



Sistema de Gestión para el Índice de Integridad Ecosistémica

Proyecto Arrayán

Imanol Luis Heredia Sánchez
imanol.heredia.96@gmail.com

Agustín Daniel Méndez Romero
agu2153@gmail.com

Diego Facundo Pígola Diaz Sanz
diegopigola06@gmail.com

Supervisores: Msc. Ing. Raquel Sosa, Dr. Ing. Antonio Mauttone

Usuarios: Oscar Blumetto, Andrés Castagna

Proyecto de Grado de Ingeniería en Computación
Instituto de Computación, Facultad de Ingeniería
Universidad de la República

RESUMEN

El funcionamiento integral de los ecosistemas es fundamental para una serie de procesos esenciales de la vida humana y silvestre. Con el objetivo de medir la integridad de los ecosistemas surge el Índice de Integridad Ecosistémica (IIE), que apunta a evaluar dicha integridad en función de datos relevados en el campo. Este índice es el resultado de un trabajo de varios años realizado por el Instituto Nacional de Investigación Agropecuaria (INIA).

Actualmente la gestión del índice y la toma de mediciones para evaluar establecimientos se realizan de forma manual. El próximo paso para la evolución del índice, es la gestión de este mediante un sistema informático, el cual debe permitir gestionar el índice y además brindar la posibilidad de ser usado en dispositivos móviles para el relevamiento de datos en campo. Lo anteriormente presentado permite calcular los resultados de distintos índices de forma automática, en contraposición a como se hace hoy día con planillas de cálculo electrónicas.

El objetivo de este proyecto es diseñar un sistema informático que atienda estas necesidades e implementar un prototipo funcional que valide su factibilidad, capaz de reemplazar el modelo de funcionamiento actual. Para llevar a cabo dicho objetivo es necesario realizar actividades como el estudio del dominio, el relevamiento de requisitos y el diseño del sistema.

La solución desarrollada se compone de una aplicación web para la gestión general y la visualización de los resultados, una aplicación móvil para la carga de datos en el campo y un servidor encargado de brindar soporte a estas dos, así como realizar los cálculos del índice.

Estos módulos fueron implementados usando distintas tecnologías. Para la aplicación web se usó ReactJs, para la aplicación móvil se usó Kotlin, para el servidor se usó SpringBoot sobre Java. Los datos son persistidos en una base de datos relacional PostgreSQL. El producto logrado funciona con la versión actual del índice y es configurable en caso de que se deseen aplicar nuevas versiones del mismo.

Palabras Clave: Servicios ecosistémicos, Evaluación ambiental, Adquisición de datos, Cálculo configurable, Sistema de información geográfica empresarial, Arquitectura distribuida.

Índice general

1	Introducción	1
1.1	Contexto y Motivación	1
1.2	Objetivos	2
1.3	Aportes del proyecto	2
1.4	Organización del documento	3
2	Fundamentos teóricos	4
2.1	Índice de Integridad Ecosistémica	4
2.1.1	Términos importantes	4
2.1.2	Índice de Integridad Ecosistémica	4
2.2	Tecnologías de la información	8
2.2.1	Definiciones Generales	9
2.3	Trabajos relacionados	15
2.3.1	Aplicaciones Web	16
2.3.2	Aplicaciones Móviles	17
2.3.3	Recopilación de datos en aplicaciones Móviles	18
3	Análisis	23
3.1	Metodología de trabajo	23
3.2	Abordaje del proyecto	24
3.2.1	Estudio de dominio	24
3.3	Relevamiento de Requisitos	25
3.3.1	Usuarios y Roles	28
3.3.2	Requisitos del Sistema	31
4	Diseño	35
4.1	Modelado del dominio	35
4.2	Arquitectura	40

4.2.1	Cliente Web	40
4.2.2	Cliente Móvil	42
4.2.3	Servidor	45
4.2.4	Base de Datos	49
5	Implementación	53
5.1	Arquitectura final	53
5.2	Tecnologías utilizadas	53
5.3	Carga de datos	57
5.4	Aplicación Web	58
5.4.1	Interfaz de usuario	58
5.4.2	Comunicación con Backend	59
5.4.3	Autenticación	59
5.4.4	Funcionalidades	60
5.5	Aplicación Móvil	62
5.5.1	Base de datos local sin ORM	62
5.5.2	Módulos de carga de datos	62
5.5.3	Listas correlacionadas	63
5.5.4	Estructura arborescente de entidades	64
5.5.5	Ayuda en carga de datos	64
5.5.6	Sincronización	65
5.5.7	Potreros en el mapa	66
5.6	Backend y Servidor	66
5.6.1	Componentes del servidor	67
5.6.2	Base de datos	69
5.6.3	Evaluador de Fórmulas	72
5.7	Pruebas de Sistema	75
5.7.1	Pruebas con el usuario	75
5.8	Problemas encontrados	76
5.8.1	Aplicación Móvil	76
5.8.2	Servidor	77
6	Gestión del proyecto	78
6.1	Gestión	78
6.1.1	Fases del proyecto	78
6.1.2	Riesgos	80

<i>ÍNDICE GENERAL</i>	IV
6.1.3 Proyección del proyecto	80
6.2 Desvíos	83
6.3 Hitos del Proyecto	84
7 Conclusiones y Trabajo futuro	86
7.1 Principales aportes y conclusiones	86
7.2 Trabajo futuro	88
Bibliografía	92

Capítulo 1

Introducción

1.1. Contexto y Motivación

Dentro de las principales actividades económicas de nuestro país se encuentran las actividades agropecuarias [1]. A nivel mundial las actividades económicas generan un gran impacto ambiental, produciendo de esta forma cambios climáticos y distintos efectos en la biodiversidad de los ecosistemas, lo que ha generado que cada vez más interés en el análisis y control [8].

El Instituto Nacional de Investigación Agropecuaria (INIA) [19] desarrolló un método de evaluación ambiental en sistemas de producción denominado Índice de Integridad Ecosistémica y está procurando ampliar su uso en el sector. Dicho índice está compuesto por el orden de decenas de variables. Siendo estas variables relativas a estructura, especies, suelo y zona ribereña de cauces de agua. Dichas entidades se agrupan en áreas y refieren a elementos que un investigador o técnico puede evaluar en salidas de campo.

Es del estudio de la integridad ecosistémica que surge el índice que da origen a este proyecto. Índice que es aplicado a establecimientos de producción agraria de los que actualmente se relevan datos utilizando planillas en papel, luego son llevados a una planilla de cálculos electrónicas, donde también se ingresan otros datos como la pendiente media del terreno, la vegetación potencial de la zona y si el terreno cuenta con cursos de agua. Esta planilla de cálculo electrónica se encarga de calcular el índice y entregar los resultados con el fin de evaluar la desviación actual con respecto al estado natural del terreno.

Este índice se encuentra en constante evolución y desarrollo. Con este proyecto se busca avanzar hacia una siguiente etapa de la evolución, apuntando

a la migración del papel y planillas de cálculo electrónicas a un sistema de gestión informático.

1.2. Objetivos

El objetivo general de este proyecto es proponer un sistema para la gestión de estudios del índice de integridad ecosistémica [47] [46] que permita realizar evaluaciones en campo, guardar un histórico de las evaluaciones realizadas y soportar la evolución del índice. En términos generales, se busca lograr una sistematización del procesamiento de la información utilizada para el cálculo del índice, que permita a los usuarios realizar las tareas correspondientes de una manera eficiente y segura.

Para llevar a cabo este objetivo, el mismo fue dividido en objetivos concretos y más pequeños.

1. **Estudiar el Dominio** Estudiar el contexto de trabajo de la investigación de INIA sobre el índice de integridad ecosistémica, así como herramientas similares que atacaran problemáticas similares.
2. **Análisis** Relevar las necesidades de los investigadores para la definición del sistema de gestión de estudios de índice de integridad ecosistémica.
3. **Diseñar** Diseñar y definir la arquitectura de un sistema para cumplir con las necesidades del cliente.
4. **Implementar** Evaluar la factibilidad de la implementación de un prototipo funcional con algunas de las funcionalidades recabadas en etapas anteriores.

1.3. Aportes del proyecto

Los principales aportes que se lograron con este proyecto son:

- **Análisis del modelo de dominio y la realidad del proyecto.** Se realiza un trabajo de investigación con respecto al dominio de la realidad,

obteniendo como resultado tanto diagramas conceptuales, como documentación de índice.

- **Trabajo de análisis de requerimientos.** El trabajo de relevamiento de requisitos realizado por parte del equipo de trabajo abarca un alcance mayor al luego desarrollado, permitiendo relevar requerimientos para un proyecto de gran porte, y obteniendo como resultado diagramas y documentación de todo el trabajo de análisis.
- **Prototipo de sistema.** Con el prototipo de sistema desarrollado como resultado del proyecto, se desarrollaron tres módulos. De estos módulos, destacan por sobre el resto la extensibilidad con la cual fue diseñado el backend y la adaptabilidad de la aplicación móvil a los datos configurados en el servidor.

1.4. Organización del documento

El documento se organiza en varios capítulos. En el capítulo 2 se desarrolla el marco conceptual del proyecto. En este se presentan los trabajos relacionados estudiados, así como los conceptos necesarios para comprender todo el proceso del proyecto. El capítulo 3 se presenta el proceso de análisis llevado a cabo. El capítulo 4 presenta la etapa de diseño del proyecto. Desde una visión de altura (arquitectura “física” del sistema), a una visión más en detalle de la arquitectura de los distintos componentes del sistema. En el capítulo 5 se presentan las tecnologías utilizadas, así como una presentación de cada módulo del sistema y problemas encontrados en la implementación. El capítulo 6 presenta la gestión del proyecto y el modo de trabajo llevado a cabo por el equipo. Por último, el capítulo 7 presenta las conclusiones y trabajos futuros. Luego de estos capítulos, se encuentra el glosario de términos usados, y se anexan los manuales de usuario.

Capítulo 2

Fundamentos teóricos

Este capítulo tiene como finalidad explicar los conocimientos previos necesarios para la comprensión del proyecto, tanto a nivel de dominio, como a nivel informático. A su vez se presenta el trabajo de investigación realizado en búsqueda de tecnologías con enfoques que resuelven problemas similares a los que se desarrollaran a lo largo del documento.

2.1. Índice de Integridad Ecosistémica

En esta sección se presentan los principales términos relacionados a la realidad del cliente.

2.1.1. Términos importantes

1. **Establecimiento:** Extensión de campo administrada por una misma entidad (personas y/o empresas). Unidad máxima en la que se aplica el índice, subdividida en potreros. También es denominado predio.
2. **Potrero:** Unidad de división de establecimientos. Un establecimiento está compuesto por varios potreros, unidad mínima en la cual se calcula el índice.

2.1.2. Índice de Integridad Ecosistémica

En esta subsección se introducen los conceptos principales del índice. Toda la información relacionada al índice es extraída de [47], [46] y [37].

El funcionamiento integral de los ecosistemas es fundamental para una serie de procesos esenciales para la vida humana y la vida silvestre. Este provee los servicios ecosistémicos de los que todos dependemos. Estos servicios se clasifican en cuatro categorías:

- servicios de aprovisionamiento
- servicios de regulación
- servicios culturales
- servicios de soporte

Entre los servicios de soporte se encuentra la conservación de la biodiversidad, tanto en su composición como en su funcionamiento, aspectos que el índice de integridad ecosistémica (IIE) apunta a evaluar para un ecosistema dado.

En el fundamento del diseño del IIE se representan las relaciones que hacen posible tener una aproximación al funcionamiento de un ecosistema. Esto se realiza mediante la evaluación del estado de los componentes del ecosistema a través de la toma de medidas. La gran ventaja que aporta este trabajo radica en que medir los servicios directamente, o aspectos finos de la composición y funcionamiento, es un proceso caro en términos de tiempo y recursos humanos, que en algunos casos implica un elevado costo en equipamiento e insumos. Esta es la principal fortaleza del IIE, ya que hace posible tener una valoración cuantificada del estado de un ecosistema, aplicable por personas con una capacitación básica, sin involucrar equipos o análisis de laboratorio.

¿Cómo funciona el índice?

El IIE asigna una calificación de cero a cinco, siendo cinco el mejor estado teórico posible para un sitio determinado en función de la ecorregión del país en la que se encuentra. La calificación disminuye en la medida que la situación se aleja de ese óptimo. La evaluación incluye cuatro dimensiones principales que son consideradas claves para el funcionamiento del ecosistema terrestre:

- el estado del suelo
- la estructura de la vegetación
- las especies vegetales que componen la comunidad
- el estado de cursos de agua (cañadas y arroyos) y su zona riparia

Es así como cada una de esas dimensiones es calificada en forma independiente y todas son promediadas en un único valor, en función del peso ponderado de cada una.

El IIE está pensado para la aplicación a nivel de potreros u otras unidades de decisión de manejo (cuadros, franjas, chacras) dentro de los establecimientos. Para el cálculo del índice de todo un establecimiento se calcula como el valor de cada uno de los potreros, ponderado por el porcentaje de área que el potrero representa sobre el total del establecimiento. De forma que se obtiene un resultado del IIE para un establecimiento, un valor para cada potrero y además se puede ver la influencia de las diferentes dimensiones evaluadas (suelo, especies, estructura y cursos de agua).

Proceso de relevamiento de datos en campo

El relevamiento de la situación en el campo se realiza siguiendo un protocolo. Esto quiere decir que quien releva los datos no califica, sino que recolecta datos e información, anotando todos los datos necesarios en una planilla. Dentro de los datos que relevan se encuentran:

Suelo

A nivel de suelo se relevan datos como:

- Erosión
- Factores predisponentes

Estructura de la vegetación

A nivel de estructura se relevan datos como:

- Presencia y distribución de los distintos estratos [2.1](#)
- Disturbios
- Especies (rangos de diversidad en los estratos)

Estratos

A nivel de estratos se releva la cantidad de especies.

Especies

A nivel de especies se relevan datos relativos a:

- Diversidad estratos
- Especies exóticas

Zona riparia

A nivel de zona riparia se evalúan aspectos similares a los elementos antes mencionados. A su vez, se documentan alteraciones como represamientos, o contaminaciones con residuos visibles, así como erosión del canal, desmoronamientos y sedimentación.



Figura 2.1: Estratos. Extraído de documentación de INIA.

Un ejemplo de una medida que se releva en campo, relativa a la dimensión de especies, es el registro de variedades de especies exóticas.

¿Cómo se visualizan los resultados?

La planilla de cálculo electrónica a la que se transfieren los datos de campo, retorna el resultado de los potreros y su respectivo establecimiento, contando con el respectivo desglose de este cálculo. Los resultados del índice se pueden visualizar a través de un plano del establecimiento. Para ello se usa un sistema de información geográfica (SIG) donde los valores son cargados como atributos de los distintos polígonos que representan a cada potrero o unidad de análisis. Esto permite una representación espacial con escala de colores que facilita la visualización. Este SIG permite además seguir agregando otra información que puede ser de interés para el productor como usos del suelo, características de pendiente, fertilidad, productividad primaria.

¿Cómo puede usarse?

El IIE es una herramienta de evaluación ambiental que brinda la posibilidad de conocer en que estado se encuentra un determinado establecimiento. Es posible usarlo para evaluaciones puntuales o como trayectorias a lo largo del tiempo. Se puede usar como herramienta de gestión, con la que se puede discriminar los aspectos evaluados que se encuentran en niveles más bajos y tomar medidas para mejorar los valores de un establecimiento o potrero. Estos valores pueden ser usados para detectar problemas ecosistémicos en donde se esté evaluando, y de esta forma tomar medidas para solucionar las problemáticas, haciendo que los valores mejoren y por ende evitando la pérdida ese suelo.

Fórmula

$$\frac{\sum_{i=1}^n (St_i + Sp_i + S_i + Zr_i) \cdot PA_i}{4FA}$$

Donde cada uno de los términos significa:

- St_i : Calificación de estructura para el potrero i
- Sp_i : Calificación de especies para el potrero i
- S_i : Calificación de suelo para el potrero i
- Zr_i : Calificación de zona riparia para el potrero i
- PA_i : Área del potrero i
- FA : Área del establecimiento
- i : Potrero i del establecimiento
- n : Cantidad de potreros que componen al establecimiento

2.2. Tecnologías de la información

En esta sección se presentan los principales conceptos informáticos a usar en el documento.

2.2.1. Definiciones Generales

Aplicación Web

Una aplicación web [43] es una herramienta que puede ser usada por usuarios. Para ello es preciso acceder a un servidor web a través de internet o de una intranet mediante un navegador. Es un software codificado en un lenguaje que pueda ser ejecutado por los navegadores web.

La popularidad de las aplicaciones web se debe a lo práctico de los navegadores web como clientes ligeros, la independencia de estas del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones sin distribuir e instalar software a miles de usuarios potenciales.

Una página web puede contener elementos que permiten comunicarse de forma activa al usuario con la información. Es decir, el usuario puede interactuar con la información a la cual está accediendo. De modo que la aplicación responde a cada una de sus acciones.

Aplicación Móvil

Una aplicación móvil [5], es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil. Entre estos dispositivos se encuentran teléfonos inteligentes, tablets, etc. A diferencia de las aplicaciones web o de escritorio, las aplicaciones móviles se alejan de los sistemas de software integrados. En cambio, cada aplicación móvil proporciona una funcionalidad aislada y limitada, como puede ser un juego, una calculadora o un navegador web móvil, etc.

Debido a los recursos de hardware limitados de los primeros dispositivos móviles, las aplicaciones móviles evitaban la multi funcionalidad. A pesar de que hoy en día se cuentan con más recursos a nivel de hardware, las aplicaciones móviles mantienen su principal característica, siguen siendo funcionales.

Mock-UP

Herramienta conceptual utilizada en el desarrollo de aplicaciones, con el objetivo de elaborar un primer borrador de una pantalla. Son utilizados para convertir ideas y conceptos en un diseño concreto. Incluyen la estructura de navegación final y elementos de diseño detallados, buscando aproximarse a la experiencia de usuario definitiva.

Software de Recolección de datos

Software cuyo objetivo es la recolección y almacenamiento de datos cuantitativos y cualitativos en base a formularios. Estos brindan múltiples ventajas a los usuarios tales como la posibilidad de exportar, analizar y presentar los datos de manera rápida y eficiente.

Protocolo HTTP/HTTPS

HTTP, (Hypertext Transfer Protocol), es un protocolo que permite realizar una petición de datos y recursos, como pueden ser documentos HTML, imágenes, etc. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor.

HTTP se basa en sencillas operaciones de solicitud-respuesta. El cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Adicionalmente las operaciones pueden adjuntar un objeto o recurso sobre el que actúan. Cada objeto en la Web, (por ejemplo documentos HTML, archivo multimedia o aplicación) es conocido mediante su URL.

Arquitectura Cliente-Servidor

Arquitectura de sistemas que permite conectar a varios clientes a los servicios que provee un servidor. La mayoría de las aplicaciones y servicios tienen como gran necesidad que puedan ser consumidos por varios usuarios de forma simultánea y una forma de resolver esta situación es haciendo uso de esta arquitectura.

Los componentes de una arquitectura Cliente/Servidor son:

1. **Red:** Una red es un conjunto de clientes, servidores, bases de datos, etc.
2. **Cliente:** Demandante de servicios. Requiere información proveniente de la red para funcionar.
3. **Servidor:** Proveedor de servicios, envía (cuando corresponde) información a los demás agentes de la red.
4. **Protocolo:** Conjunto de normas, reglas y/o pasos establecidos de manera clara y concreta sobre el flujo de información.
5. **Servicios:** Conjunto de información y procesamientos que buscan

responder las necesidades de un cliente.

6. **Base de datos:** Bancos de información ordenada, categorizada y clasificada.

API

Una API (interfaz de programa de aplicación), define la comunicación entre sistemas de software mediante la invocación de funciones o procedimientos. El objetivo de una API es exponer mediante interfaces los recursos para comunicar distintos sistemas.

RESTFul

Una API RESTful es una interfaz entre dos sistemas la cual es usada para el intercambio de información de forma segura entre los sistemas involucrados. La seguridad a la que se hace mención está dada porque las API RESTful siguen estándares de comunicación confiables, seguros y eficientes.

Los actores participantes en el consumo de una API son los siguientes:

1. Clientes:
Usuarios que desean acceder a la información provista por la API. El cliente puede ser una persona o un sistema de software que utiliza la API.
2. Recursos:
Un recurso es la información que diferentes aplicaciones proporcionan a sus clientes. Un recurso puede ser una imagen, videos, texto, números o cualquier otro tipo de objeto.
3. Servidor:
Actor encargado de entregar recursos al cliente.

Al solicitar un recurso a una API RESTful hay que tener en cuenta los siguientes elementos.

1. URL:
El servidor identifica cada recurso con identificadores únicos de recursos.
2. Método:
Las API RESTful son desarrolladas usando el protocolo HTTP. El método o verbo HTTP indica al servidor lo que debe hacer con el recurso.

Dentro de los métodos recién mencionados destacan:

1. GET:
Es usado para acceder a los recursos que están ubicados en la URL especificada en el servidor.
2. POST:
El verbo POST se usa para enviar datos al servidor. Incluyen la representación de los datos con la solicitud.
3. PUT:
Este método es utilizado para actualizar los recursos existentes en el servidor.
4. DELETE:
Verbo usado para eliminar el recurso correspondiente.

OpenAPI

La especificación OpenAPI (OAS) define una interfaz estándar independiente del lenguaje utilizado para el desarrollo de las API RESTful. OpenAPI permite que tanto los humanos como las computadoras descubran y comprendan las capacidades del servicio, interactuar con el servicio remoto con una cantidad mínima de lógica de implementación. Las herramientas de generación de documentación pueden usar una definición de OpenAPI para exponer la API, herramientas de generación de código para generar servidores y clientes en varios lenguajes de programación, herramientas de prueba y muchos otros casos de uso.

Autenticación

Se define como autenticación al proceso de verificar una (pretendida) identidad. Este proceso está determinado por las siguientes etapas:

1. Obtener la información de autenticación de una entidad.
2. Analizar los datos.
3. Determinar si la información de autenticación está efectivamente asociada a la entidad.

Autorización

Se define como Autorización al proceso que determina (luego de su autenticación) a que recursos de un sistema tiene acceso una identidad.

OAuth 2.0

OAuth 2.0 es un estándar que permite realizar flujos de autorización tanto para usuarios finales como otras aplicaciones. Para entender un poco mejor OAuth 2.0 es necesario comprender los roles que en él participan.

- Dueño del recurso: Es el propietario de el/los recursos
- Cliente: Usuario aplicación que quiere hacer uso del recurso
- Servidor de recurso: Api que expone los recursos
- Servidor de autorización: Gestiona las peticiones de autorización, verifica identidades y emite tokens de acceso

El flujo de OAuth 2.0 se presenta en la figura 2.2.



Figura 2.2: Flujo Genérico OAuth 2.0 [30]

Identity Provider

Un proveedor de identidad IdP [18], es un servicio que almacena y administra identidades digitales. Estos servicios permiten identificar a los usuarios que se conectan con los recursos que necesitan. Proporcionan

una forma estandarizada de administrar el acceso, agregando o eliminando privilegios a los usuarios.

Docker y Contenedores

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución.

Docker brinda una manera estándar de ejecutar el código de distintas aplicaciones indistintamente del lenguaje en el que estas hayan sido desarrolladas. Docker es un sistema operativo para contenedores. De manera similar a como una máquina virtual virtualiza (elimina la necesidad de administrar directamente) el hardware del servidor, los contenedores virtualizan el sistema operativo de un servidor. Docker se instala en cada servidor y proporciona comandos sencillos que son usados para crear, iniciar o detener contenedores.

Algunas de las ventajas del uso de dockers son:

- Restauración: En caso de que una nueva imagen no cumpla las expectativas, es sumamente sencillo volver a versiones anteriores.
- Modularidad: Permite separar partes de la aplicación sin necesidad de deshabilitar toda la aplicación.
- Implementación rápida: Levantar un contenedor es un proceso sumamente ágil, por lo cual agiliza los despliegues de aplicaciones.

JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. Es de fácil lectura y escritura para los usuarios, al mismo tiempo es fácil de analizar y generar por parte de las máquinas. JSON se basa en un subconjunto del lenguaje de programación JavaScript [22].

Geo-JSON

GeoJSON es un formato de intercambio de datos geoespaciales basado en JavaScript. Los objetos son representados usando formato JSON. GeoJSON define varios tipos de objetos JSON y la manera en que estos

se combinan para representar datos sobre características geográficas, sus propiedades y su extensión espacial.

APK

Un archivo APK (Android Application Package) es un archivo empaquetado, el cual contiene una aplicación móvil que puede ser instalada en un dispositivo móvil Android.

Hibernate

Hibernate [16] es un ORM. Es decir, una herramienta de mapeo objeto a entidades de base de datos.

Hibernate está bajo una licencia libre para Java. Esta herramienta facilita el mapeo de atributos en una base de datos tradicional, y el modelo de objetos de una aplicación mediante archivos declarativos o anotaciones en las entidades que permiten establecer estas relaciones. Esta herramienta agiliza la relación entre la aplicación y la base de datos SQL, de un modo que optimiza el flujo de trabajo.

SIG

SIG [40] (Sistema de Información Geográfica) es un sistema que crea, administra y mapea todo tipo de datos georreferenciados. Este tipo de sistemas conectan los datos a un mapa, integrando datos de localización con todo tipo de información descriptiva. Esto permite las bases para la construcción de mapas y análisis de datos que son aprovechados en distintas áreas de la ciencia e industrias.

2.3. Trabajos relacionados

En todo proyecto es necesario realizar un estudio de las distintas herramientas que ayudan a resolver problemáticas similares. En la siguiente subsección se presentan algunas de las herramientas analizadas. Estas herramientas tienen por objetivo recolección de datos desde aplicaciones móviles, visualización de datos en mapas tanto en aplicaciones web como aplicaciones móviles.

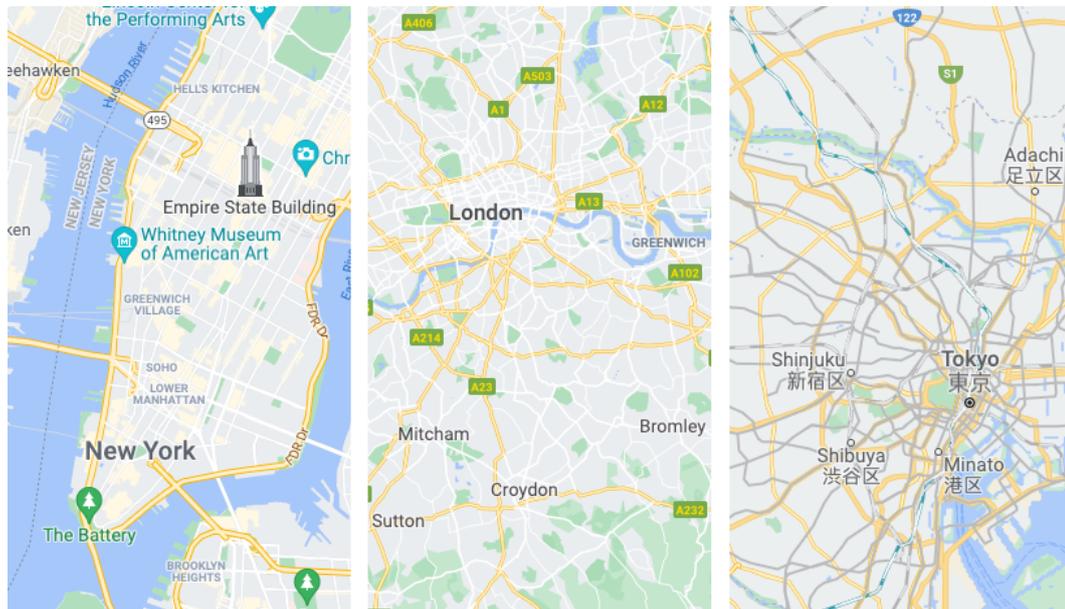


Figura 2.3: GoogleMaps. Extraído de documentación GoogleMaps.

2.3.1. Aplicaciones Web

GoogleMaps

GoogleMaps es una herramienta para posicionar información, tanto en formato JSON como GeoJSON, sobre capas en un mapa base. Haciendo uso de la API Javascript de Google Maps, se pueden insertar en un mapa marcadores, polilíneas, polígonos, etc. Facilitando de esta forma su extensión a múltiples páginas web.

OpenStreetMap

OpenStreetMap (OSM) es un proyecto colaborativo de código abierto para crear mapas libres y editables. OpenStreetMap está creado por una gran comunidad de colaboradores que con sus contribuciones al mapa añaden y mantienen datos sobre caminos, senderos, cafeterías, estaciones de ferrocarril y muchas cosas más a lo largo de todo el mundo.

Esta herramienta es utilizada por miles de sitios web, aplicaciones móviles y dispositivos de hardware.

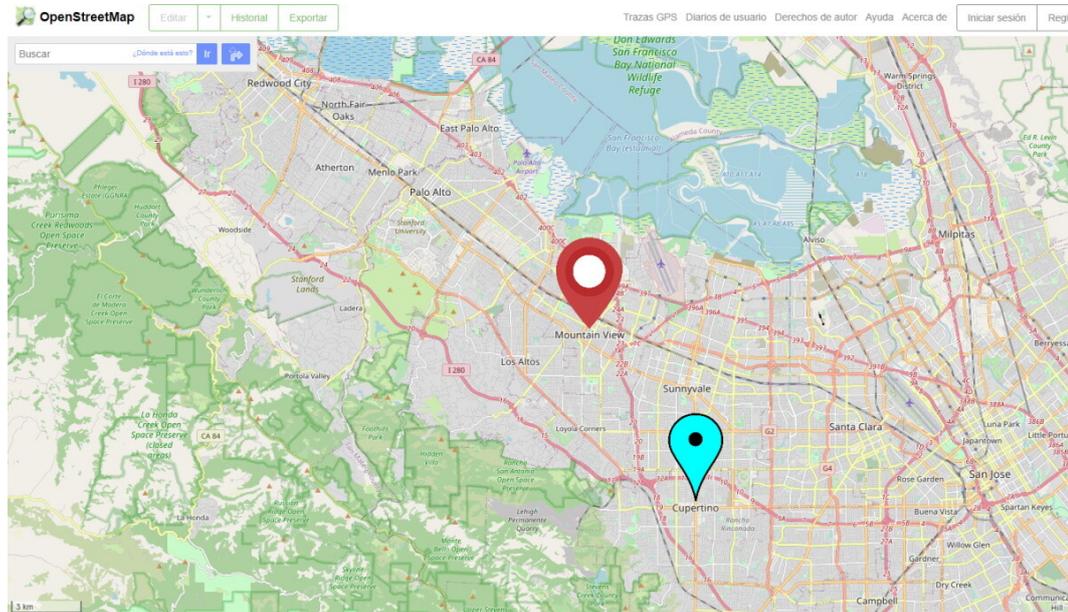


Figura 2.4: OpenStreetMap. Extraído de documentación de OSM.

2.3.2. Aplicaciones Móviles

CartoDruid

CartoDruid es una aplicación GIS para trabajar en campo directamente con dispositivos móviles. Una de sus principales funciones es resolver el problema de la utilización de información geográfica fuera de la oficina en un entorno desconectado, tanto para poder consultarla como editarla. Es una aplicación de mucha utilidad para los técnicos agrarios, agricultores, ganaderos, etc. CartoDruid está basada en CartoSAT, una aplicación con diez años desde que fue creada y cuyos principales proyectos fueron las inspecciones de campo, observatorio de plagas. A pesar de ello cuenta con varios puntos en contra, como lo son:

- Altos costos
- Sin actualizaciones
- Sin soporte para versiones nuevas
- Atado a software propietario
- Usable en todo tipo de condiciones

CartoDruid aporta a lo antes mencionado las siguientes fortalezas:

- Uso offline
- Edición don operaciones gráficas

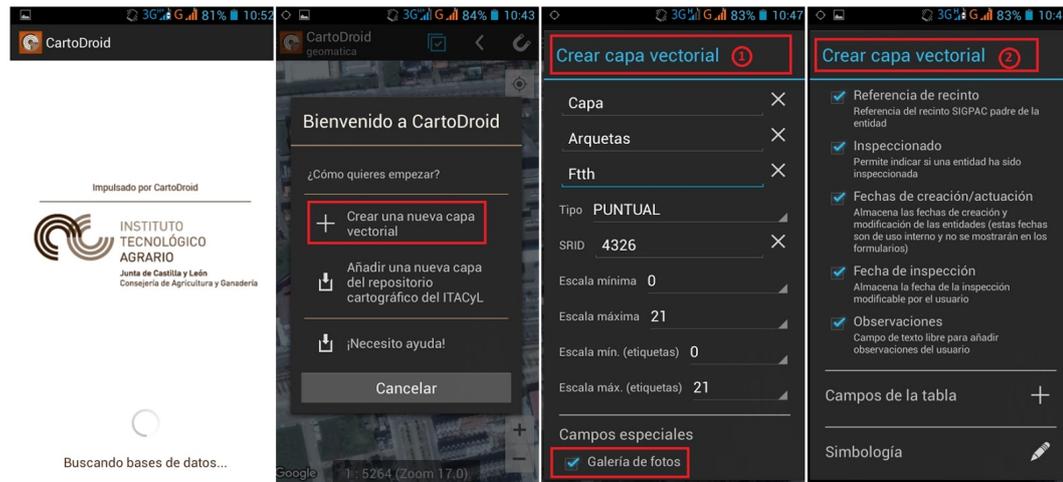


Figura 2.5: CartoDruid. Extraído de documentación de CartoDruid.

- Formularios configurables
- Bajo costo
- Uso de datos e imágenes
- Permite importar datos
- Permite exportar datos

gvSIG

gvSIG Mobile es una aplicación desarrollada para realizar relevamiento de datos sobre el terreno de una forma rápida y de calidad. Es interoperable para la recolección de datos en dispositivos Android. Es parte de la Suite gvSIG y se integra directamente con gvSIG desktop y gvSIG online.

Está orientado a la captura de datos en campo y recomendado para proyectos de inventarios, censos, revisiones, inspecciones, etc.

2.3.3. Recopilación de datos en aplicaciones Móviles

Teamscope

Teamscope [42] es una plataforma de recopilación de datos segura, desarrollada para gestionar y almacenar datos confidenciales, investigaciones

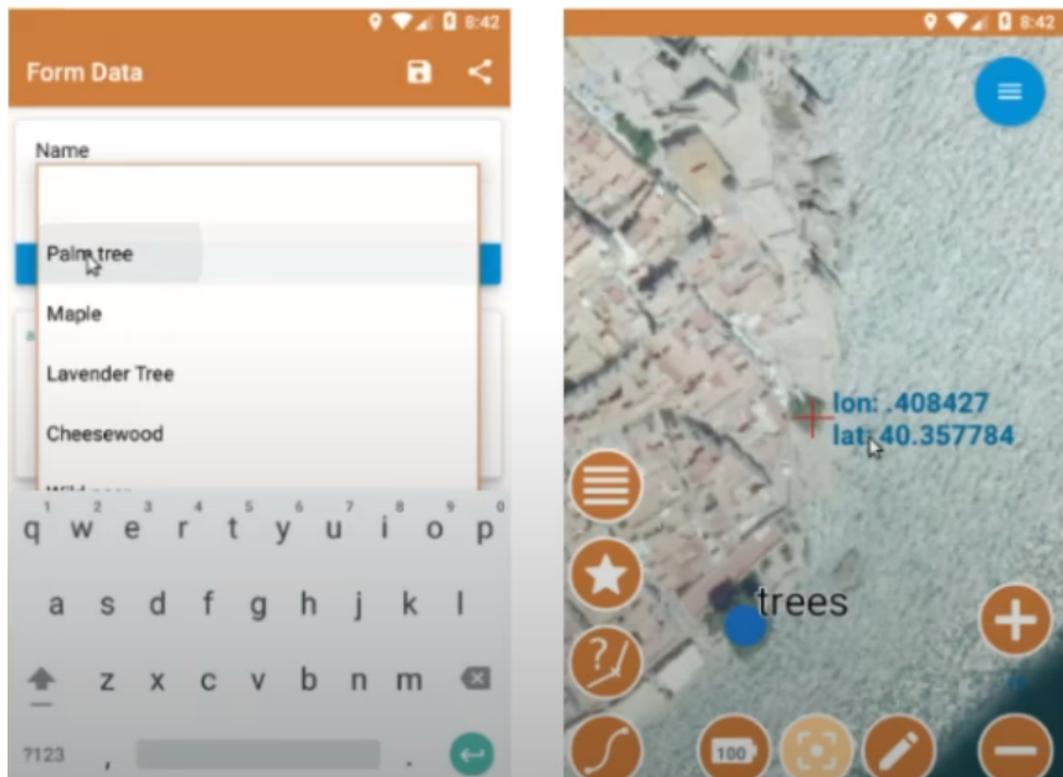


Figura 2.6: GVSig. Extraído de documentación de GvSig.

científicas, investigaciones clínicas, etc. Esta plataforma ofrece una aplicación móvil que surge en un contexto en donde la mayoría de las herramientas son aplicaciones web. Es decir, son aplicaciones que dependen de una conexión a internet y sin ella no son capaces de funcionar. Bajo esta premisa, fue que se desarrolló pudiendo funcionar de forma offline en los dispositivos y permitiendo recolectar información a los usuarios. La prioridad de esta aplicación es la seguridad de datos, por lo cual los datos son almacenados en los dispositivos móviles bajo las credenciales de los usuarios, las cuales son necesarias tanto para acceder a la aplicación como para el intercambio de datos con el servidor.

ODK

Open Data Kit (ODK) es un software de código abierto para la recolección, gestión y uso de datos en entornos con recursos limitados. El objetivo del ODK es ofrecer herramientas de código abierto basadas en estándares que sean fáciles de probar, fáciles de usar, fáciles de modificar

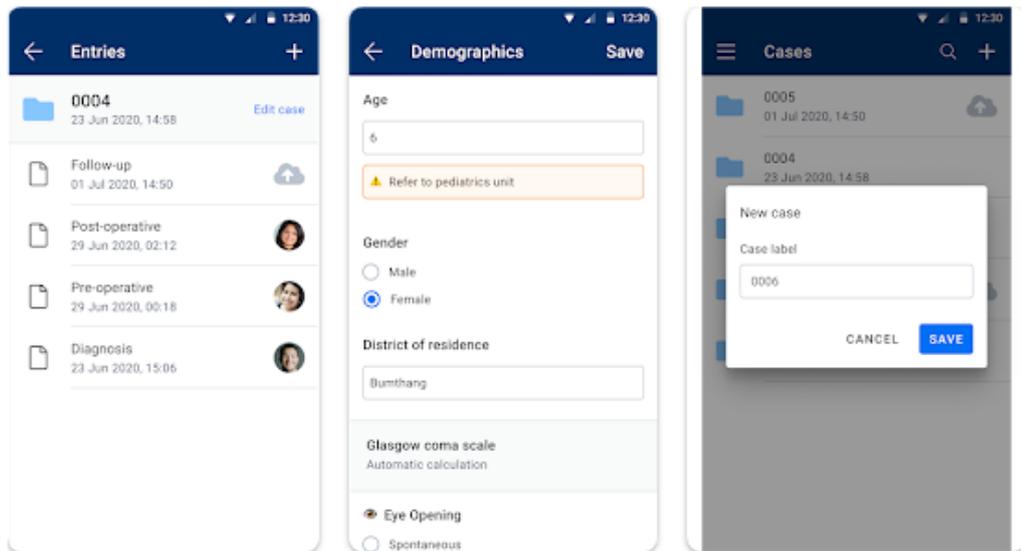


Figura 2.7: Teamscope. Extraído de documentación de TeacmScope.

y fáciles de escalar. Algunos de los tipos de datos soportados por ODK son:

1. Imágenes
2. Texto
3. Coordenadas

Además de esto, ODK soporta múltiples idiomas y brinda la posibilidad de uso offline. ODK es multiplataforma, permite tanto hacer uso de dispositivos móviles como de la aplicación web. Estas suites son ODK y ODK-X El primero brinda acceso a herramientas sencillas de despliegue a gran escala para el registro de datos móviles. Es sumamente utilizado para proyectos de pequeño y mediano porte. Por otra parte, ODK-X ofrece herramientas para flujos de trabajos más complejos, o proyectos que requieren más procesamiento de datos.

Destacan dentro de sus características que es de código abierto, gratuito, funciona de forma online y posee una gran comunidad trabajando activamente.

Este producto solo está disponible para Android.

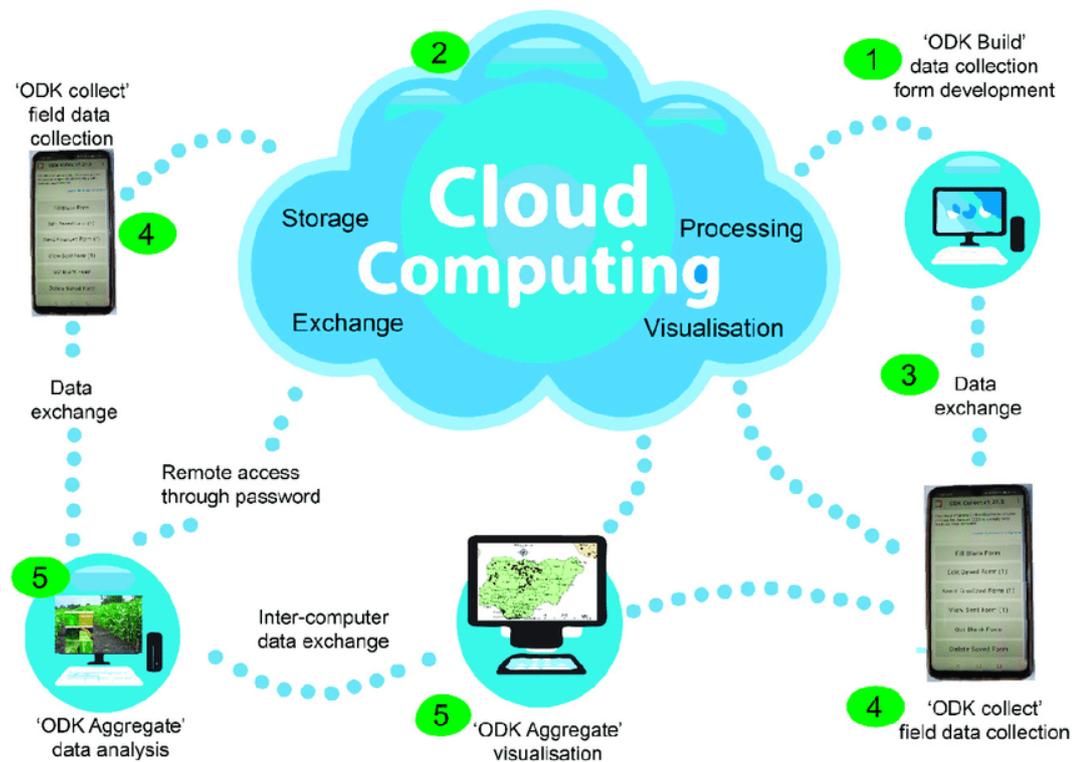


Figura 2.8: ODK. Extraído de documentación de ODK.

Magpi

Magpi es una aplicación para el almacenamiento de datos para dispositivos móviles la cual brinda a los usuarios la posibilidad de crear formularios tanto de manera offline como online. Es ampliamente usada tanto en sectores de la salud, la agricultura, el medio ambiente, etc. Es usada en sectores en donde es necesario la realización de manera rápida, barata y eficiente de encuestas móviles para llevar a cabo investigaciones escalables y directas.

Dentro de las principales funcionalidades de Magpi se encuentran el uso de imágenes, datos de GPS, entre otros. A su vez permite introducir datos de todo tipo de usuarios y se integra con hojas de cálculo de Google, bases de datos SQL, etc.

Su objetivo es sacar el máximo provecho de todo tipo de aplicación que se enfoque a recopilar datos y brindando la posibilidad de analizar estos datos.

Entre sus principales características destacan la posibilidad de uso offline, notificaciones mediante SMS, recopilación de datos de respuesta de voz

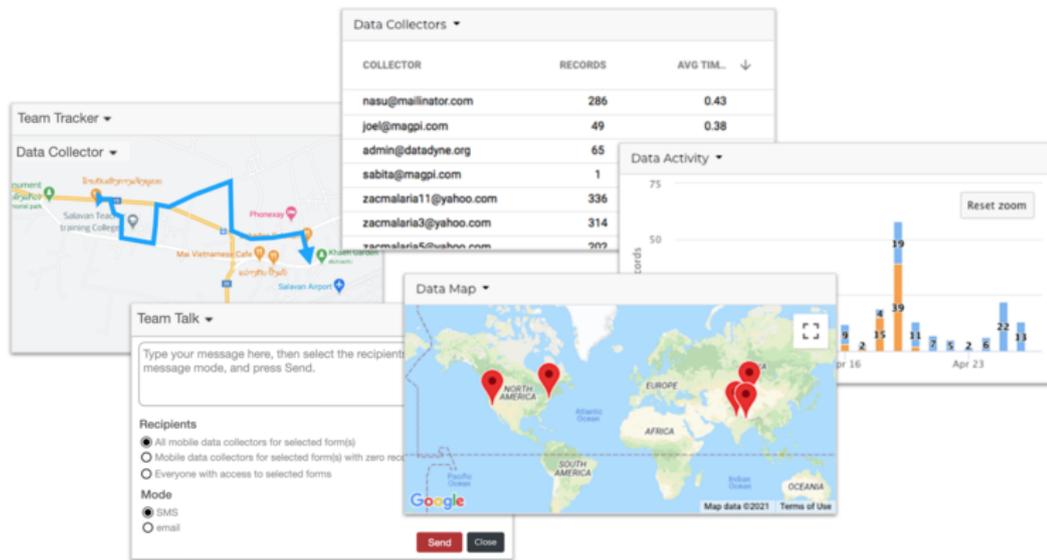


Figura 2.9: Magpi. Extraído de documentación de Magpi.

interactiva (IVR), entre otras. Como contraparte, su versión gratuita no permite el uso de gran parte de estas funcionalidades y cuenta con planes de pago y planes empresariales.

Capítulo 3

Análisis

En este capítulo se presenta la metodología de trabajo, el estudio del dominio, el proceso de relevamiento de requisitos, las técnicas usadas y los cambios que se dan a medida que avanza este proceso. Por último, se presenta la especificación de los requisitos funcionales y no funcionales del sistema.

3.1. Metodología de trabajo

Se busca involucrar al cliente en las etapas en las cuales es de suma importancia su presencia, para esto se plantea la coordinación de reuniones periódicas, con el fin de acordar y organizar las tareas, para de esta forma llevar adelante un proceso en el cual ambas partes estén presentes en la toma de decisiones.

Un resumen de la planificación del trabajo es el siguiente:

- Presentación inicial del proyecto: Reuniones periódicas para que el cliente explique tanto el dominio, como la problemática que desea resolver.
- Estudio: Fase de estudio por parte de los estudiantes.
- Requisitos: Reuniones semanales para el intercambio de entendimiento entre las partes para poder modelar los requisitos. Siendo estas mayormente expositivas, donde el equipo presenta los artefactos utilizados para relevar los requisitos, poder validarlos y en caso de ser necesario realizar correcciones.
- Desarrollo y Pruebas: Se realiza una etapa de desarrollo con menor interacción con el cliente, para luego presentarle avances del prototipo y

poder realizar pruebas en conjunto.

Si bien la metodología utilizada no se basa en ninguna de las metodologías de desarrollo de software formales, se siguió un proceso incremental, similar a un modelo en cascada [29], con iteraciones en cada etapa.

3.2. Abordaje del proyecto

El estudio del dominio es una de las primeras tareas que debe ser llevada a cabo en todo proyecto. Esta es necesaria para que el equipo de trabajo pueda entender el contexto del proyecto, así como para comprender la realidad del cliente, y de esta forma poder resolver las problemáticas planteadas, optando por las mejores alternativas. Este proceso se divide en varias etapas, las mismas son descritas en las próximas subsecciones.

3.2.1. Estudio de dominio

El estudio de dominio es la etapa en la cual se realiza el abordaje inicial del proyecto. Esta etapa consiste en reuniones frecuentes con el cliente. Los principales temas que se abordan en estas reuniones son el IIE y todos los conceptos asociados a este. El resultado de estas reuniones es parte de lo presentado en el capítulo 2. Por otra parte, en estas reuniones surgieron otras temáticas como los usos actuales del índice. Tanto el índice en sí, como el uso de este hoy en día son aspectos fundamentales para la etapa de análisis. En esta etapa no es necesario ahondar en más detalles sobre el índice, pero es importante destacar el funcionamiento actual del mismo.

Funcionamiento actual

Hoy en día el uso del índice en campo es realizado por un reducido grupo de trabajo dentro de INIA. A su vez, este grupo cuenta con planillas de cálculo electrónicas con la fórmula para calcular el índice, en función de los datos recabados en campo. Para la fase de carga de datos el grupo de trabajo, que a su vez es el encargado de ir al campo y tomar todas las mediciones, releva esta información en papel, en una planilla similar a la planilla de cálculo electrónica con la que cuenta. Una vez recabados los datos de los potreros, esas planillas son llevadas a INIA y los datos se pasan a las planillas de cálculo para que estas queden guardadas y el índice se calcule. Este proceso es principalmente manual,

siendo así uno de los principales motivos por los cuales surge la necesidad de digitalizar el proceso.

El comprender el funcionamiento actual es de utilidad para proyectar el prototipo de sistema y comenzar a entender las necesidades del cliente. Pudiendo así, sentar las bases sobre las cuales se construye el backlog de trabajo tanto en esta etapa como en las siguientes.

3.3. Relevamiento de Requisitos

La etapa de relevamiento de requisitos se lleva a cabo mediante la repetición de tres tareas, estas son:

- **Generación de requisitos** Proceso mediante el cual se definen y plasman los distintos requisitos del sistema en mock-ups.
- **Presentación y discusión de requisitos** Instancia de reunión en la cual se discuten los requisitos nuevos y los que aun no han alcanzado el nivel final de refinamiento.
- **Corrección de requisitos** Proceso en el cual se toma como parámetros de entrada requisitos y los comentarios del cliente para obtener como resultado un requisito refinado.

Antes de comenzar con la metodología de trabajo, es necesario explicar el concepto de historia de usuario. Una historia de usuario es una explicación general e informal de una función de software. Está escrita desde la perspectiva del usuario final. Su propósito es articular como proporcionará una función de software valor al cliente. En resumen, una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final.

Las historias de usuario [17] son los componentes básicos de los marcos ágiles, como los epics e iniciativas.

- **Epics** Los epics o simplemente épicas, son grandes elementos de trabajo divididos en un conjunto de historias.
- **Iniciativas** Varias épicas constituyen una iniciativa. Estas estructuras más grandes garantizan que el trabajo diario del equipo de desarrollo contribuya a los objetivos de la organización incorporados en las épicas y las iniciativas.



Figura 3.1: Ejemplo Mock-Up aplicación móvil

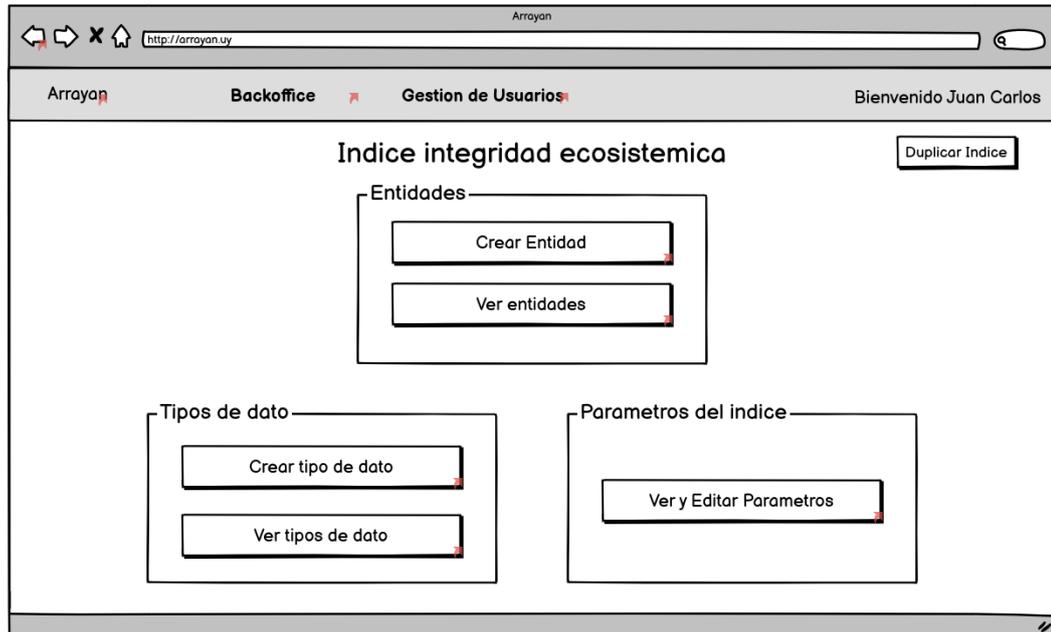


Figura 3.2: Ejemplo Mock-Up aplicación web

En la figura 3.3 se presenta un diagrama que representa lo recién mencionado.

La metodología pensada para relevar los requisitos en esta etapa fue la de uso de historias de usuario, complementando estas con mock-ups (ver figura 3.1 y 3.2), especialmente desarrollados para la discusión de los distintos escenarios y requisitos con el cliente. Luego de una primera presentación de los conceptos y la modalidad de trabajo a adoptar, el cliente no se vió a gusto con el enfoque de historias de usuario, en contraposición a lo que sucedió con los Mock-Ups. Esto lleva a un cambio de rumbo en el proceso, por lo cual luego de la primera iteración se enfatizó el desarrollo de Mock-Ups y esta fue la metodología adoptada por el equipo para continuar con este proceso.

En las siguientes tres reuniones se pudieron recolectar suficientes escenarios que reflejaban el deseo del cliente, generando un listado de requisitos lo suficientemente completo como para desarrollar el prototipo de sistema planteado. Al mismo tiempo, también se relevaron una gran cantidad de escenarios que pueden servir como punto de partida para seguir desarrollando este sistema.

Una ventaja aportada por el enfoque tomado fue que se logró independizar por completo la fase de relevamiento, de la fase de la selección de tecnologías. Por lo cual, este proceso fue llevado a cabo con el único fin de captar la mayor cantidad de casos y lograr conceptualizar lo que el cliente deseaba, sin tener

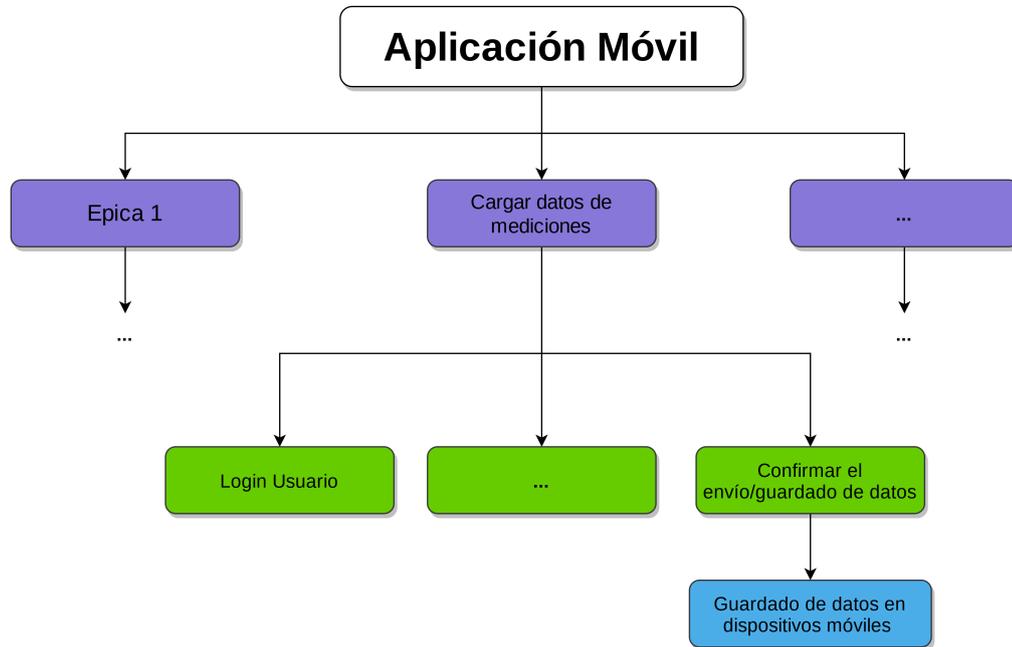


Figura 3.3: Diagrama historias de usuario.

un enfoque hacia como esto iba a ser implementado. Logrando de esta forma focalizar en esta etapa los siguientes elementos:

- Captar el detalle de los requisitos.
- Enfocar el proceso en el cliente.
- Mejorar la experiencia de usuario.

3.3.1. Usuarios y Roles

Una vez analizada y comprendida la realidad, se llega a la conclusión de que existen datos sensibles. Estos datos no pueden ser modificados e incluso no deberían si quiera ser accedidos por algunos usuarios del sistema. Esto motiva al uso de roles y usuarios para poder gestionar la información relacionada tanto al índice de integridad ecosistémica, así como la asociada a los distintos establecimientos y sus respectivos potreros.

Se identifican múltiples funcionalidades del sistema, para ello lo que se plantea es un enfoque en el cual existen varios grupos de usuarios, estos grupos tienen asociados un conjunto de funcionalidades. Por lo cual un usuario al pertenecer a un grupo tendrá acceso a distintas funcionalidades como se presenta en la figura 3.4.

3.3.1.1. Grupo de Usuarios Invitados

Todo usuario al cual se le brinda el acceso a reportes e información por parte de otros usuarios con mayores permisos. Estos usuarios solamente acceden a lo que se les permita puntualmente a revisar, y sólo en modo lectura, sin posibilidad de modificar o alterar ningún dato sea sensible o no.

3.3.1.2. Grupo de Usuarios Móviles

El grupo de usuarios móviles está compuesto por todos aquellos usuarios que trabajan en campo tomando medidas de los distintos establecimientos.

3.3.1.3. Grupo de Usuarios BackOffice

El grupo de usuarios de backoffice es un reducido grupo de usuarios que tienen como premisa el manejo de toda la información administrativa del sistema. Tienen las facultades para ayudar y corregir información relacionada a los establecimientos y potreros en caso de que sea necesario.

3.3.1.4. Grupo de Usuarios Técnicos

En vistas de que el IIE es una fórmula que ha ido evolucionando a lo largo del tiempo, que se ha vuelto sumamente compleja y sensible, se generó este grupo de usuario. Usuario con permisos para poder acceder a leer, modificar y eliminar todo tipo de dato asociado a la configuración de los índices que manejen, pudiendo ser el único grupo de usuario con acceso a esta información.

3.3.1.5. Aspectos a tener en cuenta

Para el acceso a la información de los distintos establecimientos, también se agregó una capa más de permisos, ya que cada establecimiento tiene asociado a múltiples usuarios en distintas modalidades. Los usuarios asociados son:

- Dueño. Usuario que registró el establecimiento.
- Usuarios con permisos para acceder a la información del establecimiento.

El administrador, los usuarios de backoffice y el dueño del establecimiento, tienen los permisos de asignar otros usuarios para que tengan visibilidad de los datos sensibles del establecimiento.

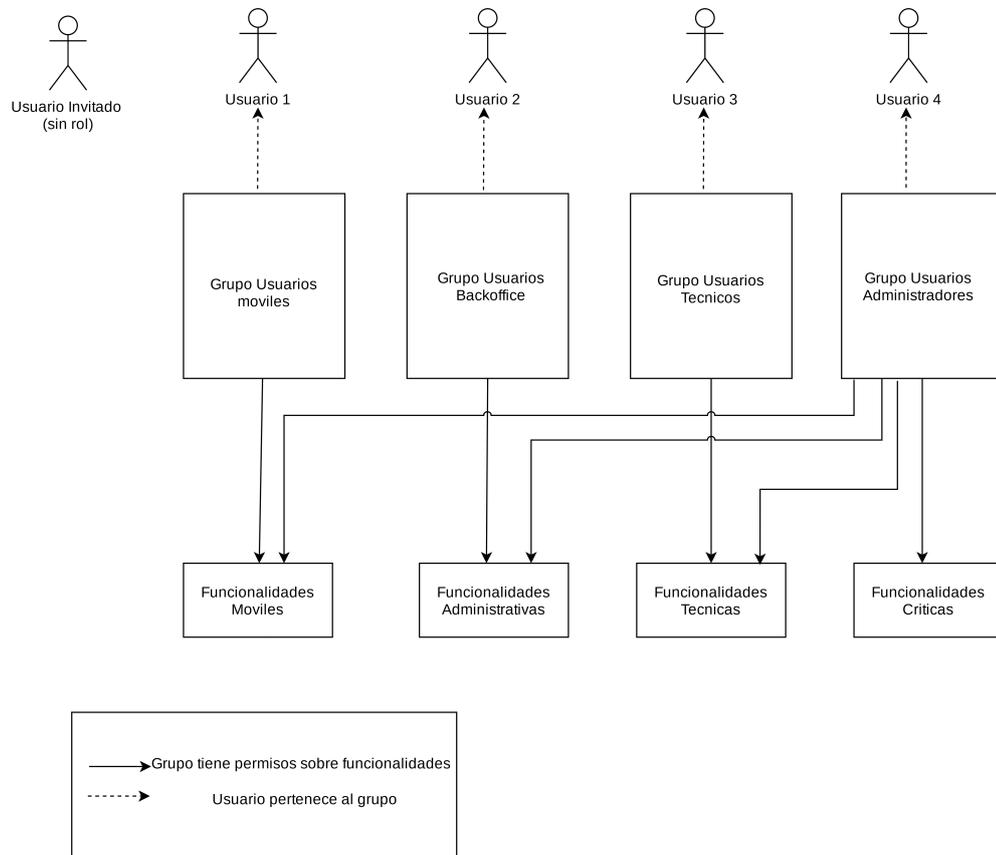


Figura 3.4: Usuarios, roles, grupos y funcionalidades

3.3.2. Requisitos del Sistema

3.3.2.1. Requisitos Funcionales

A continuación, se presentan los principales requisitos funcionales del sistema presentados en formato de historia de usuario. Para una mejor comprensión de estos, se plantea un enfoque separado por módulos. Se presentan los requisitos funcionales tanto de la aplicación móvil, como los de la aplicación web, así como los requisitos que estas tienen en común.

Requisitos en común

Login de usuarios con Google

Como usuario tanto de la aplicación web, como de la aplicación móvil quiero ingresar al sistema haciendo uso de mis credenciales de Google.

Alta, baja y modificación de establecimientos

Como usuario tanto de la aplicación web, como de la aplicación móvil quiero gestionar la información correspondiente a mis establecimientos. Pudiendo de esta forma crear, modificar y eliminar los mismos.

Brindar acceso a mis establecimientos

Como usuario tanto de la aplicación web, como de la aplicación móvil quiero poder habilitar a otros usuarios para que estos accedan a la información correspondiente a mis establecimientos.

Alta, Baja y Modificación de potreros

Como usuario dueño o con acceso a un establecimiento quiero poder agregar, quitar y modificar los potreros asociados al establecimiento.

Visualización de resultados para los distintos establecimientos

Como usuario dueño o con acceso a un establecimiento quiero poder visualizar el resultado de haber evaluado una medición para un potrero asociado al establecimiento.

Aplicación Web

Alta, baja y modificación de índices

Como usuario técnico quiero poder gestionar índices y las entidades asociadas al mismo. Esto implica el poder crear, borrar y modificar los datos de un índice, así como la fórmula de este.

Alta, baja y modificación de parámetros

Como usuario técnico quiero poder gestionar los parámetros asociados a los distintos índices. Esto implica el poder crear, borrar y modificar los distintos parámetros de un índice.

Alta, baja y modificación de tipos de datos

Como usuario técnico quiero poder gestionar los tipos de datos que reconoce el sistema. Es decir, quiero poder crear, borrar y modificar tipos de datos, enumerados, etc.

Manipulación de datos geográficos asociados a índices

Como usuario técnico quiero poder configurar los mapas tanto de vegetación potencial, como de pendientes asociados a los índices. Los antes mencionados son tenidos en cuenta a la hora de realizar el procesamiento de las medidas.

Manipulación de datos geográficos asociados a potreros y establecimientos

Como usuario dueño o con acceso a un establecimiento quiero poder gestionar la geolocalización de los establecimientos y potreros asociados para los cuales tengo acceso.

Recibir notificaciones de usuario verificado

Como usuario de backoffice quiero poder recibir notificaciones con las peticiones de usuario verificado realizada por usuarios móviles.

Habilitación de usuario verificado

Como usuario de backoffice quiero poder habilitar y deshabilitar usuarios verificados de la aplicación móvil.

Aplicación Móvil

Generación de medidas en campo

Como usuario dueño o con acceso a un establecimiento quiero poder cargar las mediciones asociadas a los potreros del establecimiento.

Envío de medidas

Como usuario dueño o con acceso a un establecimiento quiero poder enviar al servidor las medidas tomadas de mis respectivos establecimientos para que sean procesadas.

Solicitud de usuario verificado

Como usuario de la aplicación quiero solicitar la acreditación de mi usuario, para que este sea verificado.

Gestión de perfil de usuario

Como usuario de la aplicación quiero poder modificar los datos de perfil de usuario.

3.3.2.2. Requisitos No Funcionales

En esta sección se presentan los principales requisitos no funcionales del sistema.

Aplicación Móvil sin conexión

Un requerimiento primordial dada la naturaleza del relevamiento de datos en campo, es el funcionamiento sin conexión u offline. El dispositivo móvil debe poder funcionar por largos períodos sin conectarse con el servidor. Esto es debido a los lugares remotos donde se pueden llegar a recabar los datos.

El mínimo requerido en este aspecto es poder realizar una carga de datos, que esta sea almacenada para su posterior envío hacia el servidor. Esto implica que se almacenen los datos del índice que conciernen tanto a opciones que tendrá el usuario para cargar datos, así como la ayuda para esto, sumado a los datos de los establecimientos con sus respectivos potreros. Por otro lado, también se requiere la posibilidad de crear un nuevo establecimiento y nuevos potreros dentro de este o de los establecimientos ya existentes.

Procesamiento de medidas en segundo plano

El envío de los datos en el campo debe disparar el cálculo del índice, el remitente requiere solo la confirmación del guardado de los datos, para de esta forma continuar con otras tareas sin importar que el cálculo aún se este ejecutando, o a la espera de ser procesado. Por esto el procesamiento de medidas debe ser en segundo plano en el servidor, siendo transparente para el usuario móvil una vez los datos fueron enviados.

Capítulo 4

Diseño

En este capítulo se presentan las tareas de diseño desempeñadas a lo largo del proyecto. Entre ellas destacan el diseño de la arquitectura general del sistema, diseño de los componentes del sistema y por último se presenta el modelado final del dominio de trabajo.

4.1. Modelado del dominio

Como resultado del estudio del dominio, principalmente el estudio de las entidades que componen la realidad que maneja a diario el cliente, es que se vio la necesidad de modelar esta de alguna forma en la cual pudiese ser implementada mediante un sistema informático. Para ello es necesario abstraer lo representado hasta el momento en una planilla de cálculo electrónico, de una forma que sea posible almacenar la información en una base de datos. Para esta etapa de diseño es necesario tomar como punto de partida el resultado de la etapa de análisis, ya que el modelo de la solución debe replicar lo planteado en las planillas, y contemplar los requisitos funcionales y no funcionales que establece el cliente. Es por ello que se generó una estructura arborescente para representar, tanto la forma canónica del índice, así como las mediciones tomadas en campo, que luego son utilizadas para calcular el índice.

Esta estructura arborescente busca no solo solucionar la representación de los datos, si no que busca facilitar el acceso a ellos, ya que actualmente deben ser accedidos mediante funciones de las planillas de cálculo electrónicas. Como se detalla en el apartado [5.3](#).

En las figuras 4.1 y 4.2 se puede apreciar que el índice está representado como un árbol finitario, donde la raíz del mismo representa al índice, y de ella se desprenden los hijos (o entidades hijas) en los cuales se almacenan las entidades intermedias que componen al índice. A su vez, se cuenta con dos elementos más a nivel de cada entidad. Primeramente, para facilitar el cálculo del índice y el entendimiento del usuario, en vez de definir una única fórmula a nivel de índice, el mismo se define como fórmulas más cortas y sencillas a nivel de entidad. El cálculo del resultado final del índice se hace de forma recursiva, tomando los datos de cada entidad y los datos almacenados en los descendientes, o entidades en otras ramas del árbol en caso de ser necesario.

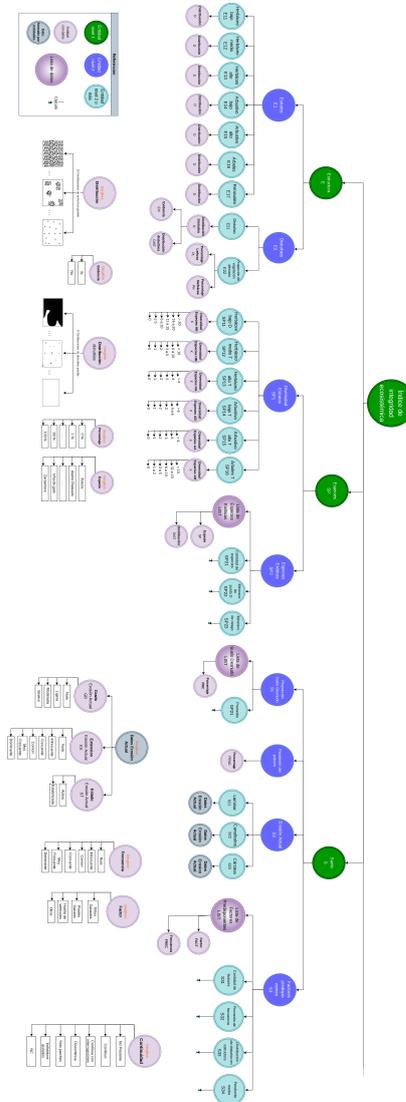


Figura 4.1: Estructura arborescente primera parte.

Por otra parte, también se puede configurar a nivel de entidad, el tipo de dato que se quiere almacenar, de esta forma los usuarios tienen permitido modificar los datos que son almacenados en cada entidad.

Todo esto refiere a la representación del índice. Las mediciones son una réplica de este árbol, pero en vez de contar con referencias a la fórmula correspondiente a la entidad y al tipo de datos que se puede almacenar en esa entidad, almacenan el dato cargado por el usuario y junto con el, el valor intermedio el cual será usado para realizar los cálculos del índice.

En lo que refiere al resto de esta subsección se hará referencia al árbol que representa al índice, quedando las mediciones para otras etapas del informe. De esta forma se cuentan con tres posibilidades utilizando el mismo tipo de dato (Entidad).

Raíz

Comienzo del índice, entidad en el que se almacena la fórmula inicial.

Hojas

Entidades finales, que no tienen hijos y en su mayoría solamente tienen datos cargados en campo.

Entidades Intermedias

Entidades intermedias del árbol, que tienen hijos y a su vez pueden tener datos.

Por como se construye esta estructura, cualquier entidad puede contener una fórmula asociada, a pesar de que no es un requisito que cuente con ella.

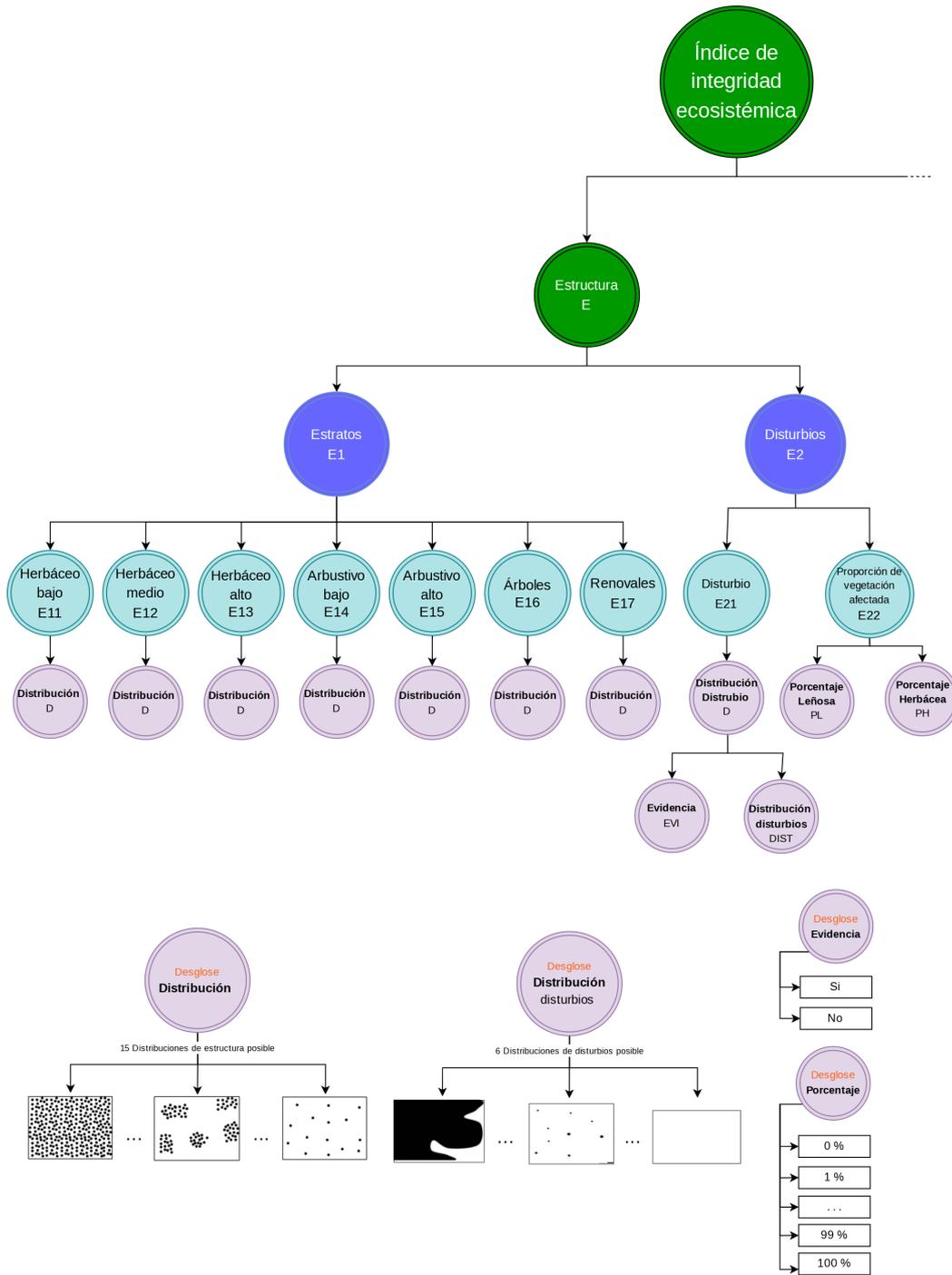


Figura 4.3: Estructura arborescente zoom-in.

Para facilitar la estructura arborescente y el funcionamiento de esta se presenta la figura 4.3 donde se puede apreciar una subsección de la estructura 4.1 y 4.2. En la parte superior, en color verde se encuentra la entidad que representa el índice. De esta entidad se desprenden varias sub-entidades a las

cuales se hará referencia como hijos de ahora más. En este caso, se aprecia como hija únicamente a la entidad **Estructura**. De la entidad **Estructura** se pueden apreciar dos entidades hijas, **Estratos** y **Disturbios**. Dentro de estratos hay siete entidades, las cuales cada una representa cada uno de los estratos de los cuales se releva la distribución. Estas entidades son finales (hojas del árbol finitario), debido a que no tienen entidades hijas asociadas. Sin embargo, como se puede ver tienen asociado un tipo de datos, la distribución. Esta distribución es el dato que contiene esa entidad y el cual indica que, a la hora de relevar datos en campo, allí son cargados datos y del tipo que en esta estructura es indicado.

Por otra parte, la otra entidad que se desprende de **Estratos**, es **Disturbios**. Dentro de los **Disturbios** se desprenden dos entidades más, **Disturbio** y **Proporción de vegetación afectada**. **Disturbio** está compuesta por dos entidades hijas, **Evidencia** y **Distribución de Disturbios**. Estas últimas son muy similares a lo antes mencionado en los estratos. Es decir, son entidades finales con un tipo de dato asociado. Por último, la **Proporción de vegetación afectada** cuenta con dos entidades hijas, las cuales son hojas del árbol cuyo tipo de dato un porcentaje, el cual se representa como se muestra en la referencia.

4.2. Arquitectura

Tomando como entrada toda la información generada en la etapa de análisis, se plantea como solución una arquitectura cliente-servidor, compuesta por cuatro módulos que se exponen en la figura 4.4.

4.2.1. Cliente Web

El objetivo de este componente es presentar las principales funcionalidades de backoffice al usuario de la forma más amigable posible dada la complejidad del dominio. Para lograr, que mediante su uso el usuario pueda realizar tareas de control, gestión de índices, parámetros, usuarios, establecimientos y potreros, así como la visualización datos y resultados.

El trabajo de diseño del cliente web abarca dos etapas principales. La primera de ellas se realiza en paralelo con la etapa de análisis del proyecto, a través del diseño de mock-ups. Estos diseños se le presentan al cliente para re-

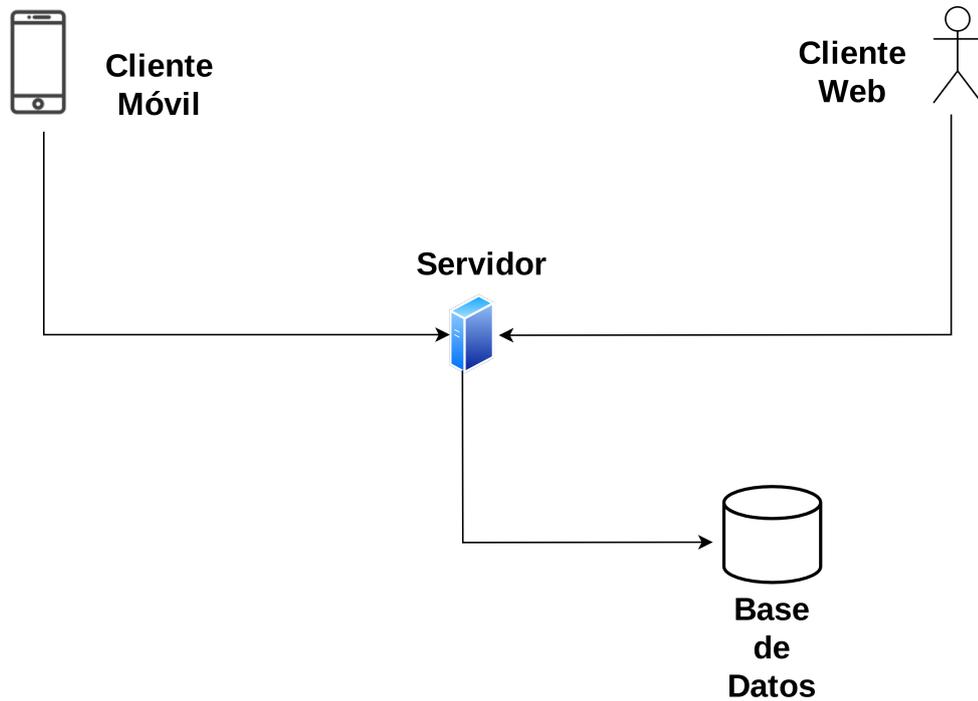


Figura 4.4: Arquitectura general del sistema.

levar requisitos, donde el foco está en comenzar a evaluar la interfaz de usuario (UI) de la aplicación y a su vez poder validarla con el cliente, evaluando aspectos tales como amigabilidad y experiencia de usuario. Las similitudes entre la UI representada por los mock-up y la UI implementada dejan ver la relevancia de esta etapa. En esta primera etapa el mayor desafío a enfrentar es el poder representar de una forma amigable y comprensible las páginas web capaces de configurar los datos, siguiendo la estructura planteada en las figuras 4.1 y 4.2. La segunda etapa, consiste en acompañar el desarrollo del resto de los módulos del sistema. El objetivo de este desarrollo en conjunto es que los distintos componentes del sistema se comuniquen de forma eficiente, usando interfaces en común. Esta etapa de diseño durante el desarrollo, se justifica por la alta dependencia entre los distintos módulos que implican tomar decisiones de diseño.

4.2.2. Cliente Móvil

4.2.2.1. Funcionamiento offline

El diseño de la solución se ve afectado en gran medida por el requisito funcional relevado en la sección 3.3.2. A continuación, se detallan los módulos resultantes de las decisiones de diseño tomadas para satisfacer esto.

Base de datos local

Se requiere tener acceso a todos los datos localmente, tanto datos de establecimientos y potreros, así como los recursos y configuraciones del índice. Todos estos son almacenados en la base de datos local del dispositivo móvil, manteniendo un paralelismo con el diseño de entidades y relaciones que se sitúa en la base de datos del servidor.

Los establecimientos y potreros pueden ser creados por el usuario en modalidad offline, para esto se almacenan en la base local para su posterior sincronización, permitiendo la carga de datos.

Se tienen los datos cargados por el usuario, estos se almacenan localmente para el funcionamiento offline, así como para persistir los datos de campo seleccionados por el usuario antes de ser enviados, ya que el usuario puede cerrar la aplicación y continuar la carga en otro momento. Estos datos son almacenados en la base de datos local en el momento que son seleccionados, permitiendo así el desentendimiento del guardado temporal por parte del usuario.

Las entidades y relaciones definidas difieren del modelo de base de datos del servidor 4.2.4, puesto que hay datos utilizados por el servidor que no son utilizados en el módulo móvil. Los datos pueden coincidir o no con el servidor, en función del escenario de sincronización como se verá a continuación.

4.2.2.2. Sincronización

La sincronización entre la base de datos local del dispositivo móvil y la base del servidor se realiza como una tarea disparada por el usuario. Durante la ejecución de la sincronización, el resto de las funcionalidades de la aplicación móvil se ven restringidas, para de esta forma asegurar la consistencia de los datos.

Se pueden tener tres escenarios en cuanto a la sincronización: pueden coincidir los datos existentes en el dispositivo móvil y el servidor, los datos del servidor pueden estar más actualizados y por último puede haber datos en el

módulo móvil que aún no se enviaron al servidor.

En el primer escenario se tienen ambas partes consistentes por lo que no se encuentran problemas. No así en el segundo escenario, donde podemos tener un cambio de configuración del índice que implique que los datos cargados por el usuario no sean válidos. Para evitar el problema de cargar datos que no sean válidos se debe realizar una sincronización siempre que haya un cambio en la configuración del índice que corresponda.

En el tercer escenario nos encontramos con datos cargados en el móvil, pero no en el servidor, nótese que pueden ser tanto datos de relevamiento de datos, como información correspondiente a establecimientos o potreros. En ambos casos el diseño presentado mantendrá un correcto funcionamiento offline. Cuando se realice una sincronización el servidor almacena los nuevos datos y realiza el procesamiento de estos en caso de corresponder.

La fecha y hora de la última sincronización es almacenada para poder transparentar al usuario este dato. Esto es particularmente útil en el caso de que se quiera agregar sincronización automática, es decir, no disparada por el usuario.

4.2.2.3. Carga de datos configurable

La carga de datos se tiene que amoldar a lo configurado para un determinado índice, desde las opciones que se tienen para un dato específico, hasta el método que se prefiere para ingresar este dato. Esto es necesario ya que para datos similares se puede tener una imagen de referencia en cada dato o en caso contrario puede seleccionarse únicamente por el nombre.

Es importante tener esto en consideración, ya que se requiere un dinamismo absoluto en la pantalla de carga de datos, esta pantalla se basa en módulos, para cada dato se muestran los módulos básicos y luego los que correspondan en función de la configuración.

Los módulos de información muestran al usuario información ubicada en el árbol de entidades (figuras 4.1 y 4.2), los módulos de navegación permiten moverse en este árbol, sea hacia la entidad raíz, la última entidad visitada, las entidades hijas o la siguiente entidad que no posee un dato cargado. A su vez, se encuentran los módulos de carga de datos, estos pueden permitir ingreso de datos más básicos como porcentajes o datos predefinidos. Las distintas configuraciones determinan estos detalles, así como la información que se muestra

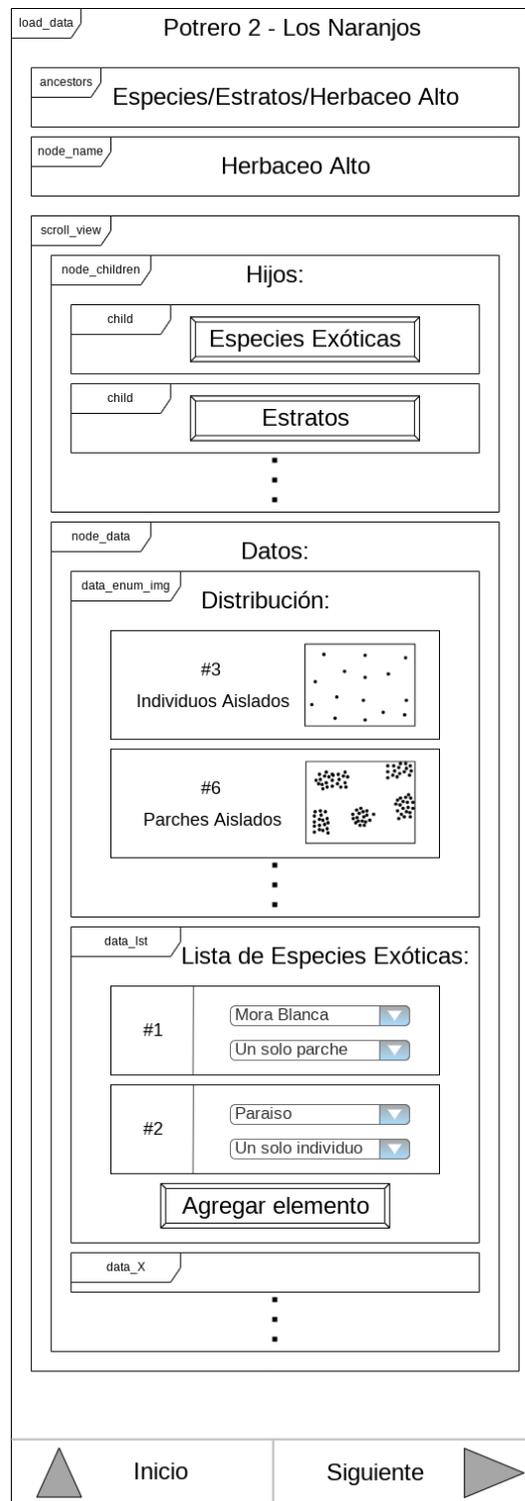


Figura 4.5: Diseño Módulos - Móvil

al usuario, y la ubicación de estos datos dentro del árbol de entidades.

En la figura 4.5 se muestra parte del diseño de los módulos mencionados.

4.2.3. Servidor

El componente servidor, como ya se mencionó en etapas anteriores es el encargado de realizar múltiples tareas de suma importancia para el funcionamiento del sistema.

Veremos al servidor como dos grandes componentes, estos componentes son:

- API
- Módulo de procesamiento

4.2.3.1. API

El componente API, o simplemente la API, es una API RESTFul que se encuentra desplegada en el servidor. Esta es la interfaz encargada de exponer al mundo (en este caso a los clientes web y móvil) los datos que son almacenados en la base de datos. Es el componente encargado de brindar las operaciones de acceso a datos para que estos puedan ser leídos, modificadas y/o borrados desde fuera del servidor.

4.2.3.2. Módulo de procesamiento

Por otra parte, en el servidor se encuentra un módulo de procesamiento de datos. El objetivo de este componente es realizar el cálculo del valor correspondiente al índice para cada uno de los potreros de los cuales se cuenta con datos almacenados en el servidor. Debido a la sensibilidad de los datos que se manejan y al hecho de que el procesamiento que se precisa realizar puede ser complejo (en función del índice configurado), es que se considera que este componente debe de estar aislado del resto de los componentes de backend.

4.2.3.3. Detalle de componentes

Como se puede apreciar en la figura 4.6 el servidor se encuentra dividido en varios módulos y a su vez cada módulo tiene sus respectivos submódulos.

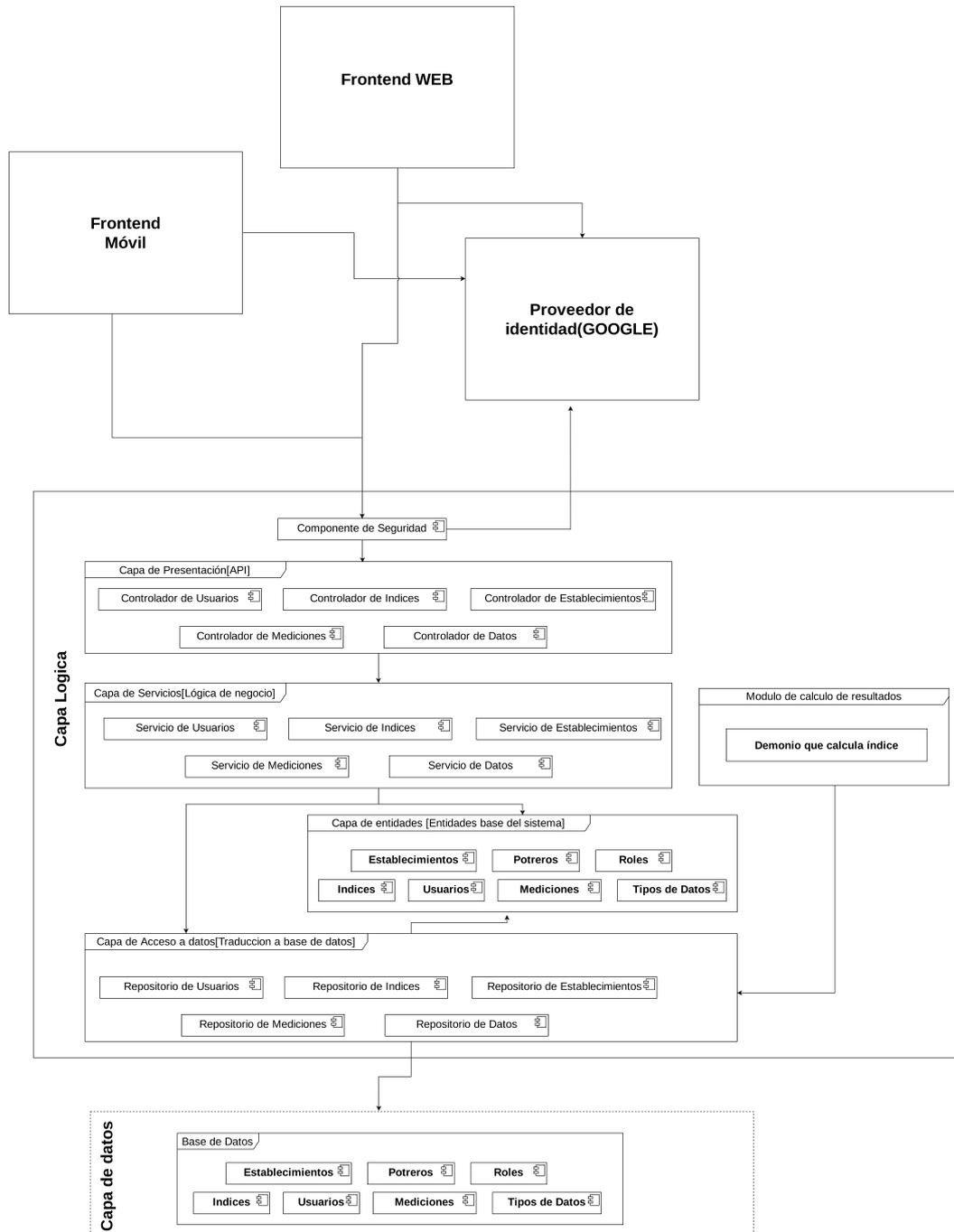


Figura 4.6: Vista en detalle servidor

Los módulos del servidor son los siguientes:

- Módulo de autenticación y autorización
- Módulo de procesamiento
- API
- Lógica de Negocio
- Entidades
- Acceso a datos

Módulo de procesamiento

En cuanto al módulo de procesamiento, no es necesario ahondar mucho más de lo que ya se hizo. Pero lo interesante, es entrar en detalle en los otros componentes.

Módulo de autenticación y autorización

Este módulo es el encargado de la seguridad a nivel de servidor. Su principal función es la de interceptar todas las peticiones HTTP que son recibidas. Cada vez que una petición es recibida en el servidor, esta es interpretada primero por este componente que se encarga de revisar que las mismas cumplan los requisitos de seguridad del sistema. Esto quiere decir que las peticiones cuenten con un cabezal HTTP de autorización (Authorization) y que el dato que contenga este cabezal sea un Bearer Token generado por Google para el usuario que está haciendo uso de la API. De esta forma se asegura la identidad del usuario, y en función de la identidad del usuario chequear que este cuente con los permisos pertinentes para realizar la acción solicitada en el sistema.

El flujo que lleva adelante este componente es el que se presenta en el diagrama de flujos del interceptor, figura 4.7. En donde se puede apreciar que todas las solicitudes son procesadas por este componente, el cual chequea los datos correspondientes a la identidad de usuario contra el IDP, validando así que las solicitudes fueron realizadas por usuarios habilitados.

API

Esta capa cumple la función de exponer entidades, servicios y operaciones de las capas inferiores al resto del mundo. Para ello expone lo que corresponda de las capas inferiores haciendo uso de un lenguaje común a cualquier cliente. Esto se debe a que expone estos recursos a través de una API RESTFul.

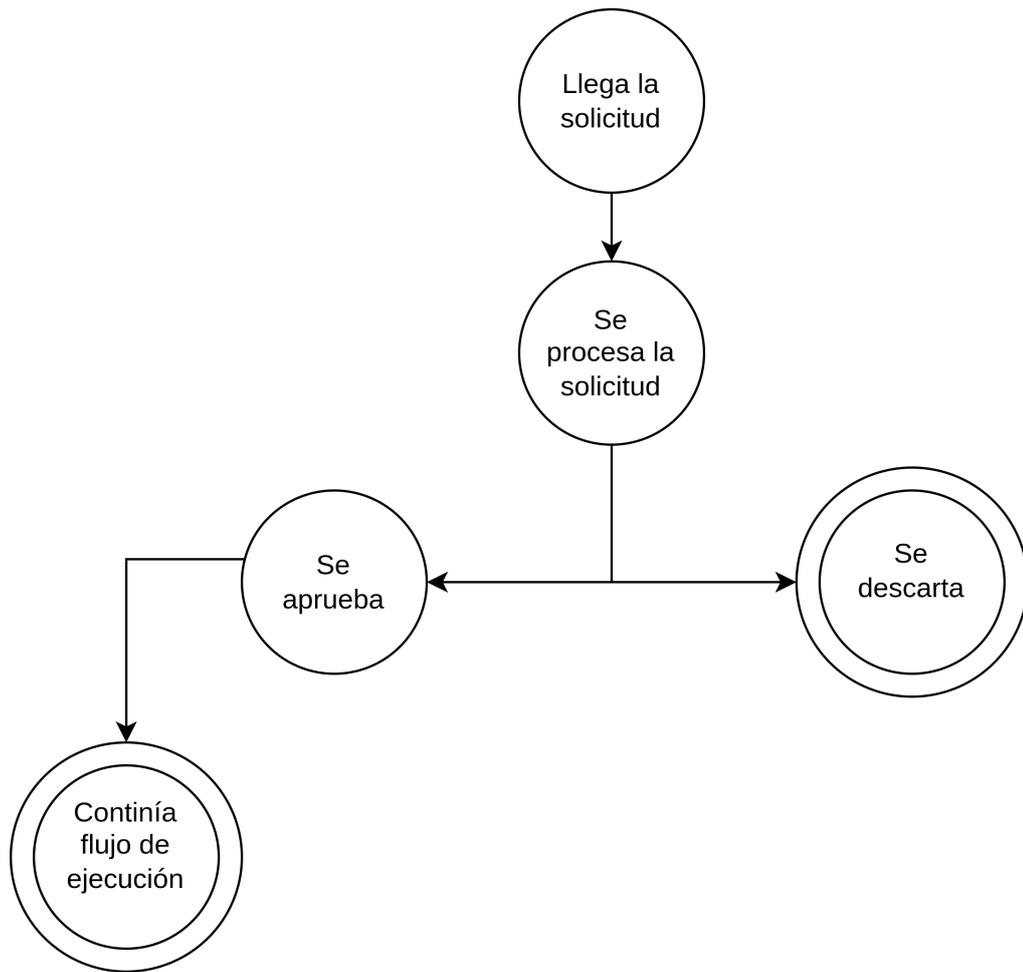


Figura 4.7: Flujo Interceptor

Lógica de negocio

Capa central de servicios, en esta se encuentra implementada toda la lógica de negocio (excluyendo el cálculo del índice). Está compuesta por distintos servicios. Cada servicio corresponde a un tipo de entidad o agrupación de estas, de forma tal que existe un determinado orden entre componentes. El objetivo de estos servicios es el de centralizar la lógica de negocio y brindarla para el resto del sistema.

Entidades del sistema

Esta capa está compuesta por las entidades que componen al sistema en sí. En ella se encuentran todos los objetos que representan la realidad dentro del componente backend.

Traducción a base de datos

Capa inferior del servidor, es la encargada de dos tareas fundamentales, traducir la información que maneja el servidor a su representación a nivel de base de datos, para de esta forma poder persistir toda la información que corresponda en la base de datos. Por otro lado, también es capaz de interpretar la información almacenada en la base de datos. Es por ello por lo que cuando el servidor necesita información almacenada previamente en la base, este componente es el encargado de traer esa información y traducirla para su uso.

4.2.4. Base de Datos

En esta subsección se presenta el diseño de la base de datos, el mapeo entre los elementos de la realidad y las entidades que en la misma se almacenan y una explicación detallada de cada una de estas entidades. Para comenzar a entender un poco más sobre las entidades de base de datos se plantea un enfoque bottom-up. En el cual se presentan primero las relaciones más relevantes de la realidad llegando finalmente a presentar el MER (Modelo Entidad Relación) de este proyecto.

4.2.4.1. Principales relaciones

Las principales relaciones que se dan en la realidad de este proyecto.

Usuario - Rol

La relación Usuario-Rol es 1 a 1..N. Esto quiere decir que un usuario va a tener asociado uno o varios roles como se muestra en la figura 4.8.



Figura 4.8: Relación Usuario-Rol

Establecimiento - Usuario

La relación Usuario-Establecimiento es doble, ya que es necesario diferenciar dos tipos de relación entre estas entidades. Motivo por el cual existen las siguientes:

- **1 a 1** Representa al usuario dueño del establecimiento.
- **1 a n** Representa a los usuarios con permiso de acceder a la información del establecimiento.

Estas relaciones se pueden visualizar en la figura 4.9

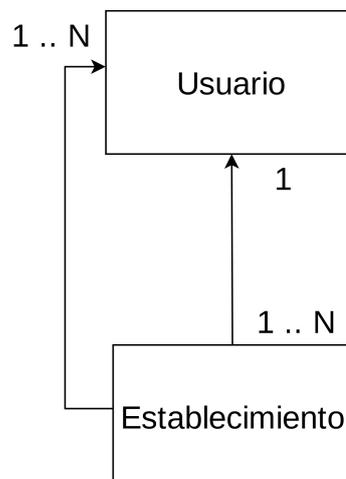


Figura 4.9: Relación Establecimiento-Usuario

Establecimiento - Potrero

Como fue planteado en capítulos anteriores, un establecimiento está compuesto por varios potreros como se muestra en la figura 4.10.

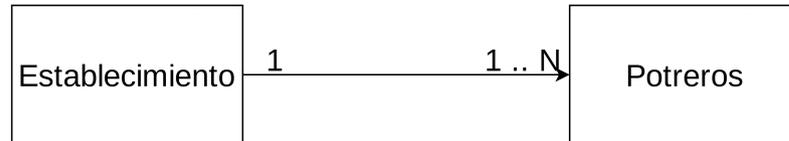


Figura 4.10: Relación Establecimiento-Potrero

DataTpye - DataTpye

Los tipos de datos tienen la posibilidad de ser configurables. Dentro de esas posibilidades se encuentra la posibilidad de que un tipo de dato representa una lista de datos. Para hacer esto posible, se permite tener la relación entre datos, como se ve en la figura 4.11.

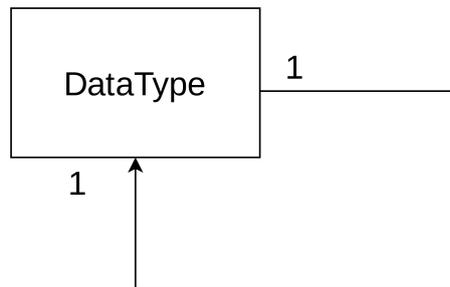


Figura 4.11: Tipo de datos Lista

DataTpye - CustomEnum - CustomValue

En el sistema se cuenta con tipos de datos enumerados. Este tipo de datos son los más interesantes a nivel de diseño, ya que para llevarlos a cabo se plantea la siguiente figura 4.12. En la figura 4.12 se puede apreciar que un tipo de dato enumerado tiene asociado un CustomEnum. Un **CustomEnum** es una entidad que representa al enumerado en sí. A su vez, esta entidad tiene asociada de 1 a n ocurrencias de la entidad **CustomValue**. La mejor forma de comprender esto es si se quisiera definir las vocales, para representarlas usando estas entidades habría que tener un CustomEnum denominado Vocales, el cual va a tener asociado a los CustomValues que representan a las vocales **A**, **E**, **I**, **O**, **U**.

Luego simplemente se define un tipo de dato en el cual se carguen como valores las vocales, y quedaría asociado este enumerado al DataType correspondiente.

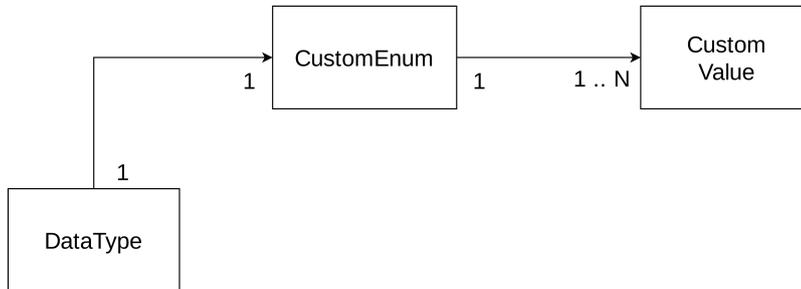


Figura 4.12: Tipo de datos Enumerado

Índices - Parámetros

Una de las principales entidades del sistema es el índice. Este se asocia con los distintos parámetros, como se puede ver en la figura 4.13.

En donde se puede apreciar que un índice se relaciona con múltiples parámetros. Estos parámetros son los encargados de almacenar la información adicional necesaria ya sea para cálculos, información sobre otras entidades e incluso configuraciones de sistema que deben estar asociadas al índice.

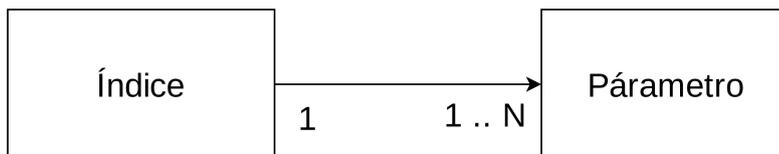


Figura 4.13: Relación entre índices y parámetros

Capítulo 5

Implementación

En este capítulo se presentan las principales tecnologías utilizadas para implementar los distintos módulos del producto desarrollado. Se menciona de los aspectos más destacados y las dificultades encontradas en la implementación de cada uno de los módulos. A su vez se presenta el trabajo de análisis sobre las planillas de cálculo electrónicas actuales para el desarrollo del módulo de cálculo del índice.

5.1. Arquitectura final

Con lo obtenido tras la etapa de diseño, se opta por entregar como solución una arquitectura en la cual, tanto el servidor como la base de datos se encuentran desplegadas en un mismo contenedor. Por otra parte, el cliente Web se encuentra en un contenedor distinto y la apk del cliente móvil fue entregada para ser distribuida por INIA.

5.2. Tecnologías utilizadas

Antes de entrar en detalle, se presentan las tecnologías utilizadas en cada componente mediante el diagrama de la arquitectura del sistema [5.1](#).

React

ReactJS [\[35\]](#) es una biblioteca JavaScript de código abierto desarrollada por Facebook. Se enfoca en facilitar la creación de interfaces de usuario. Es utilizado para crear aplicaciones SPA para la web y móviles.

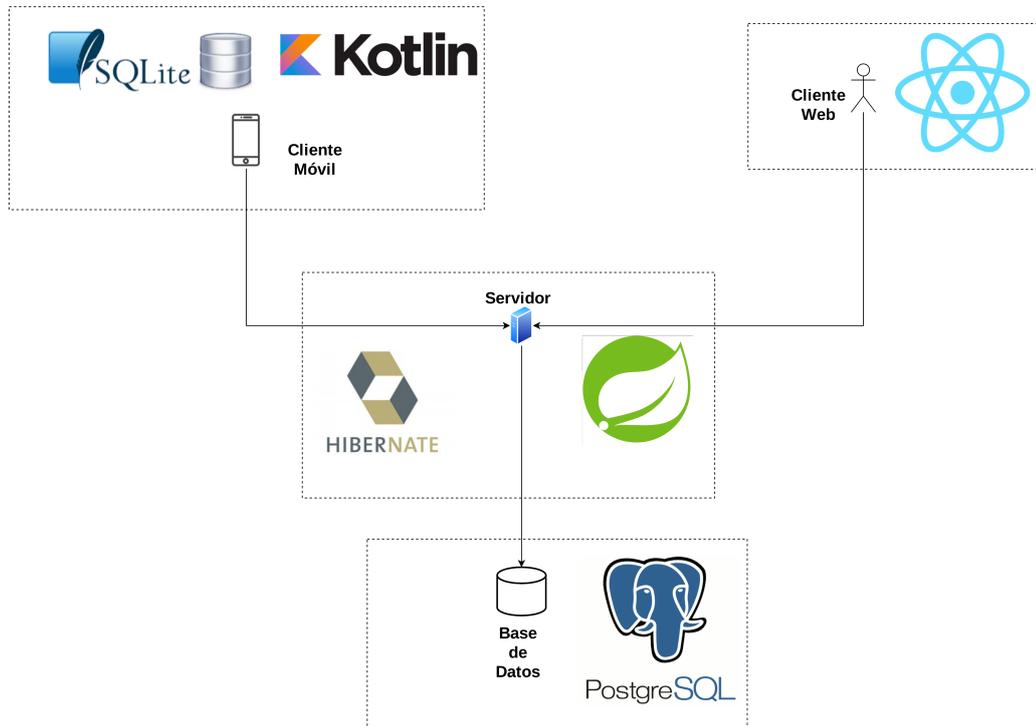


Figura 5.1: Arquitectura En Tecnologías

Kotlin [23] es un lenguaje de programación de código abierto creado por JetBrains. Este se ha popularizado debido a que se puede utilizar para programar aplicaciones Android. Es de tipado estático, sintácticamente está basado en Java y posee interoperabilidad con este, ya que es ejecutado sobre JVM.

Android

Android [2] es un sistema operativo inicialmente pensado para dispositivos móviles. Está basado en Linux y proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono.

Spring Boot

Spring Boot [41] es un framework que ayuda a crear aplicaciones independientes basadas en Spring. Spring toma de forma conveniente bibliotecas de terceros para comenzar la fase de desarrollo de forma rápida y efi-

ciente. La mayoría de las aplicaciones desarrolladas con este framework precisan muy poca configuración. A su vez, permite la creación de aplicaciones tanto web, como aplicaciones de consola. Es por estas razones y la experiencia con la herramienta en el equipo de trabajo que fue la tecnología seleccionada para el desarrollo del backend del sistema.

Maven

Apache Maven [28] es una herramienta que brinda manejo de paquetes, configuración de un proyecto en todo su ciclo de vida, desde la fase de compilación, pasando por el empaquetado y la instalación de mecanismos de distribución de librerías, para que puedan ser utilizadas por otros desarrolladores y equipos de desarrollo. También contempla temas relacionados con la integración continua, para poder realizar la ejecución de pruebas unitarias y pruebas automatizadas, pruebas de integración, etc.

Gradle

Gradle [15], es una herramienta que brinda manejo de paquetes y automatización de compilación. Es una herramienta de código abierto que se encuentra centrada en la flexibilidad y configurabilidad. Los scripts de compilación de Gradle se escriben utilizando Groovy o Kotlin DSL. Tiene una gran flexibilidad y permite hacer uso de otros lenguajes, no solamente Java (como sucede con Maven). También cuenta con un sistema de gestión de dependencias muy estable. Es el sistema de compilación oficial para Android y cuenta con soporte para diversas tecnologías y lenguajes.

PostgreSQL

Para la implementación de la base de datos se utiliza PostgreSQL [31]. PostgreSQL es un sistema de base de datos relación de objetos Open Source que cuenta con más de 30 años de desarrollo activo, se opta por esta tecnología por su confiabilidad, robustez y rendimiento.

SQLite

En la implementación de la base de datos móviles se utiliza SQLite. Un gestor de base de datos reducido, enfocado en dispositivos de pocas

capacidades de hardware, en nuestro caso un dispositivo móvil. SQLite brinda herramientas para el desarrollo que facilitan el uso de la base de datos móvil, siendo ésta un paradigma distinto al de las convencionales, de forma que en el desarrollo se mantengan estrategias y herramientas similares.

GitLab

Antes de entrar en detalles de que es Gitlab es necesario definir el término controlador de versiones [9]. Controlar versiones es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo. Una vez introducida la definición de controlador de versiones, es necesario introducir el termino Git (Sistema de control de versiones).

De esto se llega a que GitLab [14] es una herramienta de ciclo de vida y repositorio de Git.

La selección de las tecnologías no fue una decisión puntual. Por el contrario, para llegar a este stack tecnológico se tomaron en cuenta varios aspectos, tales como:

- Conocimiento previo del equipo
- Distribución de módulos a desarrollar
- Tendencias actuales
- Arquitectura del Sistema y requerimientos

Una vez contemplados los aspectos anteriores, se comienza la etapa de evaluación del stack tecnológico. El objetivo de este proceso es el de realizar pruebas de concepto de funcionalidades críticas, para evaluar su factibilidad y futura implementación. Entre estas funcionalidades destacan:

- Login
- Funcionamiento offline
- Manejo de estructura de datos arborescente

Mediante estas pruebas de concepto, no solo se evalúan las tecnologías, si no que se evalúa tanto el equipo como la velocidad de trabajo de este. Permitiendo de esta forma comenzar a plantear el alcance del proyecto.

En esta etapa se realizan varias pruebas de concepto, entre las cuales destacan:

- Aplicación móvil en React Native
- Aplicación móvil con funcionamiento offline
- Servidor con Base de datos y API que expone interfaces para interactuar con esos datos
- Manejo de archivos e imágenes a nivel de API

Gracias a esto, se pudo mitigar varios riesgos técnicos (sobre estos riesgos se hará mayor énfasis en la sección de Gestión 6). Pero como aspecto destacable, se optó por hacer un cambio de tecnología para el desarrollo de la aplicación móvil en concreto, ya que inicialmente esta iba a ser desarrollada en React Native (Para IOS y Android). Pero luego de realizar algunas pruebas de concepto y validar esto con el cliente, se descartó esta posibilidad y el equipo decidió desarrollar la aplicación móvil en Kotlin para Android.

5.3. Carga de datos

Uno de los objetivos del proyecto es el desarrollo de un prototipo, que permita evaluar la viabilidad de un sistema el cual permita gestionar el índice. En esta sección se explica en detalle el proceso previo al desarrollo de este prototipo.

Para comprender la necesidad del cliente, es necesario poder llevar el cálculo del índice, desde la realidad actual a un sistema informático. Para ello, es necesario tomar como entrada las planillas de cálculo electrónicas que se usan actualmente, y obtener la misma salida en el sistema, mediante la configuración de éste aplicando el modelo anteriormente descrito.

En principio se comienza por una sección del índice con el objetivo de validar el resultado parcial, agregando paulatinamente distintas secciones apuntando a construir el resultado final.

Una vez hecho ese análisis se procede a ver los datos fijos que hay configurados en algunas de las hojas de la planilla, debido a que estos datos

posteriormente son configurados como parámetros de sistema o como valores desplegables para los enumerados. Entre estos datos destacan algunos como:

- Pesos de ciertos valores en sus respectivas fórmulas.
- Enumerados y sus respectivas descripciones para desplegables.
- Nombres de potreros y numeración de estos.

Una vez comprendidos estos datos y el funcionamiento general de las planillas de cálculo electrónico, se procede a interpretar la fórmula. Este trabajo tuvo como resultado uno de los principales módulos desarrollados en el servidor. El módulo de procesamiento del índice, módulo en el cual se implementó un componente que es capaz de interpretar fórmulas, que respetan cierta sintaxis y que a su vez es posible configurar por parte de los usuarios. Permitiendo así que el índice pueda ser modificado.

5.4. Aplicación Web

Se implementó una aplicación web utilizando la biblioteca de JavaScript React, descrita anteriormente en la sección de tecnologías utilizadas. Para el set up inicial de la aplicación se utilizó el inicializador oficial provisto por React, CRA (“create react app”) [10]. Mediante este inicializador se genera una estructura inicial con las herramientas y librerías necesarias para generar una aplicación React.

5.4.1. Interfaz de usuario

Para facilitar el diseño de la interfaz de usuario, se usaron los componentes provistos por la librería Material UI [27], la cual provee distintos componentes pre-diseñados.

Con este diseño se buscó lograr una interfaz de sitio web de administración (BackOffice), donde la pantalla principal tiene dos componentes principales, la barra de navegación (o navbar) y la barra lateral (o sidebar) como se puede ver en la figura 5.2. Para la construcción del diseño de esta interfaz gráfica se tomaron elementos de una plantilla de tablero (dashboard) provista por Material UI llamada Minimal Dashboard [26].

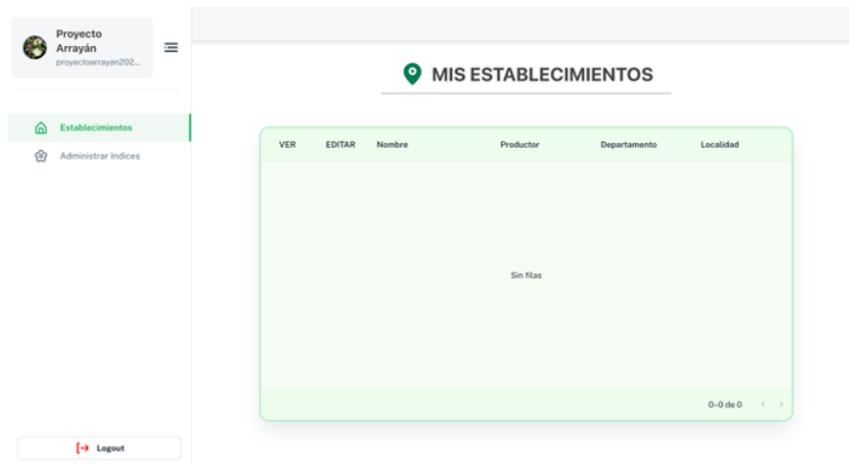


Figura 5.2: Pantalla principal

5.4.2. Comunicación con Backend

La comunicación con el Backend se realiza a través de la API expuesta por el mismo, para esta comunicación se utilizó el cliente HTTP axios [6].

Adicionalmente para agregar funciones de almacenamiento en caché y sincronización de las respuestas de la API en el estado de la aplicación, se utilizó la biblioteca React Query [34].

Una de las principales justificaciones del uso de esta biblioteca, fue el poder mejorar la performance gracias a la capacidad de almacenar en caché ciertos datos de gran volumen.

5.4.3. Autenticación

La autenticación de la aplicación se realizó utilizando cuentas de Google, solicitando al usuario que ingrese con su cuenta de Google para autenticarse en la aplicación. Esta autenticación se implementó utilizando el componente “React Google Login” [33]. A este componente se le realizaron ajustes de estilo para lograr el diseño visual deseado (ver figura 5.3).



Figura 5.3: Google login

5.4.4. Funcionalidades

Las funcionalidades provistas por la aplicación web se dividen en dos módulos principales.

5.4.4.1. Administración de establecimientos

Este módulo permite al usuario administrar los establecimientos que tiene asociados. Permitiendo realizar la edición de los datos principales de cada establecimiento, crear y editar sus potreros. Para cada potrero se puede visualizar las medidas del índice que se hayan generado.

5.4.4.2. Administración de índices

Dentro de este módulo se realiza la gestión de las distintas versiones del índice. Se listan las distintas versiones del índice que existen y dentro de cada índice se puede ver y editar los datos principales de cada uno.

Además de esto, se puede ver, agregar y editar los elementos principales que definen el mismo:

- Entidades
- Parámetros
- Tipos de dato
- Enumerados

Dentro de este módulo destaca la funcionalidad de visualizar la estructura arborescente de entidades del índice, a través de una interfaz gráfica similar a la representación de un sistema de archivos. Esta representación ayuda a un mejor entendimiento de la estructura del índice (ver figura 5.4).

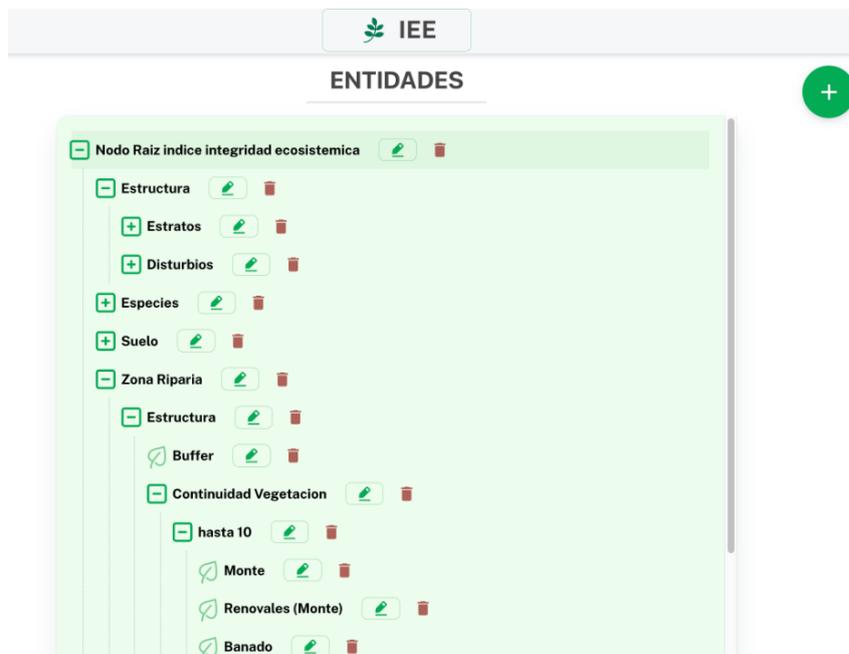


Figura 5.4: Visualización de estructura de entidades

5.5. Aplicación Móvil

5.5.1. Base de datos local sin ORM

Es implementada en SQL lite bajo la biblioteca Android Room [3]. Esta implementación no cuenta con un ORM, es por ello por lo que las relaciones se realizan usando tablas de relación, tal como se recomienda por los creadores de la biblioteca.

La falta de un ORM conlleva un trabajo extra, sobre todo a la hora de realizar acceso a datos de una tabla que están relacionados con datos de otras tablas. A esto, se le suma la dificultad de llevar estos datos (extraídos de una o varias tablas) a objetos en memoria. Estos objetos en memoria son usados tanto para la interacción con el usuario, así como para comunicarse con el servidor.

5.5.2. Módulos de carga de datos

Cada elemento que puede incorporarse de forma dinámica en función de la configuración, es implementado como un fragmento Android. De esta forma se construye la pantalla de carga de datos a medida que se navega por el índice.

Los distintos fragmentos reciben por medio de su constructor la información necesaria para la carga de datos. Por ejemplo, en el caso de un enumerado se recibe el nombre del dato a ser cargado y la lista de valores posibles para este.

Un caso particular es el de las listas, las listas son fragmentos con un nivel más de dinamismo, ya que el usuario puede agregar tantos valores como desee. Esto último provoca que tanto la interfaz gráfica, como el modelo lógico relacionado deban soportar un tamaño desconocido de elementos.

Cada fragmento tiene la responsabilidad de “escuchar” los cambios en los datos ingresados y persistir localmente cuando corresponda. Por lo cual para muchos casos la interfaz limita el ingreso de los datos, de tal modo que siempre se debe persistir esta información. No obstante, hay casos como la lista de correlación donde se deben seleccionar dos valores, o los porcentajes (ver figura 5.5) donde el valor debe ser un número entre cero y cien. Para estos últimos casos se realiza una validación antes de persistir el valor seleccionado.

Proporcion de suelo desnudo		
Datos:		
Lista de Porcentajes	4.75	%
Valor 1	5	%
Valor 2	4.5	%
Valor 3		%

AGREGAR VALOR

Figura 5.5: Aplicación - Lista de porcentajes

5.5.3. Listas correlacionadas

Por listas correlacionadas se hace referencia a todos los datos que requieren dinamismo en cuanto a cantidad de valores de las listas, y a su vez se componen de dos valores. A nivel lógico son dos listas independientes. Sin embargo, la semántica dada correlaciona los datos en función de su posición en la lista.

En la figura 5.6 se puede ver como el valor uno se compone por dos elementos, los cuales son almacenados con el modelo lógico descrito.

Lista Especies Exoticas		
Valor 1	Cirsium vulgare	▼
	Pocos parches	▼
Valor 2	Mora Blanca	▼
		▼

AGREGAR VALOR

Figura 5.6: Aplicación - Listas correlacionadas

5.5.4. Estructura arborescente de entidades

A nivel de interfaz gráfica se tiene que cada entidad del índice es una instancia de la misma pantalla de carga de datos, con la diferencia de que cada instancia se realiza con el nombre, los hijos y los datos que corresponda. Esto tiene como resultado una pantalla compleja, pero al mismo tiempo dinámica, resolviendo todas las casuísticas requeridas.

Lógicamente el árbol se representa almacenando el identificador de la entidad padre, debido a las limitantes del ORM anteriormente mencionadas. Por lo que la navegación hacia la entidad padre resulta trivial, pero al obtener los hijos se requiere una búsqueda más compleja.

Debido a la representación arborescente del índice, inspirada por la naturaleza de este, es necesario realizar recorridas recursivas. Algún caso interesante se detalla en la siguiente sección.

5.5.5. Ayuda en carga de datos

Con el objetivo de ayudar al usuario en la navegación por el árbol de entidades, y facilitar la carga de datos en sí, se implementan las funcionalidades que se detallan a continuación.

Dentro de estas funcionalidades se encuentran:

- Avance total
- Sitios en donde faltan datos
- Flechas de navegación

El avance total se muestra en una barra por cada potrero. Esta barra avanza a medida que se cargan datos. Y el envío de los datos al servidor se permitirá si solo si esta barra está completa. Para guiar al usuario hacia donde faltan datos, se cuenta con un código de colores en los botones de la navegación hacia entidades hijas. En caso de que el color sea verde, quiere decir que no restan datos por cargar, de lo contrario la tonalidad de rojo o anaranjado indican que tantos datos restan por cargar. Esto último se puede apreciar en la figura 5.7.

Los dos items mencionado se basan en un cálculo de progreso a nivel de entidad, el cual es actualizado cada vez que se carga un nuevo dato, viéndose afectada la entidad que posee este dato y todos sus ancestros en el árbol de entidades.



Figura 5.7: Aplicación - Colores de referencia

Se cuenta con dos flechas de navegación. Una hacia arriba y la otra hacia la derecha. Pulsando la flecha hacia arriba se descarta todo el avance dentro del árbol de entidades y se vuelve a la pantalla de carga de datos de la entidad raíz. El objetivo de esto es que el usuario no tenga que presionar reiteradas veces hacia atrás.

Cuando se pulsa a la derecha se navega hacia la siguiente entidad que no tenga datos cargados, y que a su vez no se haya sido visitado previamente. De esta forma se puede acceder rápidamente a los datos que restan cargar, sin necesidad de navegar manualmente por el árbol de entidades.

La implementación de esta última funcionalidad se basa en una estructura arborescente análoga al índice, la cual marca si se deben visitar o no las entidades. Esto se hace mediante una búsqueda recursiva sobre el árbol, la cual da como resultado la entidad objetivo, entidad a la cual se navega automáticamente.

5.5.6. Sincronización

La sincronización desde el servidor hacia la aplicación es ejecutada por el usuario en la pantalla principal, ésta inhabilita el resto de las funcionalidades, de forma de no permitir que se presionen los botones que las disparan.

El envío de los datos cargados de un potrero, se realiza para cada potrero siempre y cuando se hayan cargado todos los datos del mismo. La sincronización requiere la transformación de los datos recibidos y enviados, ya que los

modelos lógicos cuentan con diferencias entre sí.

Para la prueba de concepto implementada, no se vio conveniente la automatización de la sincronización, de igual forma se experimentó con tareas programadas mediante JobService de java, obteniendo buenos resultados.

Es por esto por lo que se concluye que el diseño lo soporta y no sería costoso agregar la funcionalidad.

5.5.7. Potreros en el mapa

En el apartado de potreros se puede seleccionar la vista del mapa como alternativa al listado por defecto. En esta vista el usuario es geolocalizado mediante el Gps del dispositivo móvil. Se espera que el usuario se encuentre en el establecimiento cuando acceda a este apartado, por lo que automáticamente se pone el foco del mapa en su ubicación. De todas formas, tiene la posibilidad de navegar libremente, pudiendo volver siempre a su ubicación.

Los potreros se encuentran representados como polígonos en una capa sobre un mapa base, siendo este último obtenida de internet. Por lo que de no tener conexión se muestran los polígonos sin mapa base. Este inconveniente se investigó, y puede ser solucionado guardando las teselas de la región, esta funcionalidad es provista por la biblioteca utilizada, la cual será presentada a continuación [24].

La biblioteca utilizada para implementar esta funcionalidad es MapBox Android SDK [25]. Como principales ventajas se encuentra la integración en el código, ya que está implementada de forma nativa para kotlin. La biblioteca utilizada dispone de gran cantidad de funcionalidades y buenas perspectivas de futuro.

Estas teselas son almacenadas como texto en formato Json a nivel de potrero. De forma es que los potreros son directamente entregados a la biblioteca. Esta biblioteca brinda la posibilidad de agregarlos a una capa y sobreponer esta capa al mapa base.

5.6. Backend y Servidor

En esta sección se explicará el funcionamiento del servidor. Luego se hará énfasis en la implementación de las principales funcionalidades desarrolladas, los desafíos afrontados y las distintas soluciones propuestas.

La tecnología utilizada para desarrollar la aplicación de backend fue Spring Boot. Al hacer uso de esta tecnología para el desarrollo por lo ya visto, se genera un artefacto desplegable (archivo .jar). Este desplegable al ejecutar, cuenta con un servidor embebido [4] en el cual ejecuta la aplicación. Dicho esto, y como se presentó en el capítulo de diseño, la arquitectura del servidor, esta conformada por varios componentes las cuales fueron desarrolladas haciendo uso de algunas de las tecnologías mencionadas. Estos módulos están dentro de una única aplicación que ejecuta en el servidor.

5.6.1. Componentes del servidor

Módulo de autenticación y autorización

Esté módulo fue implementado haciendo uso de bibliotecas de Spring, las cuales dan la posibilidad de interceptar todas las peticiones http que son recibidos en la aplicación de backend. El objetivo de este módulo es analizar las peticiones que llegan al servidor, analizar el Header de autorización y consultar al IDP (identity provider o proveedor de identidad) para validar la identidad de quien realiza la solicitud. Además de las funcionalidades por defecto provistas por estas bibliotecas, se utilizó la información obtenida por este módulo para asociar establecimientos a usuarios, validar que estos tengan permiso para acceder a distintas funcionalidades del sistema entre otras funcionalidades.

API

Este módulo tiene como objetivo el de exponer recursos y funcionalidades. Para ello se utilizan anotaciones de spring que exponen una API. A grandes rasgos en estos elementos llamados controladores lo que se implementa son funciones. Estas funciones retornan una respuesta HTTP en la cual le devuelven al usuario que realizó la solicitud la respuesta correspondiente a lo que se solicita. Estas funciones mencionadas hacen uso de las funcionalidades de capas inferiores para poder retornar lo solicitado. Además de los controladores, en esta capa se encuentran las interfaces de los recursos que expone la aplicación. Estas interfaces son tanto de salida como de entrada al servidor y es el lenguaje que interpreta la API.

Lógica de Negocio

El módulo de la lógica de negocio es el encargado de realizar todos los cálculos, recabar la información necesaria y su posterior procesamiento. Este módulo está compuesto por objetos denominados servicios, los cuales como bien dicen su nombre, proveen servicios para el resto de los módulos del backend. Estos servicios además de realizar los cálculos y procesamientos correspondientes, tienen la capacidad de traducir las entidades que son explicadas en el próximo ítem y las interfaces que se plantean a continuación.

Entidades

La capa de entidades está compuesta por los objetos que representan a nivel de aplicación las entidades en la base de datos. Estas entidades son las que interpretan los servicios y son usadas para realizar los cálculos correspondientes.

Acceso a datos

Este módulo es el encargado de realizar las conexiones a la base de datos, traducir de forma bilateral la información entre la base de datos y la aplicación.

Utilidades

Este módulo brinda operaciones comunes y necesarias para facilitar procesamientos al resto de los componentes de backend.

Módulo de procesamiento

El módulo de procesamiento es el encargado de realizar el cálculo del índice. Es un módulo independiente que ejecuta periódicamente en búsqueda de ejecuciones de cálculo de IIE sin procesar aún. Para la implementación de este módulo, se utilizó un componente provisto por Spring que ejecuta de forma independiente al resto de los módulos en el backend. El objetivo de esto es independizar el procesamiento del resto de los servicios de backend. Siendo dos hilos de ejecución aislados uno del otro. De esta forma se independizan los recursos usados para cálculos, procesamientos, etc. Esto se hizo de esta forma, ya que el módulo de procesamiento realiza cálculos complejos. Además, por la estructura de los datos usada, precisa

realizar accesos a base de datos en funciones recursivas, lo que conlleva a un consumo mayor de recursos.

5.6.2. Base de datos

En los siguientes apartados se detallan las partes del modelo lógico que implementan la base de datos. Este modelo fue creado por hibernate como se menciono en este capítulo.

5.6.2.1. Configuración del Índice y sus componentes

En la figura 5.8 están representados tanto el índice como sus respectivos componentes. Entre sus componentes destacan:

- **Nodos:** Entidad que representa los componentes del árbol finitario que representan a la totalidad del índice.
- **Parámetros:** Parámetros del índice en los cuales se almacena información correspondiente tanto al cálculo del índice, como a otras configuraciones necesarias para el sistema.
- **Datos:** Por último, se presenta el modelo lógico usado para la implementación de los tipos de datos. Esta estructura está compuesta por la entidad principal `data_type`, la cual representa al tipo de dato en sí (pudiendo ser un enumerado, una lista, o simplemente un número, o texto). En caso de ser una lista o un enumerado es que se cuenta con estructuras auxiliares que almacenan la información para la representación.

A su vez, también son destacables las entidades de medición y ejecuciones. La entidad de mediciones será explicada en la siguiente subsección. En cuanto a las ejecuciones, es la entidad en la cual se almacena la información necesaria para que el módulo de ejecuciones pueda realizar los cálculos del índice.

5.6.2.2. Medidas con sus datos asociados y configuración de enumerados

La figura 5.9, presenta en verde la representación de los nodos de medidas, estos son los que están relacionados a los datos en las distintas representaciones, almacenando lógicamente los datos cargados en el campo.

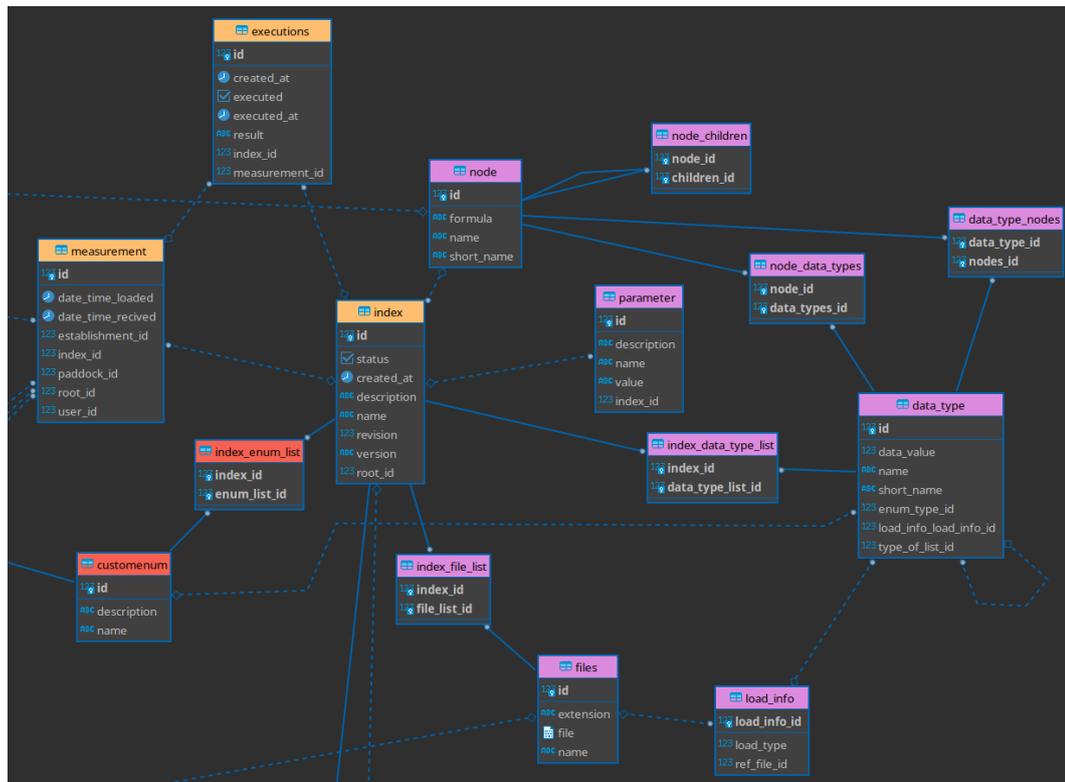


Figura 5.8: Configuración del Índice y sus componentes

La estructura arborescente que almacena los datos de las medidas se construye en función de la estructura del índice, por lo que es un espejo de esta con la diferencia que contiene los valores de los datos. Cada medida se compone de una colección de datos con la estructura expuesta anteriormente y se relaciona con el índice al cual corresponden los valores cargados.

En rojo tenemos la representación lógica de los valores posibles que puede tomar un enumerado configurable, implementando la relación definida en la etapa de diseño.

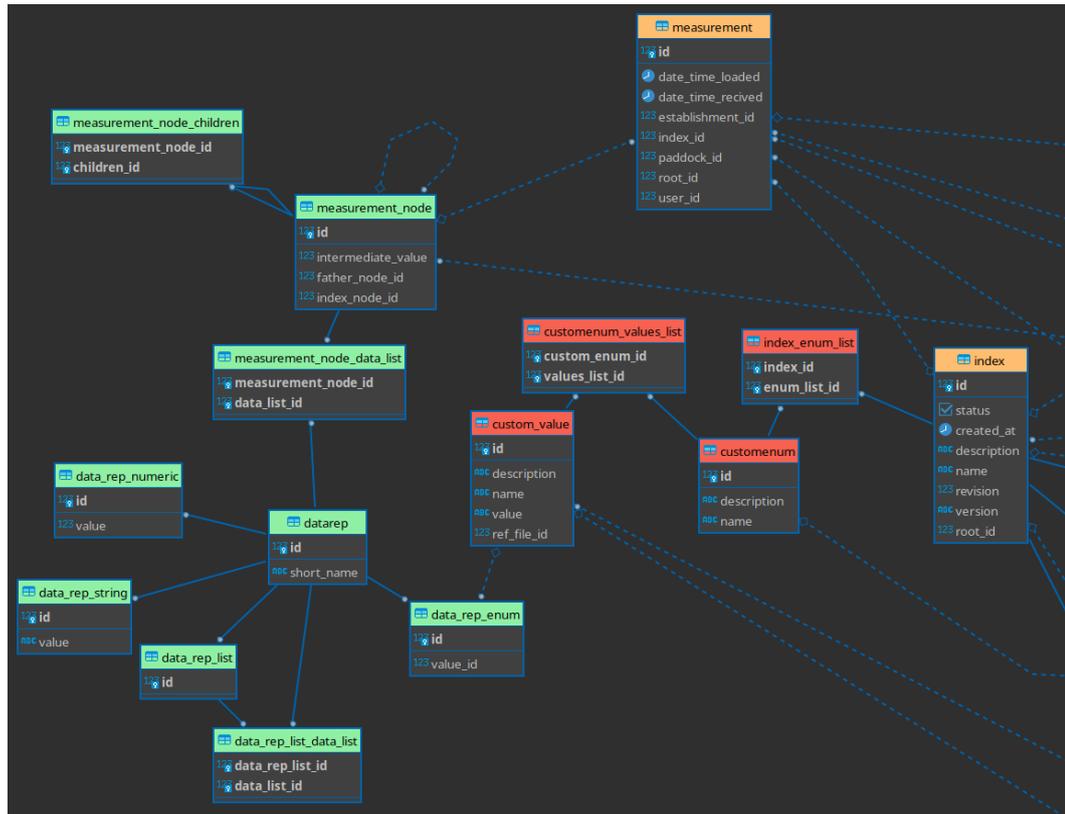


Figura 5.9: Medidas con sus datos asociados y configuración de enumerados.

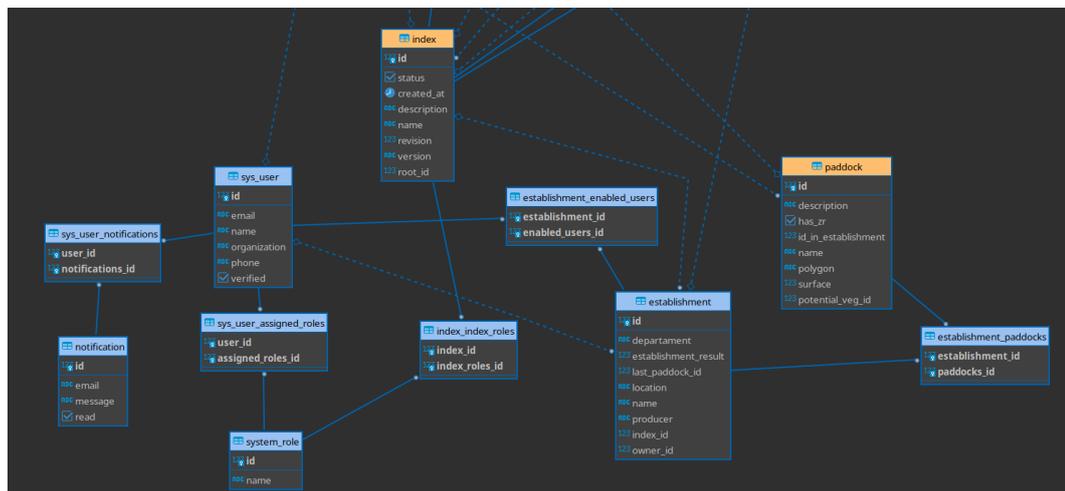


Figura 5.10: Medidas con datos asociados y enumerados

5.6.2.3. Usuarios, roles y establecimientos

En la figura 5.10 se presentan los establecimientos, los potreros, los usuarios y los roles. Estas representaciones lógicas permiten almacenar la información correspondiente a los actores del sistema, los roles que estos tienen, así como los objetos de estudio del índice (establecimientos y potreros).

5.6.2.4. Modelo de representación lógica

En las figuras 5.8, 5.9 y 5.10 se presentaron las distintas subsecciones en las que puede ser subdividido el modelo de representación lógica.

En la figura 5.11 se presenta la visión general del modelo de representación lógica.

5.6.3. Evaluador de Fórmulas

En la subsección anterior se presentaron los distintos módulos que componen el servidor. Sin embargo, hay un componente que al principio del proyecto se pensó iba a ser un componente sencillo, sin embargo a medida que se avanzó en el trabajo de relevamiento de requerimientos, este se terminó convirtiendo en un módulo con alto poder de procesamiento, al punto que este módulo ejecuta de forma independiente al resto.

Para entender el funcionamiento de este módulo es necesario introducir varios conceptos que fueron parte del proceso.

Análisis de Planillas

El primer paso fue el de analizar las planillas de cálculo electrónicas usadas actualmente por el cliente. Este proceso llevó a generar axiomas, los cuales son las funciones que el backend es capaz de interpretar para calcular el IIE. Para esto, dentro del backend hay un componente capaz de interpretar texto, identificando así las distintas funciones que brinda. Este componente está encargado de interpretar las fórmulas definidas para cada índice, y en función de los datos almacenados en la medida y la fórmula definida en el índice realiza los cálculos correspondientes.

Procesamiento de fórmulas

Como se planteó en el ítem anterior, con base en el análisis de las planillas de cálculo electrónicas se desarrolló un componente capaz de interpretar

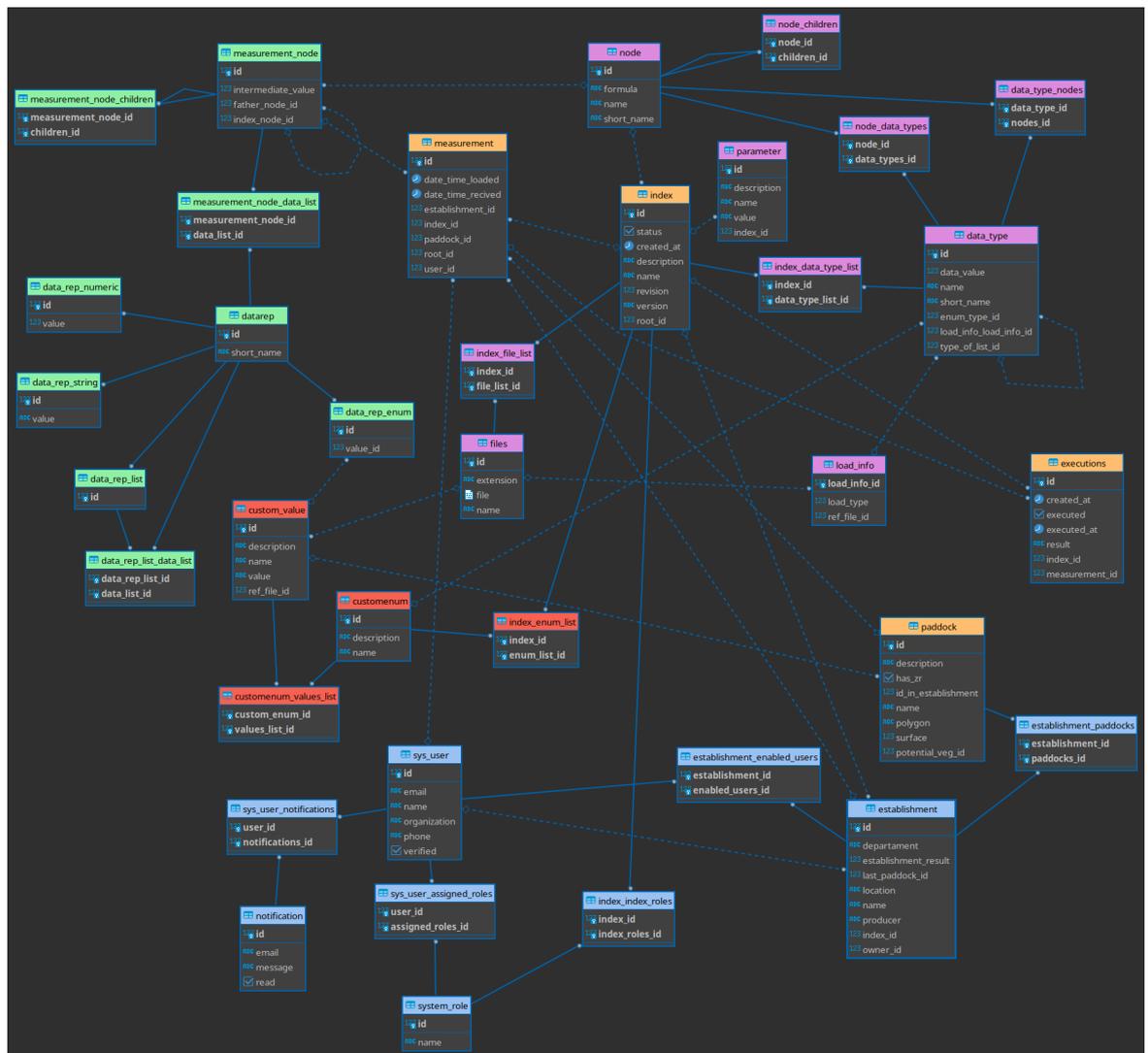


Figura 5.11: Modelo de representación lógica.

las fórmulas a nivel de índice. En resumen, este componente toma la fórmula definida en cada nodo del árbol finitario que representa el índice, toma el valor almacenado en la medida para posteriormente reemplazar los datos que correspondan y realizar los cálculos necesarios. Dentro de las funciones que interpreta este componente se encuentran:

- **SUMA:** Función n-aria que retorna la suma de todos los elementos que recibe como entrada. El resultado es un número.
- **RESTA:** Función n-aria que retorna como resultado el restarle al primer elemento, el resto de los n-1 elementos.
- **PARAMETRO:** Función unaria. Recibe como entrada el valor correspondiente a un parámetro de sistema. El resultado es el valor almacenado en ese parámetro (si existe).
- **DIVISION:** Función binaria que retorna el resultado de dividir el primer elemento entre el segundo.
- **PRODUCTO:** Función n-aria que retorna el producto de todos los elementos que recibe como entrada. El resultado es un número.
- **PROMEDIO:** Función n-aria que retorna el promedio de todos los elementos que recibe como entrada. El resultado es un número.
- **PROMEDIONONULO:** Función n-aria que retorna el promedio de todos los elementos distintos de 0 que recibe como entrada. El resultado es un número. El 0 no es considerado de ninguna forma para el cálculo del promedio, es decir no suma ni cuenta como uno de los n elementos.
- **HIJO:** Función unaria que retorna el valor correspondiente al elemento que se recibe como parámetro de entrada. Siendo que este dato se encuentra en un nodo hijo.
- **LARGO:** Función que retorna el largo de una lista.
- **MINIMO:** Función n-aria que retorna el mínimo valor de una lista. El resultado es un número.
- **MAXIMO:** Función n-aria que retorna el máximo valor de una lista. El resultado es un número.
- **SI:** Función n-aria. Función condicional, en caso de ser afirmativa la expresión que evalúa, retorna el primer valor definido, de lo contrario el segundo valor definido.
- **AND:** Función lógica binaria. Retorna el valor True en caso de ser

True ambos elementos de entrada. De lo contrario retorna false.

- OR: Función lógica binaria. Retorna el valor True en caso de ser True alguno de los elementos de entrada. De lo contrario retorna false.
- IGUALES: Función binaria. Compara los dos parámetros de entrada, si son iguales retorna true, de lo contrario retorna false.
- PARATODO: Recibe como parámetros de entrada una función y una lista. Aplica la función a todos los elementos de la lista.
- RAIZ: Función unaria que retorna el valor correspondiente al elemento que se recibe como parámetro de entrada. Siendo que este dato se encuentra en un nodo del árbol.

Para poder realizar estas funcionalidades, se hace uso de expresiones regulares, las cuales validan la sintaxis de las funciones definidas en el índice. Al coincidir la fórmula con alguna de las expresiones, esta es evaluada y se sustituyen los valores que corresponda desde los datos cargados en la medida.

5.7. Pruebas de Sistema

Las pruebas realizadas comenzaron a nivel de desarrollo con los siguientes pasos:

- Pruebas de APIS: Se realizaron pruebas de los distintos endpoints de la API a medida que estos eran desarrollados.
- Pruebas comparativas: Se cargaron datos de planillas INIA mediante la aplicación móvil. Estos datos fueron cargados y procesados en el backend. Se compararon resultados para depurar el sistema.
- Flujos básicos en Frontend: Se realizaron pruebas de los principales flujos de ejecución a nivel de Frontend. Permitiendo así corregir tanto la aplicación web como parte de los servicios del backend.

5.7.1. Pruebas con el usuario

Como instancia final del proceso de pruebas del sistema, se coordina una reunión con el cliente para validar el producto. En esta reunión se realiza una demostración de funcionalidades que era relevante probar y se le brinda

al cliente la url de la web y la APK de la aplicación móvil para que realice pruebas por su cuenta. En esta instancia se detectan algunos detalles que el usuario considera limitantes para el uso en campo de la herramienta, siendo principalmente casos no evaluados en instancias anteriores. Estos detalles son ajustados luego de la reunión, para que de esta forma la herramienta pueda ser utilizada en campo por el cliente.

Se destaca como resultado de la reunión, la respuesta del cliente, valorando positivamente la experiencia de usuario e interfaz de los módulos web y móvil, quedando interesado en realizar más pruebas por su cuenta directamente en campo.

Como conclusión final del encuentro, se destaca que el cliente considera este producto como una buena base para extender el trabajo del IIE. Esto es un hito importante para el proyecto, ya que cumple el objetivo de validar la factibilidad del prototipo.

5.8. Problemas encontrados

En esta sección se exponen los principales problemas encontrados en el transcurso de la etapa de implementación, el enfoque tomado para la presentación es presentarlos en función del módulo en el cual se enfrentó la problemática.

5.8.1. Aplicación Móvil

Durante el desarrollo de la aplicación móvil se enfrentaron varios problemas, entre ellos destacan:

Visualización de documentos PDF

Una de las funcionalidades planteadas es la de poder visualizar ayudas en el dispositivo móvil. Para ello es necesario usar aplicaciones de terceros, lo que implica tener que cortar el flujo de ejecución actual, para así poder visualizar el archivo en cuestión. Situación ante la cual se optó por mostrar imágenes de estos PDF en la misma aplicación, desligándose de esa forma de cualquier aplicación de terceros.

Ausencia de un ORM

Al no existir un ORM como el utilizado en el backend, se dificultó el

trabajo con entidades en la base de datos del dispositivo móvil. Haciendo que estas interacciones se desarrollen manualmente, y generando que el tiempo de desarrollo fuese mayor al planificado.

5.8.2. Servidor

Dentro del servidor se enfrentaron varios problemas en distintos componentes y etapas. A continuación, se enumeran estas problemáticas.

Interpretación de las fórmulas

Para la implementación del módulo de evaluación de fórmulas fue necesario interpretar las planillas de cálculo actuales. Esto en sí fue un problema, ya que son planillas con cierta dificultad y muchos años de constante evolución. Fue el primer reto a nivel de desarrollo ya que era necesario replicar su funcionamiento en el nuevo sistema.

Despliegue

La solución inicialmente estaba pensada para que su despliegue se realice en un servidor de Antel. Sin embargo, la infraestructura que brinda el servicio que se pretendía usar no era el adecuado para el despliegue de aplicaciones usando contenedores Docker. Por este motivo es que hubo que replantear esta etapa del desarrollo y evaluar soluciones alternativas, siendo la opción elegida el despliegue en el producto Elastic Cloud de Antel [12].

Módulo de autenticación

Por aspectos de seguridad, las bibliotecas que provee Spring Boot para el manejo de tokens e Identity Providers suelen actualizarse con mucha frecuencia. A la hora de implementar el módulo de seguridad en la etapa de pruebas de conceptos se alcanzó una primera versión de este módulo. Sin embargo, al intentar usar lo construido unos meses atrás hubo que volver a configurar el IdP y el módulo, debido que al haberse actualizado la biblioteca, lo que fue utilizado estaba deprecado y no servía para lo que se pretendía.

Capítulo 6

Gestión del proyecto

En este capítulo se presenta la gestión del proyecto, la planificación del tiempo contrapuesto al tiempo real, así como los riesgos y las distintas estrategias usadas para mitigar los mismos. Por último, los principales hitos del proyecto.

6.1. Gestión

En esta sección se presenta el proceso de gestión del proyecto. El cual comprende la planificación inicial del mismo. Seguida de los riesgos del proyecto y las estrategias planteadas para atacar estos riesgos.

Por último, se presenta el cronograma final del proyecto en contraposición al proyectado inicialmente.

6.1.1. Fases del proyecto

El enfoque de gestión de este proyecto tiene como eje central el proceso de desarrollo. Sin embargo, esta no es la única tarea por desempeñar, si no por el contrario es una más de las tantas tareas necesarias a realizar para llevar adelante el proyecto. Las fases que se llevan a cabo se presentan a continuación.

Fase uno

Esta etapa del proyecto consiste en realizar las siguientes tareas:

- Estudio inicial del Dominio
- Estado del arte

- Relevamiento de Requerimientos

El objetivo de esta fase del proyecto es marcar las bases para el resto de las fases de este. Es decir, se pretende alcanzar un nivel de entendimiento y conocimiento de la realidad, así como investigar proyectos con ciertas similitudes, para marcar el camino a seguir en las próximas fases.

Fase dos

La segunda fase del proyecto consiste en realizar las siguientes tareas:

- Diseño de la Arquitectura del Sistema
- Desarrollo de pruebas de concepto técnicas

Estas dos tareas tienen objetivos completamente distintos, pero que en cierta forma terminan siendo necesarias cada una a su manera para la siguiente etapa. Por un lado, está el diseño de la arquitectura del sistema, tarea que marca tanto el trabajo de desarrollo, así como el de despliegue de la solución.

Por otra parte, está el desarrollo de prueba de conceptos, en esta etapa se prueban distintas tecnologías en pos de evaluar y mitigar posibles riesgos técnicos, antes de comenzar el proceso de desarrollo.

Fase tres

La tercera etapa del proyecto consiste en una única tarea, la implementación de un prototipo del Sistema.

En esta etapa se pretende desarrollar todos o algunos de los requisitos relevados en la primer etapa, usando tanto los conocimientos adquiridos sobre la realidad en la primera etapa, así como lo diseñado y evaluado de las distintas tecnologías en la segunda fase.

Fase cuatro

Por último, se encuentra la Fase cuatro. Esta fase consiste en dos tareas principales que se realizan en paralelo.

- Pruebas del Sistema
- Documentación Final

Dentro de las pruebas de sistema, se pretende realizar pruebas de desarrollador y pruebas con el usuario, para detectar y en caso de ser posible corregir los distintos bugs del sistema.

La tarea de documentación, si bien es algo que se lleva adelante a lo largo de todo el proyecto, se pretende que en los últimos meses del proyecto se le ponga foco a la misma para finalizarla.

6.1.2. Riesgos

Evaluar los riesgos y plantear distintas alternativas para atacarlos es algo necesario en un proyecto. A continuación, se presentan algunos de los riesgos y las estrategias en caso de que se den.

Dificultades técnicas con alguna de las tecnologías

Este riesgo debe ser mitigado lo antes posible, ya que el objetivo es trabajar con un stack tecnológico en el cual el equipo de trabajo se sienta cómodo. Incluso teniendo como alternativas otras tecnologías en las cuales haya un cierto nivel de experiencia.

Evaluaciones o entregas de alguna materia

Este riesgo es aceptado, es una realidad que siendo estudiantes hay compromisos con materias que cumplir. Motivo por el cual se sabe es algo que sucederá y debe ser previsto por todos los miembros del equipo.

Indisposición de alguno de los integrantes del equipo

Este riesgo es mitigado. La estrategia a usar en caso de que esto suceda es tratar de anticipar con la mayor cantidad de tiempo posible, de forma de poder distribuir la carga del integrante para el lapso de tiempo en el cual no se encuentre, entre el resto de los integrantes de forma equitativa.

6.1.3. Proyección del proyecto

En esta subsección se presentan los diagramas de gantt del proyecto. Primero se muestra la proyección (en la figura 6.1) a comienzos de este y luego se presenta la evolución real del proyecto (en la figura 6.2), resaltando los desvíos que este sufrió. Luego de esto, se presentan los hitos del proyecto.

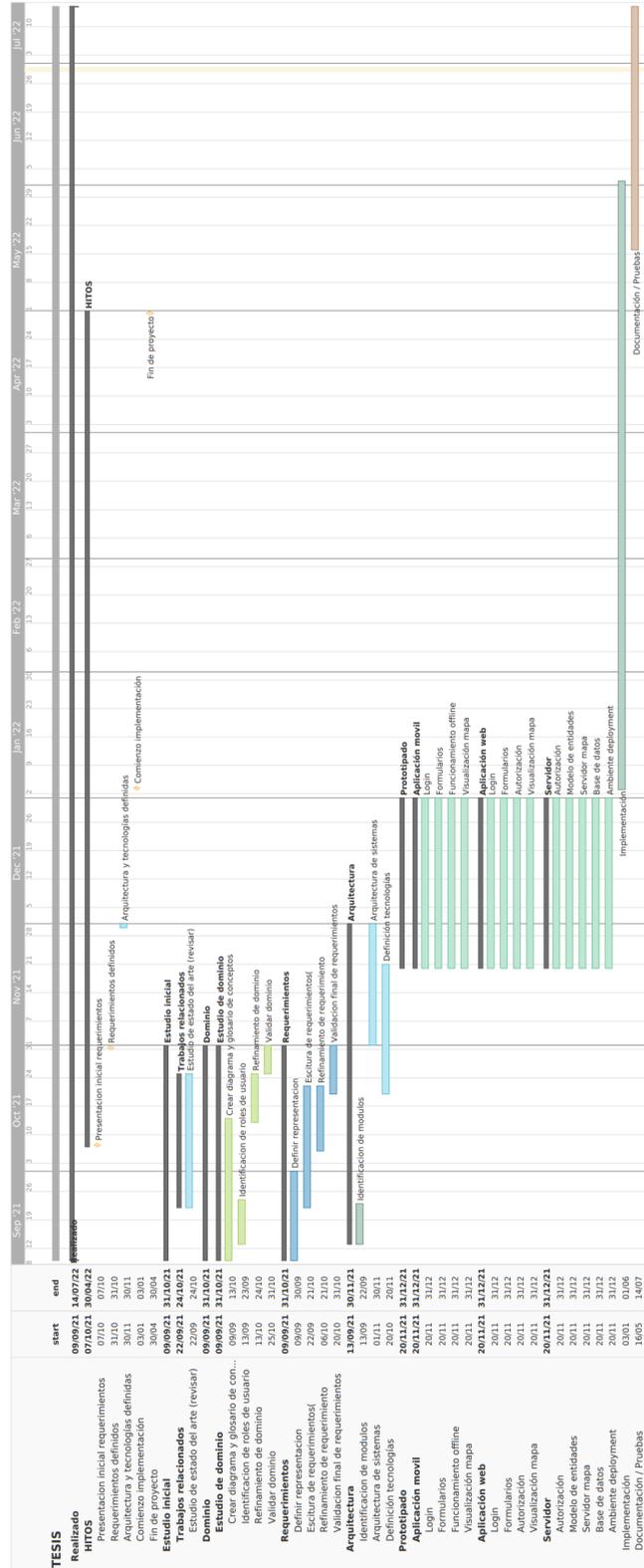


Figura 6.2: Gantt Realizado

6.2. Desvíos

En las secciones anteriores se presentan los diagramas de gantt correspondientes al proceso planificado 6.1 y en contraposición el proceso efectivo 6.2.

Al comparar ambos diagramas se pueden ver diferencias entre ellos. La principal diferencia entre ellos es la duración total del proyecto. A continuación, se presentan los motivos de desvíos, realizando un desglose por fase del proyecto.

Fase uno

Dentro de la primera fase del proyecto no se aprecian grandes desvíos, tan solo una semana que se explica debido a una iteración más en el proceso de relevamiento de requisitos. El resto de las tareas planificadas para ese periodo se realizaron en tiempo y forma de acuerdo con lo planeado.

Fase dos

En la segunda fase del proyecto, la cual está conformada por las tareas de diseño y pruebas de concepto, se encontraron algunas dificultades, las cuales plantearon ciertos desvíos del plan inicial de trabajo, pero que no implican grandes desviaciones en cuanto a tiempo, ya que se aplicaron correctamente las estrategias para afrontar los riesgos. En cuanto a diseño no hubo ningún inconveniente, las dificultades se dieron realizando pruebas de concepto y probando tecnologías. En donde por limitaciones técnicas hubo dos inconvenientes.

- App Móvil: Se descartó React-Native como tecnología, pasando a ser Kotlin la tecnología usada.
- Aplicación Web: Se contaba con una base de proyecto la cual no estaba licenciada, motivo por el cual no pudo ser usada.

En cuanto al primer ítem, fue un desvío menor, que terminó siendo positivo, ya que permitió redistribuir la carga de forma más equitativa en el equipo de trabajo, sin embargo, el segundo ítem, significó un trabajo que no se estimaba. Este desvío no impactó fuertemente en esta etapa.

Fase tres

La tercera etapa del proyecto, o etapa de implementación, en comparativa con el modelo estimado es la que sufrió más desvíos temporales. Estos se deben en parte a dos grandes sucesos. Por un lado, se subestimó el tiempo de análisis de las planillas de cálculo electrónicas y modelos usados por parte de INIA. Esto implicó un mayor esfuerzo de trabajo para modelar correctamente requisitos funcionales y no funcionales.

Por otra parte, el segundo desvío presentado en la fase anterior, que no significó demoras para la segunda fase del proyecto, aquí impactó negativamente, y junto a otras dificultades se derivó a un tiempo de desarrollo casi del doble de lo estimado, para el módulo web.

Fase cuatro

Por último, en la fase de documentación se dió el caso contrario a lo que aconteció en la etapa tres. Ya que como estrategia de trabajo se recolectó información en pos de la documentación final a lo largo del proyecto. Por lo cual los tiempos se cumplieron tal cual estaba estipulado.

6.3. Hitos del Proyecto

En esta sección se enumeran los principales hitos en el desarrollo del proyecto.

- 20 Agosto 2021, primer reunión oficial con cliente.
- 01 Setiembre 2021, comienzo oficial del proyecto de grado.
- 29 Setiembre 2021, comienzo de ciclo de reuniones para relevamiento de requisitos.
- 20 de Octubre, comienzo de diseño de la arquitectura del sistema.
- 04 de Noviembre, fin de fase de relevamiento de requisitos.
- 08 de Noviembre, comienzo de pruebas de concepto.
- 04 de Diciembre, validación de arquitectura del sistema con el cliente.
- 03 de Enero 2022. Comienzo de etapa de desarrollo.
- 15 de Febrero, reunión de seguimiento con primer demo al equipo de docentes.
- 01 Abril 2022. Reajuste del alcance del producto.

- 28 de Abril 2022, presentación del producto en el curso de TSIG edición 2022.
- 29 de Abril 2022. Fin oficial de la fase de desarrollo.
- 02 de Mayo 2022. Comienzo de fase de documentación.
- 19 de Mayo 2022, presentación formal de la aplicación al comité de aplicaciones INIA.
- 25 de Julio 2022, presentación final de la aplicación al cliente.

Capítulo 7

Conclusiones y Trabajo futuro

En este capítulo se presentan en primer lugar los principales aportes del proyecto y las conclusiones de este. Por último, se presentan los elementos mas relevantes para estudiar como trabajo a futuro de este proyecto.

7.1. Principales aportes y conclusiones

En esta sección se presentan los principales aportes del proyecto. Primero se presenta uno a uno los resultados con el objetivo planteado en las primeras secciones del informe.

Además de estos aportes, se agregan otros que se creen pertinentes enmarcar y por último se enumeran las conclusiones.

Estudiar el Dominio y Análisis

La primera etapa del proyecto fue la de análisis. En esta etapa se tomó como punto de partida todo el conocimiento e información brindada por el cliente, el estudio de aplicaciones que realizan tareas similares, para de esta forma poder modelar un conjunto de requerimientos de sistema. Estos requerimientos fueron base para poder especificar requisitos funcionales y no funcionales. Siendo todo este trabajo plasmado en mock-ups, los cuales pueden ser utilizados para futuros desarrollos que se deseen. A todo esto se suma el trabajo de entendimiento a nivel de realidad, el cual está plasmado en este documento y en varios diagramas los cuales se entregan adjuntos a este. Este aporte del proyecto es un agregado que no fue requerido por el cliente, ni planificado por docentes o alumnos, ya que la documentación generada mientras se estudió por parte de los estu-

diantes, aporta al cliente diagramas y conceptualización de conocimiento que actualmente no existía.

Diseño

En cuanto al diseño, hay que destacar que se planteó una solución cuyo objetivo es el de ser extendida a futuro. Siendo un punto de partida para el equipo que retome el proyecto.

Implementación

A nivel de implementación se alcanzó un producto configurable que cumple con lo pedido por el cliente.

Si bien no se llegó al alcance deseado por parte de los estudiantes, ya que hubo que realizar recortes por errores en las estimaciones, o por los problemas enfrentados, el producto final es un buen punto de partida para desarrollar la aplicación que INIA desea tener para el índice.

El principal aporte es el módulo de backend, el cual no solo tiene cualidades como ser configurable, y capaz de modelar la realidad. Si no que fue diseñado desde un principio con la idea de ser extensible. Es por esto por lo que a pesar de que hay funcionalidades que no estén desarrolladas o completamente desarrolladas, está dada la arquitectura interna del mismo para que pueda ser extendido.

De la mano con esto último, se suma la versatilidad de la aplicación móvil. Esta última tiene la cualidad de poder interpretar la información del servidor, almacenarla y poder ejecutar offline. Enviando los datos al servidor cuando se le indique y la conexión lo permita. Al funcionar offline tiene una gran carga lógica y esta también es un excelente puntapié para futuros desarrollos.

Carga inicial de datos

Parte del trabajo de interpretación previo al realizado a la hora de cargar el índice en la aplicación, fue el de interpretar las planillas de cálculo electrónico. Este trabajo arrojó como resultado el modelado de la información en el modelo de solución planteado, siendo esta la carga inicial de datos con la que cuenta el producto entregado al cliente, y teniendo la posibilidad de volver a este estado inicial. Este aporte es muy importante, ya que tiene un cierto nivel de complejidad y a la hora de generar nuevas

representaciones del índice es útil tener este trabajo como referencia o punto de partida.

Es por todo esto que se concluye que todos los objetivos planteados al principio del proyecto fueron cumplidos con éxito. Por lo cual se pudo desarrollar un producto que puede ser usado como base para el futuro desarrollo del sistema de gestión del IIE o alguna alternativa de este. Y no sólo eso, a su vez se logró un nivel de entendimiento de los conceptos asociados al índice. En este proceso se generó documentación la cual podrá ser usada tanto por el equipo de desarrollo que tome este prototipo para construir una solución, así como por INIA mismo en futuros desarrollos o evoluciones del índice.

7.2. Trabajo futuro

Como se mencionó en las secciones anteriores, este proyecto ha sido un hito muy importante en el desarrollo del sistema de gestión para el índice de integridad ecosistémica.

A continuación, se enumeran los principales puntos que deben ser aplicados para llevar la solución actual hacia una aplicación productiva.

Retomar trabajo de desarrollo

En cuanto a este ítem, lo que se quiere reflejar es que se realizó un trabajo de análisis más extenso que el alcance final del prototipo desarrollado en sí. Es por esto por lo que se plantea retomar el trabajo de desarrollo desde donde se dejó, tomando como punto de partida el trabajo recabado en este proyecto y el producto alcanzado.

Las principales funcionalidades que quedaron por fuera del alcance del proyecto fueron:

- Georreferenciación de potreros y establecimientos.
- Gestión de roles de usuario.
- Visualización de resultados acorde al diseño relevado.
- Páginado de todas las entidades a nivel de API.

Pruebas exhaustivas

Será necesario realizar un trabajo de pruebas evaluando las funcionalidades actualmente desarrolladas y por supuesto las nuevas funcionalidades que se desarrollen.

Workflow

Como sugerencia, para facilitar varios de los procesos de backoffice o también para requerimientos como el poder compartir la información asociada a cálculos de distintos índices para los establecimientos. Se puede estudiar la posibilidad de usar Workflows (por ejemplo Power Automate [32]) usando las API de backend como proveedoras de información para estas herramientas.

El objetivo de este tipo de herramientas low-code es el de permitir a usuarios sin conocimiento, o con poco conocimiento de desarrollo en sí, el poder generar aplicaciones con interfaces web. Pudiendo así reducir costes, tiempos de desarrollo, al mismo tiempo que se reutilizan las API ya generadas para obtener los datos que se exponen mediante estas herramientas. Es decir, la información que se brindan las API actuales, sumado a posibles extensiones del sistema, junto con este tipo de herramientas permiten potenciar de manera muy rápida y eficaz la evolución del índice.

Glosario

Backlog Se denomina backlog a la lista de trabajo ordenado por prioridades para el equipo de desarrollo. El mismo se obtiene de la hoja de ruta y sus respectivos requisitos.

Requisito Funcional Se define como una descripción de lo que el sistema debe hacer bajo condiciones específicas [36].

Requisito No Funcional Se define como una descripción de una propiedad que un sistema debe poseer o una restricción que debe respetar [36].

Modelo Cascada Modelo en Cascada [29], es una metodología que ordena las etapas del proceso de desarrollo de software, de forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Se compone de cinco etapas, el análisis de requisitos, el diseño, la implementación, la verificación y el mantenimiento.

DOM Por sus siglas en inglés “Document Object Model” es un estándar W3C [44]. DOM HTML define un estándar y una interfaz que permite a los programas y scripts acceder y modificar contenido, estructura y estilos de documentos HTML [11].

SPA Por sus siglas en inglés “Single page application”, refiere a una aplicación web de una sola página. Simula el cambio de páginas a través del cambio del contenido que muestra en una única página [39].

Javascript Javascript es un lenguaje de programación de scripting (secuencias de comandos) utilizado principalmente para desarrollo de páginas web [21].

JVM “Java Virtual Machine” es una máquina virtual utilizada para ejecutar código generado con el lenguaje de programación Java [20].

Tesela Organización de datos de terreno espacialmente coherente que permite recuperar y editar con eficacia. La definición de la tesela depende del espaciado de punto promedio de los datos de origen. [13].

Árbol finitario Un árbol general (o finitario), es un árbol en el cual el número de hijos por nodo puede variar. Es decir, cada nodo tiene una “lista” de árboles

asociados (sus subárboles) [45].

Escalamiento Horizontal El escalamiento horizontal implica tener varios servidores (denominados Nodos) funcionando como un único servidor. Se crea una red de servidores conocida como Cluster, con el objetivo de repartirse la carga de trabajo entre los distintos nodos del cluster.

Escalamiento Vertical El escalamiento vertical implica el aumento de recursos a nivel de hardware. El esfuerzo de este crecimiento es mínimo, pues no tiene repercusiones en el software, ya que solo será respaldar y migrar los sistemas al nuevo hardware.

Riesgos Un riesgo [38] es un evento o condición incierta que, de producirse, tiene un efecto positivo o negativo en uno o más de los objetivos del proyecto, tales como el alcance, el cronograma, el costo y la calidad del mismo.

Bearer Token Son el tipo de token [7] más usado por el protocolo OAuth2. Es un string opaco sin sentido para los usuarios, el cual es usado por las aplicaciones.

UI El termino UI hace referencia a la interfaz de usuario. La interfaz de usuario apunta a guiar al usuario por la aplicación cuando esta última esta en uso.

UX El termino UX hace referencia a la experiencia de usuario. La experiencia de usuario se define como la sensación que tiene un usuario al utilizar un producto o una herramienta como puede ser una aplicación web.

Zona riparia Se denomina zona riparia al espacio de transición entre el suelo y un río, arroyo u otro cuerpo de agua.

Bibliografía

- [1] Actividad economica uruguay — observatorio de complejidad económica. <https://oec.world/es/profile/country/ury>. Accedido: 2022-05-31.
- [2] Android — sitio oficial. https://www.android.com/intl/es_es/. Accedido: 2022-06-07.
- [3] Android room — definición por ashish rawat. <https://medium.com/mindorks/using-room-database-android-jetpack-675a89a0e942#:~:text=Room%20is%20an%20ORM%2C%20object,the%20full%20power%20of%20SQLite>. Accedido: 2022-07-19.
- [4] Apache tomcat — sitio oficial. <https://tomcat.apache.org/>. Accedido: 2022-06-13.
- [5] Aplicación móvil — definición por aincubator. <https://aincubator.com/que-es-una-aplicacion-movil/>. Accedido: 2022-06-20.
- [6] Axios — libreria cliente http. <https://axios-http.com/>. Accedido: 2022-06-25.
- [7] Bearer token— oauth2 bearer token. <https://oauth.net/2/bearer-tokens/>. Accedido: 2022-07-22.
- [8] Cambio climatico — organización de naciones unidas. <https://www.un.org/es/climatechange/what-is-climate-change>. Accedido: 2022-05-31.
- [9] Controlador de versiones — ¿qué es el control de versiones? por atlassian. <https://www.atlassian.com/es/git/tutorials/what-is-version-control>. Accedido: 2022-06-06.

- [10] Create react app — inicializador para aplicaciones react. <https://create-react-app.dev/>. Accedido: 2022-06-25.
- [11] Document object model — definición por w3 schools. https://www.w3schools.com/js/js_htmlDOM.asp. Accedido: 2022-07-16.
- [12] Elastic cloud — sitio oficial - antel. https://minubeantel.uy/index.php?NAME_PATH=Elastic_Cloud. Accedido: 2022-07-12.
- [13] Esri — definición de tesela. <https://support.esri.com/es/other-resources/gis-dictionary/term/c5a0ea54-3685-424a-99e5-ab44a32e653f>. Accedido: 2022-07-10.
- [14] Gitlab — acerca de gitlab. <https://about.gitlab.com/>. Accedido: 2022-06-06.
- [15] Gradle — introducción por yanina muradas. <https://openwebinars.net/blog/que-es-gradle/>. Accedido: 2022-06-06.
- [16] Hibernate — introducción por if geek then. <https://ifgeekthen.nttdata.com/es/que-es-java-hibernate-por-que-usarlo>. Accedido: 2022-07-11.
- [17] Historias de usuario — atlassian. <https://www.atlassian.com/es/agile/project-management/user-stories>. Accedido: 2022-07-29.
- [18] Identity provider — introducción por okta. <https://www.okta.com/identity-101/why-your-company-needs-an-identity-provider/>. Accedido: 2022-07-12.
- [19] Instituto nacional de investigación agropecuaria — marco institucional. <http://www.inia.uy/marco-institucional/>. Accedido: 2022-05-31.
- [20] Java virtual machine — definición por w3 schools. <https://www.w3schools.in/java/java-virtual-machine>. Accedido: 2022-07-16.
- [21] Javascript — definición por mozilla. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Accedido: 2022-07-16.

- [22] Json — definición por ibm. <https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format>.
Accedido: 2022-07-11.
- [23] Kotlin — sitio oficial. <https://kotlinlang.org/>. Accedido: 2022-06-07.
- [24] Mapbox — ejemplos de mapbox offline. <https://docs.mapbox.com/android/legacy/maps/examples/download-a-simple-offline-map/>. Accedido: 2022-07-10.
- [25] Mapbox — mapbox mobile sdk. <https://www.mapbox.com/mobile-maps-sdk>. Accedido: 2022-07-10.
- [26] Material kit react — plantilla layout dashboard. <https://github.com/minimal-ui-kit/material-kit-react>. Accedido: 2022-06-25.
- [27] Material ui — librería gráfica de componentes react. <https://mui.com/>. Accedido: 2022-06-25.
- [28] Maven — introducción por carlos yagüe. <https://openwebinars.net/blog/que-es-apache-maven/>. Accedido: 2022-06-06.
- [29] Modelo cascada — definición por ecured. https://www.ecured.cu/Modelo_en_cascada. Accedido: 2022-06-12.
- [30] OAuth 2.0 — introducción por digital ocean. <https://www.digitalocean.com/community/tutorials/una-introduccion-a-oauth-2-es>. Accedido: 2022-06-29.
- [31] PostgreSQL — acerca de postgresql. <https://www.postgresql.org/about/>. Accedido: 2022-06-06.
- [32] Power automate — sitio oficial. <https://powerautomate.microsoft.com/en-us/>. Accedido: 2022-06-27.
- [33] React google login — componente react para google login. <https://www.npmjs.com/package/react-google-login>. Accedido: 2022-06-25.
- [34] react-query — librería react para data fetching. <https://react-query.tanstack.com/>. Accedido: 2022-06-25.

- [35] Reactjs — sitio oficial. <https://es.reactjs.org/>. Accedido: 2022-06-07.
- [36] Requerimientos funcionales y no funcionales — sitio eva curso introducción a la ingeniería de software. https://eva.fing.edu.uy/pluginfile.php/283040/mod_resource/content/1/Requisitos%20-%20Clase%201.pdf. Accedido: 2022-05-31.
- [37] Revista iie — publicación inia. <http://www.inia.uy/Publicaciones/Documentos%20compartidos/Revista-INIA-67-Dic-2021-22.pdf>. Accedido: 2022-06-20.
- [38] Riesgos — sitio eva curso introducción a la ingeniería de software. https://eva.fing.edu.uy/pluginfile.php/293606/mod_resource/content/2/IIS%20Notas%20-Gestion%20de%20proyectos%20-%20Riesgos%20-%20Clase3.pdf. Accedido: 2022-06-27.
- [39] Single page application — definición por mozilla. <https://developer.mozilla.org/en-US/docs/Glossary/SPA>. Accedido: 2022-07-16.
- [40] Sistema de información geográfica — definición por esri. <https://www.esri.com/en-us/what-is-gis/overview>. Accedido: 2022-07-16.
- [41] Springboot — introducción por spring. <https://spring.io/projects/spring-boot#overview>. Accedido: 2022-06-06.
- [42] Teamscope — formularios móviles offline para estudios en campo. <https://es.teamscopeapp.com/>. Accedido: 2021-11-13.
- [43] Webapp — definición por ictea. <https://www.ictea.com/cs/index.php?rp=%2Fknowledgebase%2F4205%2FWhat-is-a-web-application.html>. Accedido: 2022-06-20.
- [44] World wide web consortium — acerca de w3c. <https://www.w3.org/Consortium/>. Accedido: 2022-07-18.
- [45] Árbol finitario — sitio eva curso programación 2. https://eva.fing.edu.uy/pluginfile.php/106171/mod_resource/content/7/Punteros_arboles.pdf. Accedido: 2022-05-31.

- [46] Oscar Blumetto, Andrés Castagna, Gerónimo Cardozo, Felipe García, Guadalupe Tiscornia, Andrea Ruggia, Santiago Scarlato, María Marta Albicette, Verónica Aguerre, and Alfredo Albin. Ecosystem integrity index, an innovative environmental evaluation tool for agricultural production systems. *Ecological Indicators*, 2019.
- [47] Oscar Blumetto, Andrés Castagna, Gerónimo Agustín Cardozo, Andrea Ruggia, Guadalupe Tiscornia, and Verónica Aguerre. Índice de integridad ecosistémica: herramienta multidimensional para la evaluación ambiental de sistemas de producción. *XXIV Reunión del Grupo Técnico en Forrajes del Cono sur (Grupo Campos)*, 2017. Tacuarembó, Uruguay.