiempo. Se ve que muestra que, inicialmente, los tiempos de respuesta son estables hasta aproximadamente el segundo 76. A partir de ese momento, los tiempos de respuesta comienzan a aumentar rápidamente y alcanzan un máximo crítico alrededor del segundo 90, llegando a cerca de 36,000 ms, lo cual es demasiado y refleja que la GeoApi en ese instante no estaba respondiendo correctamente. Después de este máximo, los tiempos de respuesta fluctúan pero permanecen elevados. Estos resultados tienen concordancia con lo visto en la Figura 54, que alcanzaba los máximos entre los segundos 90 y 100.

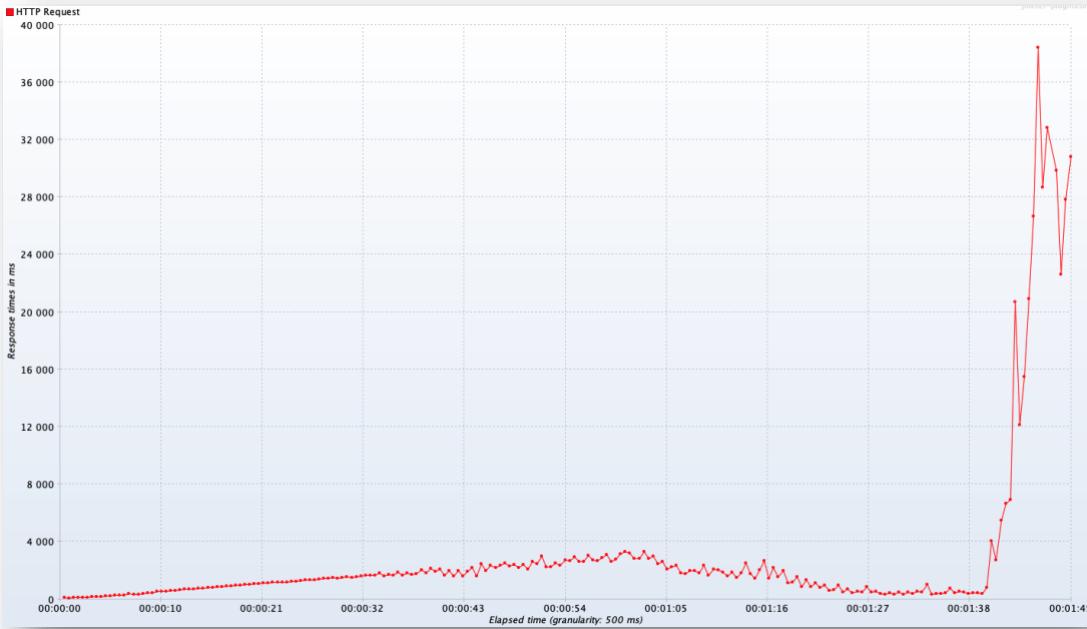


Figura 59 : Gráfico de tiempo de respuesta a lo largo del tiempo.

Para finalizar esta sección, la Figura 60 muestra un extracto de un video³², se muestra el monitoreo de la máquina virtual donde se aloja la GeoApi. Se observa que entre los segundos 55 y 60 de la prueba de carga, el monitoreo de la CPU y la RAM muestra que la CPU está casi al máximo de su capacidad, con un uso entre 92.1% y 98.7% en sus núcleos, y la RAM utiliza 5.78 de los 6 GB disponibles. La alta saturación de la CPU y la RAM indica que el servidor está sobrecargado, lo que contribuye a los problemas de rendimiento mencionados anteriormente.

Figura 60 : Extracto del monitoreo del servidor de la GeoApi

³² https://www.youtube.com/watch?v=saL5Q70TqFc&ab_channel=bastidebasti

A pesar de que los resultados obtenidos en la prueba de carga indican que existen problemas significativos al alcanzar aproximadamente 900 usuarios concurrentes, es importante tener en cuenta que este es un escenario simulado. No se espera que la GeoApi tenga ese nivel de tráfico, ya que, basándonos en lo mencionado anteriormente, el máximo de visitantes del jardín botánico es de aproximadamente 1000 personas en un día de verano. Es poco probable que todos los visitantes utilicen la aplicación simultáneamente. Por lo tanto, se cree que es casi imposible alcanzar los niveles observados en las pruebas.

Otro punto importante a destacar es que la máquina virtual donde se encuentra alojada la GeoApi tiene recursos limitados, por lo que era evidente tener estos problemas de rendimiento. Se espera que en el futuro el sistema de geoparques pueda escalar tanto horizontalmente como verticalmente.

Por lo tanto, se puede concluir que el sistema funciona adecuadamente para una carga moderada de personas y que no debería presentar problemas, dado que la aplicación se encuentra recién dándose a conocer y está siendo utilizada por un número reducido de personas.

5.3 Metodología de Validación Usabilidad

Para evaluar la usabilidad de las distintas funcionalidades que ofrece GeoParques – Visitantes y obtener percepciones y retroalimentación de los usuarios, se decidió realizar un formulario de usabilidad en el cual se solicitó a los encuestados realizar diversas actividades en la aplicación del visitante, y luego entregar su opinión sobre estas actividades.

Mas específicamente y considerando que este sistema está pensado para ser utilizado por cualquier tipo de usuario (rango etario, personas que conocen de especies botánicas, entre otras), se llevaron a cabo dos métodos para aplicar el formulario de usabilidad: pruebas de usabilidad de pasillo y un taller con personas que no conocen el proyecto y que tampoco conocen al creador de este. Esto se hizo con el fin de poder tener una apreciación no sesgada del proyecto, ya que a veces, cuando se realizan pruebas de pasillo, se presenta la aplicación a personas que conocen al creador y, por ser amables, responden el formulario de manera muy positiva, lo que podría alterar los resultados. Por este motivo, se decidió comparar los dos conjuntos de resultados.

5.3.1 Descripción del Formulario

El formulario fue desarrollado en la aplicación que ofrece Google para la creación y administración de estos, este se puede visualizar en el Anexo U: Formulario para validación de usabilidad. El formulario constaba de ocho secciones. En cada sección, cada usuario tenía que realizar tareas específicas para, posteriormente, poder expresar su opinión respecto de lo que pudo apreciar. Las sesiones definidas en el formulario fueron las siguientes:

- Preguntas para conocer al encuestado, como por ejemplo su nombre, su edad y si conoce una pregunta si conoce sobre las aplicaciones web progresivas.
- Actividades para navegar e interactuar con la vista principal (cuando el usuario entra por primera vez a la aplicación), además de explorar como cambiar el idioma en la aplicación
- Actividades para navegar en la vista mapa.
- Actividades para activar la georreferenciación.
- Actividades navegar y escanear un código QR.
- Actividades para navegar en la vista de especies.
- Retroalimentación Final del usuario.
- 10 preguntas basadas en el sistema *System Usability Scale* (SUS) creado por John Brooke en 1986, en las que los encuestados deben responder utilizando una escala Likert de cinco puntos, que va desde 'Totalmente en desacuerdo' hasta 'Totalmente de acuerdo' (Arias, s. f.). Esta escala se puede apreciar en la Figura 61.



Figura 61 : Formato de la escala de Likert utilizado en la preguntas SUS

5.3.2 Pruebas Usabilidad de Pasillo

Las pruebas se llevaron a cabo durante la semana del 13 de mayo. Se solicitó a estudiantes presentes en el edificio de informática de la Universidad Austral de Chile, así como a otros conocidos, que primero revisaran el formulario y posteriormente exploraran la aplicación por un tiempo antes de contestar el formulario. Estas actividades se realizaron a lo largo de toda la semana y se logró obtener la participación de 15 encuestados.

5.3.3 Pruebas de Usabilidad en Taller

El taller de usabilidad se llevó a cabo en el edificio de informática durante una clase de introducción a la programación, el miércoles 22 de mayo a las 15:50 horas, y estuvo dirigido a estudiantes de primer año. Durante el taller, se presentó a 26 estudiantes el proyecto, explicando la problemática y las soluciones propuestas. Se mostraron los avances de GeoParques – Administrador, pero no de GeoParques – Visitante, ya que esa era la aplicación que iban a probar y no se debía proporcionar información previa a los encuestados. El objetivo de estas pruebas es que los participantes tengan conocimiento limitado sobre la aplicación y su funcionamiento, para evitar alterar los resultados de la usabilidad.

Los estudiantes mostraron gran atención durante la presentación y se comprometieron activamente en el taller de usabilidad. Después de finalizar la explicación, se mostró un código QR para que pudieran abrir la aplicación en sus dispositivos móviles. Posteriormente, completaron el formulario de evaluación. Cabe destacar la disposición de los estudiantes para participar y retroalimentar el proyecto.

5.3.4 Resultado de las Pruebas

En total, se logró recopilar 41 respuestas mediante las pruebas de pasillo y pruebas en el taller. Jakob Nielsen, uno de los líderes en el campo de la usabilidad, ha publicado numerosos estudios y artículos en los que apoya la idea de que, para que las pruebas de usabilidad sean exitosas, se necesita contar con un mínimo de 5 personas (Nielsen, 2000). Por esta razón, la cantidad de respuestas obtenidas constituye una muestra suficiente y diversa entre los usuarios.

Es importante mencionar que se compararon las dos muestras para analizar los resultados. Esto se puede apreciar desde la Figura 62 en adelante, donde los gráficos muestran barras en las que el color naranja corresponde a las pruebas de pasillo y el azul a las pruebas en el taller. Además, el eje X representa la cantidad de participantes y el eje Y las respuestas seleccionadas por los participantes. También es relevante señalar que en las barras se muestran los porcentajes de respuestas, indicando cuántas personas respondieron cada pregunta.

5.3.4.1 Resultado Pregunta Sobre Conocimiento de una Aplicación PWA

La primera pregunta indaga sobre el conocimiento de los encuestados sobre la existencia de aplicaciones web progresiva es resultado de esta se puede apreciar en la Figura 62, en donde se muestra que, en las pruebas de pasillo, el 53.8% de los encuestados han oído hablar de las aplicaciones web, pero no en detalle. Por otro lado, en la prueba del taller, el 61.5% no conoce estas aplicaciones, el 34.6% ha oído hablar de ellas, pero sin detalles, y solo el 3.8% si las conoce. Esto indica que dentro de los encuestados no hay un conocimiento claro de estos tipos de aplicaciones y que el conocimiento que existe es superficial. Esto es debido a la poca difusión que hoy existe de este tipo de aplicaciones.

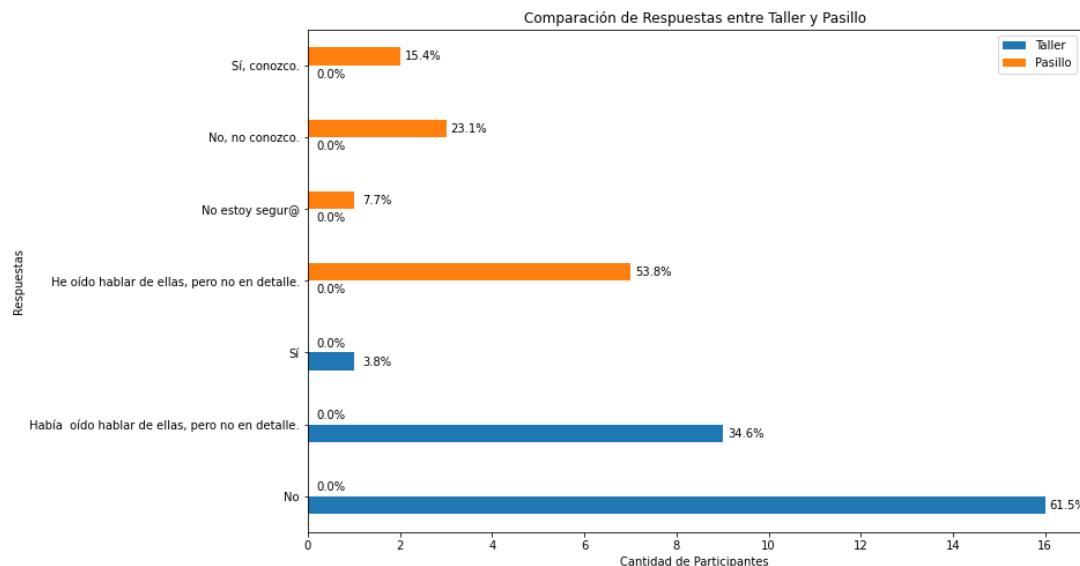


Figura 62 : Gráfico de barra de las respuestas sobre el conocimiento de las PWA

5.3.4.2 Resultado Sobre las Actividades de la Vista Principal

Esta sección del formulario contemplaba dos preguntas. Para la pregunta 2 indagaba sobre como encontrar una tarjeta con información sobre el parque los resultados obtenidos se muestra en la Figura 63. La mayoría de los encuestados encontraron la opción para ver la información sobre el parque sin dificultades, con un con un 73.3% en las pruebas de pasillo y un 61.5% en el taller. Sin embargo, un 34.6% en el taller y un 13.3% en pasillo tuvieron dificultades para encontrar la información. Un pequeño porcentaje no pudo encontrar la opción. Esto pudo haber sucedido ya que las encuestados se iban directamente al mapa y no seguían interactuando de la vista inicial, por lo tanto, se sugiere la necesidad de mejorar la accesibilidad de para esta información.

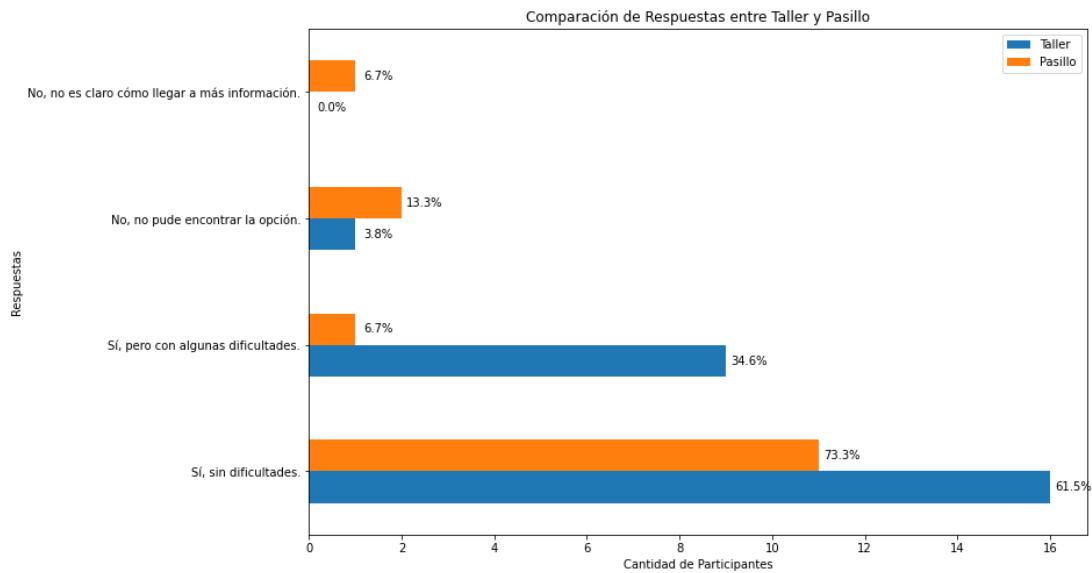


Figura 63 : Respuesta de la pregunta de tarjeta de información del parque

Para la pregunta 3, que indaga sobre la facilidad para cambiar el idioma de la aplicación a inglés, los resultados se pueden ver en la Figura 64. En las pruebas de pasillo, el 93.3% de los encuestados pudo cambiar el idioma fácilmente, mientras que el 6.7% lo logró con algunas dificultades. En las pruebas en el taller, el 88.5% pudo cambiar el idioma sin problemas, el 7.7% lo hizo con dificultades, y el 3.8% intentó, pero no pudo cambiarlo. Estos resultados indican que la mayoría de los usuarios pudo realizar esta tarea sin dificultad, aunque una persona no logró cambiar el idioma. Por esta razón, se debería analizar la forma en la que está diseñada esta opción para asegurar que el 100% de las personas puedan cambiar el idioma sin dificultades

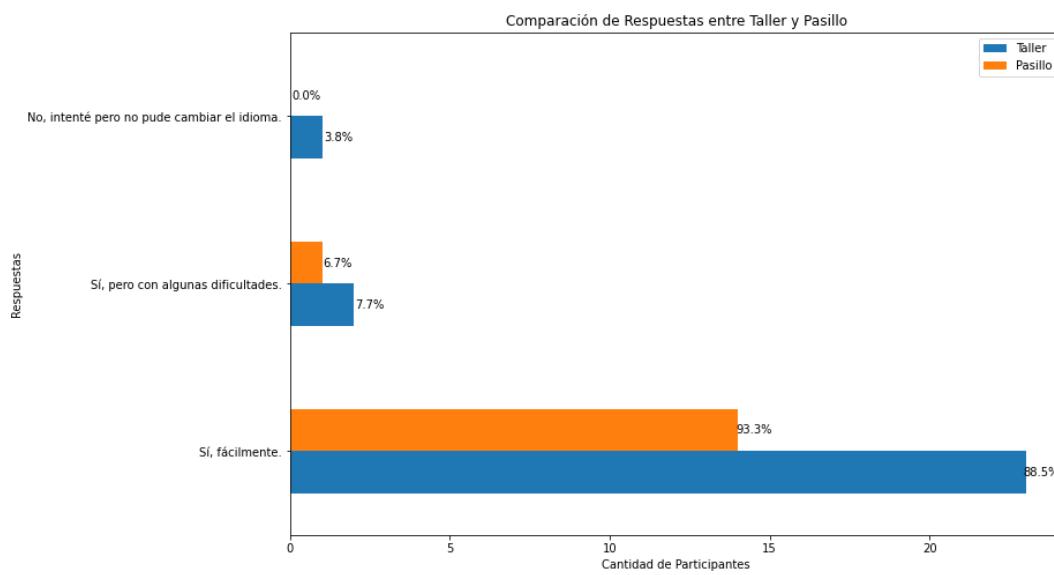


Figura 64 : Respuestas sobre la actividad de cambiar el idioma

5.3.4.3 Resultado sobre las Actividades de la Vista del Mapa

Esta sección cuanta con tres preguntas relacionadas a la interacción con las opciones y funciones que se puede realizar en el mapa.

La pregunta número 4 corresponde a que el encuestado se dirija al mapa y busque información sobre un árbol en particular. Los resultados obtenidos se pueden visualizar en la Figura 65. En las pruebas de pasillo, todos los encuestados (100%) encontraron la información del árbol sin dificultades. En las pruebas en el taller, el 96.2% también encontró la información sin problemas, mientras que un 3.8% lo logró, pero con algo de tiempo. Esto muestra que esta acción en la interfaz es muy accesible y fácil de usar, aunque hay una ligera oportunidad de mejorar para hacerla más intuitiva a la hora de buscar la información de un árbol.

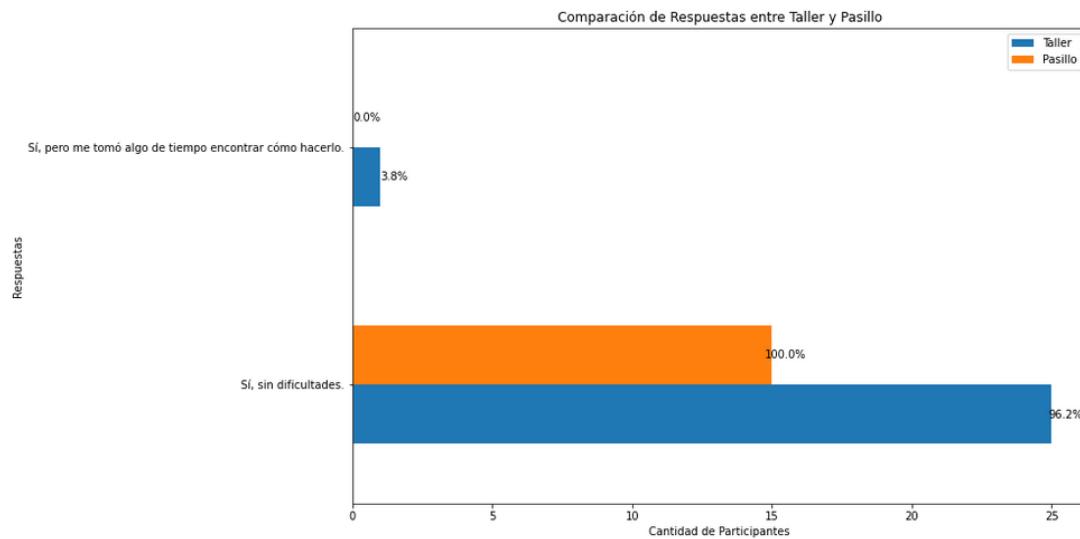


Figura 65 : Respuesta de la pregunta como ver la información de un árbol

Para la pregunta número 5, que indaga si la información mostrada en la ficha de especie resulta útil, los resultados se pueden apreciar en la Figura 66. En las pruebas de pasillo, el 60% de los encuestados indicó que toda la información es útil y comprensible, mientras que un 33.3% consideró que la mayor parte de la información es útil y comprensible, y un 6.7% señaló que algo de la información es clara o útil. En las pruebas en el taller, el 57.7% encontró toda la información útil y comprensible, el 38.5% consideró que la mayor parte de la información es útil y comprensible, y el 3.8% indicó que algo de la información es clara o útil. Esto sugiere que la mayoría de los encuestados encuentran la información útil; sin embargo, existe margen para explicar con mayor claridad y agregar más información útil para los visitantes.

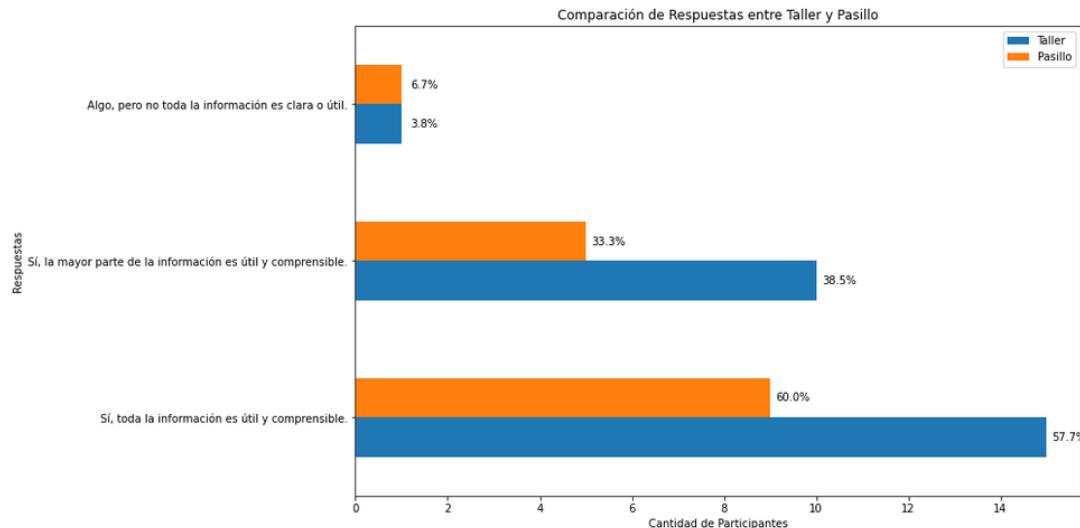


Figura 66 : Respuesta a la pregunta si le resulta útil la información de la ficha especie.

En la pregunta número 6, que indaga sobre la facilidad o dificultad para reportar información incorrecta de una especie o árbol en el parque, los resultados se pueden ver en la Figura 67. Donde, en las pruebas de pasillo, el 86.7% de los encuestado indicó que encontró rápidamente como reportarlo, mientras que un 13.3% tuvo alguna dificultad, pero pudo hacerlo. en las pruebas en el taller 80.8% encontró rápidamente como reportarlo, y un 19.2% lo lograron con alguna dificultad. Esto sugiere a que los encuestados encuentran fácil el cómo reportar errores.

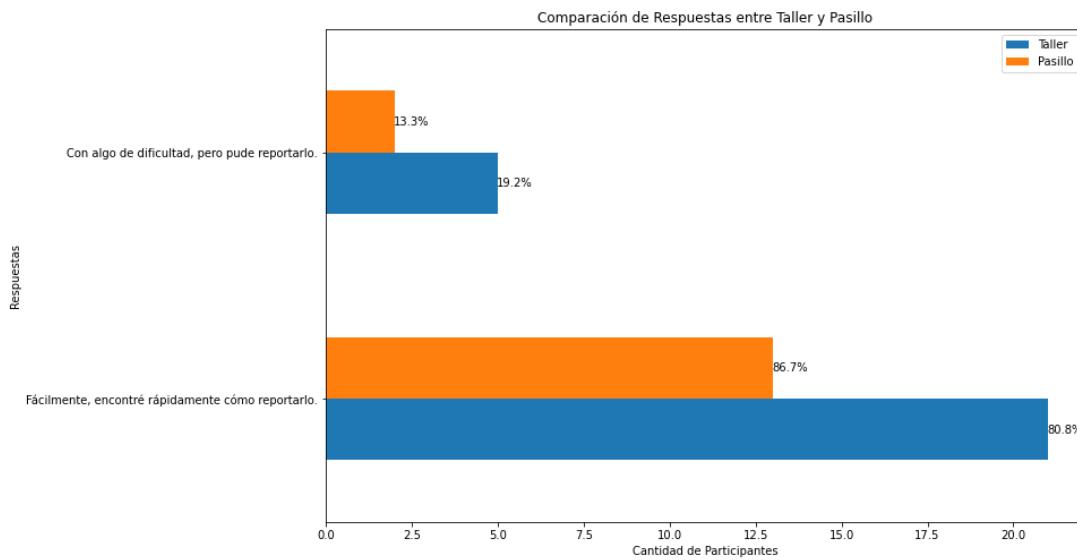


Figura 67 : Respuesta a la pregunta sobre la facilidad o dificultad de reportar una especie

Con el objetivo de obtener una retroalimentación de la actividad solicitada, se le preguntó a los encuestados, si tenía algún tipo de comentario, pregunta o sugerencia según lo realizado. Los comentarios que más se repitieron fueron parecidos a los siguientes:

“Podrían añadir lugares de referencia en dónde se encuentran los árboles que no son nativos”

“podrían colocar en la información adicional alguna curiosidad sobre la especie para atraer aún más la atención del consumidor”

“Reportar un problema fue bastante fácil e intuitivo, creo que la opción está bien posicionada, solo le agregaría color o algo aparte de la flecha que ayude a descifrar que se puede presionar sobre la pregunta para abrir la ventana de reportar un problema (pues como todo el fondo es blanco, quizás para algunas personas no quede tan claro que se puede presionar ahí mismo).”

“Si bien es fácil encontrar el botón de reporte bajo la imagen, lo dejaría con un color rojizo o ícono tipo danger. Creo que se explicaría por sí solo sin leer el texto.”

Se pueden rescatar muchos aspectos de los comentarios dejados por los encuestados, como la sugerencia de añadir en el mapa referencias sobre la ubicación de los árboles nativos, así como incluir curiosidades del parque que se puedan enviar como notificaciones. Además, se sugiere que el botón para reportar especies sea más colorido para llamar la atención, ya que en su estado actual algunas personas pueden pasarlo por alto.

5.3.4.4 Resultado Sobre la Actividad de Activar Georreferenciación del Mapa

La pregunta número 7, indaga si los encuestados pudieron activar la georreferenciación para conocer su posición en el parque, los resultados se pueden ver en la Figura 68. En las pruebas de pasillo, el 73.3% de los encuestados pudo hacerlo sin problemas, un 13.3% lo logró, pero con dificultades, y otro 13.3% intentó, pero no pudo activarla. En las pruebas en el taller, el 61.5% pudo activarla sin problemas, un 19.2% lo hizo con dificultades, y un 19.2% intentó, pero no pudo. Esto sugiere que la gran mayoría puede activar la georreferenciación de sus dispositivos móviles. Sin embargo, todavía hay una porción significativa de usuarios que tuvieron dificultades al activarla. Esto se puede deber a factores como no haber dado los permisos de localización al ingresar a la aplicación, bloqueando esta opción. Esto indica que se pueden hacer mejoras para facilitar esta función.

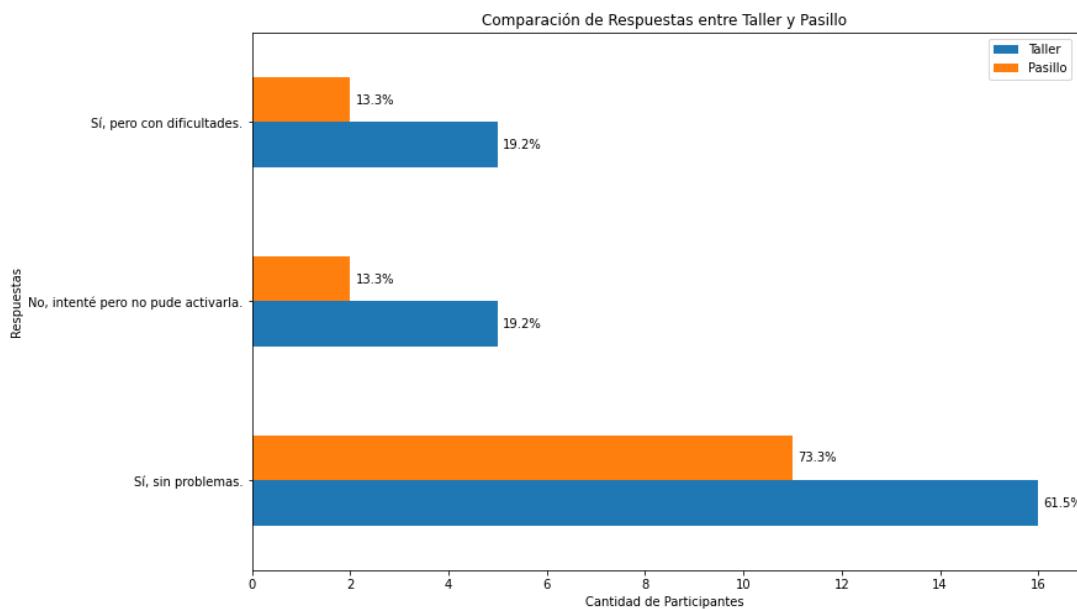


Figura 68 : Respuestas sobre la pregunta de activar georreferenciación

Para la pregunta 8, sobre la activación de la opción de Geoproximidad, que muestra las especies (árboles) más cercanas a la posición del usuario, los resultados se pueden ver en la Figura 69. En las pruebas de pasillo, el 66.7% de los encuestados pudo activar la opción sin problemas, el 20% lo logró con algunas dificultades, el 6.7% intentó pero no pudo activarla, y otro 6.7% no encontró la opción. En las pruebas en el taller, el 61.5% pudo activarla sin problemas, el 15.4% lo hizo con dificultades, otro 15.4% intentó pero no

pudo, y el 7.7% no encontró la opción. Estos resultados sugieren que, aunque la mayoría de los usuarios puede activar esta función, hay un número considerable de usuarios que encuentran dificultades o no pueden activarla, indicando la necesidad de mejorar la accesibilidad de esta funcionalidad para los usuarios finales.

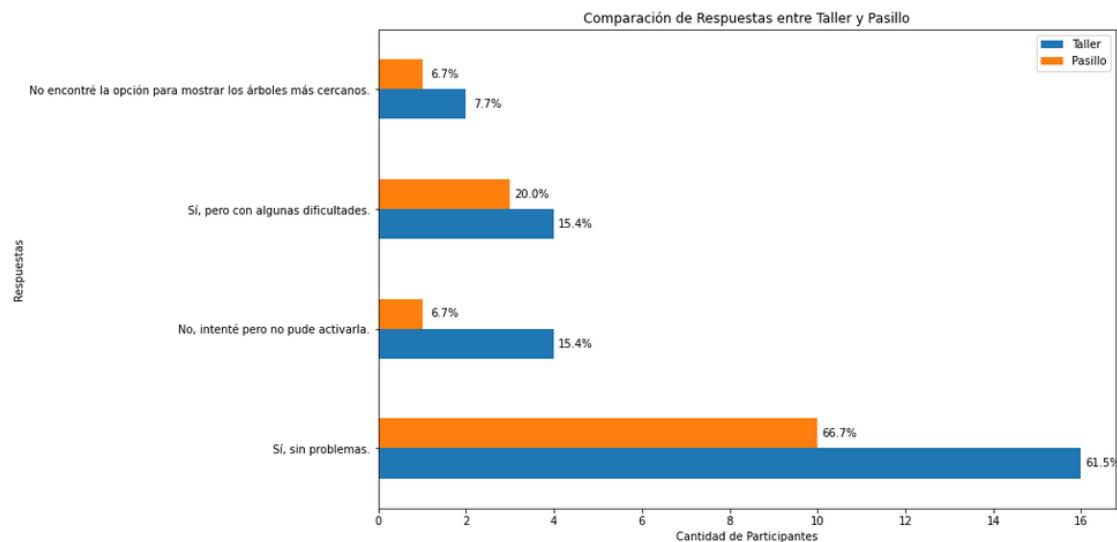


Figura 69 : Respuesta sobre la activación de la Geoproximidad

Terminando las preguntas relacionadas con activar la opción de localización y Geoproximidad, se realizó la siguiente pregunta:

“Respecto a las preguntas anteriores, ¿qué tan difícil fue encontrar lo solicitado?”

Esta pregunta contaba con una escala del 1 al 10, donde 1 era "muy fácil" y 10 era "muy difícil". Los resultados de esta pregunta se muestran en la Figura 70. En las pruebas de pasillo, el 66.6% de los encuestados calificó la tarea como fácil (puntuación de 1 y 2). En cambio, en las pruebas que se realizaron en el taller, el 26.9% calificó la tarea como fácil. Sin embargo, como se aprecia en la Figura 70, las puntuaciones fueron variadas, llegando hasta el punto donde una persona encontró estas acciones muy difíciles. Esto indica que en el taller los encuestados encontraron más difícil la tarea, lo que nos dice que estas acciones de geolocalización necesitan mejorar la accesibilidad y facilidad para su uso.

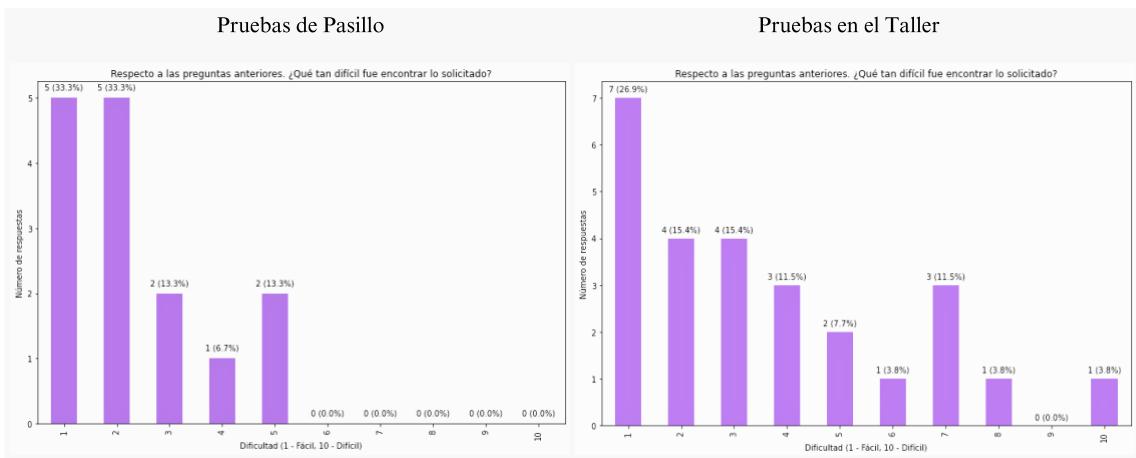


Figura 70 : Resultado de las actividades para activar localización y Geoproximidad

La retroalimentación que se obtuvo de los encuestados realizando esta actividad fueron las siguientes:

“La GeoProximidad podría tener un botón directo en el mapa”

“Yo pondría el botón para mi geolocalización más abajo y aún más visible (más grande quizás)”

“la opción de geo proximidad está dentro del menú con el símbolo que comúnmente se utiliza para filtrar por lo que puede causar confusión al consumidor”

“Está muy escondida la opción de proximidad”

Estos comentarios señalan y confirman que el botón para activar la Geoproximidad de la aplicación Geoparques está mal posicionado en el Drawer. Como solución, se espera que este botón pueda estar directamente en el mapa para su fácil activación y que el usuario final tenga conocimiento de que existe esta funcionalidad.

5.3.4.5 Resultado Sobre la Utilización de los Código QR

Está sección cuanta con dos preguntas, en donde se estudia el comportamiento del encuestado con las opciones que nos ofrece la aplicación para escanear un código QR de una especie. Para ello, se le facilitó a los encuestados un código QR y así realizar el escaneo.

La pregunta número 9 indaga sobre la facilidad para escanear un código QR y obtener información sobre una especie; los resultados se muestran en la Figura 71. En las pruebas de pasillo, el 93.3% de los encuestados indicó que fue fácil y encontraron rápidamente

cómo hacerlo, mientras que un 6.7% intentó, pero no pudo escanear el código. En las pruebas en el taller, el 92.3% encontró la tarea fácil y rápida, y un 7.7% logró realizarla, aunque les tomó algo de tiempo entender cómo hacerlo. Estos resultados sugieren que la mayoría de los usuarios encuentran fácil realizar esta acción, lo cual era de esperarse ya que el uso de los códigos QR está estandarizado. Por otro lado, la persona que no pudo escanear el código QR podría haber enfrentado problemas debido a que no otorgó los permisos necesarios para activar la cámara de su dispositivo móvil.

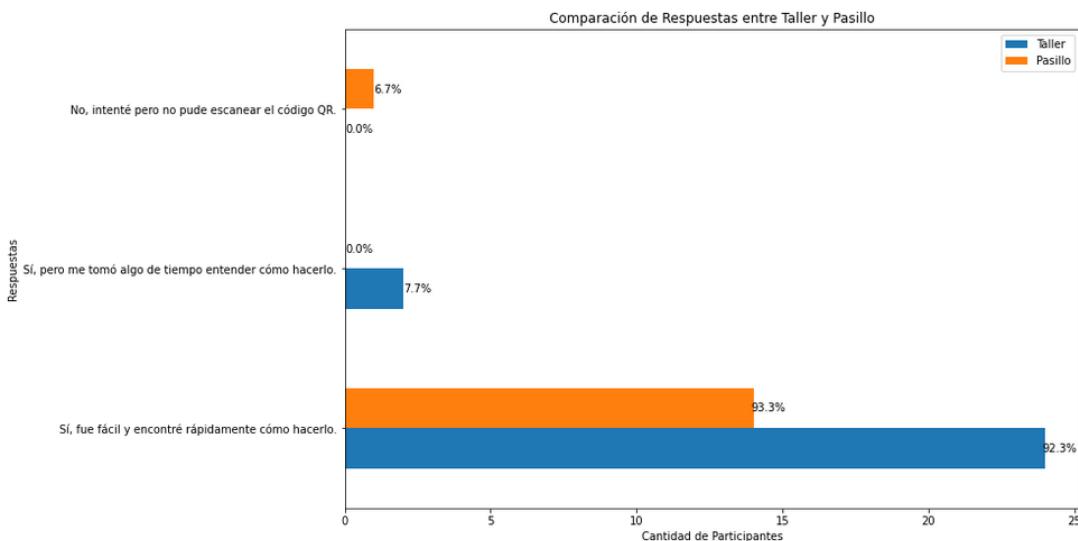


Figura 71 : Resultado de la actividad de escanear un código QR

La pregunta 10 tenía como objetivo determinar por dónde los encuestados escaneaban el código QR. Los resultados se pueden ver en la Figura 72, donde en las pruebas de pasillo, el 53.3% usó la aplicación de cámara de su celular, el 33.3% utilizó la vista de código QR y el 13.3% lo hizo a través del mapa. En las pruebas en el taller, el 50% usó la cámara del celular, el 34.6% la vista de código QR y el 15.4% el mapa. Esto indica que la mayoría de los usuarios prefirieron utilizar la cámara de su celular para escanear el QR de la especie. Esto es curioso, ya que la aplicación cuenta con dos opciones para poder escanear; sin embargo, el encuestado prefería salir de la aplicación y utilizar la cámara de su dispositivo. Esto se puede deber a que el encuestado no exploró bien las funcionalidades de la aplicación o simplemente tiene la costumbre de que, al escanear un QR, lo hace de forma casi instintiva utilizando la aplicación de su teléfono.

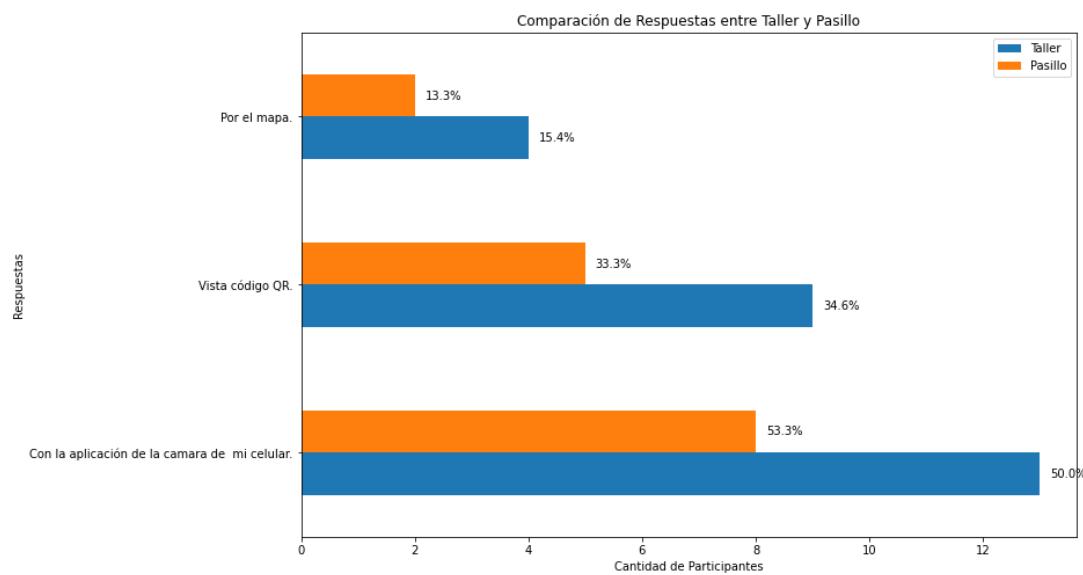


Figura 72 : Resultado de la pregunta de por qué medio se escanea el código QR

Al igual que en la sección anterior, se preguntó qué tan difícil fue encontrar lo solicitado. En la Figura 73, se pueden apreciar los resultados donde se observa que, en general, los encuestados encuentran fácil lo que se le solicitó. Sin embargo, en las pruebas de taller, algunos encuestados encontraron más difícil esto, lo que resultó en una mayor dispersión en las respuestas. En conclusión, la forma de consumir los códigos QR tanto en la aplicación como de manera general es de fácil uso.

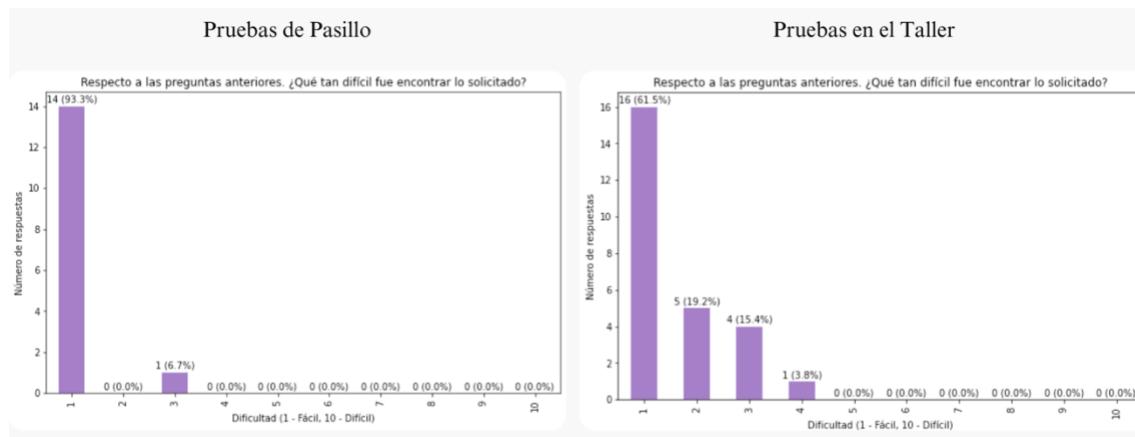


Figura 73 : Resultado de las actividades para escanear códigos QR

Para concluir esta sección, a continuación, se presentan los comentarios y retroalimentación proporcionados por los usuarios, que fueron los siguientes:

“El QR funciona perfectamente, pero en el menú de escanear QR colocaría una forma de volver al mapa rápidamente.”

“No tengo ninguna sugerencia, fue sencillo todo el proceso”

“Considero pertinente aclarar que utilicé el escáner de mi celular en lugar del integrado en la app no por dificultad para acceder a él, sino que por la costumbre de hacerlo de esa forma. Quise hacer hincapié en eso, pues pienso que la opción del escáner de la página está bien configurada de todos modos.”

“La parte del qr no funciona, se queda en [Loading...]”

Los comentarios recibidos son sumamente positivos, y confirman que las opciones para escanear códigos QR están bien implementadas. Además, validan lo comentado anteriormente acerca de que los usuarios tienden a utilizar otras aplicaciones para escanear los QR, dado que están acostumbrados a realizar esta acción con dichas herramientas. Por último, se destaca un comentario sobre un problema con la opción de escanear que no funciona y se queda en 'Loading'. Esto ocurre porque la aplicación no cuenta con los permisos necesarios para utilizar la cámara. Se podría mejorar este aspecto asegurando que, al momento de abrir el modal de la cámara, se vuelva a solicitar al usuario que permita el uso de la cámara.

5.3.4.6 Resultado Sobre la Utilización de Vista Especie

Esta sección cuenta con dos preguntas y su objetivo es validar la funcionalidad y la utilidad de la vista de especies. La pregunta 11 indaga sobre la facilidad para encontrar la opción de explorar todas las especies del parque, y los resultados se pueden ver en la Figura 74. En las pruebas de pasillo, el 100% de los encuestados encontró la opción fácilmente. En la prueba en el taller, el 88.5% encontró y usó la opción sin problemas, el 7.7% no intentó buscar esta información y el 3.8% la encontró pero con cierta dificultad. Los resultados son muy positivos y alentadores, ya que la mayoría de los encuestados no tuvo dificultad para encontrar lo solicitado, lo cual indica que esta opción está bien integrada en el sistema.

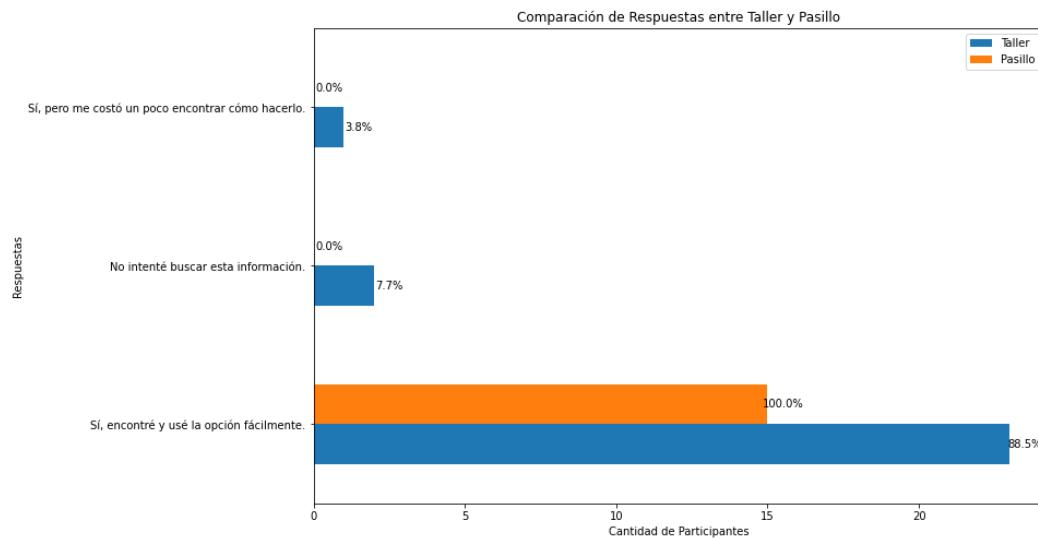


Figura 74 : Resultado sobre encontrar la opción de explorar las especies

La pregunta 12 indaga sobre la facilidad para visualizar en el mapa la ubicación de una especie que llamó la atención, y los resultados se pueden apreciar en la Figura 75. En la prueba de pasillo, el 86.7% de los encuestados pudo ver la especie en el mapa sin problemas, mientras que el 13.3% lo logró con dificultades. En las pruebas en el taller, el 84.6% pudo ver la especie en el mapa sin problemas, el 11.5% lo hizo con dificultades, y el 3.8% no intentó buscar la especie en el mapa. Estos resultados sugieren que los encuestados encontraron relativamente fácil realizar esta acción en la aplicación, lo cual nos permite concluir que esta opción cuenta con una buena usabilidad y que el diseño está bien logrado.

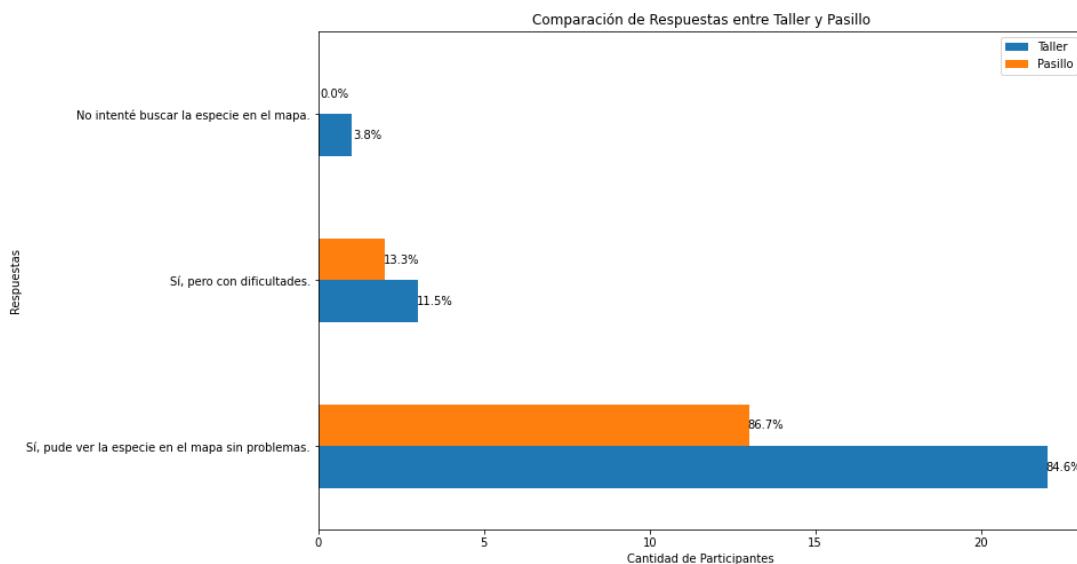


Figura 75 : Resultado sobre visualizar en el mapa una especie

5.3.4.7 Retroalimentación Final de los Encuestados

Para concluir la encuesta de validación de usabilidad, se pidió a los encuestados que dejaran sugerencias y opiniones generales sobre la aplicación y sus impresiones sobre el proyecto. Esto se hizo con el fin de recoger las percepciones de las personas acerca del proyecto. A continuación, se presentarán algunos de estos comentarios:

“Muy buena aplicación, muy útil para las personas que nos gustan las plantas”

“Se me ve muy buena la app, siento que para alguien que tiene nociones básicas de cómo usar una app es fácil de usar y eso es positivo, pensando en adultos que les guste salir al jardín botánico.”

“Muy buena app!! Sigan así, esta increíble y muy útil”

“la vista de la interfaz en el mapa podría ser más pequeña debido a que si lo buscas como página se siente una acumulación de iconos, podría abrir la interfaz con algún gesto como deslizar para descongestionar la vista”

“El ícono del pino para referenciar a los árboles o especies me parece fuera de lugar ya que esta es una especie introducida, póngale una araucaria que igual son icónicas o algo así”

“Hay muchas cosas de en inglés que aún no están traducidas. Aparte de eso, la aplicación está increíble, todo super bien hecho, visualmente placentera, intuitiva e informativa.”

Se pueden observar comentarios muy positivos sobre la aplicación, así como sugerencias para mejorar la usabilidad y hacerla más placentera. Además, algunos comentarios señalaban que había partes de la aplicación que no estaban completamente traducidas. Actualmente, se está trabajando para corregir las secciones que no están traducidas.

5.3.4.8 Resultados de Usabilidad Utilizando *System Usability Scale*

Para comprender los resultados obtenidos en las preguntas orientadas al SUS, es necesario entender el algoritmo utilizado para calcular los puntajes, el cual puede ser complejo. Primero, se debe recordar que la escala usada es Likert, que va de 1 a 5. Para calcular el puntaje de un encuestado, se sigue la siguiente secuencia de pasos:

- Sumar las respuestas de los enunciados impares y después restar 5.
- Sumar las respuestas de los enunciados pares y restar 2.5.
- Sumar ambos resultados y multiplicarlos por 2.5.

Este proceso se realiza para todos los encuestados y, posteriormente, se calcula el promedio de los puntajes obtenidos.

A continuación, se presentan los resultados obtenidos mediante el formulario aplicado tanto en las pruebas de pasillo, que se pueden apreciar en la Figura 76, como en las pruebas realizadas en el taller, visibles en la Figura 77.

Persona	0	1	2	3	4	5	6	7	8	9	SUS Score
Persona 1	4	1	4	2	5	1	5	1	5	1	92.5
Persona 2	4	1	3	2	5	1	4	2	4	2	80
Persona 3	4	3	5	3	5	2	4	3	4	1	75
Persona 4	3	2	5	2	4	1	4	2	5	2	80
Persona 5	1	3	4	4	4	3	3	2	5	2	57.5
Persona 6	5	1	4	3	5	1	4	1	5	1	90
Persona 7	5	2	4	3	5	1	4	2	5	1	85
Persona 8	3	2	5	2	5	2	3	2	5	2	77.5
Persona 9	5	1	4	2	5	1	5	2	5	1	92.5
Persona 10	4	1	4	2	5	2	5	1	5	2	87.5
Persona 11	5	1	5	1	5	1	5	2	5	1	97.5
Persona 12	5	1	5	1	5	1	5	1	4	1	97.5
Persona 13	3	1	5	1	5	1	4	1	4	1	90
Persona 14	5	1	5	1	5	1	5	1	5	1	100
Persona 15	4	1	4	1	4	1	3	1	4	1	85

Figura 76 : Resultados de preguntas SUS en prueba de pasillo

El promedio obtenido de las preguntas SUS en las pruebas de pasillo fueron: 85.83 puntos.

Por otra parte, los resultados obtenidos del formulario SUS en el taller se presentan en la siguiente Figura 77.

Persona	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	SUS Score
Persona 1	5	1	5	1	5	1	5	1	5	1	100
Persona 2	2	1	2	2	1	3	2	3	3	3	45
Persona 3	3	3	4	2	4	1	4	1	4	4	70
Persona 4	3	1	5	1	5	1	5	1	4	1	92.5
Persona 5	3	2	4	2	4	2	3	2	3	2	67.5
Persona 6	3	1	4	1	4	4	5	1	4	1	80
Persona 7	3	1	5	1	5	1	4	1	5	1	92.5
Persona 8	2	1	5	1	4	1	4	1	3	1	82.5
Persona 9	4	2	5	1	4	1	4	1	5	1	90
Persona 10	3	2	3	3	3	2	4	4	4	3	57.5
Persona 11	3	1	5	2	5	1	5	2	4	3	82.5
Persona 12	3	2	5	1	3	2	4	1	4	1	80
Persona 13	3	1	5	1	5	1	4	1	5	1	92.5
Persona 14	5	1	5	1	5	1	5	1	5	1	100
Persona 15	5	1	5	1	5	1	5	1	4	1	97.5
Persona 16	4	1	5	1	5	1	5	1	5	1	97.5
Persona 17	3	1	5	1	5	2	5	1	5	1	92.5
Persona 18	5	1	5	1	5	2	5	2	5	4	87.5
Persona 19	3	2	3	3	3	3	3	3	3	3	52.5
Persona 20	4	1	5	1	5	1	5	1	5	1	97.5
Persona 21	3	2	4	1	3	3	3	3	3	2	62.5
Persona 22	3	1	5	1	5	1	4	1	5	1	92.5
Persona 23	4	1	5	1	5	1	4	1	4	1	92.5
Persona 24	2	1	5	1	4	1	4	1	5	2	85
Persona 25	4	2	5	1	5	1	5	1	2	1	87.5
Persona 26	4	1	5	1	4	1	5	1	5	1	95

Figura 77 : Resultados de preguntas SUS en pruebas del taller

El promedio obtenido de las preguntas SUS en el taller fue de 83.56.

Se puede apreciar que la diferencia de porcentaje obtenido entre la prueba de pasillo y el taller no fue significativa, con apenas 2.27 puntos de diferencia. Esto ayuda a validar la preocupación mencionada anteriormente, que en las pruebas de pasillo se puede obtener un puntaje más alto debido a que son personas que conocen al desarrollador de la aplicación y esto podría influir en las respuestas de forma positiva. Sin embargo, los puntajes son extremadamente similares en los dos tipos de muestra, lo que permite validar la consistencia de este puntaje.

Por lo tanto, si se promedian los dos puntajes obtenidos en las dos muestras, obtendríamos un promedio aproximado de 84.70 puntos.

Ahora, para comprender los resultados obtenidos, se puede realizar mediante la normalización, la cual consiste en convertir los resultados en percentiles y clasificarlos con letras (de A - F). Esta clasificación fue desarrollada por Jeff Sauro y se puede resumir en la Tabla 20 (Sauro, 2016).

Tabla 20 : Grados para puntaje SUS

SUS	Grado	Descripción
90-100	A	Lo mejor imaginable
80-89	B	Excelente
70-79	C	Bueno
60-69	D	Ok/justo
Menos de 60	F	Malo

Por lo tanto, los resultados obtenidos promediando los puntajes se clasifican en la categoría B, la cual se caracteriza como un software excelente. Estos resultados son alentadores, ya que inicialmente se esperaba que la aplicación para visitantes estuviera en el grado C. Sin embargo, aún faltan muchos detalles por mejorar, los cuales, gracias a las pruebas de pasillo y del taller, se han identificado y están siendo resueltos. Se espera continuar mejorando estos detalles para brindar una experiencia enriquecedora a los usuarios que abren la aplicación por primera vez.

6. CONCLUSIONES Y TRABAJO FUTURO

El trabajo de título permitió al estudiante aplicar y poner en práctica conocimientos no solo del área de informática, sino también de ciencias botánicas. Pudo reforzar sus conocimientos en arquitectura de software y construcción de software en varios módulos, y además tuvo la oportunidad de relacionarse con otras áreas, como fue el caso de la Facultad de Ciencias de la Universidad Austral. Allí, pudo practicar habilidades blandas trabajando en conjunto para consolidar las ideas de GeoParques.

Sin duda, fue una experiencia enriquecedora; se pudo trabajar en grupo definiendo trabajos y, posteriormente, trabajar de manera autónoma. Además, se ampliaron los conocimiento sobre especies botánicas, tanto nativas como no nativas, las cuales anteriormente no conocía ni podía reconocer, y ahora puede identificarlas con solo verlas.

6.1 Estado Actual

Actualmente, las dos aplicaciones, GeoParques Visitante y Administrador, están en producción en fase beta y disponibles para su uso. De estas, GeoParques – Visitante fue la que generó mayor impacto, tal como se había previsto. A la fecha del 8 de junio de 2024, la aplicación puede utilizarse en el jardín botánico de la Universidad Austral de Chile, donde los visitantes pueden recorrer el parque, descubrir y buscar especies que les llamen la atención. Además, pueden buscar los códigos QR distribuidos por el parque, algunos de los cuales están muy visibles y otros escondidos por los senderos del jardín. De manera iterativa, ambas aplicaciones están siendo sujetas a correcciones de errores y mejoras para optimizar su apariencia y usabilidad. Esto se debe a que una aplicación nunca está completamente finalizada; siempre hay mejoras que realizar.

6.2 Cumplimiento de los Objetivos

Desde el principio, el objetivo general de este trabajo de título fue “enriquecer la experiencia de los visitantes de parques turísticos mediante una aplicación multiplataforma (PWA)”. Este objetivo se cumplió con éxito tras alcanzar los siguientes objetivos específicos.

6.2.1 Diseñar y Desarrollar una Aplicación Utilizando la Metodología de Desarrollo Aplicación Web Progresiva (PWA)

El desarrollo y diseño de una aplicación PWA, conocida como GeoParques – Visitantes, representó una fase esencial de este proyecto. Gracias a esta aplicación se proporciona información a los visitantes de parques turísticos, quienes son los usuarios finales y dan vida a toda la plataforma de GeoParques. El desarrollo de esta aplicación se logró con éxito, implementando las tecnologías mencionadas en este documento y agregando nuevas funcionalidades como reportar especies, visualizar los eventos más importantes en el parque, y un blog donde se pueden subir noticias del parque. Esto tiene el fin de que la aplicación entregue más valor tanto a los visitantes como a los administradores de parques

turísticos. Cabe destacar que esta aplicación rastrea al usuario para saber qué especies le llaman más la atención, lo cual alimenta a la base de datos que posteriormente puede ser analizada.

La culminación de este objetivo resultó ser una de las partes más fundamentales del proyecto y permitió obtener validación y retroalimentación de los usuarios finales. La correcta ejecución de este objetivo específico dio vida al objetivo general de este trabajo de título.

6.2.2 Integrar un Sistema de Geolocalización que Permita Rastrear la Ubicación en Tiempo Real del Visitante y Proporcionar Información Mediante su Posición.

El desarrollo e implementación de la localización del visitante en el mapa fue todo un desafío, ya que, como se mencionó anteriormente, era necesario mostrar en el mapa de la aplicación la posición del visitante en el parque y, además, integrar una funcionalidad para que, al acercarse a un área específica, se muestren las especies presentes en esa área. Aunque estas dos opciones se implementaron con éxito en la aplicación del visitante, todavía es necesario realizar mejoras profundas para optimizar los algoritmos que hacen posible estas opciones, ya que se han identificado algunos bugs con el sistema de GeoProximidad. Como ejemplo, un usuario reportó el siguiente problema:

“Hubo un momento que cuando activé la opción de GeoProximidad, todos los árboles permanecieron, pero una vez la desactivé desaparecieron, la volví a activar, aparecieron, y una vez la desactivé se quedaron en su lugar.”

Aunque este error no se ha repetido en pruebas subsiguientes, su sola ocurrencia indica la existencia de un bug, lo que motiva la necesidad de investigar más a fondo qué está provocando este comportamiento para poder solucionarlo.

Además, se han reportado problemas cuando se recorre el parque con la localización activada y se transita por senderos donde la visibilidad del cielo es obstaculizada por la maleza y ramas de los árboles, y en estos mismos lugares, la señal de internet es débil. La opción de proximidad comienza a tener fallas, no entregando la posición precisa del visitante y, en ocasiones, mostrando un error y desconectando esta opción. Este problema era de esperarse debido a la gran cantidad de árboles y plantas que obstruyen la buena recepción de señal de ambos sensores. En invierno, estos problemas se reducen debido a la falta de hojas en las ramas de los árboles, sin embargo, en períodos donde los árboles tienen hojas, esta opción tiende a fallar más.

A pesar de estos problemas, que a veces ocurren, la implementación y funcionalidad de este módulo funciona de forma estable, proporcionando a los usuarios la información de su posición y las especies cercanas.

6.2.3 Establecer un Sistema de Código QR: Diseñar y Desplegar Códigos QR en Ubicados Estratégicas Dentro del Parque, Vinculadas a las Fichas Especies Determinadas por los Administradores.

La implementación y desarrollo de este módulo fue un éxito, ya que se cumplió todo lo planificado. Se logró desde la generación de códigos QR por la aplicación del administrador hasta la capacidad de los visitantes de escanearlos mediante la aplicación de GeoParques o por otras aplicaciones de terceros. Además, se implementó la fase beta para la colocación de los 12 códigos QR en el jardín botánico, lo cual se realizó con éxito. En la Figura 78 se puede apreciar uno de los códigos QR en el jardín, colocado en un árbol estratégico. Cabe mencionar que algunos QR están posicionados en los árboles a una altura considerable para evitar que sean robados o vandalizados. Por otro lado, la recepción de las personas al utilizar los códigos QR fue muy positiva.



Figura 78 : Código QR desplegado en el jardín botánico

Se espera que con el paso del tiempo se pueda analizar si los códigos permanecen en la posición en la que fueron colocados y si están en buen estado. Además, se planea desplegar más códigos QR a lo largo del parque en especies estratégicas, con el fin de divulgar más información sobre las especies y promover la aplicación de GeoParques. En conclusión, no surgieron mayores inconvenientes al desarrollar este objetivo específico.

6.2.4 Llevar el Software a un Ambiente de Producción con el Propósito de Ponerlo a Disposición de los Usuarios Finales.

Este objetivo se llevó a cabo de forma exitosa y sin inconvenientes, cumpliendo con las fechas acordadas. Cabe mencionar que cada semana se está revisando el ambiente en producción para verificar si es necesario realizar alguna actualización al sistema operativo que aloja las máquinas virtuales. Además, estas máquinas están monitorizadas, lo que permite que, si alguno de los componentes de GeoParques falla, se envíe un correo al estudiante para notificar que la aplicación está caída. En conclusión, este objetivo se cumplió de manera exitosa, gracias a la experiencia que el estudiante ha adquirido al llevar a cabo esta tarea.

6.2.5 Llevar a Cabo Prueba de Validación de Funcionalidades de la Aplicación (PWA), Asegurando que se Cumpla con los Requisitos y Expectativas Planificadas

La validación de usabilidad de las funcionalidades de la aplicación GeoParques fue una parte crucial en el desarrollo de este proyecto, ya que son los usuarios finales quienes utilizan la aplicación y deben sentirse cómodos haciéndolo. Gracias a la retroalimentación obtenida de los usuarios, se pudieron identificar puntos de mejora en las distintas funcionalidades que ofrece GeoParques.

La participación de los usuarios en el formulario de usabilidad resultó muy positiva y alentadora para el estudiante, quien no esperaba que los resultados fuesen tan buenos como los obtenidos. Además, llamó su atención la disposición de los encuestados para ayudar y contribuir al proyecto. Los resultados de este formulario sirvieron no solo para mejorar la aplicación, sino también para observar el comportamiento de las personas al utilizar aplicaciones. Esto permitirá tomar ciertas precauciones en el futuro al momento de agregar nuevas funcionalidades a la plataforma.

6.3 Trabajo Futuro

Para el futuro de GeoParques, se espera corregir los bugs que surjan en la aplicación del visitante y continuar mejorando la aplicación del administrador, dado que aún necesita mejoras para funcionar correctamente, ya que esta aplicación no tuvo toda la atención necesaria. Es importante mencionar que estar a cargo del desarrollo de dos aplicaciones y de la GeoApi no es una tarea fácil, razón por la cual algunas funcionalidades en la aplicación del administrador no se pudieron implementar al 100%.

Además, se planea fusionar todas las aplicaciones desarrolladas por el equipo en una sola, ya que cada una aporta valor y presenta la información de manera innovadora y atractiva. Para esto, se debe estudiar y planificar cómo realizar esta fusión, que podría lograrse consolidando las aplicaciones en una sola o implementando lo que hoy en día se conoce como frontend por microservicio.

Para futuras versiones de GeoParques, se planea implementar varias características que no se incluyeron en la primera versión. Una de estas es la integración de notificaciones en la aplicación del visitante, destinadas a aquellos que tengan la aplicación anclada en sus dispositivos. Por ejemplo, se enviarán consejos informativos cuando el usuario se encuentre visitando el parque, con mensajes que se activarán según su posición actual en el parque. Otra función planeada permite a los usuarios agregar a favoritos las especies que les interesen, con un panel donde puedan ver sus especies favoritas y su ubicación en el parque. En cuanto a la aplicación del administrador, se espera que los administradores puedan crear más hitos informativos personalizados, como por ejemplo un hito de 'precaución' para advertir a los visitantes de tener cuidado en ciertas áreas, o un hito de 'prohibido pasar'. La idea es que los administradores puedan crear y personalizar estos hitos a su estilo, eligiendo los iconos que ellos prefieran.

Por otro lado, se espera continuar colaborando con el jardín botánico de la Universidad Austral de Chile, con el fin de seguir probando la aplicación y obteniendo retroalimentación de los usuarios que la utilicen, además de atraer a nuevos parques interesados en probar la plataforma.

Se espera obtener financiamiento de alguna entidad para seguir realizando mejoras a la aplicación, en particular mejoras visuales, contratando a un diseñador con experiencia en UX/UI para ofrecer la mejor experiencia visual y de usabilidad posible. Además, se planea desplegar el sistema de GeoParques en un entorno cloud donde el sistema pueda escalar según la demanda de los usuarios. Junto a esto, se espera que esta plataforma se pueda ofrecer como SaaS³³ (*Software as a Service*), donde los parques paguen una suscripción para utilizar las distintas aplicaciones que ofrece GeoParques. Para esto, se espera que la aplicación adapte su arquitectura para poder desplegarse como una arquitectura multi-

³³ Es un modelo de distribución de software donde las aplicaciones se alojan en la nube y se acceden a través de internet, eliminando la necesidad de instalación y mantenimiento local (microsoft.com, s.f.)

tenant³⁴, permitiendo que el sistema funcione en una sola infraestructura en los diversos parques que deseen contratar la plataforma de GeoParques.

Por último, se ofrecen algunas recomendaciones con el objetivo de proporcionar la máxima experiencia a los visitantes que utilicen la aplicación.

Recomendaciones

Como se ha mencionado, en ciertos lugares del parque la localización no funciona correctamente debido a la maleza y la densidad de los árboles. Se recomienda que en estas áreas se complemente esta función con la colocación de más códigos QR. De esta manera, si la localización falla, el visitante puede escanear los códigos QR para asegurarse de no quedarse sin acceso a la información.

Además, se recomienda que al fusionar GeoParques con los otros modulos que están en construcción, se aproveche el módulo de Geosensores en los lugares donde la localización no funcione correctamente. Instalando estos sensores, se activarán automáticamente para mostrar la información relevante en la aplicación. Esto asegurará que el visitante nunca se quede sin acceso a la información mientras usa la aplicación.

³⁴ Es una arquitectura de software donde una única instancia de una aplicación sirve a múltiples clientes, cada uno con datos aislados pero compartiendo la misma infraestructura, lo que reduce costos y facilita la escalabilidad (ibm.com, s.f.).

7. REFERENCIAS

- ABAMobile. (s. f.-a). *Diferencias entre sistema operativo Android e iOS*. Accedido el 12 de abril, 2024, desde <https://abamobile.com/web/diferencias-sistemas-operativos-android-ios/>
- ABAMobile. (s. f.-b). *Tecnología móvil para empresas en la era postcovid*. Accedido el 12 de abril, 2024, desde <https://abamobile.com/web/tecnologia-movil-para-empresas-postcovid/>
- Acuña, G. (s. f.). *Programa Jardín Botánico UMCE organiza visita guiada por códigos QR*. Accedido el 7 de abril, 2024, desde <https://www.umce.cl/index.php/facultades/instituciones/entomologia/item/1040-jardin-botanico-qr>
- Apache.org. (s. f.). Apache JMeter - User's Manual: Building a Simple Database Test Plan. *jmeter.apache.org*. Accedido el 2 de junio, 2024, desde <https://jmeter.apache.org/usermanual/build-db-test-plan.html>
- Arbotag. (s. f.). *Quienes somos - Arbotag*. Accedido el 12 de abril, 2024, desde <https://web.arbotag.cl/quienes-somos/>
- Arias, J. (s. f.). *Likert. Escala de calificación en encuestas a usuarios - UXABLES / Blog*. Accedido el 13 de junio, 2024, desde <https://www.uxables.com/investigacion-ux/likert-escala-de-calificacion-en-encuestas-a-usuarios/>
- Arimetrics. (2022). *Qué es Geolocalización - Definición, significado y ejemplos*. Accedido el 11 de abril, 2024, desde <https://www.arimetrics.com/glosario-digital/geolocalizacion>
- Baseline. (2024). *Baseline Cybersecurity Solutions*. <https://www.base-line.cl/#servicios>
- C4mode. (s. f.). *The C4 model for visualising software architecture*. Accedido el 16 de mayo, 2024, desde <https://c4model.com/>
- Cadena, C., Sánchez, S., & Johanna, V. (2021). *Colección Viva del Jardín Botánico de Bogotá, Colombia*.
- Campos, R. (2023). *Tipos de Pruebas: Pruebas de rendimiento / by Rodrigo Antonio Campos Tapia / Medium*. <https://medium.com/@rcampos.tapia/tipos-de-pruebas-pruebas-de-rendimiento-dc5d38dd1f32>
- Comunicaciones UACH. (s. f.). *Proyecto de Cultura busca recuperar y proyectar el legado del Jardín Botánico de la UACH – Noticias UACH*. 2020. Accedido el 10 de abril, 2024, desde <https://diario.uach.cl/proyecto-de-cultura-busca-recuperar-y-proyectar-el-legado-del-jardin-botanico-de-la-uach/>
- DataScientest. (2022). *MongoDB: todo sobre la base de datos NoSQL orientada a documentos*. Accedido 10 de mayo, 2024, desde <https://datacientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos>
- Deloitte. (s. f.). *¿Qué es Scrum? / Deloitte España / DXD*. Accedido de 21 de abril, 2024, desde <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-scrum.html>
- Díaz, B. (s. f.). *El Museo del Louvre mejora su estrategia digital y la experiencia de usuario con la ayuda de Accenture Interactive*. Accedido 12 de abril, 2024, desde <https://newsroom.accenture.es/es/news/2019/company-news-release-museo-louvre-mejora-estrategia-digital>

- Docker. (s. f.). *What is a Container? / Docker*. Accedido el 21 de junio, 2024, desde <https://www.docker.com/resources/what-container/>
- Donetonic. (s. f.). *Qué es un tablero Kanban - Metodología ágil - DoneTonic*. Accedido el 22 de abril, 2024, desde <https://donetonic.com/es/que-es-un-tablero-kanban/>
- EDteam. (2021). *¿Cuáles son las diferencias entre página web, sitio web y aplicación web? / EDteam*. Accedido el 11 de abril, 2024, desde <https://ed.team/blog/cuales-son-las-diferencias-entre-pagina-web-sitio-web-y-aplicacion-web>
- Eluniversal. (s. f.). *En promedio una persona mira su celular 150 veces en el día / EL UNIVERSAL - Cartagena*. Accedido el 7 de abril, 2024, desde <https://www.eluniversal.com.co/tecnologia/en-promedio-una-persona-mira-su-celular-150-veces-en-el-dia-194126-FCEU294100>
- Euemuseografia. (2021). *Experiencia Emocional en las Visitas a los Museos - EVE Museos + Innovación*. Accedido el 13 de abril, 2024, desde <https://evemuseografia.com/2021/04/14/experiencia-emocional-en-las-visitas-a-los-museos/>
- Fernández, A., & González, R. (2019). ANÁLISIS DEL USO DE LA TECNOLOGÍA EN LOS MUSEOS: LOS MUSEOS INTELIGENTES. ESTUDIO DE CASOS EN LA CIUDAD DE MADRID. *REVISTA INTERNACIONAL DE TURISMO, EMPRESA Y TERRITORIO*, 3(1). <https://doi.org/10.21071/riturem.v3i1.11190>
- Hamilton, T. (s. f.). *¿Qué son las pruebas de picos en las pruebas de software? Aprende con el ejemplo*. Accedido el 21 de junio, 2024, desde <https://www.guru99.com/es/spike-testing.html>
- Hamilton, T. (2024). *¿Qué es la prueba de volumen? Aprende con ejemplos*. Accedido el 22 de junio, 2024 desde <https://www.guru99.com/es/volume-testing.html>
- Honour, R. (2016). *Jardín Botánico UTalca estrena aplicación de realidad aumentada que enseña biodiversidad - CodeXVerde*. Accedido el 08 de abril, 2024, desde <https://codexverde.cl/22027-2/>
- Hostinger. (2023, junio 29). *Qué es React: definición, características y funcionamiento*. Accedido el 17 de abril, 2024, desde <https://www.hostinger.es/tutoriales/que-es-react>
- Hubspot. (2023a). *Qué es JavaScript, para qué sirve y cómo funciona*. Accedido el 17 de abril, 2024, desde <https://blog.hubspot.es/website/que-es-javascript#que-es>
- Hubspot. (2023b). *Qué es un diseño responsive: características y ejemplos*. Accedido el 24 de mayo, 2024, desde <https://blog.hubspot.es/website/diseno-responsive#que-es>
- ibm.com. (s. f.). *What Is Multi-Tenant? / IBM*. Accedido el 8 de junio, 2024, desde <https://www.ibm.com/topics/multi-tenant>
- Ionos. (s. f.). *¿Qué es un código QR? Guía definitiva sobre su uso y funcionamiento - IONOS*. Accedido el 8 de abril, 2024, desde <https://www.ionos.es/digitalguide/online-marketing/vender-en-internet/que-es-un-codigo-qr/>
- Ken, A. (2023). *Arquitectura de software: ¿Qué es y qué tipos hay?*. Accedido el 10 de abril 2024, desde <https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay>
- Kinsta. (2023a). *Qué es Node.js y por qué deberías usarlo*. Accedido el 10 de mayo, 2024, desde <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>

- Kinsta. (2023b, diciembre 19). *Crea y Despliega una Aplicación Node.js en 5 Minutos con Express* - Kinsta®. Accedido el 17 de abril, 2024, desde <https://kinsta.com/es/blog/crea-aplicacion-express/>
- Knowmadmood. (2021). *Qué Tailwind CSS y por qué usarlo - Las grandes Ventajas*. Accedido el 17 de abril, 2024, desde <https://www.knowmadmood.com/es/blog/que-es-tailwind>
- Ladino, J. (2022). *Implementación de una aplicación PWA orientada a la gestión de ruta del sendero agroecoturístico Zoratama en el municipio de Pasca*. www.ucundinamarca.edu.co
- Lisbeth, K. (2020). “IMPLEMENTACIÓN DE UN SISTEMA DE GEOLOCALIZACIÓN Y APLICACIÓN MÓVIL PARA LAS RUTAS TURÍSTICAS DEL CANTÓN LA MANÁ”. <http://repositorio.utc.edu.ec/handle/27000/6892>
- microsoft.com. (s. f.). *What is SaaS? Software as a Service / Microsoft Azure*. Accedido el 8 de junio, 2024, desde <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-saas/>
- Mongodb. (s. f.-a). *Geometry - MongoDB Manual v7.0*. Accedido el 21 de abril de 2024, de <https://www.mongodb.com/docs/manual/reference/operator/query/geometry/>
- Mongodb. (s. f.-b). *MongoDB Limits and Thresholds - MongoDB Manual v7.0*. Accedido el 9 de junio, 2024, desde <https://www.mongodb.com/docs/manual/reference/limits/#bson-documents>
- Mozilla. (s. f.-a). *Clickjacking - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web / MDN*. Accedido el 24 de abril, 2024, desde <https://developer.mozilla.org/es/docs/Glossary/Clickjacking>
- Mozilla. (s. f.-b). *MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web / MDN*. Accedido el 10 de abril, 2024, desde <https://developer.mozilla.org/es/docs/Glossary/MVC>
- Nielsen, J. (2000). *Why You Only Need to Test with 5 Users*. Accedido el 21 de junio, 2024, desde <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- NPMjs. (s. f.). *@yudiel/react-qr-scanner - npm*. Accedido de 24 de mayo, 2024, desde <https://www.npmjs.com/package/@yudiel/react-qr-scanner>
- npmjs.com. (2024). *react-i18next - npm*. <https://www.npmjs.com/package/react-i18next>
- Patrimoniocultural.gob. (s. f.). *Directores de museos conversan sobre nuevas tecnologías en el patrimonio cultural / Servicio Nacional del Patrimonio Cultural*. Accedido el 7 de abril, 2024, desde <https://www.patrimoniocultural.gob.cl/noticias/directores-de-museos-conversan-sobre-nuevas-tecnologias-en-el-patrimonio-cultural>
- Puromarketing. (s. f.). *El 74% de los consumidores ya prefiere realizar sus compras a través del móvil*. Accedido 12 de abril, 2024, desde <https://www.puromarketing.com/76/213363/consumidores-prefiere-realizar-compras-traves-movil>
- Quiroz, A. (2021). *¿Qué es una Aplicación Web Progresiva (PWA)? - B2Chat*. Accedido el 13 de abril, 2024, desde <https://www.b2chat.io/blog/aplicaciones/aplicacion-web-progresiva-pwa/>
- Sauro, J. (2016). *Measuring Usability With The System Usability Scale (SUS)*. Accedido el 3 de junio, 2024 desde <https://www.userfocus.co.uk/articles/measuring-usability-with-the-SUS.html>

- Scrum. (2022). *Tus primeros pasos en User Story Mapping* / Scrum.org. Accedido el 22 de abril, 2024, desde <https://www.scrum.org/resources/blog/tus-primeros-pasos-en-user-story-mapping>
- Sernatur. (s. f.). *Valdivia–Corral se posiciona como el destino con mayor ocupación en el país con un histórico 91,1%* - Sernatur. Accedido 23 de mayo, 2024, desde <https://www.sernatur.cl/valdivia-corral-se-posiciona-como-el-destino-con-mayor-ocupacion-en-el-pais-con-un-historico-911/>
- sqs. (s. f.). *Pruebas de carga y rendimiento para garantizar la estabilidad del sistema - SQS*. Accedido el 2 de junio, 2024, desde <https://www.sqs.es/pruebas-de-carga-y-rendimiento-para-garantizar-la-estabilidad-del-sistema/>
- zaproxy.org. (s. f.). *ZAP*. Accedido el 13 de junio, 2024, desde <https://www.zaproxy.org/>

ANEXOS

Anexo A: Base de datos que se realizó en conjunto con los integrantes de los distintos subproyectos

En el Figura 79, se puede apreciar la primera base de datos construida durante la fase de planificación por todos los integrantes del equipo.

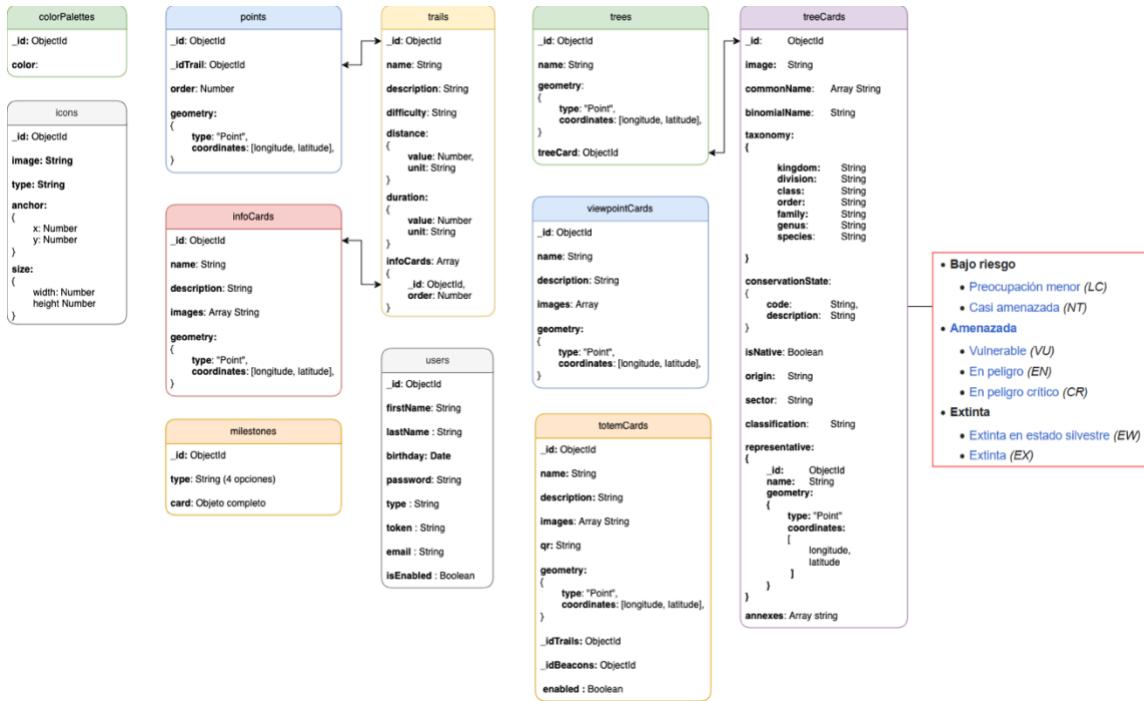


Figura 79 : Base de datos construida por todo el equipo

Anexo B: Descripción de la colección treeCards (Ficha especie)

En la Tabla 21, se puede apreciar una descripción de los atributos que tiene la colección de treeCards, proporcionando un mayor entendimiento de los parámetros que se guardan para cada especie.

Tabla 21 : Descripción de los atributos de colección treeCards

Atributo	Descripción
_id	Identificador Único.
image	Link de imagen de la especie correspondiente.
commonName	Este atributo es un arreglo con los nombres comunes de la especie.
binomialName	Nombre científico de la especie.
Taxonomy	<p>Corresponde a un objeto con atributos de clasificación de las distintas especies.</p> <ul style="list-style-type: none"> • Kingdom: es una característica que asocia a los seres vivos por su naturaleza en común. • Division: se utiliza para subdividir los reinos plantae y fungi • Class: grupa a los seres vivos con semejanzas entre sí. • Order: es una agrupación de individuos e una clase que tienen características entre sí. • Family: agrupación de seres vivos que se encuentran en un orden, por características comunes entre ellos. • Genus: De las familia proviene los géneros, conjunto de especies relacionadas entre sí por la evolución • Species: grupo de individuos con las mismas características, que permiten relacionarse entre sí y tener descendencia.
conservationState	Este objeto contiene información del estado de conservación de la especie, su código y una pequeña descripción.

isNative	Este atributo es un booleano, si el árbol es nativo o no.
origin	Cuál es el origen de la especie correspondiente. Ejemplo, España, Chile.
sector	En qué sector del hemisferio se encuentra: Norte , sur.
classification	Si es un árbol existe una clasificación para identificarlo. Si la especie no es un árbol este atributo no existe en el documento.
representative	Es un objeto que tiene las geoposición de un árbol o especie de ejemplo para la ficha especie.
annexes	Es un lista que cuanta con distintos URL de investigaciones que se han realizado a la especie.

removed	Booleano para el eliminado lógico de la especie. Esto significa que no se elimina de la base de datos pero no se refleja en el sistema.
---------	---

Anexo C: Descripción de la colección ingreso Persona

En la Tabla 22, se detalla la descripción de los atributos que contiene la colección de la base de datos Torobayo, donde se rastrea el ingreso del visitante a la aplicación de GeoParques.

Tabla 22 : Descripción de los atributos de la colección ingresoPersona

Atributo	Descripción
_id	Identificador Único.
idUsuario	ID del usuario que entró a la aplicación. Si no aceptó las cookies, este campo quedará como anónimo.
posiciónVisitante	Trackea la posición del visitante, esto es, si aceptó compartir su ubicación.
navegador	Es un objeto que tiene dos campos, los cuales son los siguientes: <ul style="list-style-type: none"> • nombre: es el nombre del navegador en donde se está ejecutando la aplicación. • versión: es la versión del navegador.
os	Este es un objeto que cuenta con dos campos, los cuales son los siguientes: <ul style="list-style-type: none"> • nombre: nombre del sistema operativo en donde se está ejecutando la aplicación. • versión: versión del sistema operativo.
dispositivo	Este es un objeto que cuenta con 3 campos, los cuales son los siguientes:

	<ul style="list-style-type: none"> • tipoDispositivo: identifica si es un dispositivo móvil, una tableta o un computador de escritorio. • fabricante: la marca del dispositivo en el cual se ejecutó la aplicación. • modelo: el modelo del dispositivo.
fechaHora	<p>Es un objeto que cuenta con 2 campos:</p> <ul style="list-style-type: none"> • hora: la hora en que ingresó a la aplicación. • fecha: la fecha en que ingresó a la aplicación.

Anexo D: Primera Carta Gantt

En la Figura 80 se puede apreciar la primera carta Gantt donde se planificó el trabajo y desarrollo de este proyecto de título.

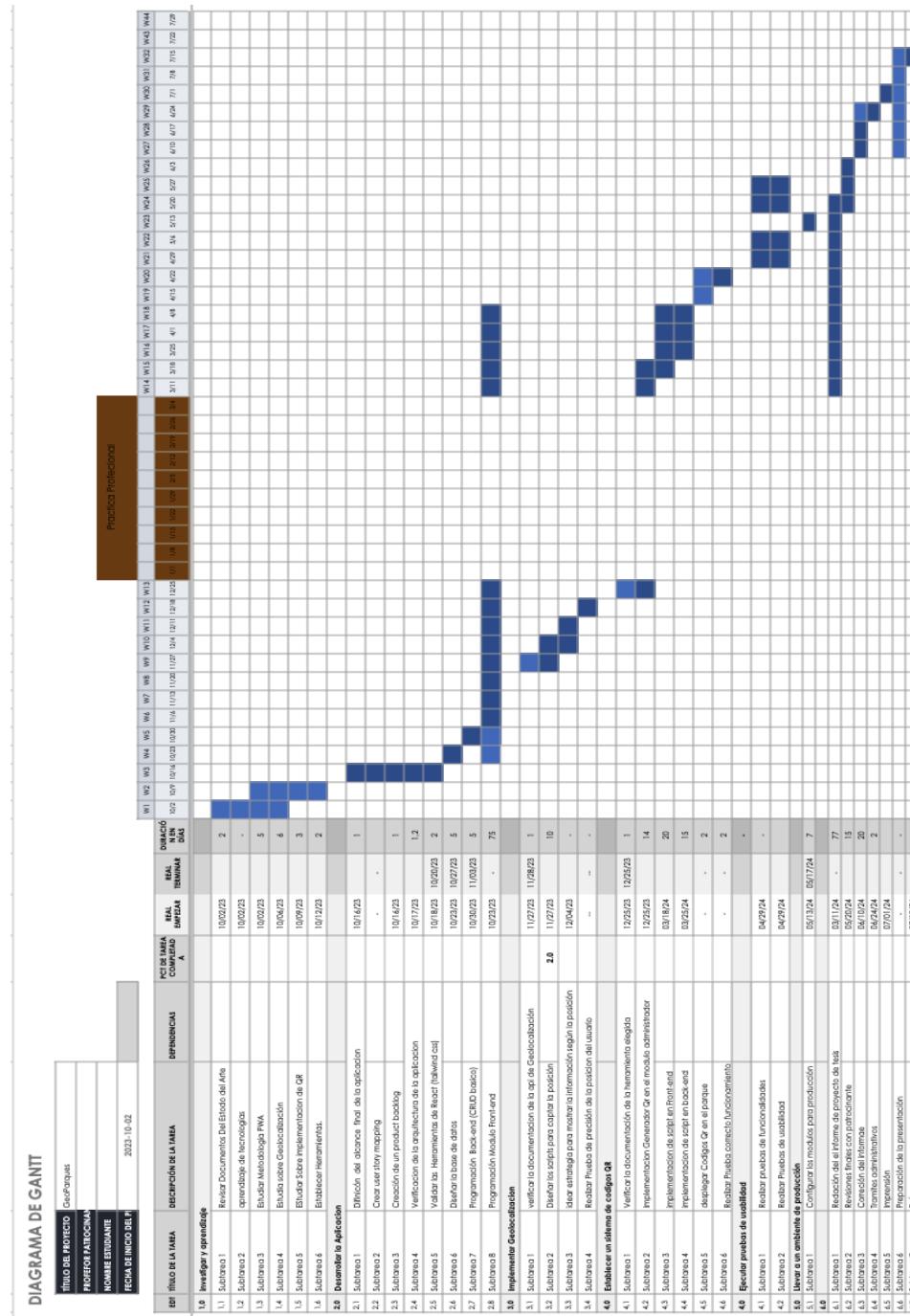


Figura 80 : Primera carta Gantt

Anexo E: *User history mapping* de la aplicación administrador

En la Figura 81 se pueden observar las historias de usuarios organizadas para el administrador.

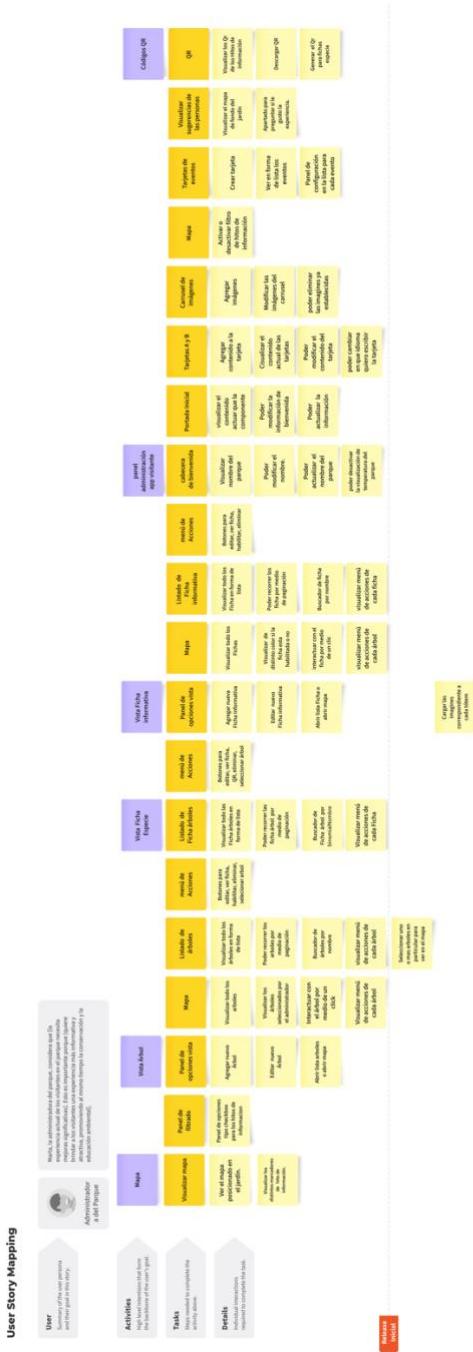


Figura 81 : User history mapping de la aplicación administrador

Anexo F: Tablero Kanban

En la

Figura 82 se puede apreciar el tablero donde se organizaban las tareas durante el proceso de desarrollo.

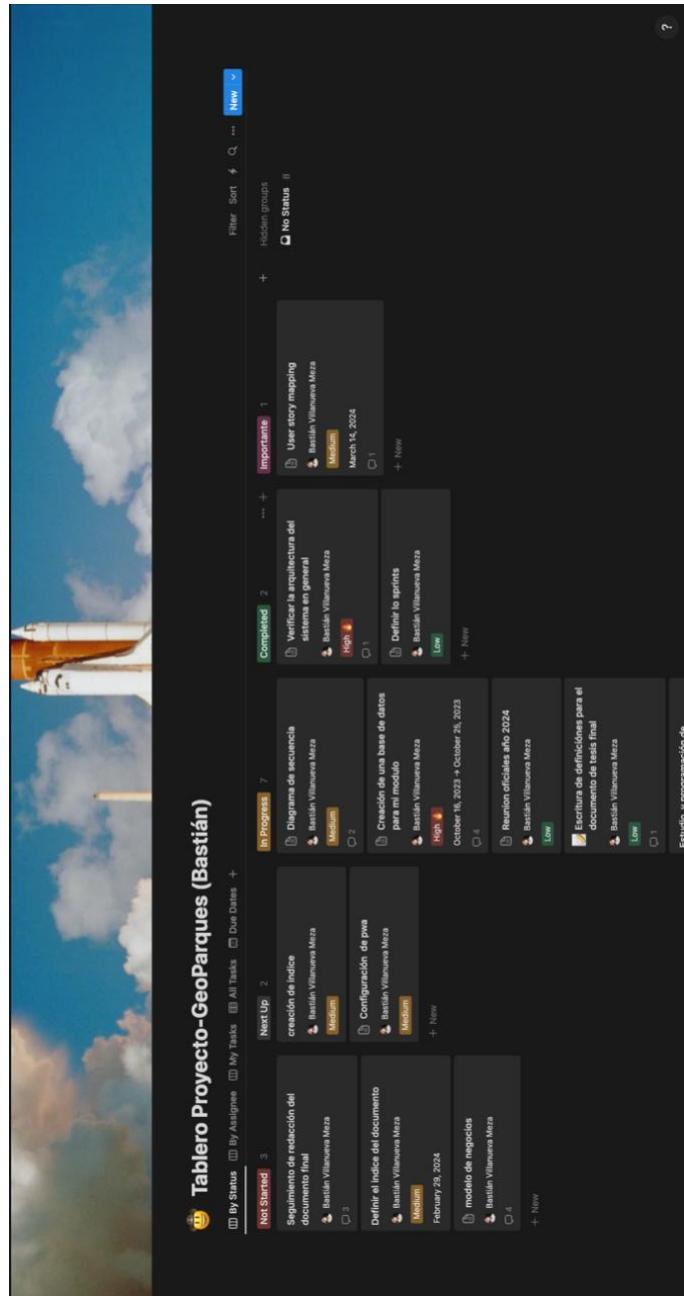


Figura 82 : Tablero Kanban

Anexo G: Bitácora

- https://drive.google.com/file/d/1-QgpJIXiZUv7M_VknWK8nWYvtkvSBxnH/view

Anexo H: Rutas (endpoint) de información de la GeoApi

En la Tabla 23 se pueden apreciar los distintos endpoints que cuenta la GeoApi, con el propósito de proporcionar información tanto a la aplicación del visitante como a la del administrador.

Tabla 23 : Rutas de información de la GeoApi

	Tipo	Ruta	Descripción
Árboles	POST	/api/arboles	Permite crear un árbol.
	GET	/api/arboles	Obtiene todos los árboles.
	GET	/api/arboles-habilitado	Obtiene los árboles que están habilitados en sistema.
	GET	/api/arboles-filter-ficha/:id	Obtiene los árboles según una ficha árbol asociada (obtiene los árboles de una especie correspondiente).
	GET	/api/arboles/:id	Obtiene un árbol según su id.
	PUT	/api/arboles/:id	Actualiza la información de un árbol según su id.
	PUT	/api/arboles-deac/:id	Desactiva un árbol según su id.
	PUT	/api/arboles-acti/:id	Activa un árbol según su id.
	DELETE	/api/arboles/:id	Elimina el árbol del sistema según su id.
Fichas	POST	/api/fichas	Permite crear una nueva ficha especie.
	GET	/api/fichas	Obtiene todas las fichas especie.
	GET	/api/fichas/:id	Obtiene una ficha según.
	PUT	/api/fichas/:id	Actualiza la información de una ficha especie según su id.
	DELETE	/api/fichas/:id	Elimina la ficha especie.
Fichas informativas	POST	/api/ficha-informativa	Permite crear una ficha informativa.
	GET	/api/ficha-informativa	Obtiene todas las fichas informativas.
	GET	/api/ficha-informativa-habilitado	Obtiene las fichas informativas habilitadas en el sistema.

a - I n f o r m a t i v a	GET	/api/ficha-informativa/:id	Obtiene una ficha informativa según su id.
	PUT	/api/ficha-informativa-deac/:id	Desactiva una ficha informativa según su id.
	PUT	/api/ficha-informativa-acti/:id	Activa una ficha informativa según su id.
	DELETE	/api/ficha-informativa/:id	Elimina una ficha informativa según su id.
T ó t e m	POST	/api/totems	Permite crear un tótem.
	GET	/api/totems	Obtiene todos los tótems.
	GET	/api/totems-habilitados	Obtiene todos tótems que estén habilitados en el sistema.
	GET	/api/totems/:id	Obtiene un tótem según su id.
	PUT	/api/totems/:id	Actualiza la información de un tótem mediante su id.
	PUT	/api/totems-deact/:id	Desactiva en el sistema el tótem mediante su id.
	PUT	/api/totems-acti/:id	Activa en el sistema el tótem mediante su id.
	DELETE	/api/totems/:id	Elimina en el sistema el tótem mediante su id.
Mir ado r	POST	/api/viewpointcard	Permite crear un mirador.
	GET	/api/viewpointcard	Obtiene todos los miradores.
	GET	/api/viewpointcard-habilitados	Obtiene todos los miradores que están habilitados en el sistema.
	GET	/api/viewpointcard/:id	Obtiene un mirador por su id.
	PUT	/api/viewpointcard-deact/:id	Desactiva en el sistema un mirador mediante su id.
	PUT	/api/viewpointcard-acti/:id	Activa en el sistema un mirador mediante su id.
	PUT	/api/viewpointcard/:id	Actualiza la información de un mirador mediante su id.
	DELETE	/api/viewpointcard/:id	Elimina en el sistema el mirador por su id.
Sen der o	POST	/api/senderos	Permite Crear un sendero.
	GET	/api/senderos	Obtiene todos los senderos.
	GET	/api/senderos/:id	Obtiene un sendero mediante su id.
	PUT	/api/senderos/:id	Actualiza la información de un sendero mediante su id.
	DELETE	/api/senderos/:id	Elimina en el sistema el sendero mediante su id.
Pun	POST	/api/puntos	Crear un punto.

tos	GET	/api/puntos	Obtener todos los puntos.
	GET	/api/puntos/:id	Obtiene un punto mediante su id.
	PUT	/api/puntos/:id	Modifica la información de un punto según su id.
	DELETE	/api/puntos/:id	Elimina en el sistema un punto mediante su id.
Sist em a	POST	/api/single-img	Guardar una imagen en la carpeta compartida.
	POST	/api/multiple-img	Guardar hasta 10 imágenes en la carpeta compartida.
	GET	/api/image/:imageName	Obtener una imagen mediante su nombre.
	GET	/api/images/:imagesNames	Obtener imágenes mediante sus nombres.

Anexo I: Rutas de la capa controladora de la aplicación del administrador

En la Tabla 24 se pueden apreciar los distintos endpoints de información que cuenta la capa controladora de la aplicación del administrador.

Tabla 24 : Rutas de información de la capa controladora administrador

	Tipo	Ruta	Descripción
U s u a r i o	POST	api/usuarios/login	Permite iniciar sesión en el sistema al administrador de parque.
	POST	api/usuarios/registrarAdmin	Permite crear un usuario administrador en la aplicación de geoparques-administrador (solo un administrador puede crear otra cuenta para otro administrador)
	GET	api/usuarios/confirmar_/:token	Permite confirmar el token de seguridad de una cuenta administradora.
	POST	api/usuarios/activar_cuenta/:token	Permite activar una cuenta al momento de registrar un nuevo administrador.
	POST	api/usuarios/olvide-password	Permite iniciar el proceso para recuperar contraseña.
	POST	api/usuarios/olvide-password/:token	Permite realizar el cambio de contraseña
	GET	api/usuarios/olvide-password/:token	Permite por medio del token enviado al iniciar proceso para recuperar contraseña realizar el cambio.

	POST	api/usuarios/deshabilitar-usuario	Permite al administrador del parque desactivar la cuenta de otro administrador.
	POST	api/usuarios/habilitar-usuario	Permite al administrador del parque activar un usuario deshabilitado.
	POST	api/usuarios/eliminar-usuario	Permite al administrador eliminar la cuenta de otro administrador.
	GET	api/usuarios/perfil	Obtiene los datos de un administrador.
	GET	api/usuarios/obtener-usuario	Permite obtener la información de todos los administradores

Anexo J: Manual de uso aplicación GeoParques Visitantes

- https://drive.google.com/file/d/1dg6d7o4DFi3OCmolx3F_IZEYYkW-9rbK/view

Anexo K: JSON real de la configuración del manifiesto

En la Figura 83 se puede apreciar la configuración del manifiesto que permite transformar una aplicación en una PWA.

```
1  {
2     "name": "Geoparques", // Nombre completo de la aplicación web.
3     "short_name": "Geopark", // Nombre corto de la aplicación web. Se utiliza cuando no hay suficiente espacio para mostrar el nombre completo.
4     "start_url": "/", // URL de inicio de la aplicación web. Esta es la página que se abre cuando se lanza la aplicación desde un dispositivo.
5     "display_override": ["fullscreen", "minimal-ui"], // Modos de visualización preferidos. Si el navegador no puede utilizar el primer modo, intentará utilizar el siguiente.
6     "display": "standalone", // Modo de visualización por defecto. "standalone" significa que la aplicación se abrirá en su propia ventana, como una aplicación nativa.
7     "dir": "ltr", // Dirección de texto. "ltr" significa de izquierdo a derecha.
8     "background_color": "#fffffe", // Color de fondo de la aplicación. Se utiliza en la barra de estado del sistema operativo.
9     "icons": [ // Ícones de la aplicación. Se utilizan en la pantalla de inicio, en la barra de tareas, etc.
10        {
11            "src": "android-chrome-192x192.png", // Ruta del ícono.
12            "sizes": "192x192", // Tamaño del ícono.
13            "type": "image/png" // Tipo de medio del ícono.
14        },
15        {
16            "src": "/android-chrome-192x192.png",
17            "sizes": "192x192",
18            "type": "image/png"
19        },
20        {
21            "src": "/android-chrome-512x512.png",
22            "sizes": "512x512",
23            "type": "image/png"
24        },
25    ],
26    "orientation": "portrait-primary", // Orientación preferida de la aplicación. "portrait-primary" significa que la aplicación debería mostrarse en modo retrato.
27    "scope": "/" // El alcance define el conjunto de URL a las que se aplica el manifiesto.
28 ]
```

Figura 83 : Configuración manifiesto

Anexo L: Rutas (endpoint) capa controladora de la aplicación visitante

Tabla 25 : Rutas de información de la capa controladora visitante

	Tipo	Ruta	Descripción
Encuesta inicial al visitante.	GET	/api-geo-info/encuesta-inicial	Obtiene todas las encuestas realizadas. Con el objetivo de analizarlas.
	GET	/api-geo-info/encuesta-inicial/:id	Obtiene una encuesta por su id.
	POST	/api-geo-info/encuesta-inicial	Sirve para crear encuesta al momento de que el usuario rellena todos los campos correspondientes a la encuesta inicial.
Módulo Vista Inicial	GET	/api-geo-info/portada-inicial/:idioma	Obtiene la portada inicial mediante el idioma
	PUT	/api-geo-info/portada-inicial/:id	Modifica la portada inicial según su id.
	GET	/api-geo-info/tarjeta-inicial/:tipo	Obtiene una tarjeta de la vista inicial según su tipo (esta puede ser A o B).
	GET	/api-geo-info/tarjeta-inicial-id/:id	Obtiene una tarjeta mediante su id-
	POST	/api-geo-info/tarjeta-inicial	Crear una tarjeta.
	PUT	/api-geo-info/tarjeta-inicial/:tipo	Actualiza la información de una tarjeta (esta puede ser A o B)
Módulo Evento	GET	/api-geo-info/eventos	Obtiene todos los eventos.
	GET	/api-geo-info/eventos-habilitados	Obtiene todos los eventos habilitados en el sistema.
	GET	/api-geo-info/eventos/:id	Obtiene un evento mediante su id.
	GET	/api-geo-info/eventos-no-eliminados	Obtiene los eventos no eliminados del sistema (el eliminado es lógico por los eventos se mantiene en la base de datos).
	POST	/api-geo-info/eventos	Crea un evento.
	PUT	/api-geo-info/eventos/:id	Actualiza la información de un evento mediante su id.
	PUT	/api-geo-info/eventos-activate/:id	Activa un evento en particular mediante su id.
	PUT	/api-geo-info/eventos-deactivate/:id	Desactiva un evento en particular mediante su id.

	DEL ETE	/api-geo-info/eventos/:id	Eliminado lógico de un evento por su id.
Módulo Post	GET	/api-geo-info/post	Obtiene todos los posts.
	GET	/api-geo-info/post-habilitado	Obtiene todos los posts habilitados en el sistema.
	GET	/api-geo-info/:id	Obtiene un post mediante su id.
	POST	/api-geo-info/post	Crea un post en el sistema.
	PUT	/api-geo-info/post	Modifica un post mediante su id.
	PUT	/api-geo-info/post-activate/:id	Activar en el sistema un post.
	PUT	/api-geo-info/post-deactivate/:id	Desactivar en el Sistema un post.
	DEL ETE	/api-geo-info/post/:id	Eliminado lógico de un post.
Módulo Reporte fallo visitante	GET	/api-geo-info/reporte-fallo-visitante	Obtiene todo los reportes realizado por los visitantes.
	GET	/api-geo-info/reporte-fallo-visitante-no-vistos	Obtiene los reportes que no han sido visto por los administradores de parque.
	GET	/api-geo-info/reporte-fallo-visitante/:id	Obtiene un reporte mediante su id.
	POST	/api-geo-info/reporte-fallo-visitante	Crea un reporte al sistema.
	PUT	/api-geo-info/reporte-fallo-visitante-visto	Modifica un reporte como visto en el sistema.
	PUT	/api-geo-info/reporte-fallo-visitante-modificado	Actualiza la información de un reporte. (este sirve para marcar que el reporte está solucionado)
	DEL ETE	/api-geo-info/reporte-fallo-visitante-eliminado	Eliminado lógico de un reporte en el sistema.
Módulo Sistema	GET	/api-geo-info/sistema-direccion	Obtiene la dirección ip de usuario (este endpoint no se está utilizando)
Módulo Suggerencia	GET	/api-geo-info/sugerencia-no-leidas	Obtiene las sugerencias no leídas por un administrador.
	GET	/api-geo-info/sugerencia/:id	Obtiene una Sugerencia por su id.
	POST	/api-geo-info/sugerencia	Permite crear una sugerencia.
	PUT	/api-geo-info/sugerencia-marca-leida/:id	Permite marca una sugerencia como leída. (uso para el administrador)
Módulo Track	GET	/api-geo-info/track-sistema	Obtiene todo los registros trackeados en el sistemas (escanear un QR, realizar clic

Sistema			en una especie en el mapa, entre otros)
	GET	/api-geo-info/track-sistema/:id	Obtener un registro por medio de su id.
	POST	/api-geo-info/track-sistema	Crear un registro de track en el sistema(se trakea cuando el escanea un QR, cuando realiza un clic en una especie que le llamo la atención)
Módulo Seguimiento	GET	/api-geo-info/seguimiento-ingreso	Obtiene todo los registros de seguimiento de las personas que ingresaron a la aplicación del visitante.
	GET	/api-geo-info/seguimiento-ingreso/:id	Obtiene un registro de seguimiento mediante la id.
	POST	/api-geo-info/seguimiento-ingreso	Crea un registro de seguimiento cuando la persona ingresa a la aplicación
	PUT	/api-geo-info/seguimiento-ingreso/:id	Permite modificar el registro de seguimiento (este endpoint no se utiliza).

Anexo M: JSON con las variables en idioma inglés

En la Figura 84 se puede apreciar el archivo JSON que ofrece las traducciones necesarias para que la aplicación pueda utilizarse en inglés.

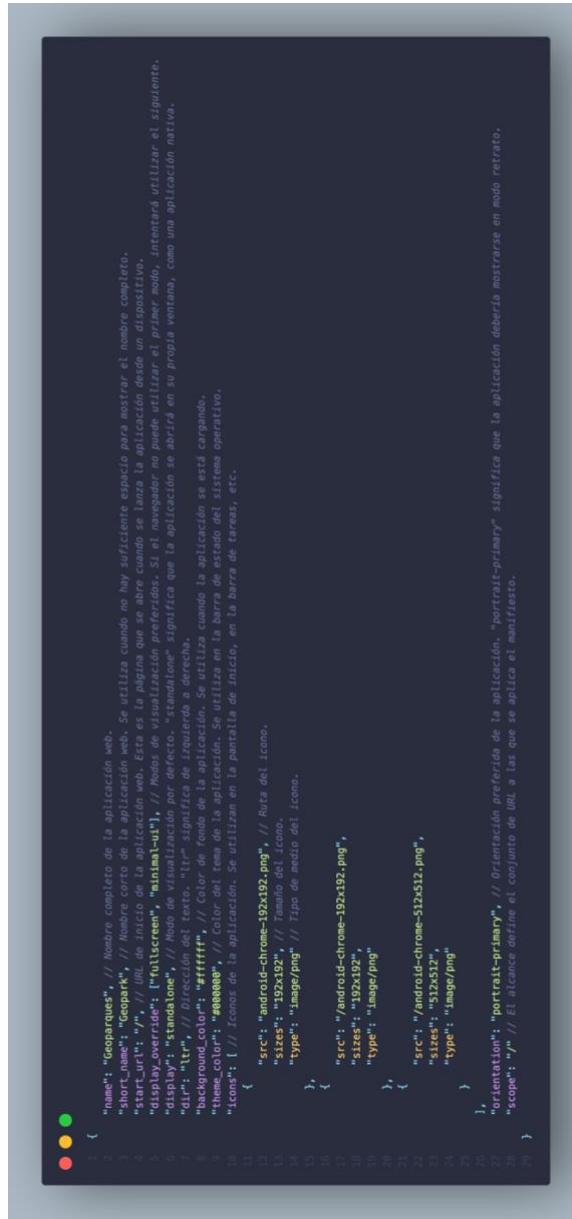
A screenshot of a terminal window on a Mac OS X system, indicated by the red, yellow, and green window control buttons at the top. The terminal displays a JSON configuration file with code numbers on the left. The JSON structure includes sections for suggestions, messages, text inputs, buttons, and configuration settings.

```
1 "sugerencia" : {
2     "mensajesAviso": {
3         "msgWarningSelect": "Please select an option",
4         "msgWarningDescription": "Please enter a description",
5         "msgWarningSend": "Suggestion sent successfully"
6     },
7     "texto": {
8         "textoH2": "Your opinion matters!",
9         "textoP": "Tell us what you thought about the experience at the UACh botanical garden."
10    },
11    "textInput": {
12        "textoCorreo": "Your email",
13        "textoExperiencia": "Did you enjoy the GeoParks experience?",
14        "textoParque": "Did you like the park?",
15        "textoComentarios": "Comments"
16    },
17    "textoBoton": {
18        "botonEnviar": "Send",
19        "botonAgendar": "Do you want to support us with a usability survey of geoparks? Click here."
20    },
21    "check": {
22        "checkSi": "Yes",
23        "checkNo": "No"
24    }
25},
26
27 "textoConfiguracion" : {
28     "titulo": "Settings",
29     "idioma": "Language",
30     "espanol": "Spanish",
31     "ingles": "English",
32     "restaurarDatos": "Restore data",
33     "boton": {
34         "restaurarDatos": "Restore data",
35         "cancelar": "Cancel",
36         "aplicar": "Apply change"
37     }
38},
```

Figura 84 : JSON con las variables en inglés

Anexo N: JSON real de la configuración del manifiesto

En la Figura 85 se puede apreciar la configuración real del archivo manifiesto para convertir la aplicación Geoparques–Visitante en PWA.



```
1  {
2     "name": "Geoparques", // Nombre completo de la aplicación web. Se utiliza cuando no hay suficiente espacio para mostrar el nombre completo.
3     "short_name": "Geoparque", // Nombre corto de la aplicación web. Se utiliza cuando se lanza la aplicación desde un dispositivo.
4     "start_url": "/", // URL de inicio de la aplicación web. Esta es la página que se abre cuando se lanza la aplicación desde un dispositivo.
5     "display_overrides": ["fullscreen", "minimal-ui"], // Modos de visualización preferidos. Si el navegador no puede utilizar el primer modo, intentará utilizar el siguiente.
6     "display": "standalone", // Modo de visualización por defecto.
7     "display": "block", // Dirección del texto.
8     "dir": "ltr", // Dirección de fondo de la aplicación. Se utiliza cuando la aplicación se está cargando.
9     "background_color": "#f0f0f0", // Color de fondo de la aplicación. Se utiliza en la barra de estado del sistema operativo.
10    "theme_color": "#e6e6fa", // Color del tema de la aplicación. Se utilizan en la pantalla de inicio, en la barra de tareas, etc.
11    "icons": [ // Íconos de la aplicación. Se utilizan en la pantalla de inicio, en la barra de tareas, etc.
12        {
13            "src": "android-chrome-192x192.png", // Ruta del ícono.
14            "sizes": "192x192", // Tamaño del ícono.
15            "type": "image/png" // Tipo del medio del ícono.
16        },
17        {
18            "src": "android-chrome-192x192.png",
19            "sizes": "192x192",
20            "type": "image/png"
21        },
22        {
23            "src": "android-chrome-512x512.png",
24            "sizes": "512x512",
25            "type": "image/png"
26        }
27    ],
28    "orientation": "portrait-primary", // Orientación preferida de la aplicación. "portrait-primary" significa que la aplicación deberá mostrarse en modo retrato.
29    "scope": "/" // El alcance define el conjunto de URLs a las que se aplica el manifiesto.
30 }
```

Figura 85 : JSON del Manifiesto

Anexo O: Carta Gantt actualizada

En la Figura 86, se muestra la Carta Gantt, la cual cuenta con la planificación de tiempos del proyecto de título.



Figura 86 : Carta Gantt actualizada

Anexo P: Documentación de despliegue GeoApi

- https://drive.google.com/file/d/1svhrPC-buGmuuE-7rtrHd9d_y3jzFiaS/view

Anexo Q: Documentación de despliegue Aplicación GeoParques - Visitante

- <https://drive.google.com/file/d/1bTgyHrTiNZMjudmPrP8WUc-6SiyhWQWI/view>

Anexo R: Documentación de despliegue Aplicación GeoParques – Administrador

- <https://drive.google.com/file/d/1aUO1AfajVC5bVRwGXCK-3TrR1vI4vVYZ/view>

Anexo S: Tabla de las distintas alertas de seguridad

En la Tabla 26 se muestran los tipos de alertas de seguridad de una vulnerabilidad detectada.

Tabla 26 : Tipos de alertas de vulnerabilidades

Tipo de alerta	Descripción
Baja (Low)	Alertas que indican posibles problemas de seguridad que tienen un impacto mínimo o que son difíciles de explotar.
Media (Medium)	Alertas que señalan problemas de seguridad que podrían ser explotados bajo ciertas condiciones y que tiene un impacto moderado en la seguridad de la aplicación.
Alta (High)	Alertas que representan problemáticas de seguridad significativas y que pueden ser fácilmente explotados, lo que resulta un impacto severo en la seguridad
Critica (Critical)	Alertas que indican problemas de seguridad graves y explotables que pueden condicionar a la compromiso completa de la aplicación o del sistema subyacente.

Anexo T: Informe de ciberseguridad generado por ZAP para GeoParques.

- https://drive.google.com/file/d/1ilbYv4BUcEyNFsWVZe5Gm_h1N3Cu2Zod/view

Anexo U: Formulario para validación de usabilidad

- <https://forms.gle/2MkQUagcAun9tFqC8>