

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“MYMICROBEAPP: DESARROLLO DE UNA APLICACIÓN
PARA LA INTEGRACIÓN Y VISUALIZACIÓN DE DATOS
DERIVADOS DE MICROORGANISMOS MARINOS CON
POTENCIAL BIOTECNOLÓGICO”

RAÚL IGNACIO CRUZ TORRES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Carlos Buil Aranda
Profesor Correferente: Beatriz Cámara

Enero - 2024

DEDICATORIA

A mi familia, quienes siempre me apoyaron durante este largo proceso.

AGRADECIMIENTOS

A mi familia, mi madre Verónica, mi padre Raúl, mi hermana Catalina y mi hermana Valentina, por la paciencia y el apoyo que me brindaron durante toda mi etapa universitaria.

A los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología "Dr. Daniel Alkalay Lowitt" de la Universidad Técnica Federico Santa María, por como me recibieron en su equipo, su orientación sobre conceptos y temáticas que desconocía, el trabajo en conjunto y continuas reuniones de retroalimentación que hicieron posible esta memoria.

A mis profesores guía (Carlos Buil) y correferente (Beatriz Cámara) por haber aceptado mi solicitud respecto al tema de memoria que publicaron e introducirme a él.

A mis amigos Esteban, Ignacio, Alonso, Felipe, Ariel y Nicolás, por tantas jornadas de alegría y entretención.

Y a mis compañeros de Ingeniería Civil Informática, Roberto, Francisco, Darinka e Ignacio, con quienes formé un agradable grupo durante mi estadía en la carrera.

RESUMEN

Resumen— El Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología ‘Dr. Daniel Alkalay Lowitt’ de la Universidad Técnica Federico Santa María tiene el objetivo de encontrar antibióticos novedosos. No obstante, su trabajo se dificulta ya que sus investigadores no cuentan con un medio digital para registrar y/o acceder a sus datos.

En este trabajo se expone el proceso de diseño y desarrollo de MyMicrobeApp, una aplicación web que permite integrar y visualizar datos de microorganismos marinos, siguiendo la metodología Personal Kanban. La aplicación fue evaluada por sus usuarios, siendo validada como una herramienta fácil de utilizar. Esta se encuentra disponible para su uso y su código está en repositorios públicos.

Palabras Clave— MyMicrobeApp; Actinobacteria; Bioprospección; Aplicación; Laboratorio de Bioprospección de Actinobacterias Marinas.

ABSTRACT

Abstract— The Marine Actinobacteria Bioprospecting Laboratory of the Centro de Biotecnología ‘Dr. Daniel Alkalay Lowitt’ at Universidad Técnica Federico Santa María aims to find novel antibiotics. However, their job is hindered as their researchers lack a digital platform to record and/or access their data.

This work presents the design and development process of MyMicrobeApp, a web application that allows the integration and visualization of marine microorganism data, following the Personal Kanban methodology. The application was evaluated by its users and validated as an easy-to-use tool. It is available for use, and its code is in public repositories.

Keywords— MyMicrobeApp; Actinobacteria; Bioprospecting; Application; Marine Actinobacteria Bioprospecting Laboratory.

GLOSARIO

ACID: Atomicity, Consistency, Isolation, Durability.

API: Application Programming Interface.

BacDive: Bacterial Diversity Metadatabase.

BASE: Basically Available, Soft state, Eventual consistency.

BGCs: Biosynthetic Gene Clusters.

BSON: Binary JavaScript Object Notation.

CB-DAL: Centro de Biotecnología Dr. Daniel Alkalay Lowitt.

CRUD: Create, Read, Update and Delete.

CSS: Cascading Style Sheets.

CSV: Comma-Separated Values.

DBMS: Database Management System.

DI: Departamento de Informática.

GCM: Global Catalogue of Microorganisms.

gcType: Global Catalogue of Type Strain.

HTML: HyperText Markup Language.

HTTP: HyperText Transfer Protocol.

JSON: JavaScript Object Notation.

MVC: Model-View-Controller.

MVP: Minimum Viable Product.

NCBI: National Center for Biotechnology Information.

NewSQL: New Structures Query Language.

NoSQL: Not Only Structured Query Language.

PCR: Polymerase Chain Reaction.

PDF: Portable Document Format.

PHP: Hypertext Preprocessor.

REST: Representational State Transfer.

RDBMS: Relational Database Management System.

rRNA: Ácido Ribonucleico ribosomal.

SQL: Structured Query Language.

SUS: System Usability Scale.

UI: User Interface.

URL: Uniform Resource Locator.

UTFSM: Universidad Técnica Federico Santa María.

UX: User Experience.

WIP: Work In Progress.

XML: Extensible Markup Language.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Contexto	2
1.2 Organización	3
1.3 Situación actual	4
1.4 Objetivos	5
1.4.1 Objetivo General	5
1.4.2 Objetivos específicos	6
1.4.3 Alcance	6
CAPÍTULO 2: MARCO CONCEPTUAL	7
2.1 Estado del arte	7
2.2 Herramientas de trabajo	9
2.2.1 Framework	9
2.2.2 Base de datos	10
2.2.3 Front-end	11
2.2.4 Back-end	14
2.2.5 Herramientas seleccionadas	17
2.3 UX/UI	18
2.3.1 UX	18
2.3.2 UI	19
2.3.3 UX vs UI	19
2.3.4 Heurísticas de Nielsen	20
2.4 Metodología de trabajo	21
2.4.1 Metodologías populares	21
2.4.2 Metodología seleccionada: Personal Kanban	22
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	25
3.1 Procesos del laboratorio	25
3.1.1 Caracterización de los procesos del laboratorio	25
3.1.2 Definición de los procesos a implementar en la aplicación	33
3.2 Diseño	37

3.2.1	Diseño de la interfaz de usuario	37
3.2.2	Diseño de la base de datos	39
3.2.3	Definición de roles de usuario	46
3.3	Desarrollo	47
3.3.1	Desarrollo de la base de datos	47
3.3.2	Desarrollo de la API	48
3.3.3	Desarrollo de la interfaz de usuario	50
3.4	Resultados	51
3.4.1	Base de datos	51
3.4.2	API	52
3.4.3	Interfaz	52
3.4.4	Menús y vistas de la aplicación	52
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		56
4.1	Sistema de escalas de usabilidad	56
4.2	Resultados	57
CAPÍTULO 5: CONCLUSIONES		59
5.1	Conclusiones generales	59
5.2	Cumplimiento de objetivos	59
5.3	Recomendaciones	60
5.4	Trabajo futuro	61
ANEXO A: Mockup de la aplicación		62
ANEXO B: Interfaz de la aplicación		89
REFERENCIAS BIBLIOGRÁFICAS		103

ÍNDICE DE FIGURAS

1	Principales métodos utilizados para la bioprospección de microorganismos . . .	3
2	Árbol del problema	5
3	Relación entre UX y UI.	19
4	Ejemplo de tablero Kanban.	23
5	Matriz de Personal Kanban	24
6	Etapas del PCR	27
7	Perfil térmico de PCR para amplificación del gen 16S rRNA específico para Actinobacterias	28
8	Fermentación de la cepa <i>Streptomyces</i> sp. A-B3.3 en 3 medios de cultivo . . .	29
9	Ensayos de actividad antibacteriana de <i>Streptomyces</i> sp. A-B3.3 contra STAU (<i>Staphylococcus aureus</i>) y LIMO (<i>Listeria monocytogenes</i>).	30
10	Ejemplo de una red de similitud de agrupamientos de genes biosintéticos. . .	33
11	Idea original de MyMicrobeApp.	34
12	Relación entre los procesos del laboratorio y las categorías en las que se almacenan los datos de las actinobacterias.	36
13	Modelo de Assembly en la base de datos.	48
14	Estructura de carpetas de la API.	49
15	Estructura de carpetas de la interfaz de usuario.	50
16	Menús y vistas de usuario con rol <i>user</i>	53
17	Menús y vistas de usuario con rol <i>manager</i>	54
18	Menús y vistas de usuario con rol <i>admin</i>	55
19	Mockup Inicio de sesión	62
20	Mockup Actinobacteria List	63
21	Mockup Actinobacteria Show	64
22	Mockup Actinobacteria Create	65
23	Mockup Actinobacteria Edit	66
24	Mockup Assembly List	67
25	Mockup Assembly Show	68
26	Mockup Assembly Create	69
27	Mockup Assembly Edit	70
28	Mockup Processed Data List	71
29	Mockup Processed Data Show	72
30	Mockup Processed Data Create	73
31	Mockup Processed Data Edit	74
32	Mockup Users List	75
33	Mockup Users Edit	76
34	Mockup Genera List	77
35	Mockup Genera Create	78
36	Mockup Genera Edit	79
37	Mockup Culture Media List	80
38	Mockup Culture Media Create	81
39	Mockup Culture Media Edit	82

40	Mockup Antimicrobial List	83
41	Mockup Antimicrobial Create	84
42	Mockup Antimicrobial Edit	85
43	Mockup Enzyme List	86
44	Mockup Enzyme Create	87
45	Mockup Enzyme Edit	88
46	Interfaz Inicio de sesión	89
47	Interfaz Actinobacteria List	89
48	Interfaz Actinobacteria Show	90
49	Interfaz Actinobacteria Create	90
50	Interfaz Actinobacteria Edit	91
51	Interfaz Assembly List	91
52	Interfaz Assembly Show	92
53	Interfaz Assembly Create	92
54	Interfaz Assembly Edit	93
55	Interfaz Processed Data List	93
56	Interfaz Processed Data Show	94
57	Interfaz Processed Data Create	94
58	Interfaz Processed Data Edit	95
59	Interfaz Users List	95
60	Interfaz Users Edit	96
61	Interfaz Genera List	96
62	Interfaz Genera Create	97
63	Interfaz Genera Edit	97
64	Interfaz Culture Media List	98
65	Interfaz Culture Media Create	98
66	Interfaz Culture Media Edit	99
67	Interfaz Antimicrobial List	99
68	Interfaz Antimicrobial Create	100
69	Interfaz Antimicrobial Edit	100
70	Interfaz Enzyme List	101
71	Interfaz Enzyme Create	101
72	Interfaz Enzyme Edit	102

ÍNDICE DE TABLAS

1	Campos de la biotecnología y sus productos. Fuente: Elaboración Propia. Basada en la tabla 1 de [Dionisi <i>et al.</i> , 2012]	2
2	Información de las bases de datos incluidas en el estado del arte. Fuente: Elaboración Propia.	9
3	Ventajas y desventajas de los frameworks de front-end más populares. Fuente: Elaboración Propia. Basada en la información de [Ivanovs, 2023].	12

4	Ventajas y desventajas de los frameworks de back-end más populares para la construcción de una API REST. Fuente: Elaboración Propia. Basada en la información de [Kaur, 2022].	15
5	Esquema de los usuarios en la base de datos. Fuente: Elaboración Propia. . .	41
6	Esquema de los géneros bacterianos en la base de datos. Fuente: Elaboración Propia.	42
7	Esquema de los medios de cultivo en la base de datos. Fuente: Elaboración Propia.	42
8	Esquema de los ensamblajes en la base de datos. Fuente: Elaboración Propia.	42
9	Esquema de las cepas tipo en la base de datos. Fuente: Elaboración Propia. .	43
10	Esquema del procesamiento de datos en la base de datos. Fuente: Elaboración Propia.	43
11	Esquema de las enzimas en la base de datos. Fuente: Elaboración Propia. . .	44
12	Esquema de las actinobacterias en la base de datos. Fuente: Elaboración Propia.	44
13	Acceso a los menús según el rol de usuario. Fuente: Elaboración Propia. . . .	46

INTRODUCCIÓN

Los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología 'Dr. Daniel Alkalay Lowitt' de la Universidad Técnica Federico Santa María trabajan en pos de encontrar actinobacterias capaces de producir antibióticos novedosos.

En dicho contexto, los investigadores se ven afectados por la no existencia de un medio digital que les permita registrar y/o acceder a los resultados de sus estudios. Por consecuencia, surge la iniciativa de desarrollar una herramienta que permita satisfacer sus necesidades y solucionar la problemática actual mediante la cooperación con el programa de memorias del Departamento de Informática de la Universidad Técnica Federico Santa María.

En la presente memoria se diseña y desarrolla MyMicrobeApp, una aplicación web que permite integrar y visualizar datos derivados de microorganismos marinos. Para su desarrollo se utiliza la metodología de trabajo 'Personal Kanban', que permite que una persona sea capaz de gestionar un proyecto por medio de un tablero con una serie de tarjetas y columnas, que representan tareas individuales y sus estados respectivamente.

Este trabajo comienza por la definición del problema que origina la necesidad de MyMicrobeApp. Dicha definición consta de una contextualización (que aborda el concepto de bioprospección), identificación de la organización para con que se trabaja, un análisis de su situación y el establecimiento de objetivos y un alcance para la solución.

Posteriormente, se expone un marco conceptual, el cual engloba el estado del arte de bases de datos bacterianas, las herramientas a utilizar en la etapa de desarrollo, conceptos correspondientes a la experiencia de usuario y las interfaces de usuario y la selección de una metodología de trabajo.

Tras lo anterior, se genera una propuesta de solución, en la que se identifican y caracterizan los procesos llevados a cabo en el Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología 'Dr. Daniel Alkalay Lowitt' de la Universidad Técnica Federico Santa María, lo que permite seleccionar cuales se implementarán y posteriormente diseñar y desarrollar una base de datos, API, interfaz de usuario y roles de usuario para la aplicación.

Luego, se exhiben los resultados de la validación de la solución por medio de la utilización de un método centrado en los usuarios.

Finalmente, se exponen las conclusiones obtenidas, las cuales se centran en el desarrollo y despliegue de MyMicrobeApp, el cumplimiento de los objetivos de la presente memoria, recomendaciones y el planteamiento de ideas para posibles trabajos futuros que podrían mejorar y/o complementar la aplicación.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. Contexto

El 60 % de la biomasa presente en el planeta tierra corresponde a microorganismos. Su importancia reside en su papel de cumplir con ciclos biogeoquímicos esenciales para la vida, como lo son la regulación de la composición de la atmósfera, su influencia sobre el clima, el reciclaje de nutrientes y la degradación de contaminantes [Dionisi *et al.*, 2012].

En suma, los microorganismos proporcionan productos y procesos valiosos para distintos campos de la biotecnología (como se puede ver en la tabla 1), por lo que durante años han sido objeto de interés tanto para la comunidad científica como para las distintas industrias.

Tabla 1: Campos de la biotecnología y sus productos.
Fuente: Elaboración Propia. Basada en la tabla 1 de [Dionisi *et al.*, 2012]

Campo	Productos
Biotecnología industrial	Químicos finos, aminoácidos, vitaminas, ácidos orgánicos, detergentes, biocatalizadores.
Biotecnología farmacéutica	Fármacos antitumorales, nutracéuticos, compuestos antibacterianos, inmunosupresores.
Biotecnología agrícola	Biopesticidas, agentes antiparasitarios, agentes de procesamiento de alimentos, fertilizantes.
Biotecnología ambiental	Biorremediación, producción de bioenergía.

Para obtener los productos de la tabla 1 es crucial el proceso de bioprospección, que se define como una búsqueda sistemática y organizada de productos útiles derivados de biorecursos, incluidas plantas, microorganismos, animales, etc., que pueden desarrollarse aún más para la comercialización y los beneficios generales de la sociedad.

La figura 1 permite esquematizar el proceso de bioprospección en sus diversas etapas. La primera consiste en el cultivo de microorganismos por medio de una muestra obtenida en el medio ambiente; dicho cultivo puede realizarse utilizando métodos tradicionales o de alto rendimiento. La segunda etapa, que es alternativa a la primera, corresponde a la extracción directa de ADN de las muestras. Posteriormente, en la tercera etapa se clona el ADN de la muestra ambiental o de cultivos puros para generar bibliotecas genómicas o metagenómicas. Luego, se puede utilizar métodos alternativos para la búsqueda de biomoléculas de interés, los cuales corresponden al cribado funcional y cribado molecular de las bibliotecas genómicas correspondientes a las etapas 4 y 5 respectivamente. Finalmente, en la etapa 6 se

realiza la secuenciación de escopeta¹, que facilita el diseño de métodos de cribado molecular [Kueete, 2017].

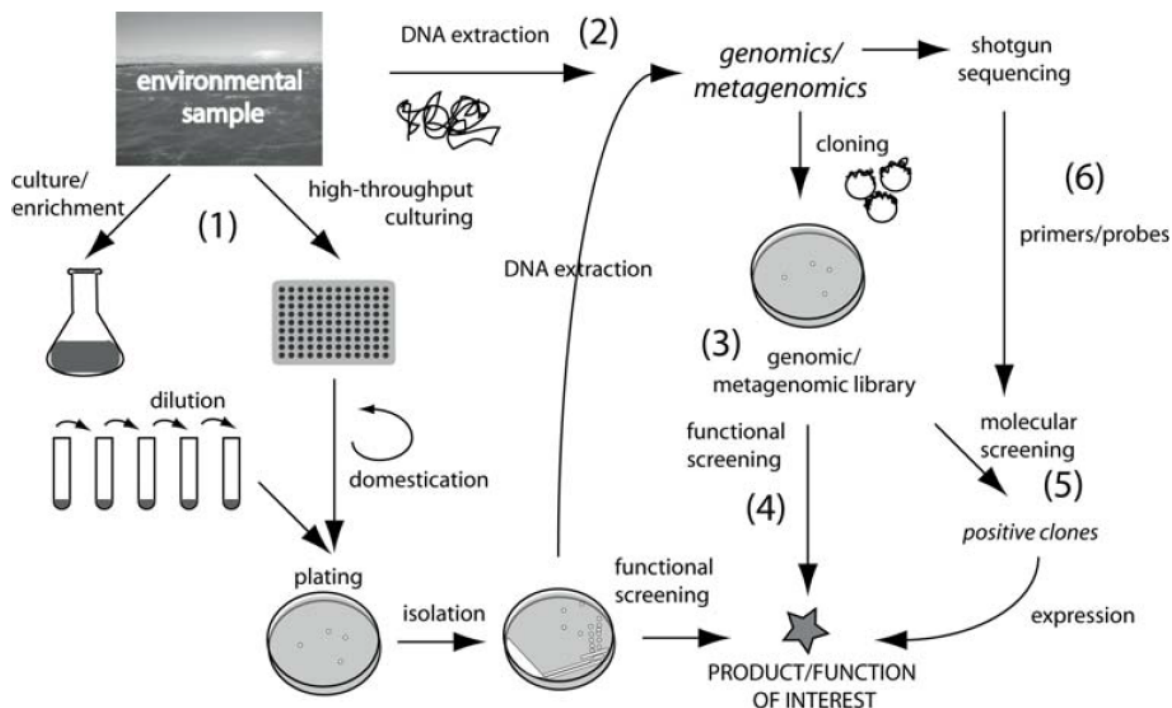


Figura 1: Principales métodos utilizados para la bioprospección de microorganismos
Fuente: *Bioprospection of marine microorganisms: biotechnological applications and methods* [Dionisi et al., 2012].

1.2. Organización

El CB-DAL de la Universidad Técnica Federico Santa María fue creado en el año 2000. Su misión es 'incorporar la dimensión biotecnológica en todas las áreas en que la Universidad Técnica Federico Santa María define su quehacer en ciencia y tecnología'. Además, declara en su visión 'ser referente nacional e internacional en Biotecnología'.

Sus áreas de investigación son:

- Biotecnología de Bioprocesos.
- Biotecnología Ambiental.

¹Técnica de laboratorio para determinar la secuencia de ADN del genoma de un organismo. El método consiste en dividir aleatoriamente el genoma en pequeños fragmentos de ADN que se secuencian individualmente. Un programa de computadora busca superposiciones en las secuencias de ADN, usándolas para volver a ensamblar los fragmentos en su orden correcto para reconstituir el genoma [Green, 2023].

- Biotecnología Asociada a Salud.
- Biotecnología de los Alimentos.
- Biotecnología Apícola.
- Bionanotecnología.

Centrado en las áreas de Biotecnología Ambiental y Biotecnología Asociada a Salud, el Laboratorio de Bioprospección de Actinobacterias Marinas del CB-DAL realiza estudios con el fin de encontrar microorganismos marinos con potencial biotecnológico. Específicamente, sus investigadores ejecutan ensayos sobre distintos cultivos de actinobacterias² para identificar su genoma, metabolismo, actividad antimicrobiana, actividad enzimática y caracterizar su fisiología, con el fin de hallar candidatos a generar antibióticos.

Dichos ensayos generan datos metabolómicos, de identificación, aislamiento, características fisiológicas, de secuenciación genómica y de actividad tanto antibacteriana como enzimática.

1.3. Situación actual

Actualmente, no existe una herramienta que permita registrar los datos mencionados previamente, por lo que cada investigador debe hacerlo por cuenta propia.

Lo anterior, sumado a la falta de organización respecto a la manera de almacenar los datos, la inexistencia de un medio único para realizar dicha tarea y las diferentes maneras de archivar la información; ha tenido como consecuencia el surgimiento de efectos negativos. Dichos efectos corresponden a la dificultad para acceder y/o registrar datos remotamente y la necesidad de que cada persona recuerde donde ha guardado los resultados de sus estudios, lo que va de la mano con la existencia de obstáculos para poder obtener datos de investigaciones realizadas por colegas, ya que como se mencionó, no existe una herramienta que permita realizar consultas de los resultados de ensayos efectuados en actinobacterias y cada investigador tiene un medio propio de almacenamiento. Por lo mismo, no hay consistencia en la forma en que se realizan registros.

La situación planteada se puede observar en el árbol del problema, que fue generado para simplificar las problemáticas del Laboratorio de Bioprospección de Actinobacterias Marinas y evidenciar sus causas, como se puede observar en la figura 2.

²Las actinobacterias son a un grupo de bacterias Gram positivas que pueden ser terrestres y acuáticas. Estas juegan un importante rol en la descomposición de la materia orgánica y una de sus aplicaciones más importantes se da dentro de la farmacología, siendo la generación de antibióticos [Anandan et al., 2016].

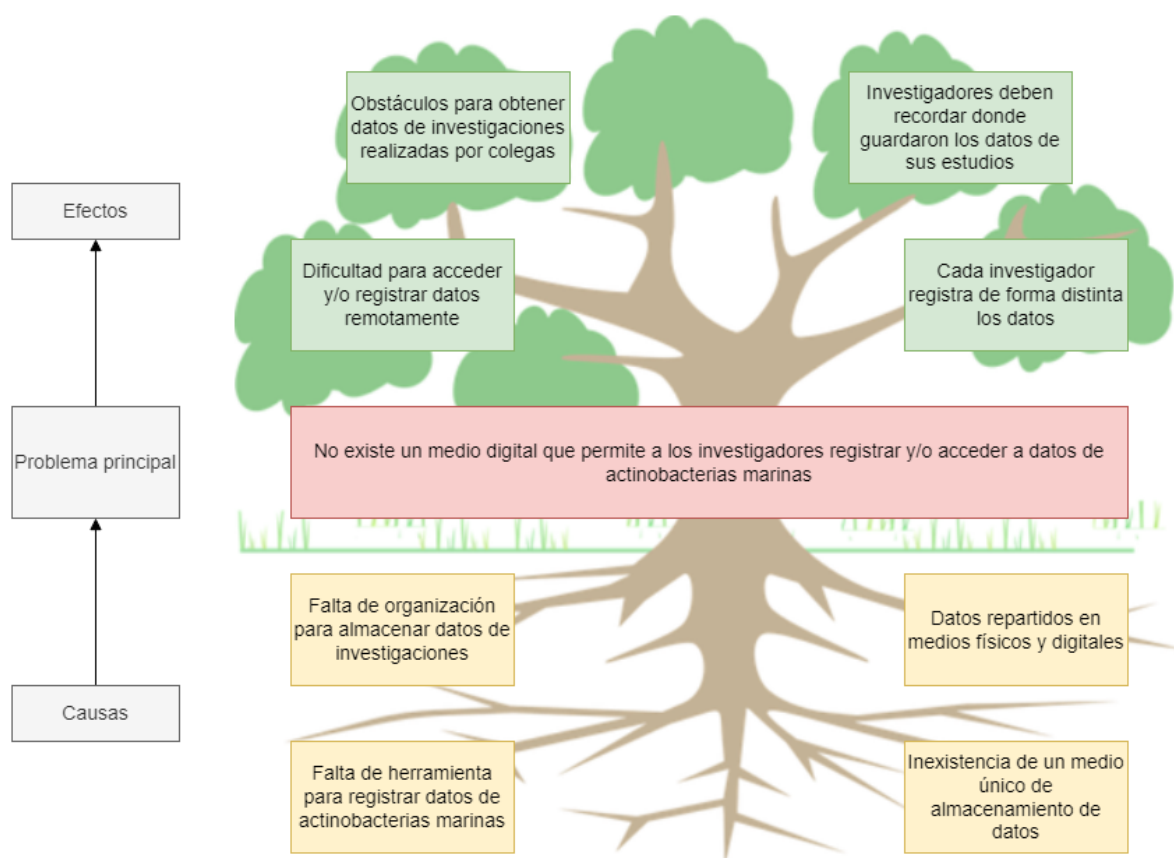


Figura 2: Árbol del problema
Fuente: Elaboración propia.

Es así, que el principal problema corresponde a la no existencia de un medio digital que permita el acceso y/o registro de datos de actinobacterias marinas a los investigadores. Por lo que surge la necesidad de desarrollar *MyMicrobeApp*, una aplicación con el objetivo de integrar y visualizar los datos de interés para los miembros del laboratorio.

1.4. Objetivos

Los objetivos enunciados a continuación corresponden a los de la solución desarrollada y explicada en los siguientes capítulos de la memoria.

1.4.1. Objetivo General

Diseñar y desarrollar una aplicación que permita integrar y visualizar datos derivados de microorganismos marinos.

1.4.2. Objetivos específicos

- Caracterizar los procesos llevados a cabo por el Laboratorio de Bioprospección de Actinobacterias Marinas.
- Definir los procesos a implementar en la aplicación.
- Construir la aplicación en base al diseño realizado previamente.
- Validar la interfaz desarrollada utilizando técnicas centradas en usuarios.

1.4.3. Alcance

La solución comprometida para la presente memoria implica el desarrollo y validación de una aplicación web que permita integrar y visualizar los datos derivados de microorganismos marinos con potencial biotecnológico. Dicha tarea considera el diseño y desarrollo de una base de datos, una interfaz de usuario y una API que cumpla el rol de servidor para responder a las consultas hechas desde la interfaz de usuario.

CAPÍTULO 2

MARCO CONCEPTUAL

Para lograr un mayor entendimiento tanto de la definición del problema como de la propuesta de solución, es relevante conocer trabajos previos describiendo el estado del arte. A su vez, se explican conceptos claves dentro de las sub-secciones de herramientas de trabajo, UX/UI y metodología de trabajo.

2.1. Estado del arte

La construcción de la aplicación *MyMicrobeApp* ocupa el diseño e implementación de una base de datos de actinobacterias que cumpla con las necesidades de sus usuarios. En consecuencia, es de utilidad conocer bases de datos dedicadas a bacterias. A continuación se presentan dos de las bases de datos más importantes en la actualidad.

- BacDive

Es la base de datos más grande del mundo para información procariota³ estandarizada. Permite indagar en la taxonomía, morfología, medios de cultivo, metabolismo, origen y secuenciación genética de más de ochenta mil cepas⁴ de bacterias y arqueas⁵. Para brindar dicha información es provista de datos por otras bases de datos especializadas en algunos de los campos nombrados previamente.

Cuenta con dos medios para acceder a sus datos: una API documentada que ofrece ejemplos en R⁶ y Python⁷; y una interfaz gráfica de usuario que posee diversas herramientas de búsqueda avanzada y un *dashboard* que resume los datos almacenados.

Dichos datos están disponibles de forma gratuita y pueden descargarse en formatos CSV, JSON y PDF, entre otros [Reimer *et al.*, 2022].

- gcType

Es una plataforma que integra los datos del GCM con información disponible públicamente de otras bases de datos. En esta plataforma se asocia información taxonómica,

³Se conoce como procariota a organismos unicelulares que tienen células procariotas. Una célula procariota es un tipo de célula que no tiene un núcleo verdadero ni orgánulos unidos a la membrana [Buckley, 2020].

⁴Grupo de microorganismos que pertenecen a la misma especie y comparten ciertas características que no se encuentran en otros miembros de la especie [Clinical Info, 2022].

⁵Grupo de microorganismos procariotas unicelulares que, al igual que las bacterias, no presentan núcleo ni orgánulos membranosos internos, pero son fundamentalmente diferentes a estas, de tal manera que conforman su propio dominio o reino [Wikipedia contributors, 2022a].

⁶Lenguaje de programación enfocado al análisis estadístico.

⁷Lenguaje de programación de alto nivel e interpretado.

fenotípica y fisiológica con las cepas tipo para que los usuarios puedan realizar análisis genómicos y funcionales.

Ofrece opciones de búsqueda avanzada, que permiten acceder de forma gratuita a sus datos, los cuales cuentan con listas de especies válidamente publicadas, listas de cepas tipo, una lista de cepas tipo por colección de cultivo y una lista completa de secuencias genómicas, y del gen 16S rRNA⁸.

Así, la plataforma permite buscar información de microorganismos, ensamblar el genoma microbiano, identificar nuevas especies y mostrar el estado actual de secuenciación de microorganismos procariotas por diferentes órdenes, apoyándose en datos propios del GCM y de catálogos de cultivo de renombre internacional [Shi *et al.*, 2021].

Por otra parte, en el ámbito nacional se destaca 'Atacama Database', una base de datos que comprende la rica diversidad de microorganismos del desierto de Atacama. Esta cuenta con más de 2300 cepas de microorganismos, los cuales incluyen arqueas, bacterias, hongos, microalgas y microorganismos no identificados taxonómicamente.

La base de datos está organizada en varias categorías, de manera que se genera un registro por cada microorganismo, cubriendo su clasificación, datos de aislamiento, morfología, fisiología, genoma e información metabólica.

Sus fuentes de datos incorporan más de 100 publicaciones literarias y fuentes de libre acceso. Por lo mismo, contiene enlaces a bases de datos externas para acceder directamente a sus datos.

Para acceder a sus datos cuenta con una interfaz gráfica de usuario que provee diversas herramientas para la búsqueda. Estas son:

- **Búsqueda básica**

Herramienta que permite acceder a los datos de un microorganismo buscándolo por su nombre o identificador único.

- **Visor de árbol taxonómico**

Mecanismo de búsqueda implementado a través de un árbol filogenético. Este contiene los datos de los microorganismos separados por dominio. Permite visualizar los diversos microorganismos en el árbol y acceder a los datos de uno en específico tras pincharlo en el árbol.

- **Búsqueda avanzada**

Permite la búsqueda por exploración taxonómica (dominio, filo, clase, orden, familia, género y especie), características del sitio de recolección como lo son el entorno de aislamiento, características del entorno o tipos de muestra; y cepas con características

⁸Gen utilizado para la identificación filogenética de procariotas [Wikipedia contributors, 2022b].

específicas como lo son cepas tipo, requerimientos de oxígeno y productos metabólicos.

- **Búsqueda por geolocalización**

Se pueden visualizar los nichos geológicos del desierto de Atacama por medio de polígonos generados con la API de *Google Maps*. Además, se pueden ubicar las colecciones de microorganismos en el mapa por medio de banderas, que fueron creadas indicando su posición a través de su latitud y longitud.

Aquellos datos son de libre acceso y pueden descargarse en formato CSV [Contador *et al.*, 2020].

El tipo de DBMS y DBMS en específico utilizado para cada una de las bases de datos descritas se presentan en la tabla 2.

Tabla 2: Información de las bases de datos incluidas en el estado del arte. Fuente: Elaboración Propia.

Base de datos	Tipo de DBMS	DBMS utilizado
BacDive	RDBMS	MariaDB
gcType	RDBMS	MySQL
Atacama Database	NoSQL	MongoDB

2.2. Herramientas de trabajo

La presente memoria comprende el desarrollo de una aplicación web. Por consiguiente, es necesaria la definición de diversos términos y herramientas asociados a dicha actividad.

2.2.1. Framework

Un *framework* consiste en una estructura base que se puede utilizar para desarrollar *software* sin tener que lidiar con la implementación de lógica adicional e innecesaria desde cero. Los *frameworks* proveen una plantilla, lo que permite centrarse en el desarrollo agregándole características y/o funcionalidades para generar un proyecto personalizado.

Las principales razones para utilizar un *framework* son:

- Es fácil colaborar con otros desarrolladores debido a que se cuenta con una estructura establecida.
- Ayuda a evitar el generar código duplicado o redundante.

- Facilita la depuración a desarrolladores que no han escrito el código.
- Es confiable, ya que continuamente es mantenido.
- Reduce el tiempo de desarrollo de una aplicación, ya que permite centrarse en la escritura de código específico del proyecto [Olawanle, 2022].

2.2.2. Base de datos

Una base de datos es una colección organizada de información estructurada, o datos, comúnmente almacenados electrónicamente en un sistema computacional. Usualmente, una base de datos es controlada por un sistema de gestión de bases de datos (DBMS), el cual es un software que sirve como interfaz entre la base de datos y sus usuarios finales o programas. Este permite recuperar, actualizar y administrar como se organiza y optimiza la información. Además, facilita la supervisión y control de la base de datos [Oracle, 2022].

Hay tres clases de DBMS basados en el modelo relacional de E.F.Codd [Codd, 1970] Estos son:

1. RDBMS

Cuenta con un esquema de datos estructurado, es decir, hay un formato estandarizado para almacenar los datos, el cual no puede ser modificado.

Sus transacciones se rigen por el principio ACID:

- *Atomicity* (atomicidad): Las transacciones son completas; o se finalizan en su totalidad o no se hacen.
- *Consistency* (consistencia): Con o sin transacciones la base de datos se mantiene en un estado estable, es decir, se garantiza la integridad de los datos.
- *Isolation* (aislamiento): Múltiples transacciones no interfieren entre ellas.
- *Durability* (durabilidad): Los cambios generan un efecto permanente en la base de datos.

Además, utiliza SQL como lenguaje para consultar y mantener la base de datos.

2. NoSQL

Esta clase, a diferencia de la anterior se caracteriza por su falta de esquema. Por consiguiente, soporta datos estructurados, semi-estructurados y no estructurados.

Sus transacciones se rigen por el principio BASE, el cual asegura la confiabilidad de las bases de datos a pesar de la pérdida de consistencia, ya que la información solo es actualizada cuando es necesario.

Los 4 tipos de bases de datos NoSQL más comunes son:

- a) De familia de columnas: Almacenan registros de datos en columnas, accediendo a ellos por medio de un identificador único.
- b) Documental: Guardan datos con archivos de documentos semi-estructurados que tienen formatos específicos como XML y JSON.
- c) Clave-valor: Almacenan datos bajo la estructura clave-valor, en donde la clave es un *string*⁹ único y el valor corresponde al dato real, que puede ser un número, *string*, archivo JSON, etc.
- d) De grafo: Guardan datos basándose en una estructura de grafo [Chen y Lee, 2018].

Por otro lado, su ventaja se centra en la escalabilidad, ya que permiten manejar grandes volúmenes de datos, los cuales van en aumento.

Cabe destacar que el prefijo 'No' de NoSQL no indica que esta clase de bases de datos no utilice SQL como lenguaje de consultas, sino que el prefijo 'No' es una sigla en inglés para *not only*, que quiere decir 'no solo'. Es así, que las bases de datos de la clase NoSQL no tienen como principal lenguaje de consultas a SQL; lo que no quiere decir que algunas no puedan soportarlo.

3. NewSQL

Esta preserva las propiedades del principio ACID de la clase RDBMS, pero a su vez brinda la escalabilidad de NoSQL. Para lograrlo, se utilizan mecanismos de replicación [Gill, 2022].

2.2.3. Front-end

El *front-end* o desarrollo del lado del cliente se encarga de dar forma a la parte frontal de un sitio web o aplicación, es decir, se encarga de construir lo que los usuarios perciben al explorar un sitio, con lo que pueden interactuar. Es así, que el *front-end* sirve para realizar la interfaz de un sitio web, desde su estructura hasta los estilos [Coppola, 2022].

Para el desarrollo de la interfaz de un sitio web se utilizan tecnologías web estándar, como lo son JavaScript¹⁰, HTML¹¹ y CSS¹². Además, frecuentemente los desarrolladores se sirven de algún framework o biblioteca de JavaScript para facilitar su construcción. Los más populares según [State of JavaScript, 2022] son:

1. React

⁹Tipo de dato. Es una secuencia de caracteres usada para representar texto.

¹⁰Lenguaje de programación que permite implementar complejas características en páginas web [Contributors, 2022].

¹¹Lenguaje de marcado que permite estructurar el contenido de una página web.

¹²Lenguaje de estilo utilizado para aplicar estilos al contenido del HTML.

Es una biblioteca de JavaScript para la construcción de interfaces de usuario. Se basa en el uso de componentes reutilizables, que manejan su propio estado y de vistas declarativas¹³ que hacen el código más predecible y fácil de depurar [React, 2023].

2. Angular

Es un framework multiplataforma basado en componentes construido en TypeScript¹⁴, que permite desarrollar aplicaciones escalables. Cada componente cuenta con una plantilla HTML en la que se indica como el contenido debe ser renderizado [Angular, 2022].

3. Vue

Es un framework de JavaScript para construir interfaces de usuario. Al igual que React usa un modelo de programación declarativo y basado en componentes [Vue, 2023].

4. Svelte

Es un framework que a diferencia del resto cuenta con una etapa de compilación al momento de construir la aplicación, en vez de realizar su trabajo en el navegador. Esto hace que sea más eficiente [Svelte, 2023].

Sus ventajas y desventajas se presentan en la tabla 3.

Tabla 3: Ventajas y desventajas de los frameworks de front-end más populares. Fuente: Elaboración Propia. Basada en la información de [Ivanovs, 2023].

Biblioteca/Framework	Ventajas	Desventajas
React	Fácil comienzo debido a los múltiples recursos disponibles, como tutoriales y cursos. Elementos fáciles de utilizar producto de su estructura de componentes. Su control de versiones proporciona avisos sobre código obsoleto.	Falta de documentación. Solamente se puede usar para el desarrollo UI.

¹³Declarativo hace referencia a que no se indica como se debe renderizar la interfaz por medio de instrucciones, si no que la misma biblioteca o framework se encarga del renderizado.

¹⁴Lenguaje de programación fuertemente tipado construido sobre JavaScript, al que le agrega una sintaxis adicional [TypeScript, 2023].

Angular	Las plantillas modulares permiten un escalado dinámico de las aplicaciones. Incorpora gestión de estado, ruteo, servicios, etc. Multiplataforma.	Alta curva de aprendizaje. Inadecuado para proyectos pequeños. No es muy querido por la comunidad en comparación al resto.
Vue	Enlace de datos reactivo para aplicaciones en tiempo real. Componentes reutilizables. Fácil de usar.	Complicada implementación multiplataforma. La falta de rigor puede conducir a un código poco estructurado y difícil de mantener. Pocos complementos.
Svelte	Se necesita menos código para construir un componente. Mejor rendimiento. Todo el código es escrito en JavaScript (o TypeScript), HTML y CSS nativos.	Falta soporte por parte de la comunidad. Puede dificultar la escalabilidad. Falta de página dedicada a la seguridad de la aplicación.

Por otra parte, son de utilidad los *frameworks*/bibliotecas de CSS, ya que permiten agilizar el proceso de desarrollo de la interfaz de usuario al proporcionar una base de elementos de diseño listos para usar y personalizar, los cuales permiten crear diseños web coherentes y responsivos. Los 3 *frameworks*/bibliotecas de CSS más populares (en base a la cantidad de estrellas que poseen en GitHub) son:

1. Bootstrap

En su quinta versión, es el *framework* de CSS más popular para el desarrollo de sitios web responsivos. Este permite la construcción de sistemas de grilla, formularios, botones, entre otras posibilidades.

Además, en la actualidad se encuentran disponibles millones de plantillas de sitios web hechas en Bootstrap, lo que facilita el agilizar aún más el desarrollo.

2. Material UI

Material UI es una biblioteca de CSS que implementa el sistema de diseño de Google *Material Design*, que se basa en la aplicación de conceptos científicamente probados en un concepto de diseño web simple y conciso.

3. Tailwind CSS

Tailwind CSS cuenta con miles de plantillas y elementos. Además, es uno de los *frameworks* más livianos y es completamente modular, es decir, el desarrollador puede usar solo las partes que necesita [Silkalns, 2023].

Cabe destacar que los 3 *frameworks*/bibliotecas de CSS descritos cuentan con una documentación muy buena y detallada.

2.2.4. Back-end

El *back-end* o desarrollo del lado del servidor se encarga de procesar toda la información que alimenta al *front-end*, es decir, corresponde al código que permite el correcto funcionamiento de una página web o aplicación. Este se compone de marcos, bases de datos o códigos, con los que el usuario no interactúa directamente y no se percata de su existencia [Coppola, 2022].

Los frameworks de back-end más populares para construir una API REST¹⁵ son:

- Spring Boot

Es un framework de código abierto que permite a los desarrolladores construir aplicaciones web y móviles. Se puede utilizar sin tener que configurar sus bibliotecas.

- Ruby on Rails

Fue originalmente desarrollado como framework MVC¹⁶. Ofrece aplicaciones con alto rendimiento. Muchos de sus conceptos fueron replicados en frameworks de Python y PHP.

- Flask

Es un framework de Python más explícito que Django y más fácil de utilizar.

- Django Rest

Es una herramienta personalizable que facilita la creación de APIs. Se basa en las vistas basadas en clases de Django.

- Express.js

Es un framework de código abierto para Node.js que facilita el proceso de desarrollo ofreciendo un conjunto de herramientas, funciones y complementos útiles.

¹⁵API que utiliza peticiones HTTP para acceder a los datos [Gillis, 2020].

¹⁶Estilo de arquitectura de software que separa en tres capas los datos de la aplicación, la interfaz de usuario y la lógica de control

- Fastify

Es un framework web dedicado a proveer la mejor experiencia posible al desarrollador. Se caracteriza por su poderosa arquitectura de complementos y una sobrecarga mínima.

- Play Framework

Es un framework de aplicaciones web para crear aplicaciones modernas y robustas. Se basa en un tipado dinámico.

- Gin

Es un framework para la construcción de aplicaciones web y micro-servicios. Contiene funcionalidades comunes de los frameworks de desarrollo web como lo son el ruteo, soporte de *middleware*, renderizado, etc.

- Phoenix

Es un framework de elixir que implementa el modelo MVC. Tiene canales para funciones en tiempo real que pre-compilan plantillas, lo que hace al sitio fluido y fácil de recorrer.

- FastAPI

Es un framework de Python centrado en la creación de APIs REST. Soporta completamente programación asíncrona¹⁷ [Kaur, 2022].

Las ventajas, desventajas y el lenguaje de programación que se utiliza para cada uno de los frameworks recientemente descritos se presentan en la tabla 4.

Tabla 4: Ventajas y desventajas de los frameworks de back-end más populares para la construcción de una API REST. Fuente: Elaboración Propia. Basada en la información de [Kaur, 2022].

Framework	Lenguaje de programación	Ventajas	Desventajas
Spring Boot	Java	Rápido de cargar por su asignación de memoria mejorada.	No es compatible con proyectos de Spring anteriores y no hay herramientas de migración.

¹⁷La programación asíncrona otorga la capacidad de diferir la ejecución de una función a la espera de que se complete una operación y así evitar que se bloquee la ejecución hasta que se haya completado la tarea en cuestión [Corzo, 2017].

		Fácilmente configurable con anotaciones y configuraciones XML. Tiene un servidor incorporado por lo que es fácil de ejecutar.	El tamaño binario puede crecer por las dependencias predeterminadas.
Ruby on Rails	Ruby	Permite un rápido desarrollo con una mínima cantidad de errores. Código abierto con muchas herramientas y bibliotecas disponibles. Diseño modular con un sistema de gestión de paquetes eficiente.	Difícil de escalar. Soporte limitado de múltiples subprocesos para algunas bibliotecas. Escasa documentación.
Flask	Python	Servidor de desarrollo incorporado y depurador rápido. Soporte integrado para <i>testing</i> unitario. Única base de código.	Falta de herramientas y extensiones incluidas. Usualmente se debe personalizar el código. Riesgos de seguridad. Código difícil de mantener.
Django Rest	Python	API accesible por la web. Permite autenticación de usuarios con OAuth2 ¹⁸ . Extensa documentación y fácil despliegue.	Alta curva de aprendizaje. No soporta llamadas asíncronas. Los serializadores son lentos y poco prácticos para validar JSON.
Express.js	JavaScript	Buena documentación. Rápido escalado. Ampliamente usado y buen soporte de la comunidad.	Falta de seguridad. Problemas en los <i>callbacks</i> .

¹⁸Estándar de autorización de APIs que permite compartir información entre sitios sin tener que compartir la identidad.

Fastify	JavaScript	Fácil para desarrollar. Buen rendimiento y altamente escalable. Bajos costos de operación.	Falta de documentación y soporte de la comunidad. Poco uso en la industria.
Play Framework	Java, Scala	Interfaz de usuario intuitiva. <i>Testing</i> de la aplicación simplificado. Desarrollo rápido en múltiples proyectos.	Alta curva de aprendizaje. Muchos complementos inestables. Falta de compatibilidad con versiones anteriores.
Gin	Golang	Buen rendimiento. Fácil obtención del código de estado del método HTTP. Fácil validación para JSON.	Falta de documentación. Syntaxis extensa.
Phoenix	Elixir	Filtra datos de forma segura y eficiente. Buen rendimiento.	Necesita un tiempo de procesamiento. Se necesita conocimiento previo del lenguaje de programación Erlang.
Fast API	Python	Alto rendimiento. Fácil de programar y en poco tiempo. Soporta programación asíncrona.	Pobre validación de solicitudes. No soporta instancias únicas. Archivo principal lleno de código.

2.2.5. Herramientas seleccionadas

Para la base de datos se escoge MongoDB, un DBMS NoSQL de código abierto que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar diversas formas de datos, brindando un entorno altamente escalable para aplicaciones y servicios multiplataforma. Sus documentos tienen un formato JSON binario (BSON), que le permite almacenar varios tipos de datos y distribuirse en diferentes sistemas [IBM, 2023].

Algunas de las ventajas de los datos en documentos de tipo BSON son:

- Eficiencia

Al ser BSON un formato binario de almacenamiento de datos, su tamaño es mucho menor en comparación a documentos con formato de texto, como lo es JSON. Esto se traduce en un mejor rendimiento del DBMS.

- Representación natural de los datos

Los documentos son capaces de representar las relaciones complejas y jerárquicas que se dan en el mundo real.

- Enriquecimiento de consultas a la base de datos

MongoDB permite manejar documentos anidados, es decir, documentos que cuenten con propiedades cuyo valor referencie a otro documento. Es así, que permite realizar consultas complejas sin la necesidad de realizar uniones como en las RDBMS.

Por otra parte, para el back-end de la aplicación se decide utilizar Node.js con Express.js, debido a su buena documentación, facilidad para escalar aplicaciones y que el memorista cuenta con experiencia previa en su uso.

Finalmente, para el front-end se selecciona React, debido a los múltiples recursos de aprendizaje que hay disponibles en plataformas *online*, a su estructura de componentes que permite la reutilización de código y que el memorista es experimentado en dicha biblioteca de JavaScript. Por su parte, se elige Material UI como biblioteca de CSS por las amplias posibilidades que ofrece en diseño responsivo, plantillas y componentes.

2.3. UX/UI

UX y UI son términos usualmente utilizados en el front-end al momento de diseñar y desarrollar una interfaz de usuario. Sin embargo, comúnmente son confundidos o no queda en evidencia su diferencia, por lo que se procede a su definición.

2.3.1. UX

UX hace referencia a la experiencia emocional que tienen los usuarios al interactuar con una marca, producto o servicio. Los productos diseñados pensando en la experiencia del usuario son fáciles de usar y brindan una experiencia positiva. Para lograrlo, se debe conocer tanto al usuario como a sus necesidades, objetivos y dificultades. En lo digital a UX se le asocia con la facilidad de uso y que el usuario consiga cumplir un objetivo.

La experiencia de usuario puede ser positiva, neutra o negativa. Para generar una experiencia de usuario positiva se incluye elementos como la gestión de proyectos, investigación de usuarios, arquitectura de la información, diseño visual, estrategia de contenido, análisis de accesibilidad y UI.

2.3.2. UI

UI es un enfoque más técnico que busca optimizar la interacción entre el usuario y la plataforma. Se basa en el diseño visual, arquitectura de la información y el diseño de la interacción. Así, UI trata de diseñar interfaces, haciéndolas lo más intuitivas posible, para mejorar la experiencia de usuario [MDS, 2022].

2.3.3. UX vs UI

De lo anterior se desprende que UX se enfoca en como la marca, producto o servicio hace sentir al usuario, mientras que UI pone énfasis en como luce la interfaz y viene a ser un subconjunto de UX, lo que se observa en la figura 3.

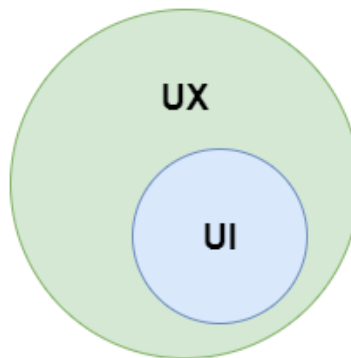


Figura 3: Relación entre UX y UI.
Fuente: Elaboración propia.

2.3.4. Heurísticas de Nielsen

En el ámbito de la usabilidad¹⁹ las heurísticas²⁰ de Nielsen corresponden a 10 principios para el diseño de interfaces de usuario, las cuales sirven como reglas o pautas para generar tanto una buena interfaz de usuario (UI) como una buena experiencia de usuario (UI). Aquellos principios son:

1. Visibilidad del estado del sistema

El usuario siempre debe estar informado sobre el estado del sistema, es decir, de lo que está pasando. Esto se logra por medio de retroalimentación, la cuál debe ser entregada en un tiempo razonable.

2. Similitud entre el sistema y el mundo real

El sistema debe hablar el idioma de los usuarios haciendo uso de términos que les sean familiares, de manera que la información aparezca naturalmente y siguiendo un orden lógico.

3. Control y libertad del usuario

Los usuarios pueden elegir funciones del sistema por error y necesitan una 'salida de emergencia' claramente identificada. Por tanto, el sistema debe brindarle al usuario la capacidad de hacer o deshacer acciones.

4. Consistencia y estándares

Se debe seguir convenciones de la plataforma para que el usuario no tenga que adivinar si diferentes palabras, situaciones o acciones significan lo mismo.

5. Prevención de errores

Es preferible un diseño que evite problemas a buenos mensajes de error. Se debe eliminar condiciones que lleven al error u otorgar opciones de confirmación a los usuarios antes de que cometan errores.

6. Reconocimiento más que recuerdo

El usuario no debería tener que recordar información de un apartado a otro. Se debe minimizar su carga de memoria haciendo visibles los objetos, acciones y opciones que le permitan instruirse sobre el uso del sistema.

¹⁹Según el estándar ISO 9241-11 se define la usabilidad como "la medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado".

²⁰En ingeniería, una heurística es un método basado en la experiencia que puede utilizarse como ayuda para resolver problemas de diseño. Mediante su uso, es posible resolver más rápidamente problemas conocidos o similares a otros conocidos [Nebot, 2019].

7. Flexibilidad y eficiencia de uso

Es recomendable ofrecer aceleradores para las interacciones de los usuarios. Se debe permitir a los usuarios personalizar sus acciones frecuentes.

8. Diseño estético y minimalista

Hay que evitar entregar información irrelevante o que rara vez se necesite. Cada unidad de información adicional compite con las unidades de información relevantes y disminuye su visibilidad relativa.

9. Ayudar a los usuarios a diagnosticar, reconocer y recuperarse de errores

Los mensajes de error deben ser claros, precisos y sugerir constructivamente una solución.

10. Ayuda y documentación

Aunque es mejor que el sistema pueda ser utilizado sin documentación, podría ser necesaria su ayuda. Por lo que este tipo de información debe ser fácilmente accesible, enfocada en la tarea del usuario, listar pasos a seguir y no ser muy extensa [Nielsen, 2005].

Dichos principios sirven como lineamientos a tener en cuenta en las etapas de diseño y construcción de la interfaz de la aplicación.

2.4. Metodología de trabajo

2.4.1. Metodologías populares

Para diseñar y desarrollar *MyMicrobeApp* es importante seleccionar la metodología más adecuada para cumplir con los objetivos establecidos. En consecuencia, a continuación se explican las cinco metodologías más populares para la construcción de una aplicación.

1. Metodología en cascada

Corresponde a una metodología secuencial, en la que hasta no haber terminado una etapa de trabajo no se puede avanzar a la siguiente. Su mayor inconveniente es que no hay vuelta atrás, por lo que si se desea realizar cambios solo se pueden hacer en la etapa de mantención, que surge después del lanzamiento del producto o servicio.

2. Metodología de prototipos

Es una mejora de la metodología en cascada, que permite realizar cambios en la etapa de diseño por medio de la implementación de prototipos que faciliten el entendimiento de los requerimientos del proyecto.

3. Metodología en espiral

Es una combinación de la metodología en cascada y la aplicación de iteraciones, que depende en todo momento de los riesgos del proyecto.

El proceso a seguir pasa por las etapas de conceptualización, desarrollo, mejoras y mantenimiento. Dentro de cada etapa se tiene las fases de planificación, análisis de riesgos, desarrollo y evaluación [Roche, 2020].

4. Metodología ágil

Es una metodología flexible enfocada en iteraciones, las cuales buscan la reducción de errores. Por ello, se divide el proyecto en una serie de cortos periodos de tiempo.

Durante el ciclo de vida del desarrollo pueden realizarse cambios a los requisitos, los cuales surgen de la colaboración con el cliente.

5. Metodología Lean Startup

Se utiliza un MVP para comenzar con el desarrollo de la aplicación. Esto permite crear, realizar seguimiento y aprender de la aplicación que se está desarrollando [KMS Solutions, 2022].

2.4.2. Metodología seleccionada: Personal Kanban

Kanban es un *framework* de metodología ágil usado para visualizar el flujo de trabajo y maximizar la eficiencia.

Un tablero Kanban consiste en una serie de tarjetas y columnas, que representan tareas individuales y sus estados respectivamente [Hakoune, 2022]. Un ejemplo de esto se da en la figura 4, en donde las columnas representan los estados *To Do* (Por Hacer), *In Progress* (En Progreso) y *Done* (Hecho).

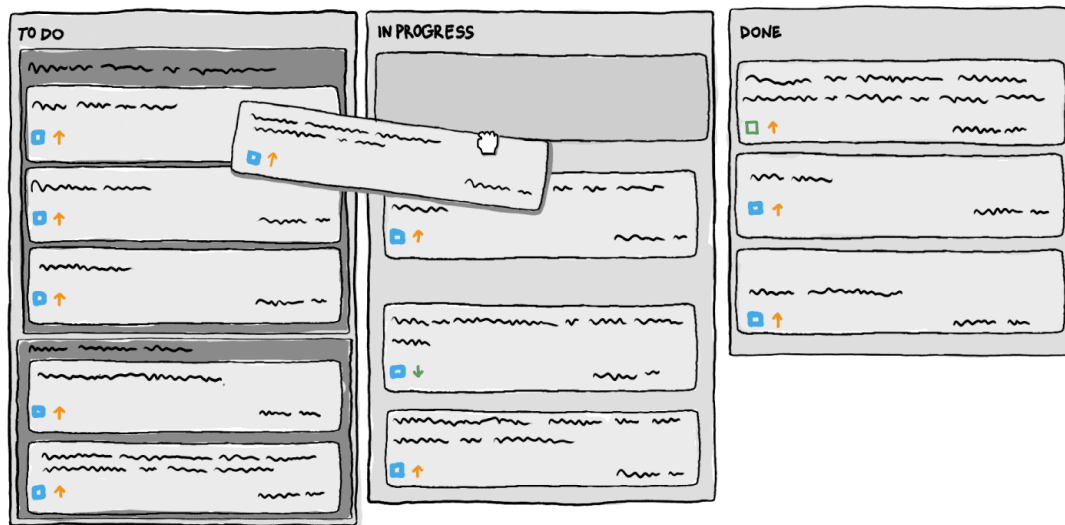


Figura 4: Ejemplo de tablero Kanban.

Fuente: Visualiza tu flujo de trabajo con tableros en Jira [Cooke, 2022].

Para el caso de la presente memoria se selecciona específicamente *Personal Kanban* una adaptación de Kanban que permite que un proyecto sea gestionado y llevado a cabo por una sola persona.

Personal Kanban tiene dos reglas:

1. Visualizar el *backlog*²¹

Es sumamente necesario dedicar una cantidad considerable de tiempo y energía mental a identificar las tareas que no se han hecho, que permitirán cumplir con los objetivos del proyecto. Esto permite que se entienda como las tareas están interrelacionadas, agrupar eficientemente tareas similares y priorizar efectivamente.

2. Limitar el trabajo en progreso (WIP)

Se debe limitar el número de tarjetas 'En Progreso' a un máximo de 2 o 3, para minimizar el estrés.

Además, para gestionar el tiempo adecuadamente es relevante entender la Matriz de *Personal Kanban* de la figura 5. Esta cuenta con cuatro cuadrantes:

1. Cuadrante de pánico (*Quadrant of Panic*)

²¹Hace referencia a la lista de tareas sin terminar que deben completarse para lograr un objetivo estratégico.

Corresponde a emergencias, que deben ser tratadas lo más rápido posible. Lo ideal es que no se deba dedicar mucho tiempo a este.

2. Cuadrante de *Kaizen*²² (*Quadrant of Kaizen*)

El enfoque debe estar centrado en este cuadrante, que corresponde a las tareas principales del proyecto.

3. Cuadrante de inversión social (*Quadrant of Social Investment*)

Forman parte de estas actividades urgentes, pero no tan importantes como lo son llamadas telefónicas, mensajes y reuniones.

4. Cuadrante orgánico (*The Organic Quadrant*)

Hace referencia a tareas menores que aportan valor, pero no son tan productivas [Fryrear, 2022]. Sin embargo, dedicar tiempo de vez en cuando a dicho cuadrante permite relajarse al desligarse de tareas más demandantes.

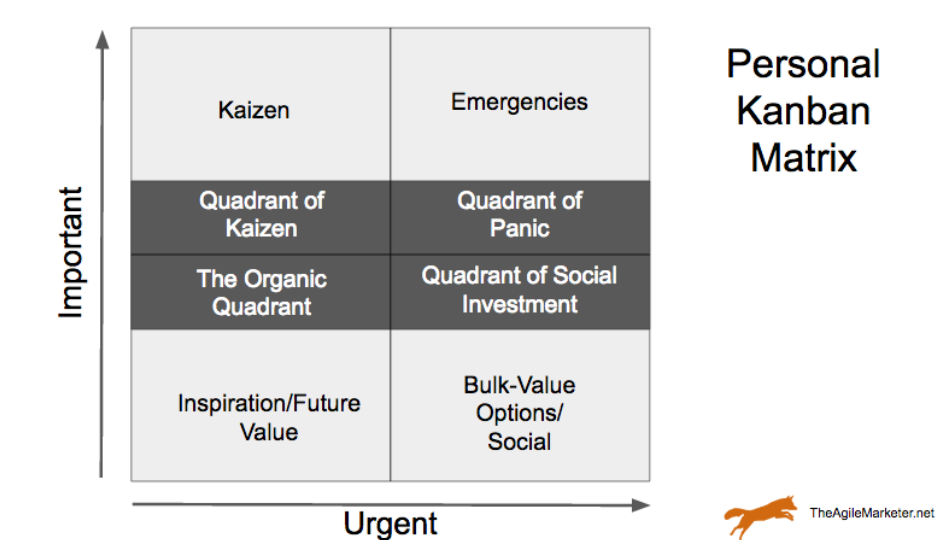


Figura 5: Matriz de Personal Kanban
Fuente: Vuelvete ágil por tu cuenta [Fryrear, 2022].

²²Concepto que hace referencia a la mejora continua [Singh y Singh, 2009].

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. Procesos del laboratorio

Para que MyMicrobeApp sea útil para los miembros del CB-DAL es crucial caracterizar los procesos que comprenden la bioprospección de actinobacterias marinas. El identificarlos y describir en que consisten permitirá diseñar y desarrollar una aplicación que cumpla adecuadamente con las necesidades de sus usuarios.

3.1.1. Caracterización de los procesos del laboratorio

A continuación se caracterizan 12 procesos llevados a cabo en el Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología 'Dr. Daniel Alkalay Lowitt' (CB-DAL) en base a la etapa de experimentación presentada en [Claverías, 2014], [Cumsille, 2015], [González, 2015], [Rojas, 2016], [Vergara, 2016], [González, 2021] y [Vargas, 2022], las cuales corresponden a tesis de los miembros del laboratorio.

1. Muestreo

Al ser los microorganismos de interés actinobacterias marinas, generalmente se obtienen muestras de agua de mar, esponjas o sedimentos marinos. El proceso de muestreo comprende la recopilación de múltiples muestras en distintos sectores de interés. Dichas muestras son recolectadas en tubos Falcón estériles de 50 [mL] y almacenadas en cajas térmicas. Durante este proceso se registran datos como lo son el tipo de muestra, sigla de identificación, lugar de obtención, coordenadas geográficas y profundidad.

2. Aislamiento

Esta etapa consiste en aislar las bacterias desde las muestras recolectadas. Para ello, se utilizan placas Petri con diferentes medios de cultivo sólido donde las muestras se laquean tanto de forma directa como en dilución. Además, con el fin de favorecer el crecimiento de actinobacterias y evitar el crecimiento de hongos y otras bacterias no deseadas, los medios se suplementan con ácido nalidíxico (25 [$\mu\text{g}/\text{mL}$]) y cicloheximida (100 [$\mu\text{g}/\text{mL}$]). Posteriormente, las placas se incuban a 20°C o 30°C por periodos de entre 1 a 4 meses, hasta observar la aparición de colonias bacterianas.

3. Conservación

Para la conservación de las cepas bacterianas aisladas, se realiza un cultivo de las mis-

mas en medio de cultivo TSA²³ complementado con ASW²⁴. Las placas sembradas se incuban a 30°C de 3 a 5 días, dependiendo del crecimiento de cada cepa. Luego, la biomasa obtenida es suspendida en tubos eppendorf y criotubos con una solución de conservación compuesta de medio TSB²⁵ (10 %), ASW y glicerol (20 %). Posteriormente, se crean dos ceparios de mantención, a -20°C y -80°C.

4. Identificación preliminar de Actinobacterias

Para la identificación preliminar de actinobacterias se realiza tanto la prueba de hidróxido de potasio (KOH) como la amplificación por PCR (reacción en cadena de la polimerasa) del gen 16S rRNA con partidores específicos para actinobacterias. No obstante, estos pasos se pueden omitir para un subconjunto de este grupo de bacterias que poseen un crecimiento miceliar característico en placas Petri y que por lo tanto, se identifican fácilmente mediante la observación de su crecimiento. De todas formas, dichos pasos serán descritos a continuación.

- Prueba del hidróxido de potasio (KOH)

Consiste en un método rápido para distinguir entre bacterias Gram negativas y Gram positivas como lo son las actinobacterias. Para esto, se dispone de biomasa de una colonia de bacterias proveniente de un medio sólido en un portaobjetos y se mezcla con una gota de KOH al 3 % [p/v]. Si se observa una solución viscosa significa que la prueba dio positiva y la cepa es Gram negativa, en cambio si la solución es acuosa la cepa es Gram positiva.

- Amplificación por PCR del gen rRNA con partidores específicos para actinobacterias

Macromoléculas comunes a todos los seres vivos como lo son ácidos nucleicos y proteínas, cambian con el tiempo. Por lo que se les considera cronómetros moleculares. Es así, que las diferencias en la secuencia de monómeros (nucleótidos o aminoácidos) que integran macromoléculas homólogas, presentes en dos formas de vida, reflejan su distancia evolutiva. Dicha idea ha sido utilizada durante décadas para establecer relaciones filogenéticas entre seres vivos [del Rosario Rodicio y del Carmen Mendoza, 2004].

El 16S rRNA es una macromolécula ampliamente utilizada en estudios filogenéticos e identificación taxonómica, ya que cuenta con regiones conservadas que son comunes a un gran número de especies bacterianas y regiones variables, que son compartidas por menos especies [LabSerGen, 2022].

En base a lo anterior, el proceso de amplificación por PCR del gen 16S rRNA con partidores específicos para actinobacterias se utiliza para replicar un fragmento de ADN (en este caso de la región V3 a V5) en miles o millones de copias y así poder determinar si las bacterias aisladas corresponden a actinobacterias.

El proceso, representado en la figura 6, cuenta con las siguientes etapas:

²³Medio de cultivo sólido a base de soja [Cherwell Labs, 2023].

²⁴Agua de mar artificial.

²⁵Medio de cultivo líquido a base de soja [Cherwell Labs, 2023].

- a) Desnaturalización inicial
Separación de las cadenas de ADN.
- b) Ciclos de amplificación
Se realizan 30 a 40 ciclos de desnaturalización (separación de las cadenas de ADN), hibridación (unión de los partidores al fragmento de interés en la cadena desnaturalizada) y elongación (replicación del fragmento de la región, por medio de la extensión de la cadena incorporando nucleótidos).
- c) Elongación final
Última extensión de la cadena tras el proceso cíclico de amplificación [Pérez de Castro, 2011].

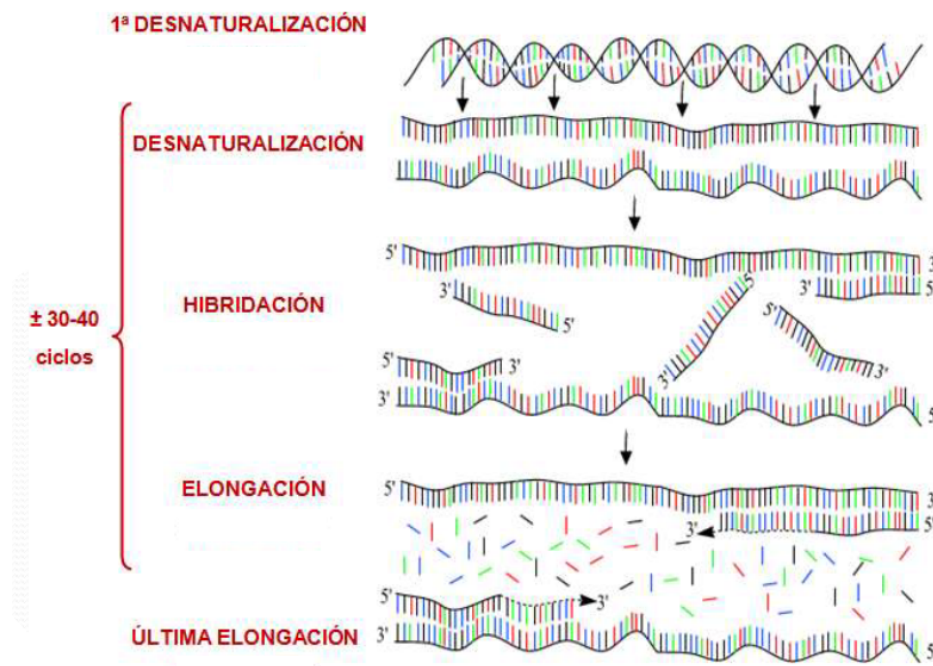


Figura 6: Etapas del PCR

Fuente: PCR: Reacción en cadena de la polimerasa [Viéitez, 2013].

Cada etapa del PCR es realizada a una temperatura y durante un tiempo específico, los cuales se declaran en un perfil térmico, como se puede ver en la figura 7.

Etapa	Temperatura	Tiempo	Ciclos
Desnaturalización Inicial	95°C	5 min	
Desnaturalización	95°C	1 min	
Hibridación	70°C	1 min	35 Ciclos
Elongación	72°C	1,5 min	
Elongación Final	72°C	10 min	

Figura 7: Perfil térmico de PCR para amplificación del gen 16S rRNA específico para Actinobacterias

Fuente: [Rojas, 2016].

5. Amplificación y secuenciación del gen 16S rRNA y análisis filogenético

El proceso de amplificación del gen 16S rRNA es similar al anterior. Su diferencia es que en vez de utilizar partidores específicos para actinobacterias se utilizan partidores universales (específicamente, los partidores 27F y 1492R).

Tras esto, el resultado de la amplificación es purificado y enviado a secuenciar a una empresa externa (en general se encarga a MacroGen Inc en Corea del Sur).

Posteriormente, se determina la relación filogenética de cada cepa bacteriana comparando los productos secuenciados con las secuencias individuales del gen 16S rRNA publicadas en la base de datos del Centro Nacional para la Información Biotecnológica (NCBI). Con esta información se puede determinar el género de cada cepa bacteriana, y si pertenece al filo Actinobacteria.

6. Fermentaciones

El objetivo de este proceso es obtener fermentaciones a partir de cepas de actinobacterias seleccionadas. Para lograrlo se cuenta con varios matraces con distintos medios de cultivo líquido en los que se agrega la cepa. Luego, los matraces se dejan incubando a 30°C y 200 rpm para obtener las fermentaciones periódicamente. El tiempo de cultivo variará dependiendo del crecimiento de cada cepa. En este periodo, las bacterias cultivadas sintetizarán diferentes metabolitos que serán secretados al medio de cultivo.

Un ejemplo de esto se da en la figura 8, en donde se dejó fermentar la cepa *Streptomyces* sp. A-B3.3 en 3 medios de cultivo diferentes, durante 9 días. Dentro de ese periodo de tiempo se realizó el cultivo para obtener las fermentaciones en los días 3, 6 y 9.

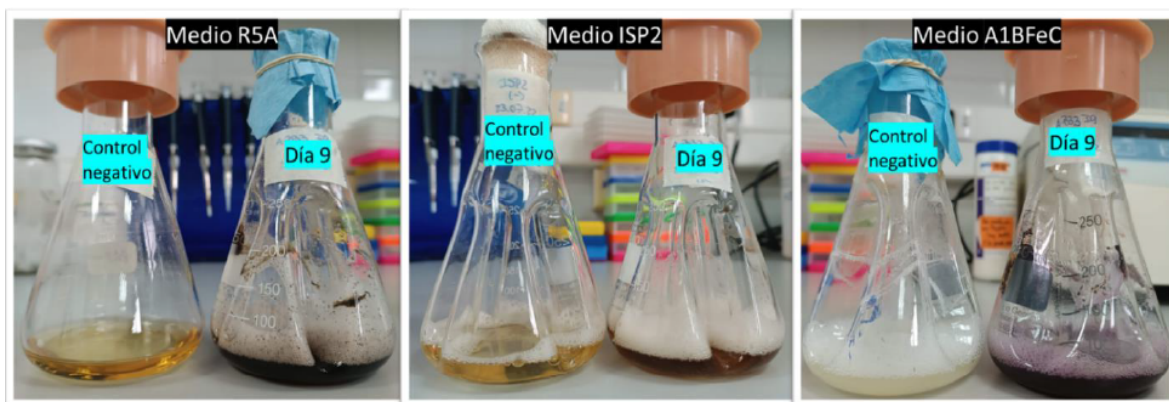


Figura 8: Fermentación de la cepa *Streptomyces* sp. A-B3.3 en 3 medios de cultivo
Fuente: Comparación de cultivos de A-B3.3 en el día 9 respecto a sus controles negativos por cada medio de cultivo utilizado (R5A, ISP2 y A1BFeC) [Vargas, 2022].

7. Extracciones con solventes orgánicos

La extracción con solventes orgánicos tiene por objetivo extraer y concentrar los metabolitos sintetizados por las actinobacterias. Para esta metodología se utiliza como materia prima las fermentaciones obtenidas en el proceso anterior. Dichas fermentaciones son mezcladas con un solvente orgánico, que por lo general es acetato de etilo. Producto de las diferencias de polaridad, se separan los metabolitos secundarios (que pueden o no ser compuestos bioactivos) del cultivo. Posteriormente, se seca la mezcla obteniendo como resultado un extracto crudo (seco), correspondiente a los metabolitos secundarios.

Es importante señalar que, de acuerdo al objetivo de estudio relacionado con la búsqueda de compuestos bioactivos, aquellos extractos que presentan propiedades antibióticas serán de mayor interés para el equipo de investigación, y permitirán seleccionar a las actinobacterias interesantes para una caracterización en mayor profundidad.

8. Ensayos de actividad antibacteriana

Los ensayos de actividad antibacteriana se realizan en base a los extractos crudos obtenidos, producto de las extracciones con solventes orgánicos.

Para realizar los ensayos, previamente se diluye los extractos crudos a una concentración de 5 [mg/mL] en metanol 50 % [v/v]. Luego, estos son puestos en placas Petri con medio LB previamente sembradas con cepas modelo (ej: *Escherichia coli*). En cada placa se puede evaluar la actividad antibacteriana de varios extractos. Para esto, las placas son rotuladas previamente formando cuadrantes de manera que en las líneas horizontales se disponen los extractos de acuerdo al medio de cultivo utilizado y en las líneas verticales, de acuerdo al día de cultivo en que se hizo la extracción, lo que se puede visualizar de mejor manera en la figura 9.

Posteriormente, se analiza la actividad antibacteriana en cada medio de cultivo y por cada día de extracción, para cada cepa modelo.

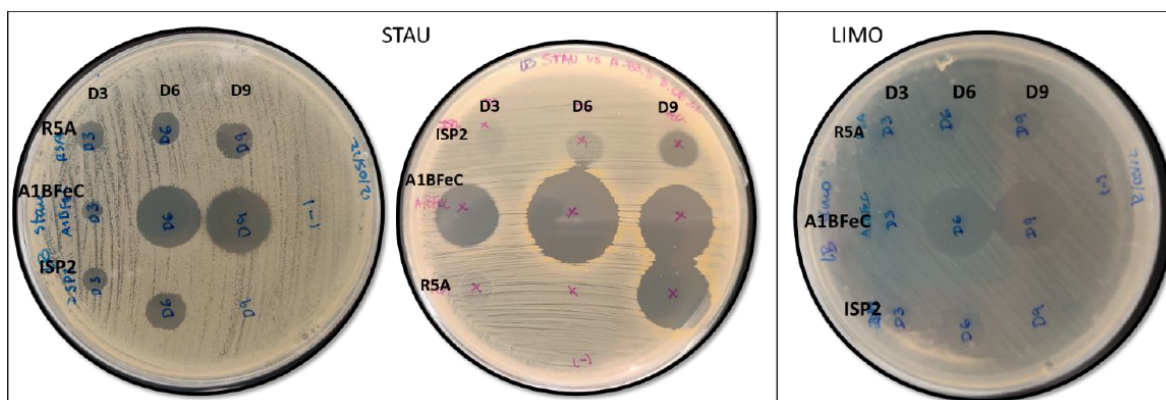


Figura 9: Ensayos de actividad antibacteriana de *Streptomyces* sp. A-B3.3 contra STAU (*Staphylococcus aureus*) y LIMO (*Listeria monocytogenes*).

Fuente: [Vargas, 2022].

9. Ensayos de actividad enzimática

Para evaluar la actividad enzimática de las actinobacterias identificadas se realizan ensayos cualitativos en placas Petri con un sustrato específico de acuerdo a la enzima que se quiera evaluar. Por ejemplo, para evaluar la síntesis de lipasas²⁶ y proteasas²⁷ extracelulares, los medios son complementados con una fuente de lípidos (como tributirina que es un triglicérido) o de proteínas (como gelatina, leche descremada entre otros), respectivamente.

Las actinobacterias se inoculan en las placas Petri que luego se incuban a una temperatura específica entre 24 y 72 horas, dependiendo del crecimiento de cada cepa. Luego se evalúa la producción de enzimas de cada cepa, observando la aparición de un halo alrededor de la colonia. La presencia de un halo es indicativo de la hidrólisis del sustrato, y así se evidencia la síntesis de lipasas extracelulares para el caso del ensayo en placas de lipasas y de proteasas extracelulares para el caso de proteasas.

10. Desreplicación química utilizando cromatografía líquida acoplada a espectroscopía de masas

La desreplicación química corresponde al proceso de identificación de compuestos presentes en una muestra problema, o en el caso del Laboratorio de Bioprospección de Actinobacterias Marinas del CB-DAL, en un extracto crudo.

Para poder realizar la desreplicación química, las muestras son enviadas a laboratorios externos (Fundación Medina de España) para determinar los perfiles metabólicos por cromatografía líquida acoplada a espectroscopía de masas²⁸ de alta resolución (LCHR-MS/MS). Esta técnica analítica entrega como resultado, cromatogramas

²⁶Enzima digestiva producida principalmente en el páncreas, que tiene como función deshacer la grasa de los alimentos en moléculas más pequeñas para que puedan ser absorbidas por el intestino [Lemos, 2022].

²⁷Enzimas que rompen los enlaces peptídicos de las proteínas [Kuddus, 2019].

²⁸Proceso de análisis que combina las técnicas de cromatografía líquida para la separación y espectroscopía de masas para la detección, identificación y cuantificación de compuestos orgánicos en una muestra. En pocas

con datos exactos de la masa/carga de los compuestos identificados, además de espectros UV-Visibles de cada *peak* del cromatograma. Así, cada compuesto presente en los perfiles metabólicos de las muestras puede ser comparado con metabolitos de la base de datos de productos naturales de la empresa para identificar compuestos novedosos (que no muestran coincidencias con ningún compuesto conocido). Luego de realizar el análisis, la entidad externa envía como respuesta los datos obtenidos, los cuales son analizados con técnicas quimioinformáticas.

Cabe destacar, que el grupo de investigación de la Dra. Beatriz Cámara se enfoca en la búsqueda de antibióticos novedosos, por lo que la desreplicación química es clave para determinar si una actinobacteria está produciendo metabolitos no reportados previamente. De esta manera, si una actinobacteria ha presentado actividad inhibitoria de alguna cepa modelo se puede confirmar que su acción es producto de un antibiótico novedoso y no de uno reportado.

11. Secuenciación del genoma

La secuenciación del genoma puede tener diversos objetivos como lo son la determinación de características genómicas presentes en la cepa analizada (síntesis de enzimas, reconstrucción de rutas metabólicas, etc) o la predicción de genes biosintéticos de metabolitos secundarios (como los antibióticos).

Para poder realizar este proceso los principales pasos son:

a) Extracción del ADN genómico

Para extraer el ADN genómico de las actinobacterias seleccionadas primero se realiza un cultivo de las cepas en estudio. Luego, la biomasa de cada cepa es resuspendida en un *buffer* adecuado, disgregada con un vástago y separada por centrifugación. Después, las células disgregadas son tratadas por métodos químicos y enzimáticos para romper su pared celular.

Posteriormente, el ADN es separado del resto de componentes celulares, purificado y precipitado. Finalmente, las muestras son hidratadas y almacenadas hasta su envío al servicio de secuenciación.

b) Secuenciación, ensamblaje y anotación de los genomas.

En esta etapa se utilizan varias herramientas bioinformáticas para cumplir con lo requerido. Primero se lleva a cabo la secuenciación del genoma con plataformas como Illumina NextSeq o PacBio, realizada por servicios externos (Macrogen, Corea del Sur). Luego, las secuencias de ADN obtenidas en '*reads*' son ensambladas utilizando softwares como SPAdes y Unicycler. Subsecuentemente, se evalúa la calidad del ensamblaje junto a la integridad y contaminación del genoma utilizando el software CheckM. Finalmente, se realiza la anotación del genoma²⁹ utilizando el software Prokka.

palabras, se separa las moléculas de la muestra a través de un proceso de ionización y luego se identifican en base a su relación masa/carga [Brajovic, 2020].

²⁹La anotación del ADN de un genoma consiste en la identificación estructural de genes, regiones codificantes y motivos, así como la identificación funcional de esas regiones [Brenes-Guillén, 2019].

12. Predicción de agrupamientos de genes biosintéticos

Tras haber secuenciado, ensamblado y anotado el genoma de la cepa en estudio, es posible predecir y analizar los agrupamientos de genes biosintéticos (BGCs, por su sigla en inglés de Biosynthetic Gene Clusters). Estos agrupamientos o '*clusters*' de genes, corresponden a un conjunto de genes que se encuentran en un mismo contexto genético y tienen el objetivo de codificar, en conjunto, complejos enzimáticos que se encargan de la biosíntesis de metabolitos secundarios.

El proceso de predicción de los BGCs se lleva a cabo ingresando un archivo del genoma (obtenido en el proceso de anotación) al servidor web antiSMASH. Este cuenta con una base de datos interna de agrupamientos de genes biosintéticos, que le permite al usuario realizar una comparación con el genoma provisto y entregar como resultado una lista con todos los agrupamientos de genes biosintéticos detectados por familia en la cepa de estudio, clasificados por familia (ej: policétido sintasa (PKS, polyketide synthase), sintetasa de péptido no ribosomal (NRPS, non ribosomal peptide synthetase), terpenos, etc).

Dicho resultado describe como están agrupados los genes, su tamaño y función. Además, se entrega un porcentaje de similitud entre los agrupamientos de genes biosintéticos detectados respecto a los de la base de datos. Estos resultados, son de gran utilidad para el objetivo de investigación del Laboratorio de Bioprospección de Actinobacterias Marinas, ya que se buscan BGCs novedosos, es decir, aquellos que tienen un bajo porcentaje de similitud.

Para complementar el análisis y poder identificar gráficamente si es que los agrupamientos de genes biosintéticos detectados son novedosos se diseña una red de similitud utilizando un software llamado BiG-SCAPE en la que los BGCs son representados como nodos, y las conexiones entre estos, como líneas (ver figura 10). Los agrupamientos detectados que son menos cercanos a los BGCs de la base de datos son más novedosos.

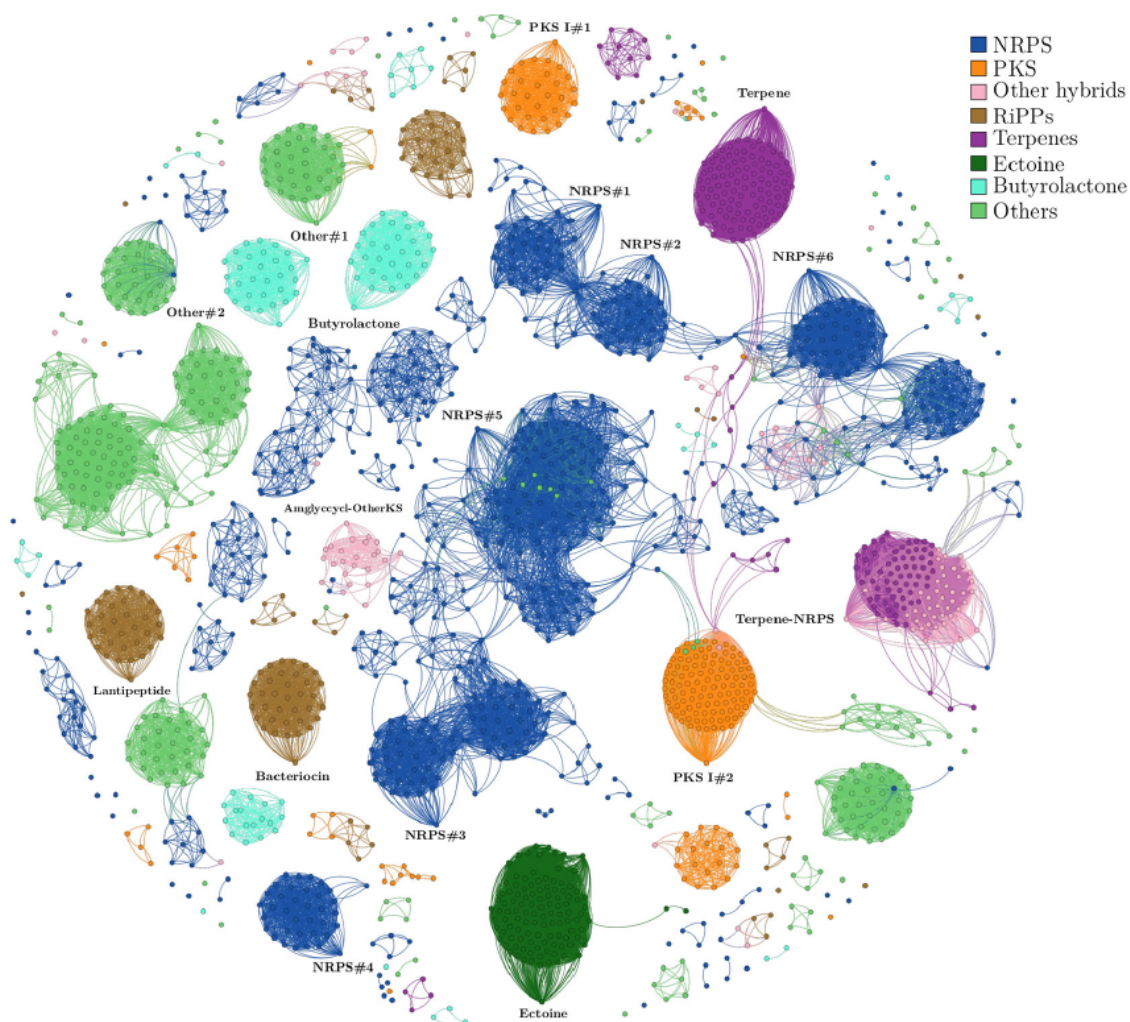


Figura 10: Ejemplo de una red de similitud de agrupamientos de genes biosintéticos.
Fuente: [Undabarrena *et al.*, 2021].

Otra herramienta bioinformática de utilidad para el análisis de los BGCs es CORASON. Este software permite analizar los contextos genéticos entre BGCs similares a los identificados en las cepas de estudio. Esto permite comprobar la novedad de los BGCs, y diferencias estructurales y evolutivas respecto a especies cercanas.

3.1.2. Definición de los procesos a implementar en la aplicación

Para seleccionar los procesos de la bioprospección de actinobacterias marinas a implementar en la MyMicrobeApp los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas del CB-DAL presentaron como idea inicial el esquema de la figura 11. En dicho esquema exponen los datos que les interesaría almacenar y visualizar, agrupándolos por categoría.

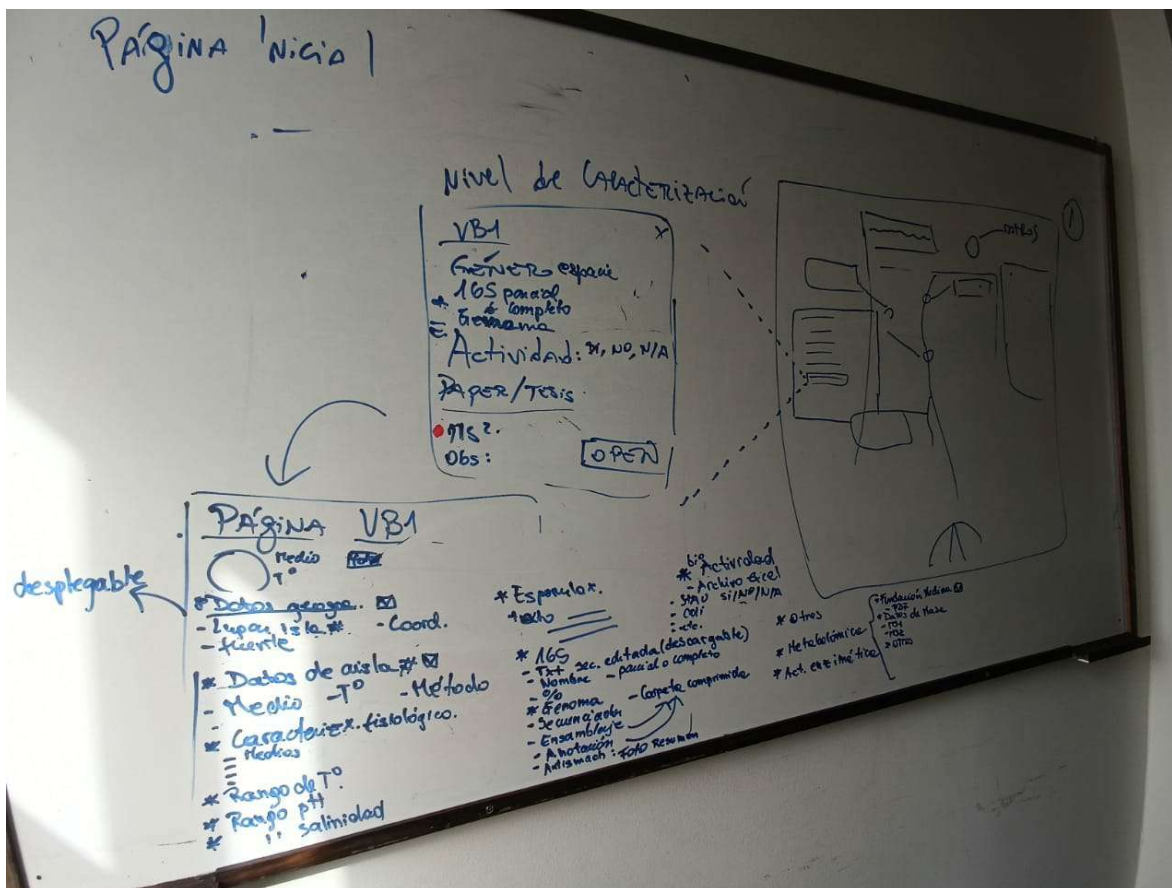


Figura 11: Idea original de MyMicrobeApp.

Fuente: Miembros del Laboratorio de Bioprospección de Actinobacterias Marinas del CB-DAL.

Tras una serie de reuniones enfocadas en la planificación y diseño de la aplicación, la idea de la figura 11 es refinada, por lo que se obtienen las siguientes categorías en las que se agruparán los datos:

1. Identificación

Como su nombre lo dice, esta categoría permite identificar y diferenciar las cepas de actinobacterias.

2. Datos geográficos

Se utiliza para registrar las características del lugar en que se realizó el muestreo.

3. Aislamiento

Permite describir el proceso de aislamiento de las cepas en un medio de cultivo y condiciones específicas.

4. Gen 16S rRNA

Facilita describir el estado del gen 16S rRNA y acceder a él.

5. Caracterización fenotípica y fisiológica

Reúne tanto los rasgos observables de una actinobacteria como su comportamiento.

6. Genoma

Agrupar los datos del genoma de la cepa.

7. Bioactividad

Permite precisar sobre cuales cepas tipo la actinobacteria presenta actividad antibacteriana.

8. Metabolómica

Reúne datos de los perfiles metabolómicos de las actinobacterias.

9. Enzimas

Se utiliza para registrar la producción de enzimas por parte de las cepas.

Posteriormente, en la figura 12 se relaciona los procesos de la sección 3.1.1 (a la izquierda) con las categorías descritas recientemente (a la derecha). Las flechas apuntan a las categorías en las que se almacenan los datos del proceso de origen.

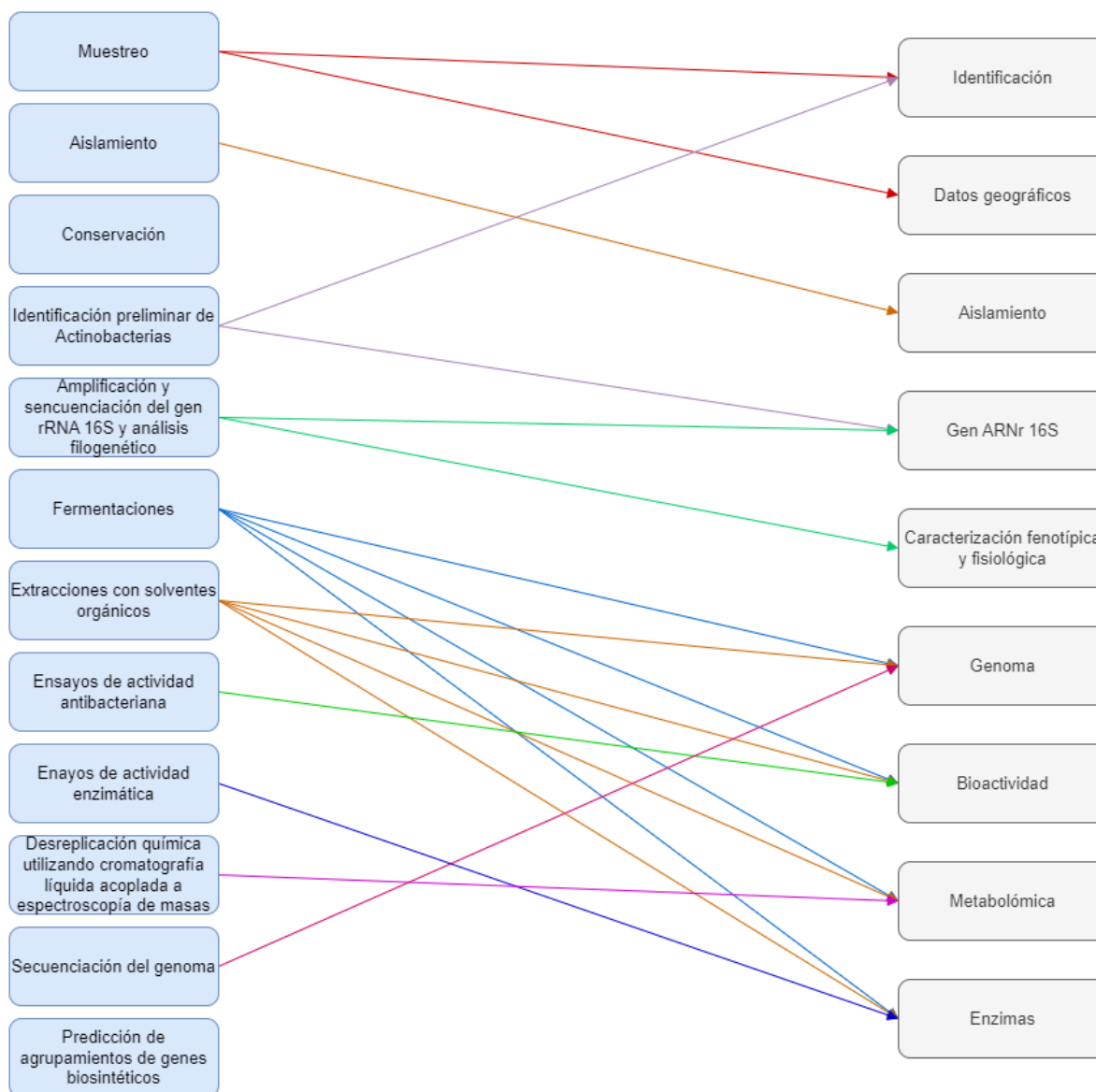


Figura 12: Relación entre los procesos del laboratorio y las categorías en las que se almacenan los datos de las actinobacterias.

Fuente: Elaboración propia.

Finalmente, de la figura 12 se obtiene que los procesos a implementar en la aplicación corresponden a aquellos que se relacionen con al menos una de las categorías de almacenamiento de datos de actinobacterias, es decir, los procesos a implementar en la aplicación son:

- Muestreo
- Aislamiento
- Identificación preliminar de Actinobacterias

- Amplificación y secuenciación del gen 16S rRNA y análisis filogenético
- Fermentaciones
- Extracciones con solventes orgánicos
- Ensayos de actividad antibacteriana
- Ensayos de actividad enzimática
- Desreplicación química utilizando cromatografía líquida acoplada a espectroscopía de masas
- Secuenciación del genoma

3.2. Diseño

El proceso de diseño de MyMicrobeApp comprende tanto el diseño de la interfaz de usuario como de la base de datos. Además, en base a las necesidades de los usuarios se definen roles para restringir el acceso a distintos menús de la aplicación.

3.2.1. Diseño de la interfaz de usuario

El diseño de la interfaz de usuario es realizado en colaboración con los investigadores del Laboratorio de Bioprospección de Actinobacterias Marinas. Este comprende reuniones de planteamiento de ideas, retroalimentación y consultas. Adicionalmente, se definen las vistas de la aplicación junto a sus menús y acciones correspondientes.

El primer aspecto a considerar corresponde a la disposición de la aplicación. A continuación se describen los componentes presentados en cada vista tras haber iniciado sesión.

1. Barra lateral

Es una barra vertical posicionada a la izquierda de la pantalla. Se compone de tres partes: El encabezado, que contiene el nombre de la aplicación y un icono para reducir/aumentar el tamaño de la barra lateral. El menú, en el que se muestran los menús de la aplicación. Y la opción de cierre de sesión.

A continuación, se listan los menús.

- a) Actinobacteria
- b) Assemblies
- c) Processed Data

- d) Users
- e) Genera
- f) Culture Medium
- g) Antimicrobial
- h) Enzyme

Cabe destacar que la barra lateral permite la opción de contraerse y tiene un comportamiento responsivo, de manera que se oculta al haber un tamaño de pantalla reducido.

2. Barra superior

Es una barra horizontal posicionada en la parte de arriba de la pantalla y a la derecha de la barra lateral. Esta cuenta con un icono que permite cambiar la paleta de colores de la interfaz entre el modo claro y modo oscuro. Además, cuenta con el avatar de la cuenta de Google del usuario.

3. Contenido

Como su nombre lo dice, es el contenido específico de cada vista perteneciente a los distintos menús de la barra lateral. Se presenta como un recuadro posicionada a la derecha de la barra lateral y bajo la barra superior de la aplicación.

El segundo aspecto a considerar corresponde a las vistas de la aplicación, las cuales cumplen acciones específicas. Se definen dos clases distintas en las que están comprendidas las vistas: 'Vistas externas', que son aquellas que no son parte del menú, y 'Vistas internas', que corresponden a las vistas del menú de la aplicación. A continuación, se define cada vista indicando su contenido y acciones que permite realizar.

Vistas externas

■ Sign In

Cuenta con un botón, que al ser presionado permite iniciar sesión en la aplicación con una cuenta de Google.

Vistas internas

■ Actinobacteria

Es un mantenedor de actinobacterias. Está compuesto por un listado con todas las actinobacterias de la base de datos y una casilla de filtrado por creador de la actinobacteria. Además, permite visualizarlas en profundidad, crearlas, actualizarlas y eliminarlas.

- **Assemblies**

Es un mantenedor de ensamblajes correspondientes al genoma de una actinobacteria. Contiene un listado con todos los ensamblajes de la base de datos y una casilla de filtrado por creador y actinobacteria. Permite crear ensamblajes, actualizarlos y eliminarlos.

- **Processed Data**

Es un mantenedor de procesamientos de datos correspondientes a la metabolómica de una actinobacteria. Cuenta con un listado con todos los procesamientos de datos de la base de datos y una casilla de filtrado por creador y actinobacteria. Permite crear procesamientos de datos, actualizarlos y eliminarlos.

- **Users**

Es un mantenedor de usuarios. Cuenta con un listado con todos los usuarios de la aplicación. Permite modificar su rol.

- **Genera**

Es un mantenedor de géneros bacterianos. Cuenta con un listado de géneros bacterianos y permite crearlos, actualizarlos y eliminarlos.

- **Culture Media**

Es un mantenedor de medios de cultivo. Contiene un listado de medios de cultivo y permite crearlos, actualizarlos y eliminarlos.

- **Antimicrobial**

Es un mantenedor de cepas tipo. Cuenta con un listado de cepas tipo y permite crearlas, actualizarlas y eliminarlas.

- **Enzyme**

Es un mantenedor de enzimas. Contiene un listado de enzimas y permite crearlas, actualizarlas y eliminarlas.

El diseño completo de la aplicación se halla en el anexo Anexo A: Mockup de la aplicación.

3.2.2. Diseño de la base de datos

Para diseñar la base de datos de la aplicación se realizan reuniones entre los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas y el memorista, teniendo en cuenta los procesos planteados en la sección 3.1.2.

Considerando que el DBMS seleccionado corresponde a MongoDB, se diseñan las entidades de la plataforma como documentos de tipo BSON. Por ende, para un mayor entendimiento,

por cada documento se presentará una tabla que representa el esquema de su modelo de datos. Dichas tablas cuentan con las columnas campo (en la que se muestran los nombres de los campos del documento), tipo de dato (que es el tipo de dato de MongoDB al que pertenece el valor del campo) y características (las cuales son cualidades que poseen dichos campos).

A continuación se describen los tipos de datos utilizados.

- **ObjectId**

Es una sigla hexadecimal de 12 bytes, compuesta por 4 bytes correspondientes a una marca de tiempo, 5 bytes generados de manera aleatoria y 3 bytes correspondientes a un contador auto-incremental que comienza con un valor aleatorio. Así, este tipo de dato genera un identificador único para los documentos.

Cabe destacar, que cada documento de MongoDB posee un campo `_id` de tipo `ObjectId` generado al momento de su creación, el cual le sirve de identificador.

- **String**

Corresponde a texto.

- **Boolean**

Es un tipo de dato lógico que solo puede tener los valores *true* (verdadero) y *false* (falso).

- **Number**

Es un número entero.

- **Decimal128**

Es un número decimal.

- **Date**

Corresponde a una fecha.

- **[String]**

Es un arreglo con elementos de tipo *string*, es decir, elementos de texto.

- **[ObjectId]**

Es un arreglo con elementos de tipo *ObjectId*, es decir, referencias a otros documentos.

A su vez, se explican las características que poseen los campos de los documentos.

- Único

Que un campo sea único implica que su valor es único para todos los documentos del esquema, es decir, no puede existir otro documento que para dicho campo tenga el mismo valor dentro del esquema.

- Requerido

Que un campo sea requerido implica que no se puede crear/actualizar un documento sin indicar un valor para dicho campo.

- min

Corresponde al valor mínimo aceptado para un campo numérico.

- match

En el caso de los campos de tipo String se utiliza la característica match para corroborar la coincidencia del texto con una expresión regular/patrón, como puede ser un correo electrónico o URL.

- ref

Hace referencia a un documento del esquema indicado. Por ejemplo, ref: 'User' hace referencia a un documento de tipo 'User'.

Tras haber descrito los tipos de datos y características utilizadas dentro de los documentos, se procede a realizar una breve reseña de cada esquema de los modelos de la base de datos:

- User

Es un esquema para los usuarios de la plataforma, como puede verse en la tabla 5. Está pensado para contener los datos del usuario, manejar sesiones y ser referenciado en otros modelos de la base de datos.

Tabla 5: Esquema de los usuarios en la base de datos. Fuente:
Elaboración Propia.

User		
Campo	Tipo de dato	Características
_id	ObjectId	Único
name	String	Requerido
email	String	Único, requerido
avatar	String	Requerido
role	string	Requerido

- Genera

Es un esquema con el fin de almacenar los géneros de actinobacterias que se mantienen en el cepario (ver tabla 6).

Tabla 6: Esquema de los géneros bacterianos en la base de datos. Fuente: Elaboración Propia.

Genera		
Campo	Tipo de dato	Características
_id	ObjectId	Único
name	String	Único, requerido
creator	ObjectId	Requerido, ref: 'User'

■ Culture medium

Es un esquema para registrar los medios de cultivos utilizados para las distintas cepas de actinobacterias (ver tabla 7).

Tabla 7: Esquema de los medios de cultivo en la base de datos. Fuente: Elaboración Propia.

CultureMedium		
Campo	Tipo de dato	Características
_id	ObjectId	Único
name	String	Único, requerido
creator	ObjectId	Requerido, ref: 'User'

■ Assembly

Esquema utilizado para almacenar los datos de los ensamblajes correspondientes al genoma de una actinobacteria en específico (ver tabla 8).

Tabla 8: Esquema de los ensamblajes en la base de datos. Fuente: Elaboración Propia.

Assembly		
Campo	Tipo de dato	Características
_id	ObjectId	Único
creator	ObjectId	Requerido, ref: 'User'
actinobacteria	ObjectId	Requerido, ref: 'Actinobacteria'
date	Date	Requerido
bgcs	String	match: URL
sequencingTechnology	string	Requerido
softwareTrimming	String	Requerido
softwareAssembly	String	Requerido

parametersAssembly	String	Requerido
qualityFinal	String	Requerido
sequencingTechnology	String	
accessionNumber	String	
paper	String	
comments	String	
link	String	match: URL

■ Type Strain

Es un esquema para registrar las cepas tipo utilizadas para hacer ensayos de bioactividad respecto a los productos obtenidos de las actinobacterias (ver tabla 9).

Tabla 9: Esquema de las cepas tipo en la base de datos. Fuente: Elaboración Propia.

TypeStrain		
Campo	Tipo de dato	Características
_id	ObjectId	Único
name	String	Único, requerido
creator	ObjectId	Requerido, ref: 'User'

■ ProcessedData

Esquema utilizado para almacenar los datos de los datos procesados correspondientes la metabolómica de una actinobacteria en específico (ver tabla 10).

Tabla 10: Esquema del procesamiento de datos en la base de datos. Fuente: Elaboración Propia.

ProcessedData		
Campo	Tipo de dato	Características
_id	ObjectId	Único
creator	ObjectId	Requerido, ref: 'User'
actinobacteria	ObjectId	Requerido, ref: 'Actinobacteria'
dataSource	string	Requerido
equipment	string	Requerido
fileName	string	Requerido
massDetection	String	
chromatogramBuilder	String	
deconvolution	String	
isotope	String	

filtered	String	
identification	String	
alignment	String	
gapFilling	String	
massIVEID	String	
link	String	match: URL
comments	String	

■ Enzyme

Es un esquema para registrar las enzimas utilizadas para hacer ensayos de actividad enzimática respecto a los productos obtenidos de las actinobacterias (ver tabla 11).

Tabla 11: Esquema de las enzimas en la base de datos. Fuente: Elaboración Propia.

Enzyme		
Campo	Tipo de dato	Características
_id	ObjectId	Único
name	String	Único, requerido
creator	ObjectId	Requerido, ref: 'User'

■ Actinobacteria

Es un esquema utilizado para poder almacenar toda la información obtenida del proceso de bioprospección de actinobacterias marinas (ver tabla 12).

Tabla 12: Esquema de las actinobacterias en la base de datos. Fuente: Elaboración Propia.

Actinobacteria		
Campo	Tipo de dato	Características
_id	ObjectId	Único
creator	ObjectId	Requerido, ref: 'User'
identifierStrain	String	Único, requerido
identifierGenera	String	Requerido, ref: 'Genera'
identifierSpecies	String	
identifierMainPhoto	String	Requerido, match: URL
identifierPhotos	String	match: URL
identifierLocalStorage	String	Requerido
identifierInternationalStorage	String	
identifierComments	String	

geographyIsolationSite	String	Requerido
geographyCoordinates	String	
geographyIsolationSource	String	Requerido
geographyAltitude	Decimal128	
geographyComments	String	
isolationMedium	String	Requerido
isolationTemperature	Decimal128	Requerido
isolationMethod	String	Requerido
isolationResponsible	String	Requerido
isolationThesisPaper	String	
isolationThesisPaperLink	String	match: URL
isolationComments	String	
arnr16sCompleteness	String	Requerido
arnr16sSize	Decimal128	Requerido
arnr16sSequenceFile	String	Requerido, match: URL
arnr16sMacroGenFile	String	match: URL
arnr16sComments	String	
characterizationGrowingMedia	[String]	Requerido, ref: 'CultureMedium'
characterizationNotGrowingMedia	[String]	ref: 'CultureMedium'
characterizationMycelial	String	Requerido
characterizationColoniesDay	Number	min: 0
characterizationSporulationDay	Number	min: 0
characterizationBiomassDay	Number	min: 0
characterizationShape	String	
characterizationBorder	String	
characterizationElevation	String	
characterizationSurface	String	
characterizationColor	String	
characterizationTransparency	String	
characterizationBrightness	String	
characterizationComments	String	
genomeRawData	String	match: URL
genomeComments	String	
bioactivityFile	String	match: URL
bioactivityYes	[ObjectId]	ref: 'TypeStrain'
bioactivityNo	[ObjectId]	ref: 'TypeStrain'
bioactivityNa	[ObjectId]	ref: 'TypeStrain'
bioactivityComments	String	
metabolomicsMedinaFoundationReports	String	match: URL
metabolomicsRawData	String	match: URL
metabolomicsComments	String	
enzymesYes	[ObjectId]	ref: 'Enzyme'

enzymesNo	[ObjectId]	ref: 'Enzyme'
enzymesNa	[ObjectId]	ref: 'Enzyme'
enzymesComments	ObjectId	

3.2.3. Definición de roles de usuario

Con el fin de restringir el acceso de algunos usuarios a los distintos menús de la aplicación se definen roles de usuario, los que se diferencian principalmente por la capacidad de gestionar y/o administrar información. A continuación, se describen los roles de usuario establecidos.

- user

Este rol permite acceder a los menús Actinobacteria, Assembly y Processed Data.

Está ideado para usuarios comunes de la aplicación, quienes no tienen influencia en datos más sensibles del laboratorio, como pueden ser practicantes y/o visitantes.

- manager

Al igual que el rol user permite acceder a los menús Actinobacteria, Assembly y Processed Data. Por otra parte, también permite el acceso a los menús Genera, Culture Media, Antimicrobial y Enzyme.

Este rol corresponde al que tendrán la mayoría de los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas, quienes tienen permitido gestionar información sensible para las cepas de actinobacterias.

- admin

Este rol tiene acceso a todos los menús de la aplicación y se diferencia con el rol *manager* en que si tiene acceso al menú 'Users', de manera que puede administrar información concerniente a los usuarios de la aplicación.

Está pensado para uno o pocos miembros del laboratorio que sean designados para poder modificar el rol de otros usuarios.

En la tabla 13 se resume el acceso a cada menú de la aplicación para cada rol de usuario.

Tabla 13: Acceso a los menús según el rol de usuario. Fuente:
Elaboración Propia.

Menú	Rol		
	User	Manager	Admin
Actinobacteria	Si	Si	Si

Assembly	Si	Si	Si
Processed Data	Si	Si	Si
Users	No	No	Si
Genera	No	Si	Si
Culture Medium	No	Si	Si
Antimicrobial	No	Si	Si
Enzyme	No	Si	Si

3.3. Desarrollo

El proceso de desarrollo de MyMicrobeApp comprende el desarrollo de la base de datos, API e interfaz de usuario en base a los diseños propuestos en la sección 3.2.

3.3.1. Desarrollo de la base de datos

El primer paso para desarrollar la base de datos consiste en crear un *cluster* o contenedor en MongoDB Atlas³⁰. Luego, se utiliza Mongoose³¹ para esquematizar un modelo de la base de datos, que puede o no requerir validación en algunos campos. Finalmente, se ejecuta el código de Node.js de manera que el modelo creado se convertirá en una colección en la base de datos creada en el *cluster* de MongoDB Atlas.

Un ejemplo de esto se da en la figura 13. En la primera línea de código se hacen las importaciones necesarias desde mongoose. Después, desde la tercera línea hasta la decimocuarta se define el esquema para un Assembly. Finalmente, desde la decimosexta hasta la decimocava línea se crea el modelo de un Assembly para la base de datos.

³⁰MongoDB Atlas es una base de datos como servicio de MongoDB, es decir, ellos se encargan de la configuración y alojamiento de la base de datos. De manera que el desarrollador tiene como única responsabilidad diseñar y llenar la base de datos de colecciones [Contreras, 2021].

³¹Mongoose es una biblioteca de programación orientada a objetos que permite conectar MongoDB y Node.js, proporcionando una solución sencilla basada en esquemas para modelar los datos de una aplicación. Incluye conversión de tipos, validación, creación de consultas, entre otras opciones [Mon, 2023].

```
1 import { Schema, InferSchemaType, model } from 'mongoose';
2
3 const assemblySchema = new Schema({
4   creator: { type: Schema.Types.ObjectId, required: true, ref: "User", select: false },
5   actinobacteria: { type: Schema.Types.ObjectId, required: true, ref: "Actinobacteria", select: false },
6   date: { type: Date, required: true },
7   bgcs: { type: String, match: [/^(http|https):\/\/\[^\s"]+$/, 'Please enter a valid URL.'],
8   softwareTrimming: { type: String, required: true },
9   softwareAssembly: { type: String, required: true },
10  parametersAssembly: { type: String, required: true },
11  qualityFinal: { type: String, required: true },
12  comments: { type: String },
13  link: { type: String, match: [/^(http|https):\/\/\[^\s"]+$/, 'Please enter a valid URL.'],
14  sequencingTechnology: { type: String, required: true },
15  accessionNumber: { type: String },
16  paper: { type: String },
17 }, { timestamps: true });
18
19 type Assembly = InferSchemaType<typeof assemblySchema>;
20
21 export default model<Assembly>("Assembly", assemblySchema);
```

Figura 13: Modelo de Assembly en la base de datos.

Fuente: Elaboración propia.

3.3.2. Desarrollo de la API

Para el desarrollo de la API se utiliza la estructura de carpetas de la figura 14. La carpeta principal corresponde a 'server', la cual cuenta con varias subcarpetas. Sin embargo, para el caso del desarrollo es relevante profundizar únicamente en el contenido de la carpeta 'src'. A continuación, se explica en detalle el contenido de las subcarpetas de 'src'.

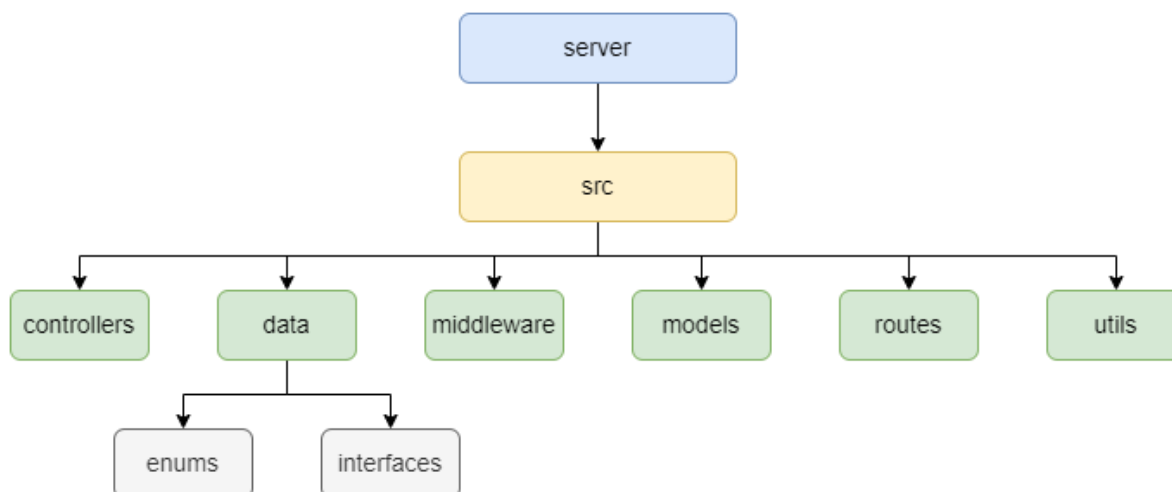


Figura 14: Estructura de carpetas de la API.

Fuente: Elaboración propia.

- controllers

Dentro de esta carpeta están los controladores para cada modelo de la base de datos. Cada modelo cuenta con los métodos necesarios para responder a las consultas de sus respectivas rutas en la API. En general, dentro de los controladores se encuentran los métodos *Create* (permite crear un documento en una colección de la base de datos), *GetById* (obtiene un documento de una colección de la base de datos por medio de su `_id`), *GetPagination* (obtiene una determinada cantidad de documentos de una colección de la base de datos, dependiendo del número de página y tamaño especificado), *EditById* (permite actualizar los campos del documento que tenga el `_id` indicado) y *DeleteById* (elimina el documento que tenga el `_id` indicado).

- data

Cuenta con las estructuras de datos necesarias para el funcionamiento de los controladores. Se divide en dos subcarpetas `enums` e `interfaces`. La primera cuenta con `enums`³² de utilidad. Por su parte, la segunda cuenta con `interfaces`³³, que se utilizan para definir la estructura del cuerpo de las peticiones (*body*), sus parámetros (*params*) y elementos de consulta (*query*), para cada modelo de la base de datos.

- middleware

Middleware es un tipo de *software* que se coloca entre las diferentes aplicaciones y el sistema operativo que las tiene que ejecutar, con el fin de facilitar la comunicación de datos entre ellos [Hiberus, 2023].

³²Un `enum` en `typescript` corresponde a una estructura que permite definir un conjunto de constantes con valor numérico o de cadena de caracteres.

³³Una interfaz en `typescript` define la estructura de un objeto.

La carpeta *middleware* contiene un método que verifica que el usuario de la aplicación esté autenticado antes de poder utilizar los métodos de los controladores, tras llamar a sus respectivas rutas.

- **models**

Contiene los esquemas de los modelos de la base de datos definidos en 3.3.1.

- **routes**

Cuenta con las rutas para responder a las peticiones a la API realizadas desde el *front-end*. Cada ruta está asociada a un método de un controlador específicamente.

- **utils**

Dentro de esta carpeta hay varias funciones utilizadas en los archivos de las subcarpetas de 'src'.

3.3.3. Desarrollo de la interfaz de usuario

Para el desarrollo de la interfaz de usuario se utiliza la estructura de carpetas de la figura 15. La carpeta principal corresponde a '*client*', la cual cuenta con varias subcarpetas. Sin embargo, para el caso del desarrollo es relevante profundizar únicamente en el contenido de la carpeta 'src'. A continuación, se explica en detalle el contenido de las subcarpetas de 'src'.

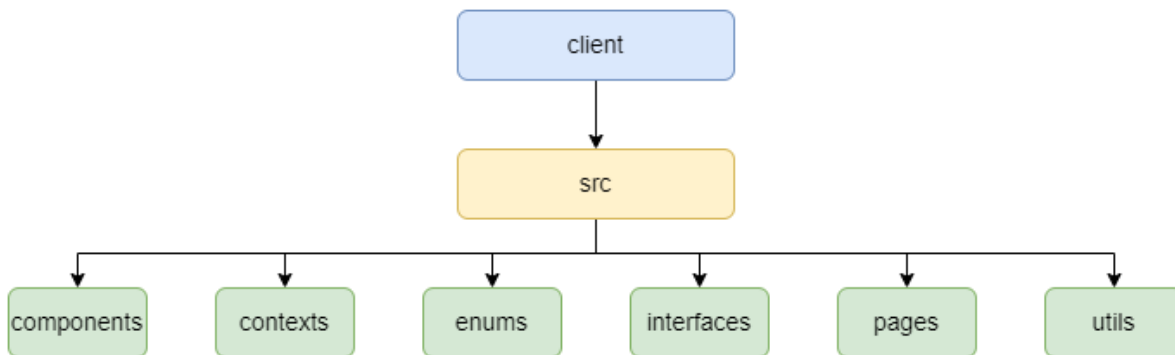


Figura 15: Estructura de carpetas de la interfaz de usuario.

Fuente: Elaboración propia.

- **components**

Dentro de esta carpeta hay componentes de uso general en la aplicación, como lo son el *header* (encabezado), el icono de la aplicación y componentes personalizados.

- **contexts**

Un *context* o contexto es un elemento que se puede crear en React para establecer una comunicación directa entre un componente en un nivel muy alto y uno en un nivel

mucho más bajo, evitando el traspaso de propiedades por todos sus componentes intermedios [KeepCoding, 2022].

En este caso se utiliza un context para traspasar el modo de color utilizado actualmente en la aplicación, que puede ser claro u oscuro.

- **enums**

Al igual que para el caso de la API, esta carpeta contiene enums de utilidad. Por ejemplo, el enum `UserRoles`, que define constantes para identificar a los distintos roles de usuario de la aplicación.

- **interfaces**

Contiene las interfaces utilizadas para definir la estructura de las peticiones a la API. Además, se emplean en la construcción de las tablas, filtros y formularios de los distintos menús de la aplicación.

- **pages**

Cuenta con las páginas con las que interactúa el usuario de la aplicación.

Se define una carpeta de páginas por cada menú. Las distintas páginas que puede contener cada carpeta son: *list* (listado o tabla con paginación y filtros), *show* (tabla que muestra los campos de un documento), *create* (formulario para la creación de un documento), *edit* (formulario para la edición de los campos de un documento).

Además, hay una página para el inicio de sesión.

- **utils**

Al igual que para el caso de la API, la interfaz de usuario cuenta con esta carpeta, que provee de funciones que pueden ser de utilidad en los distintos archivos de las subcarpetas de 'src'.

3.4. Resultados

MyMicrobeApp está disponible en la dirección web <https://mymicrobeapp.onrender.com> y el código del proyecto se encuentra en los repositorios públicos <https://github.com/RaulCruzT/mma-dashboard-server> y <https://github.com/RaulCruzT/mma-dashboard-client>.

3.4.1. Base de datos

La base de datos se desarrolló según el diseño establecido en 3.2.2. Es así que se obtuvo un cluster de MongoDB Atlas que cuenta con todas las colecciones ideadas en su etapa de diseño.

Para garantizar su seguridad, se restringe el acceso a la red en donde se encuentra el cluster, indicando que solamente se puede acceder a ella desde las direcciones IP en donde se encuentra alojada la API de la aplicación.

3.4.2. API

La API, que funciona como servidor para la aplicación, fue desarrollada de manera que permite iniciar sesión a través del sistema de autenticación de Google Cloud. Además, cuenta con los controladores y rutas necesarias para responder a las peticiones realizadas desde el *frontend*. Respecto a esto último, es relevante destacar que todas las peticiones CRUD han sido aseguradas, es decir, no pueden ser realizadas sin que el usuario esté autenticado.

Dicha API es alojada como un *web service* (servicio web) en <https://render.com/>, el cual está conectado con el repositorio público <https://github.com/RaulCruzT/mma-dashboard-server>. Además, para el despliegue de la aplicación se habilitó la opción de despliegue continuo en base a los cambios realizados en una rama en específico del repositorio de GitHub, es decir, el servicio web será actualizado y desplegado automáticamente cada vez que se suba un cambio a dicha rama del repositorio.

3.4.3. Interfaz

La interfaz de la aplicación se desarrolló según el diseño planteado en 3.2.1 y la estructura de carpetas de 3.3.3. El resultado puede ser observado en el anexo Anexo B: Interfaz de la aplicación, en donde se exhiben todas las vistas disponibles en MyMicrobeApp.

Al igual que para la API, la interfaz es alojada en <https://render.com/>, pero como *static site* (sitio web estático), el cual está conectado con el repositorio público <https://github.com/RaulCruzT/mma-dashboard-client>. Igualmente, se habilitó la opción de despliegue continuo en base a los cambios realizados en una rama en específico del repositorio de GitHub.

Por otra parte, se indicó en el servicio de autenticación de Google Cloud que este solo puede ser accedido desde la dirección web de MyMicrobeApp.

3.4.4. Menús y vistas de la aplicación

Los menús y vistas de la aplicación se establecen cuando el usuario accede a ella. En dicho momento se obtienen tanto sus datos, como lo son su correo electrónico o su nombre, como su rol de usuario.

Dichos roles son aquellos que fueron definidos en 3.2.3.

A continuación se listan los menús, páginas y funcionalidades obtenidos tras desarrollar My-MicrobeApp, por cada rol de usuario.

■ *user*

En la figura 16 se puede observar al usuario con rol *user* (en azul). Este cuenta con los menús *Actinobacteria*, *Assembly* y *Processed Data* (en amarillo). Cada uno de esos menús comprende cuatro vistas, que son: *List*, *Show*, *Create* y *Edit* (en gris); las cuales tienen las funcionalidades de listar, visualizar/mostrar, crear y editar/actualizar documentos de la base de datos respectivamente. Además, en las vistas *List* está la opción *Delete* (en rojo) para poder eliminar el documento seleccionado.

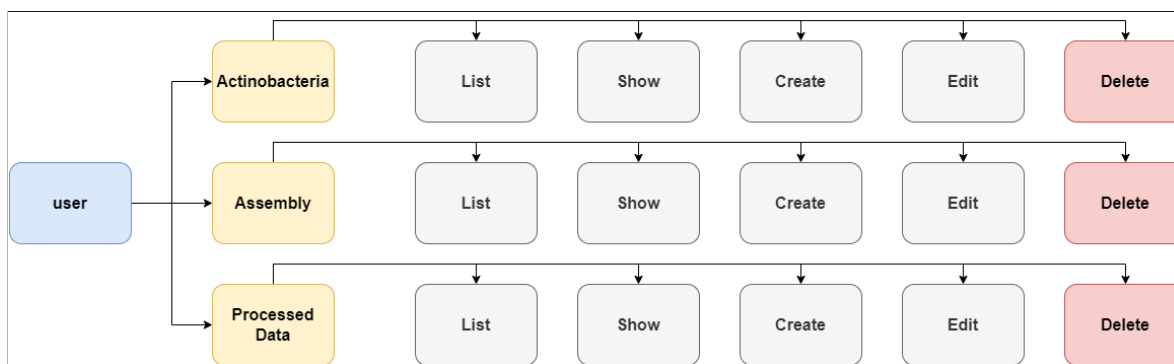


Figura 16: Menús y vistas de usuario con rol *user*

Fuente: Elaboración propia.

■ *manager*

En la figura 17 se puede observar al usuario con rol *manager* (en azul). Este cuenta con los menús *Actinobacteria*, *Assembly*, *Processed Data*, *Genera*, *Culture Media*, *Antimicrobial* y *Enzyme* (en amarillo). Los primeros tres menús comprenden cuatro vistas, que son: *List*, *Show*, *Create* y *Edit* (en gris); las cuales tienen las funcionalidades de listar, visualizar/mostrar, crear y editar/actualizar documentos de la base de datos respectivamente. Por su parte, los últimos cuatro menús comprenden 3 vistas, que son *List*, *Create* y *Edit* (en gris); las cuales tienen las funcionalidades de listar, crear y editar/actualizar documentos de la base de datos respectivamente. Además, en las vistas *List* está la opción *Delete* (en rojo) para poder eliminar el documento seleccionado.



Figura 17: Menús y vistas de usuario con rol *manager*
Fuente: Elaboración propia.

■ *admin*

En la figura 18 se puede observar al usuario con rol *admin* (en azul). Este cuenta con los menús *Actinobacteria*, *Assembly*, *Processed Data*, *User*, *Genera*, *Culture Media*, *Antimicrobial* y *Enzyme* (en amarillo). Los primeros tres menús comprenden cuatro vistas, que son: *List*, *Show*, *Create* y *Edit* (en gris); las cuales tienen las funcionalidades de listar, visualizar/mostrar, crear y editar/actualizar documentos de la base de datos respectivamente. Por su parte, el cuarto menú (*Users*) comprende 2 vistas, que son *List* y *Edit* (en gris); las cuales tienen las funcionalidades de listar y editar/actualizar documentos de la base de datos respectivamente. Por otro lado, los últimos cuatro menús comprenden 3 vistas, que son *List*, *Create* y *Edit* (en gris); las cuales tienen las funcionalidades de listar, crear y editar/actualizar documentos de la base de datos respectivamente. Además, en las vistas *List*, exceptuando la del menú *Users*, está la opción *Delete* (en rojo) para poder eliminar el documento seleccionado.

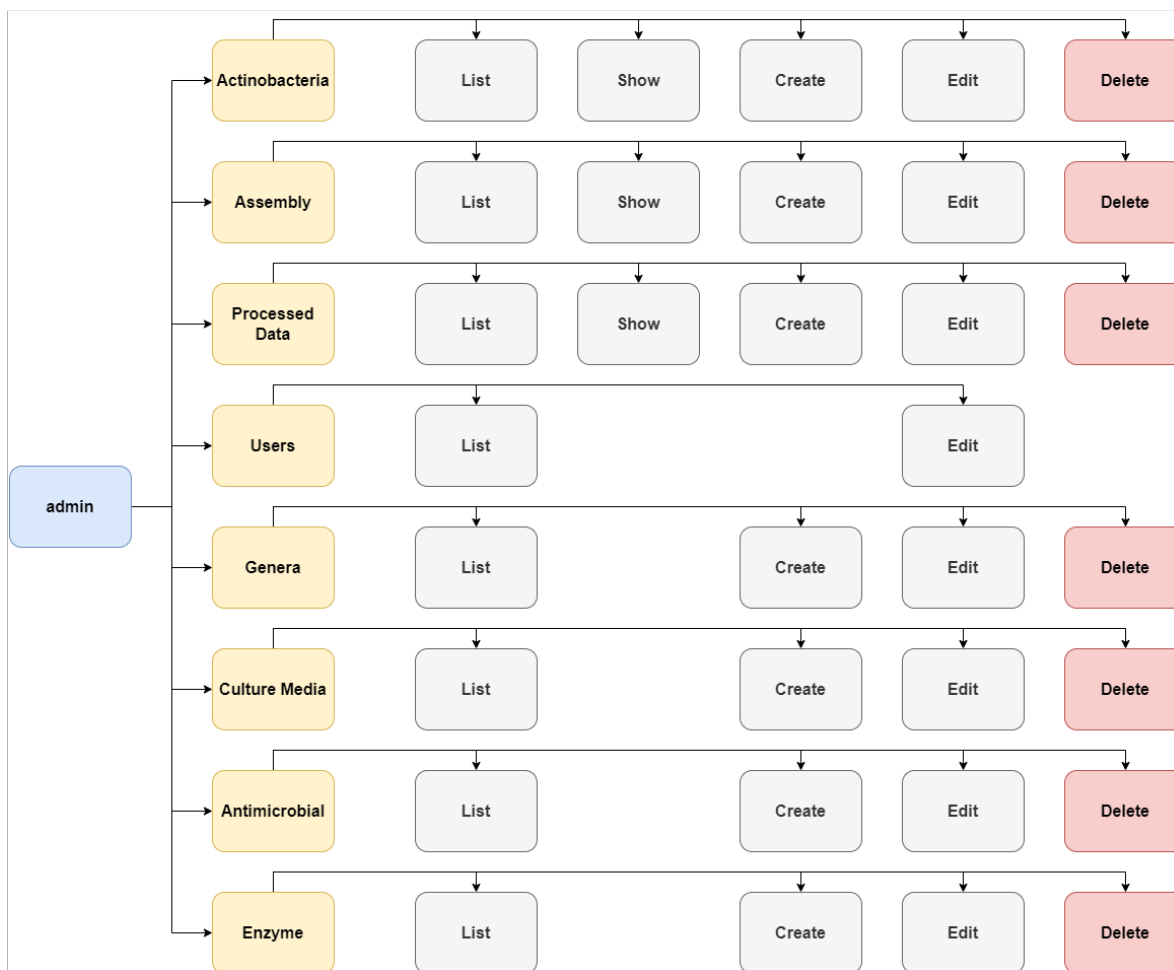


Figura 18: Menús y vistas de usuario con rol *admin*
Fuente: Elaboración propia.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

Para validar la solución desarrollada se define el medio utilizado y posteriormente se exponen los resultados obtenidos.

4.1. Sistema de escalas de usabilidad

El sistema de escalas de usabilidad (SUS), es un método desarrollado por John Brooke en 1986 para evaluar la usabilidad de cualquier sistema. Permite evaluar su eficacia, eficiencia y la satisfacción de los usuarios.

Este método consiste en una encuesta de cara a los usuarios con diez enunciados predefinidos:

1. Creo que me gustaría utilizar este sistema con frecuencia.
2. Encontré el sistema innecesariamente complejo.
3. Pensé que el sistema era fácil de usar.
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.
5. Encontré que las diversas funciones de este sistema estaban bien integradas.
6. Pensé que había demasiada inconsistencia en este sistema.
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.
8. Encontré el sistema muy complicado de usar.
9. Me sentí muy seguro usando el sistema.
10. Necesitaba aprender muchas cosas antes de empezar con este sistema.

Cada enunciado se responde siguiendo una Escala de Likert bipolar³⁴, de manera que hay 5 opciones (el número de la respuesta corresponde a su valor en la Escala de Likert):

³⁴Es un método utilizado para evaluar la opinión de los usuarios. Una Escala de Likert puede ser polar o bipolar. La primera permite que las respuestas de los usuarios sean positivas, negativas o neutras. En cambio, la segunda solo ofrece respuestas positivas o negativas [Busquets, 2022].

1. Totalmente en desacuerdo.
2. En desacuerdo.
3. Neutro.
4. De acuerdo.
5. Totalmente de acuerdo.

De los diez enunciados, los cinco impares son positivos y los cinco pares son negativos, lo que permite calcular el resultado de la encuesta. Dicho valor se obtiene de la siguiente forma:

Resultado = [(Suma de las respuestas de los enunciados impares - 5) + (25 - Suma de las respuestas de los enunciados pares)] * 2,5

El valor del resultado puede variar entre 0 a 100. Sin embargo, es importante destacar que dicho valor no es un porcentaje. [Busquets, 2021].

En [Sauro, 2011], Jeff Sauro realiza un estudio en más de 500 sitios web y aplicaciones, en donde concluye que el puntaje promedio es 68. Es así que se considera desde 68 (a 100) un valor aceptable; un resultado bajo esa cifra indica que hay varios aspectos a corregir.

4.2. Resultados

Se implementó SUS por medio de un formulario de *Google* enviado a los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología 'Dr. Daniel Alkalay Lowitt' de la Universidad Técnica Federico Santa María.

Actualmente, el Laboratorio de Bioprospección de Actinobacterias Marinas cuenta con 9 miembros, quienes son los usuarios a los que está destinada la aplicación MyMicrobeApp. Por ende, son ellos a quienes se les solicita contestar la encuesta.

Tras haberse cumplido el plazo establecido para responder la encuesta, 5 de los 9 miembros del laboratorio la contestaron, lo que representa un 55,6 % del equipo aproximadamente.

Luego, se exportó los resultados a una planilla de excel, en donde cada columna correspondía a un enunciado y cada fila agrupaba las respuestas de uno de los encuestados a cada enunciado.

Después, se calculó el resultado de SUS por cada encuestado y se obtuvo un promedio, cuyo valor fue 96.

Basándose en lo descrito en 4.1, el resultado supera el valor mínimo necesario (68) para que el producto desarrollado (en este caso MyMicrobeApp) se considere aceptable en términos de usabilidad.

CAPÍTULO 5

CONCLUSIONES

5.1. Conclusiones generales

En la presente memoria se diseñó y desarrolló MyMicrobeApp, una aplicación web que permite integrar y visualizar datos de microorganismos con potencial biotecnológico.

Para ello, se analizó las necesidades de los miembros del Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología 'Dr. Daniel Alkalay Lowitt' de la Universidad Técnica Federico Santa María, permitiendo definir cuál era su problema, sus causas y sus efectos.

En base a lo anterior, se definieron los objetivos de este trabajo, los cuales sirvieron como ruta para obtener el producto requerido. Dicha ruta comenzó por la caracterización de los procesos llevados a cabo en el Laboratorio de Bioprospección de Actinobacterias Marinas y posterior selección para su implementación, continuó con el diseño y desarrollo de la solución, en este caso la aplicación y concluyó con la validación de su interfaz por parte de sus usuarios.

Como se mencionó en 3.4, la aplicación está disponible en la dirección web <https://mymicrobeapp.onrender.com> y su código se encuentra en los repositorios públicos <https://github.com/RaulCruzT/mma-dashboard-server> y <https://github.com/RaulCruzT/mma-dashboard-client>.

Se espera que la herramienta desarrollada sea utilizada para que los investigadores puedan acceder y/o registrar datos de actinobacterias marinas de forma más fácil.

5.2. Cumplimiento de objetivos

El resultado final de esta memoria se puede medir en base a cada uno de los objetivos específicos planteados en 1.4.2, los cuales determinan el cumplimiento del objetivo general. A continuación se expone los resultados obtenidos por cada objetivo específico.

1. Caracterizar los procesos llevados a cabo por el Laboratorio de Bioprospección de Actinobacterias Marinas.

En la sección 3.1.1 se identifican y caracterizan 12 procesos llevados a cabo en el Laboratorio de Bioprospección de Actinobacterias Marinas del Centro de Biotecnología 'Dr. Daniel Alkalay Lowitt', a través del análisis de la etapa de experimentación de las tesis de varios miembros del laboratorio. Por lo que se considera cumplido el objetivo.

2. Definir los procesos a implementar en la aplicación.

En la sección 3.1.2 se definen categorías de datos en las que se almacenan los datos de las actinobacterias. Posteriormente, se muestra la relación entre los procesos llevados a cabo por el Laboratorio de Bioprospección de Actinobacterias Marinas y dichas categorías. Finalmente, se define como procesos a implementar a aquellos que se relacionen con al menos una de las categorías. Por lo que se considera cumplido el objetivo.

3. Construir la aplicación en base al diseño realizado previamente.

En la sección 3.2 se diseñan la interfaz de usuario, base de datos y roles de usuario de la aplicación.

Después, en la sección 3.3 se explica el proceso de desarrollo la base de datos, API e interfaz de usuario de MyMicrobeApp.

Finalmente, en la sección 3.4 se exponen los resultados obtenidos, lo que básicamente se traduce en la construcción de la aplicación. Por lo que se considera cumplido el objetivo.

4. Validar la interfaz desarrollada utilizando técnicas centradas en usuario.

En la sección 3.4.4 se define un método para validar la solución en base a la opinión de sus usuarios. Tras aplicarlo se evidenció un alto grado de satisfacción por parte de los usuarios, quienes identificaron a la aplicación como un sistema fácil de usar. Por lo que se considera cumplido el objetivo.

5.3. Recomendaciones

A continuación se presentan una serie de recomendaciones que podrían mejorar el resultado obtenido si es que los usuarios lo requirieran.

1. En caso de que se almacenen muchos documentos en la base de datos y su espacio de almacenamiento restante se vea reducido, se recomienda modificar el plan utilizado en MongoDB Atlas a uno con mayor espacio de almacenamiento.
2. En caso de que la demanda de usuarios de la aplicación crezca demasiado y su respuesta se ralentice, está la opción de mejorar el plan del *web service* (servicio web) en <https://render.com/>. Dentro de las opciones de mejora se encuentran: una mayor cantidad de almacenamiento y/o memoria RAM y más núcleos de procesador en la máquina en la que está alojada la aplicación.
3. Se podría dar la opción de subir archivos directamente en la aplicación (en vez de guardar un enlace a su dirección web), para almacenarlos en un servicio en la nube. Para este punto se debe tener en cuenta que el tamaño de dichos archivos debe ser limitado, ya que podría afectar el rendimiento de la aplicación.

4. Podrían agregarse más etapas del proceso de bioprospección de actinobacterias marinas que las comprendidas actualmente por MyMicrobeApp.

5.4. Trabajo futuro

A continuación se describen dos ideas que podrían ser desarrolladas como posibles trabajos de título sobre los resultados de la presente memoria.

1. En el Laboratorio de Bioprospección de Actinobacterias Marinas cuentan actualmente con datos de varias cepas, por lo que podría generarse una automatización para subirlos rápidamente a la base de datos.

Adicionalmente, sería bueno crear un nuevo rol de usuario que pudiera solamente visualizar los datos de todos los menús, a excepción del menú 'Users', pero no crear, actualizar y/o eliminar.

Siguiendo con los roles de usuario, sería bueno permitir que el rol de usuario 'user' pudiera visualizar (no crear, actualizar y/o eliminar) los datos de los menús: Genera, Culture Media, Antimicrobial y Enzyme. Esto le facilitaría el flujo de creación de actinobacterias a los usuarios con dicho rol.

Por último, se podrían añadir restricciones al sistema de acceso de la aplicación y/o modificar el registro de manera que los usuarios con rol 'admin' tengan un nuevo menú exclusivo para registrar nuevos usuarios y ya no pueda acceder a la aplicación cualquier persona.

2. A pesar de que la aplicación tiene un comportamiento responsivo, es decir, su interfaz se adapta al tamaño de pantalla; se podría desarrollar una aplicación móvil para hacer MyMicrobeApp aún más accesible.

ANEXO A: Mockup de la aplicación

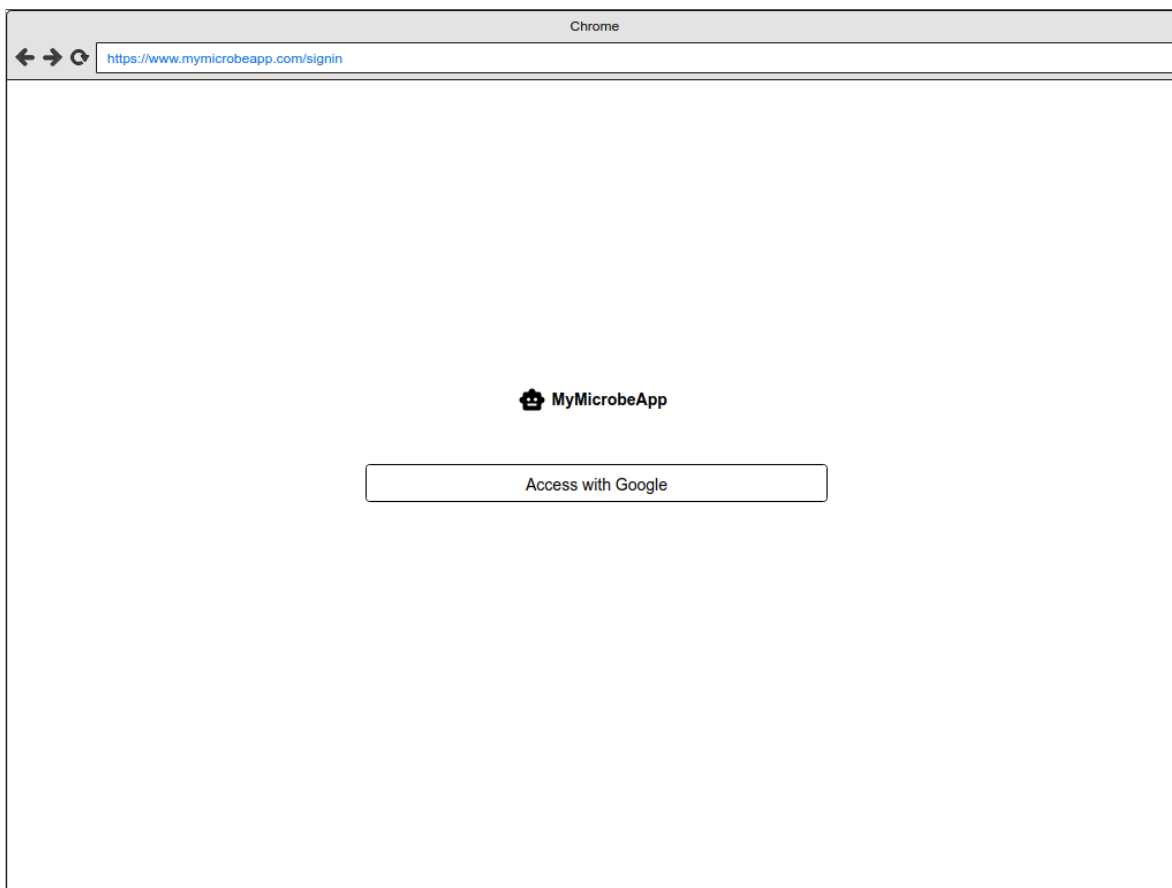


Figura 19: Mockup Inicio de sesión
Fuente: Elaboración propia.

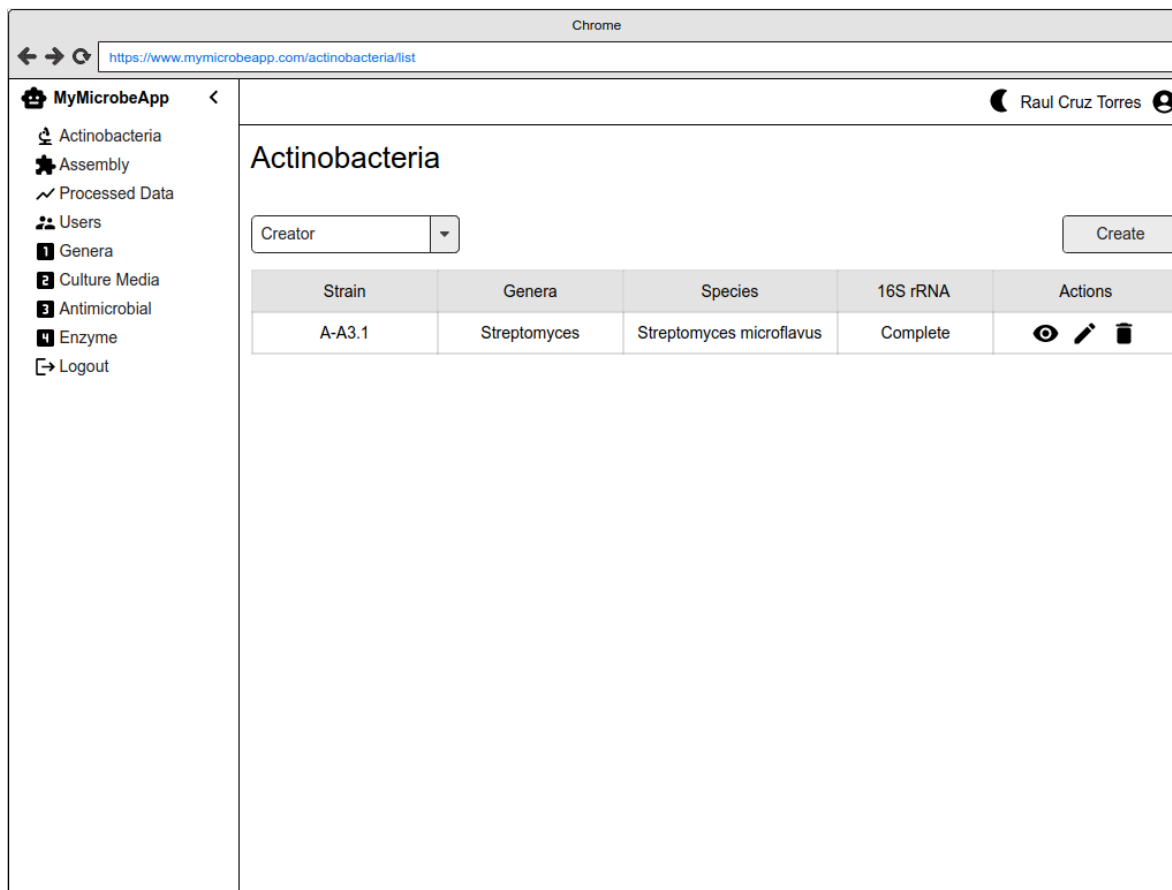


Figura 20: Mockup Actinobacteria List
Fuente: Elaboración propia.

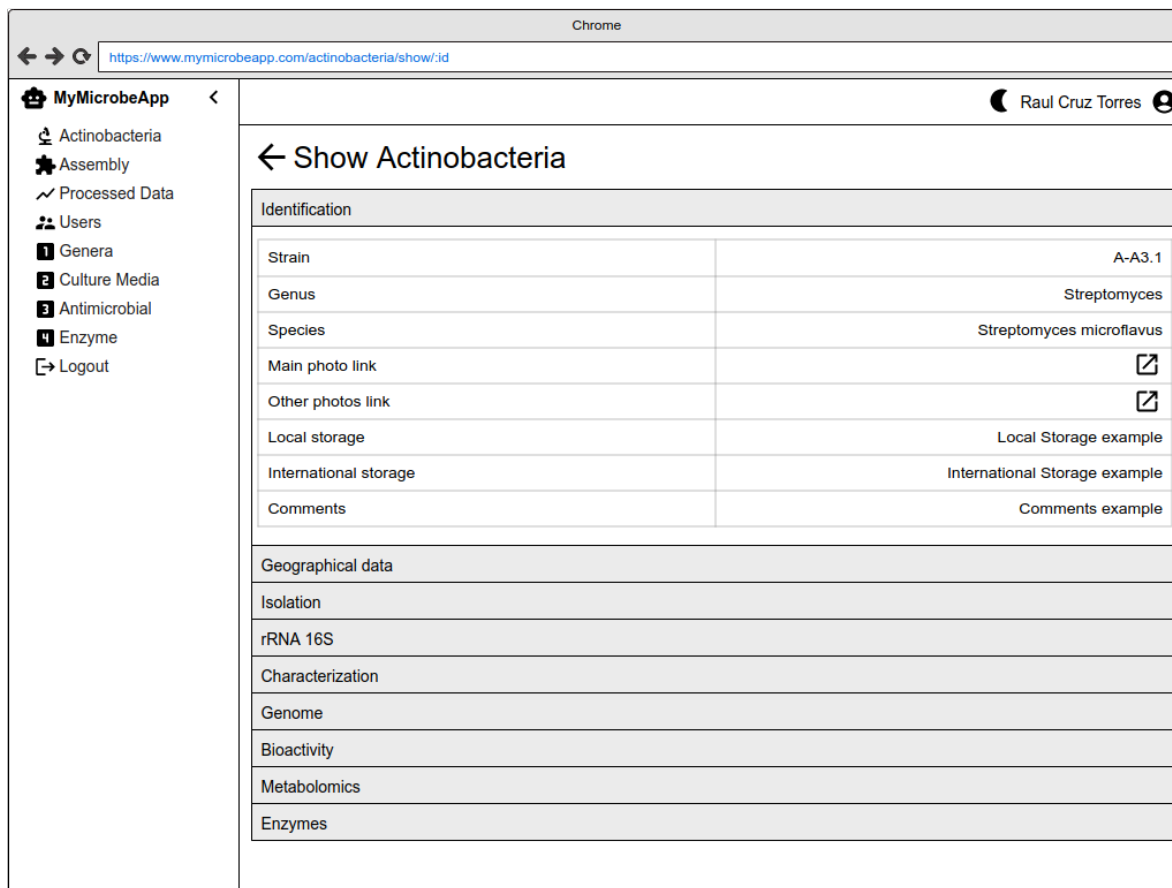


Figura 21: Mockup Actinobacteria Show
Fuente: Elaboración propia.

The mockup shows a web browser window with the URL <https://www.mymicrobeapp.com/actinobacteria/create>. The application header includes the 'MyMicrobeApp' logo, a navigation menu on the left, and the user profile 'Raul Cruz Torres' on the right. The navigation menu lists: Actinobacteria, Assembly, Processed Data, Users, Genera (selected), Culture Media, Antimicrobial, Enzyme, and Logout. The main content area is titled 'Create Actinobacteria' and contains several sections: 'Identification' with fields for Strain, Genera (a dropdown menu), Species, URL Main Photo, URL Photos Folder, Local Storage, and International Storage; a 'Comments' text area; 'Geographical data'; and a list of data categories: Isolation, rRNA 16S, Characterization, Genome, Bioactivity, Metabolomics, and Enzymes. A 'Create' button is located at the bottom right of the form.

Figura 22: Mockup Actinobacteria Create
Fuente: Elaboración propia.

Chrome

← → ↻ https://www.mymicrobeapp.com/actinobacteria/edit/id

MyMicrobeApp

Actinobacteria

Assembly

Processed Data

Users

1 Genera

2 Culture Media

3 Antimicrobial

4 Enzyme

Logout

← Edit Actinobacteria

Identification

A-A3.1

Streptomyces

Streptomyces microflavus

http://example.com/folder1/image.png

http://example2.com/folder2/

Local Storage example

International Storage example

Comments example

Geographical data

Isolation

rRNA 16S

Characterization

Genome

Bioactivity

Metabolomics

Enzymes

Edit

Figura 23: Mockup Actinobacteria Edit
Fuente: Elaboración propia.

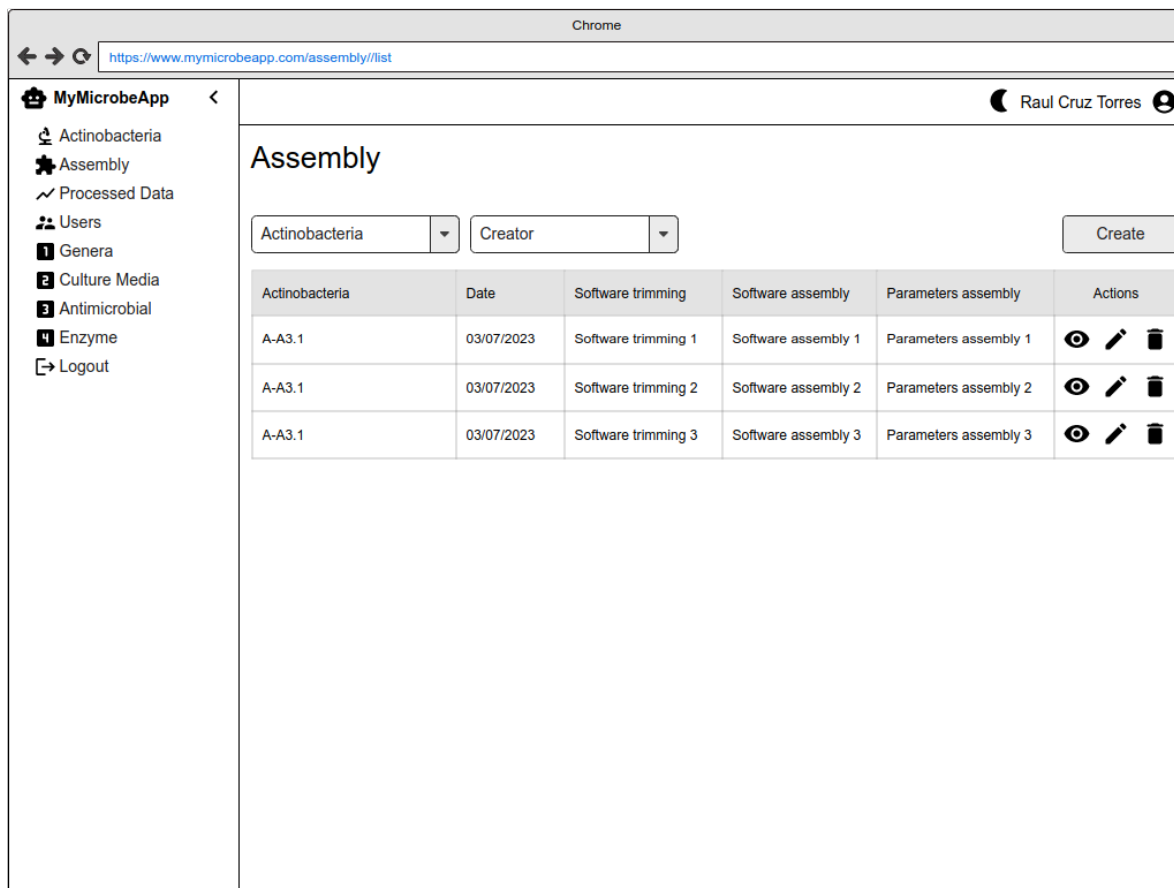


Figura 24: Mockup Assembly List
Fuente: Elaboración propia.

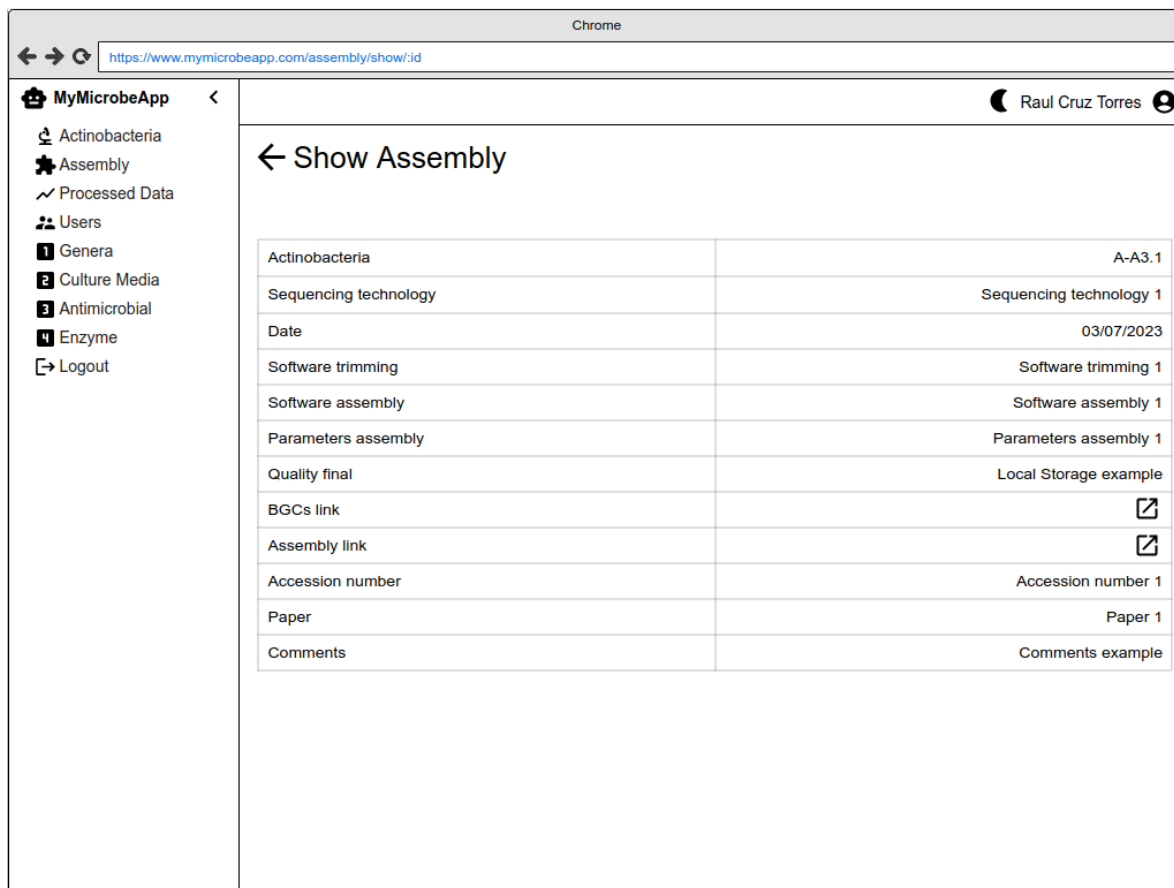


Figura 25: Mockup Assembly Show
Fuente: Elaboración propia.

The image shows a web browser window with the URL <https://www.mymicrobeapp.com/assembly/create>. The page title is "Create Assembly". On the left, there is a sidebar menu for "MyMicrobeApp" with the following items: Actinobacteria, Assembly, Processed Data, Users, Genera (highlighted with a '1'), Culture Media (highlighted with a '2'), Antimicrobial (highlighted with a '3'), Enzyme (highlighted with a '4'), and Logout. The main content area contains a form with the following fields: a dropdown menu for "Actinobacteria", a text input for "Sequencing technology", a date input for "DD/MM/YYYY", a text input for "Software trimming", a text input for "Software assembly", a text input for "Parameters assembly", a text input for "Quality final", a text input for "BGCs link", a text input for "Assembly link", a text input for "Accession number", a text input for "Paper", and a text area for "Comments". A "Create" button is located at the bottom right of the form.

Actinobacteria	
Sequencing technology	DD/MM/YYYY
Software trimming	Software assembly
Parameters assembly	Quality final
BGCs link	Assembly link
Accession number	Paper
Comments	

Create

Figura 26: Mockup Assembly Create
Fuente: Elaboración propia.

The mockup shows a web browser window with the URL `https://www.mymicrobeapp.com/assembly/edit/id`. The application header includes the 'MyMicrobeApp' logo and a user profile for 'Raul Cruz Torres'. A left sidebar contains navigation links: 'Actinobacteria', 'Assembly', 'Processed Data', 'Users', 'Genera', 'Culture Media', 'Antimicrobial', 'Enzyme', and 'Logout'. The main content area is titled 'Edit Assembly' and contains a form with the following fields:

- Assembly ID:** A dropdown menu with the value 'A-A3.1'.
- Sequencing technology 1:** A text input field.
- Date:** A date picker showing '03/07/2023'.
- Software trimming 1:** A text input field.
- Software assembly 1:** A text input field.
- Parameters assembly 1:** A text input field.
- Quality final 1:** A text input field.
- URL 1:** A text input field with the value 'https://example.com/bgcs/'.
- URL 2:** A text input field with the value 'https://example.com/assembly/'.
- Accession number 1:** A text input field.
- Paper 1:** A text input field.
- Comments:** A large text area with the placeholder text 'Comments example'.

An 'Edit' button is located at the bottom right of the form.

Figura 27: Mockup Assembly Edit
Fuente: Elaboración propia.

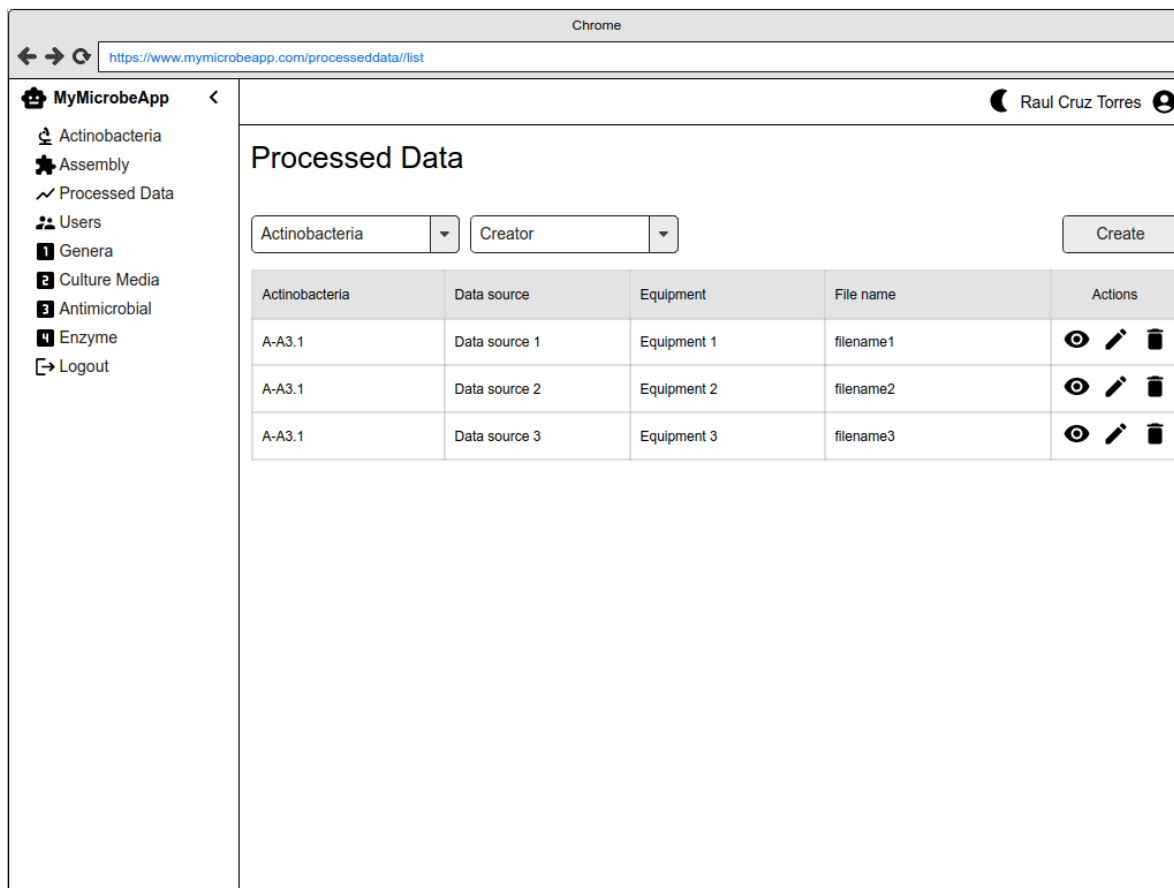


Figura 28: Mockup Processed Data List
Fuente: Elaboración propia.

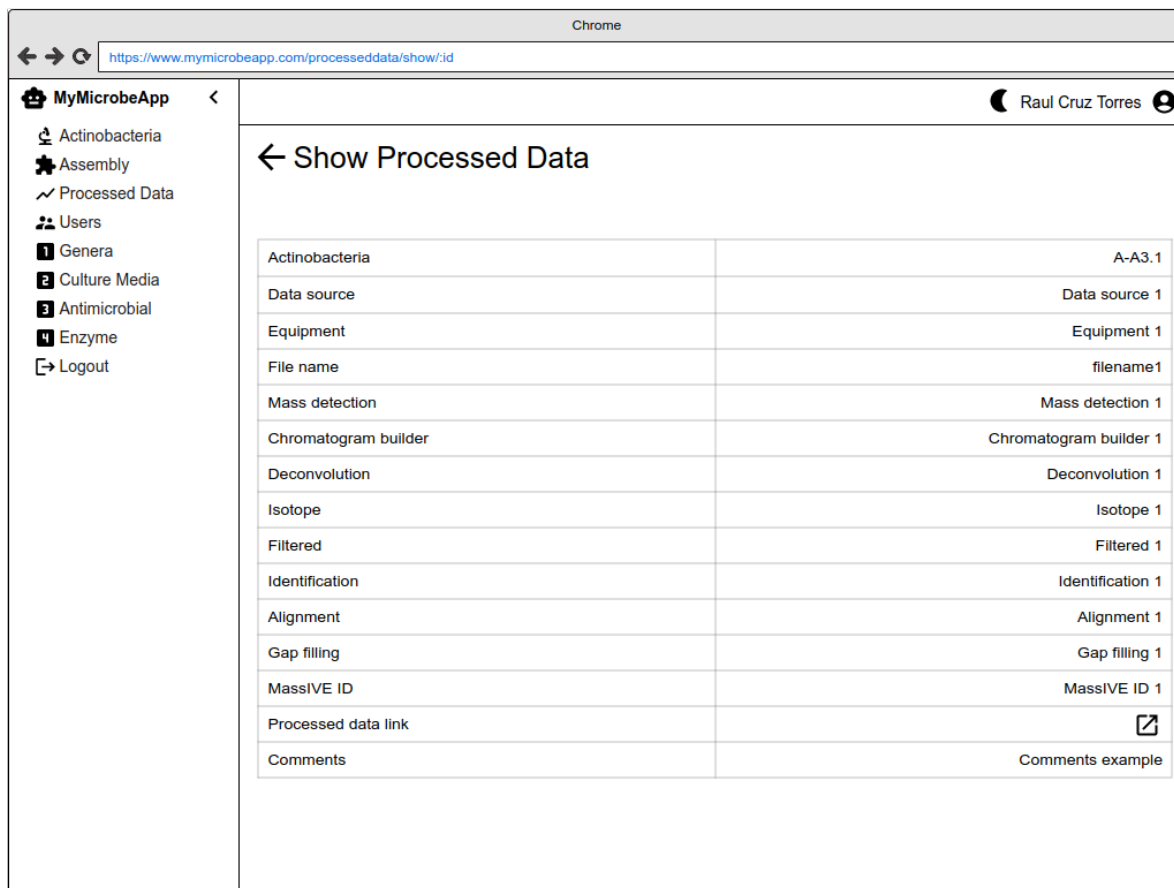


Figura 29: Mockup Processed Data Show
Fuente: Elaboración propia.

The image shows a web browser window with the URL <https://www.mymicrobeapp.com/processeddata/create>. The page title is "Create Processed Data". On the left is a sidebar menu for "MyMicrobeApp" with options: Actinobacteria, Assembly, Processed Data (selected), Users, Genera, Culture Media, Antimicrobial, Enzyme, and Logout. The main content area contains a form with the following fields: "Actinobacteria" (a dropdown menu), "Data source", "Equipment", "File name", "Mass detection", "Chromatogram builder", "Deconvolution", "Isotope", "Filtered", "Identification", "Alignment", "Gap filling", "MassIVE ID", "Processed data link", and "Comments" (a text area). A "Create" button is located at the bottom right of the form. The user's name "Raul Cruz Torres" is displayed in the top right corner.

Figura 30: Mockup Processed Data Create
Fuente: Elaboración propia.

Chrome

← → ↻ <https://www.mymicrobeapp.com/processeddata/edit/id>

MyMicrobeApp

Actinobacteria

Assembly

✓ Processed Data

Users

1 Genera

2 Culture Media

3 Antimicrobial

4 Enzyme

Logout

← Edit Processed Data

A-A3.1

Equipment 1

Mass detection 1

Deconvolution 1

Filtered 1

Alignment 1

MassIVE ID 1

Data source 1

filename1

Chromatogram builder 1

Isotope 1

Identification 1

Gap filling 1

<https://example.com/processeddata/>

Comments example

Edit

Figura 31: Mockup Processed Data Edit
Fuente: Elaboración propia.

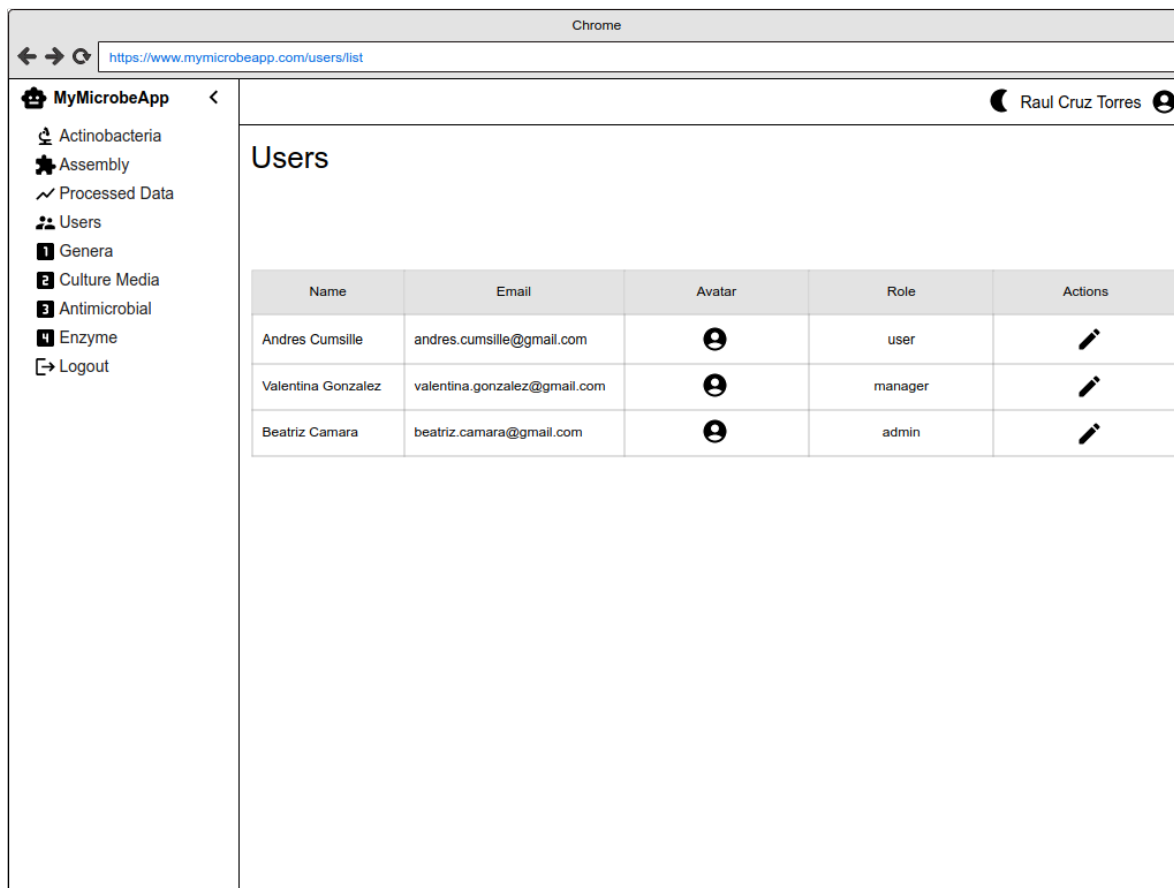


Figura 32: Mockup Users List
Fuente: Elaboración propia.

Chrome

https://www.mymicrobeapp.com/users/edit/:id

MyMicrobeApp

- Actinobacteria
- Assembly
- Processed Data
- Users
- 1 Genera
- 2 Culture Media
- 3 Antimicrobial
- 4 Enzyme
- Logout

Raul Cruz Torres

← Edit User

Andres Cumsille

andres.cumsille@gmail.com

user

Edit

Figura 33: Mockup Users Edit
Fuente: Elaboración propia.

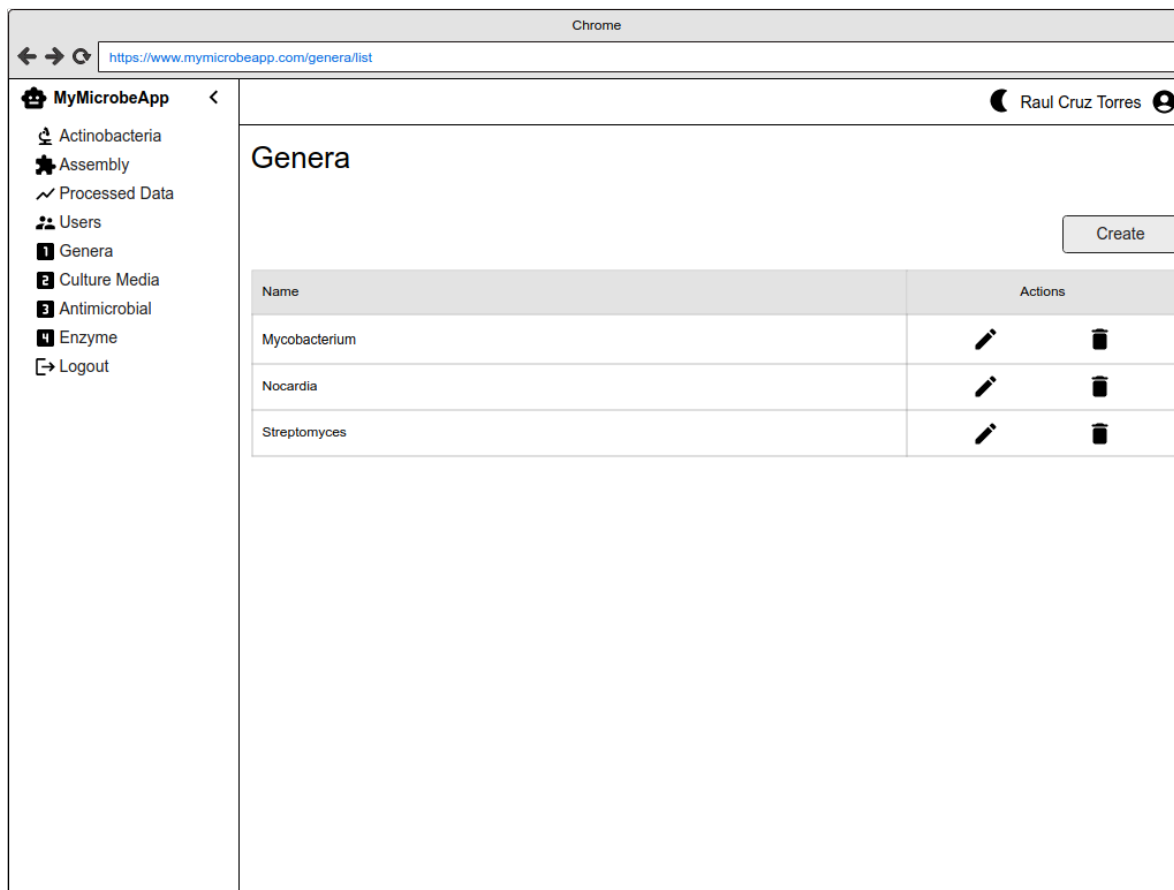


Figura 34: Mockup Genera List
Fuente: Elaboración propia.

The image shows a web browser window with the address bar displaying <https://www.mymicrobeapp.com/genera/create>. The browser is identified as Chrome. The application interface has a dark sidebar on the left with the logo 'MyMicrobeApp' and a list of menu items: Actinobacteria, Assembly, Processed Data, Users, Genera (highlighted with a red square), Culture Media (highlighted with a red square), Antimicrobial (highlighted with a red square), Enzyme (highlighted with a red square), and Logout. The main content area has a header with a back arrow and the title 'Create Genus'. Below the title is a large text input field with the placeholder text 'Name'. At the bottom right of the main area is a 'Create' button. The top right of the application shows a user profile icon and the name 'Raul Cruz Torres'.

Figura 35: Mockup Genera Create
Fuente: Elaboración propia.

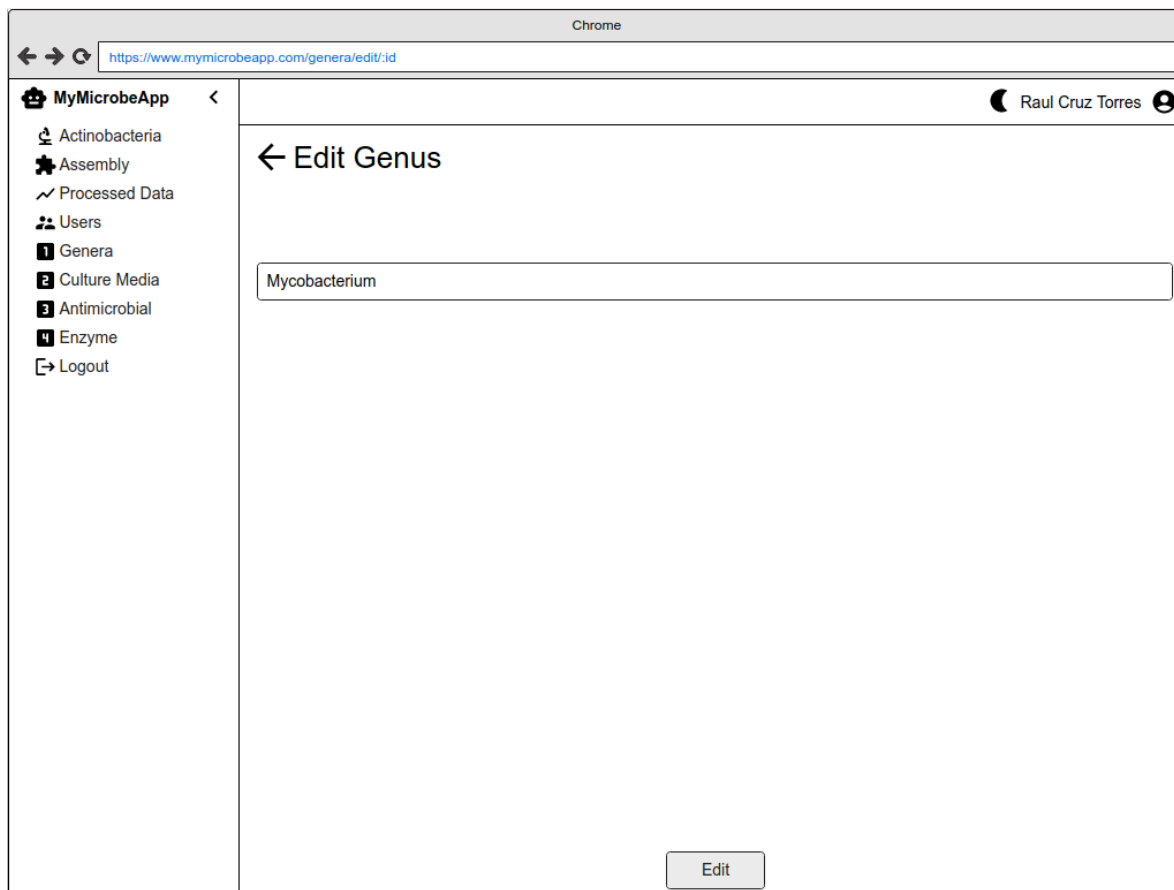


Figura 36: Mockup Genera Edit
Fuente: Elaboración propia.

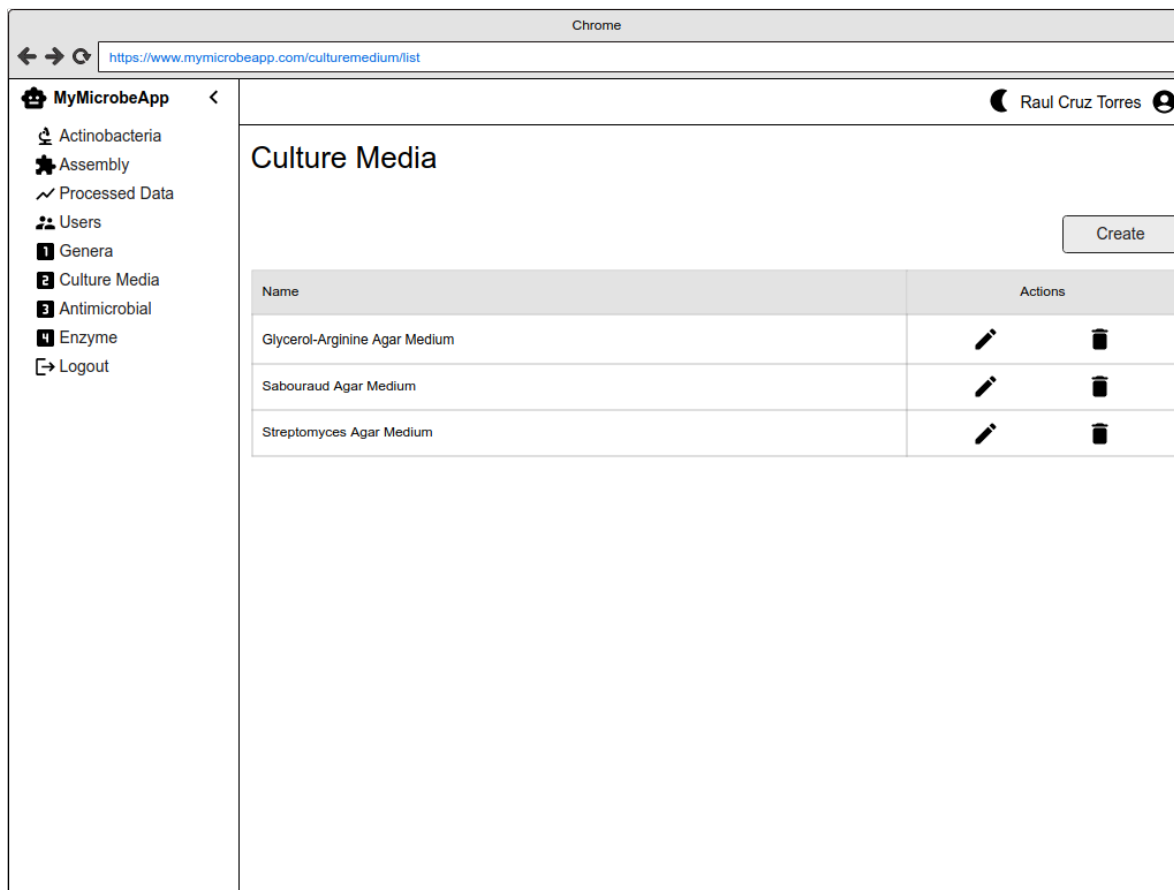


Figura 37: Mockup Culture Media List
Fuente: Elaboración propia.

The image shows a web browser window with the address bar displaying <https://www.mymicrobeapp.com/culturemedium/create>. The browser is identified as Chrome. The application interface features a sidebar on the left with the logo 'MyMicrobeApp' and a list of navigation items: 'Actinobacteria', 'Assembly', 'Processed Data', 'Users', '1 Genera', '2 Culture Media' (highlighted), '3 Antimicrobial', '4 Enzyme', and 'Logout'. The main content area is titled '← Create Culture Media'. It contains a single text input field labeled 'Name'. At the bottom right of the main area is a 'Create' button. The top right of the application shows a user profile for 'Raul Cruz Torres'.

Figura 38: Mockup Culture Media Create
Fuente: Elaboración propia.

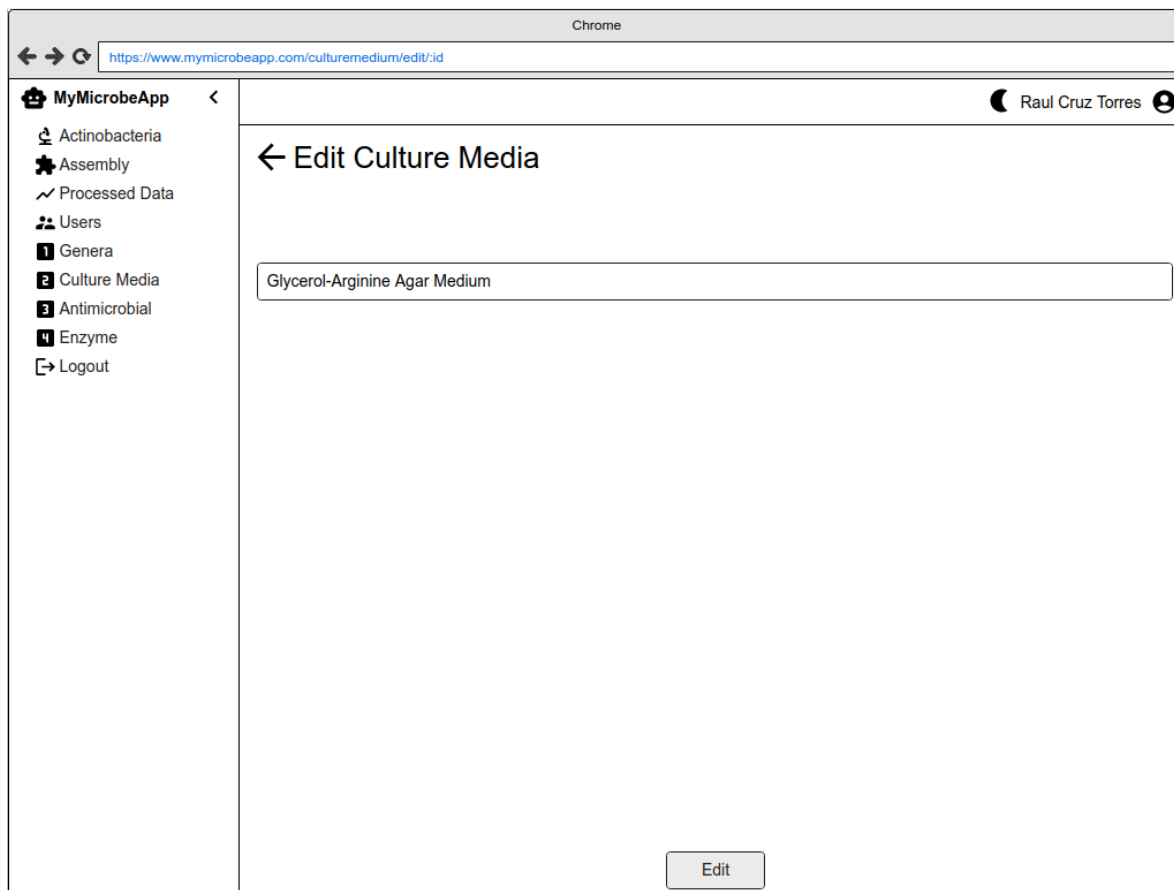


Figura 39: Mockup Culture Media Edit
Fuente: Elaboración propia.

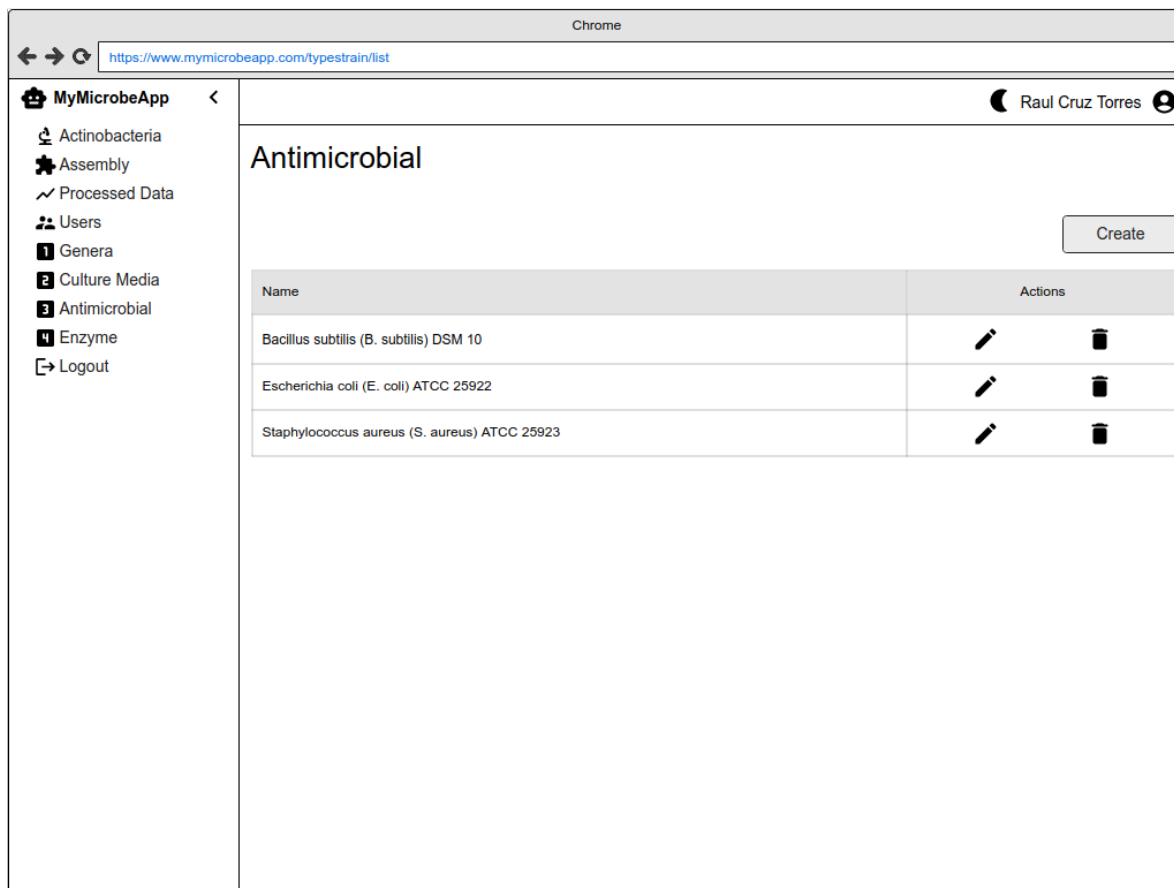


Figura 40: Mockup Antimicrobial List
Fuente: Elaboración propia.

The image shows a web browser window with the address bar displaying <https://www.mymicrobeapp.com/typetrain/create>. The browser is identified as Chrome. The application interface features a sidebar on the left with the logo 'MyMicrobeApp' and a list of navigation items: Actinobacteria, Assembly, Processed Data, Users, Genera (highlighted with a '1'), Culture Media (highlighted with a '2'), Antimicrobial (highlighted with a '3'), Enzyme (highlighted with a '4'), and Logout. The main content area is titled 'Create Antimicrobial' with a back arrow. It contains a single text input field labeled 'Name'. At the bottom right of the main area is a 'Create' button. The top right of the application shows a user profile for 'Raul Cruz Torres'.

Figura 41: Mockup Antimicrobial Create
Fuente: Elaboración propia.

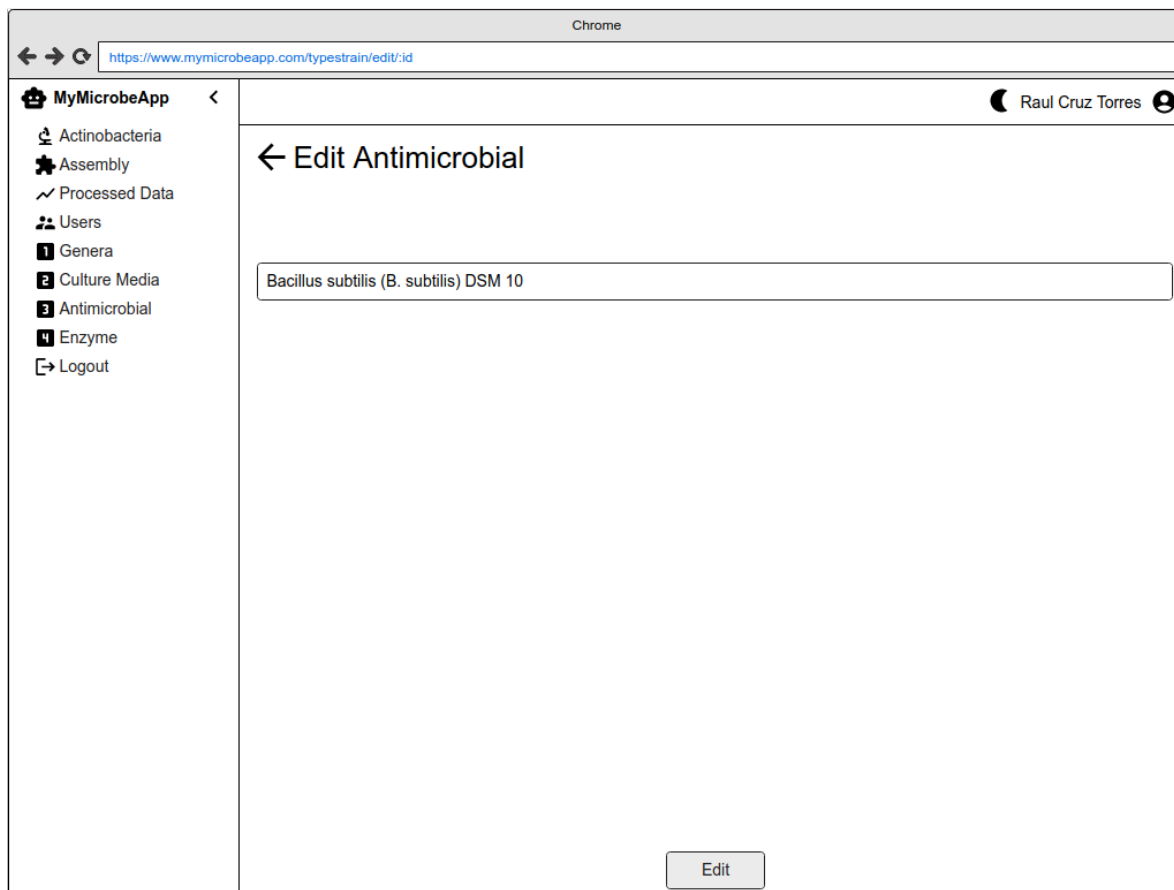


Figura 42: Mockup Antimicrobial Edit
Fuente: Elaboración propia.

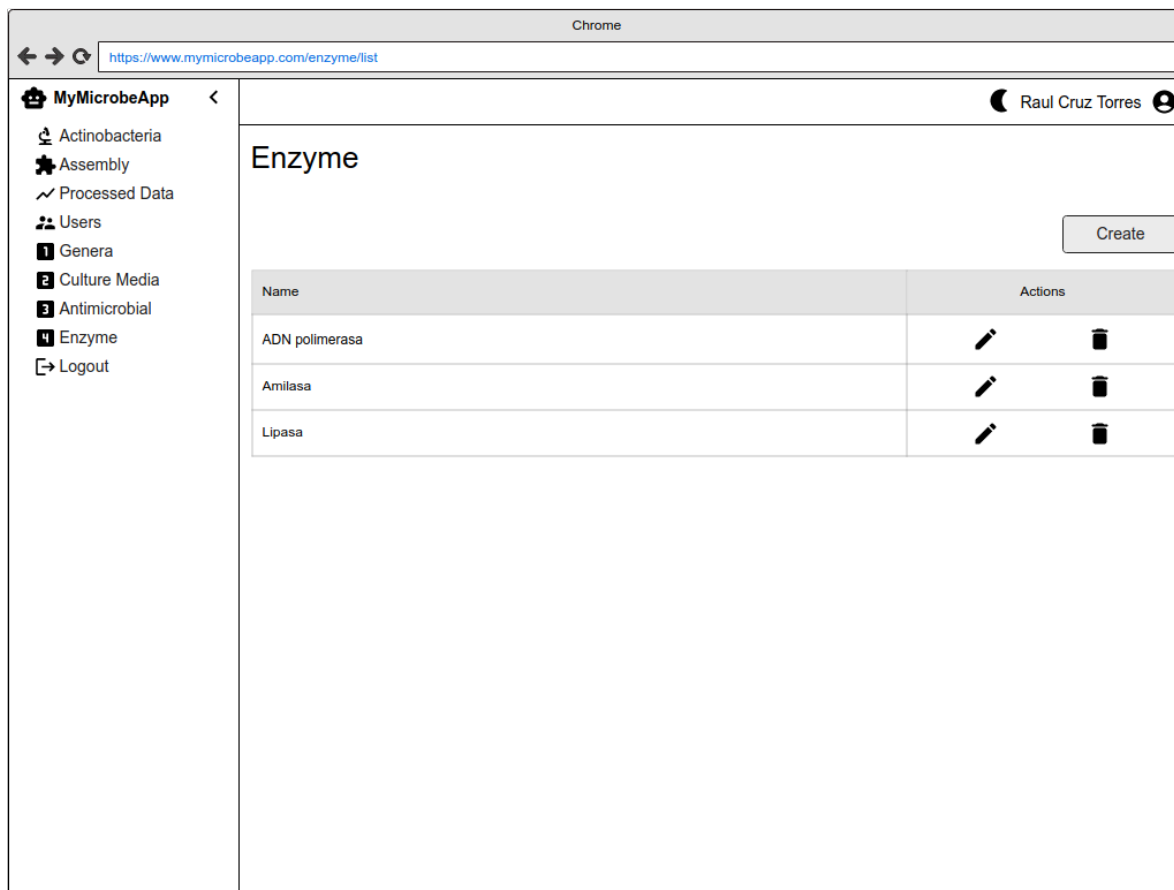


Figura 43: Mockup Enzyme List
Fuente: Elaboración propia.

The image shows a web browser window with the address bar displaying <https://www.mymicrobeapp.com/enzyme/create>. The page title is "Create Enzyme". On the left, there is a sidebar menu for "MyMicrobeApp" with the following items: Actinobacteria, Assembly, Processed Data, Users, Genera (highlighted with a '1'), Culture Media (highlighted with a '2'), Antimicrobial (highlighted with a '3'), Enzyme (highlighted with a '4'), and Logout. The main content area has a heading "Create Enzyme" with a back arrow. Below the heading is a text input field labeled "Name". At the bottom right of the main area is a "Create" button. The top right of the page shows a user profile for "Raul Cruz Torres".

Figura 44: Mockup Enzyme Create
Fuente: Elaboración propia.

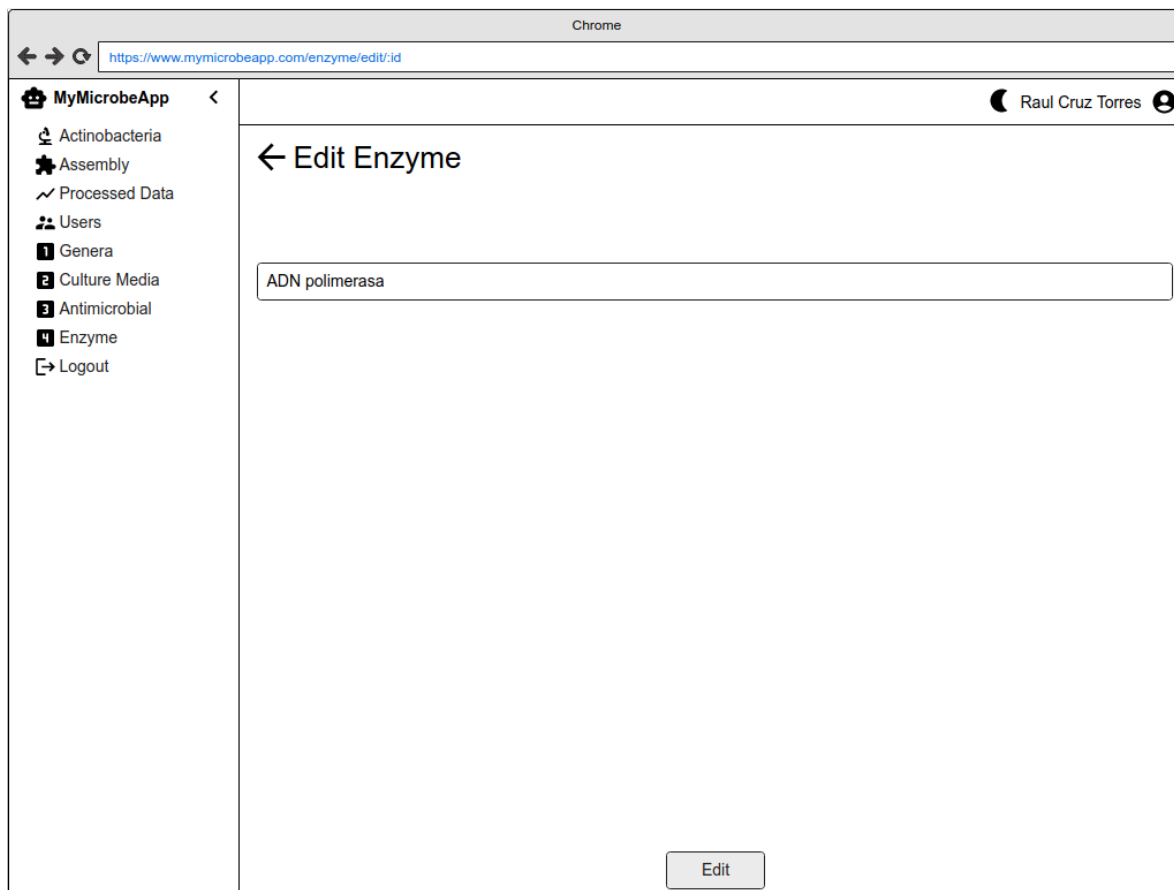


Figura 45: Mockup Enzyme Edit
Fuente: Elaboración propia.

ANEXO B: Interfaz de la aplicación

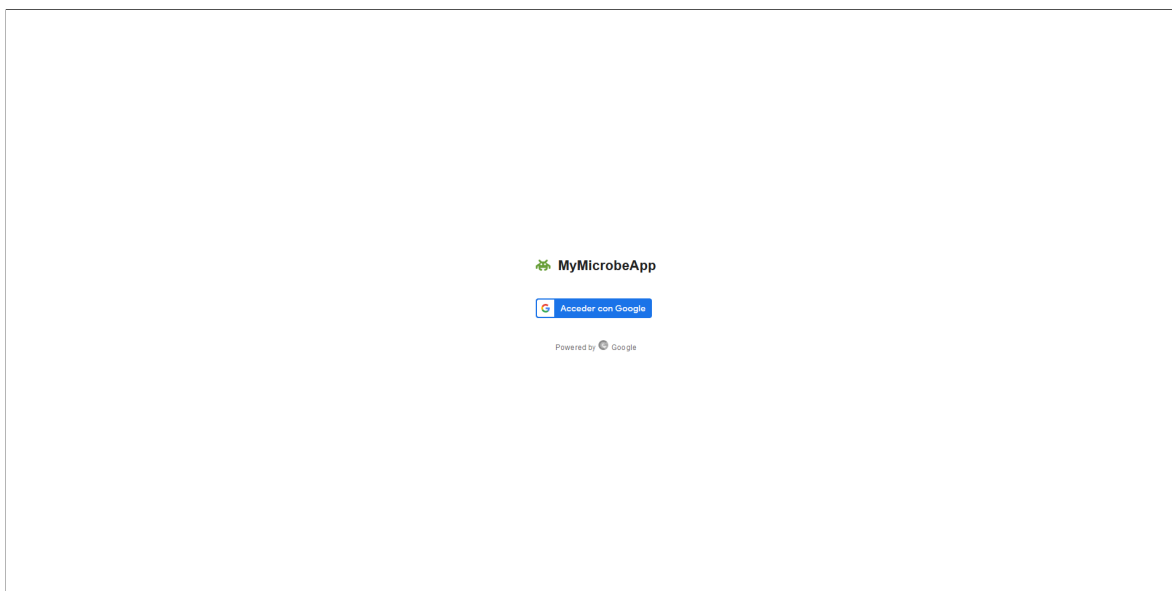


Figura 46: Interfaz Inicio de sesión
Fuente: Elaboración propia.

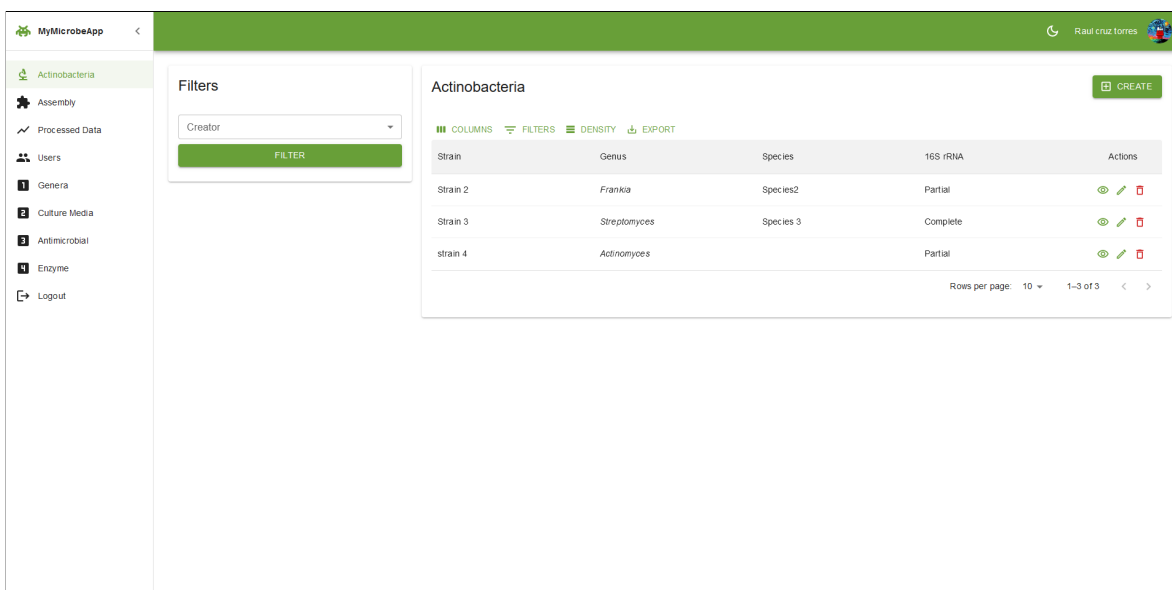


Figura 47: Interfaz Actinobacteria List
Fuente: Elaboración propia.

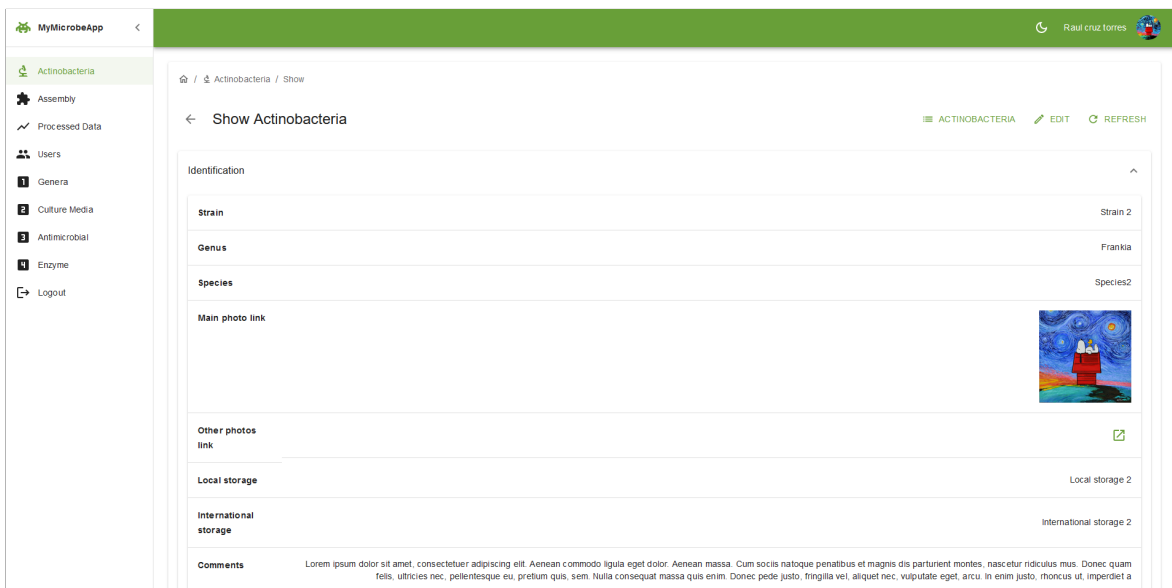


Figura 48: Interfaz Actinobacteria Show
Fuente: Elaboración propia.

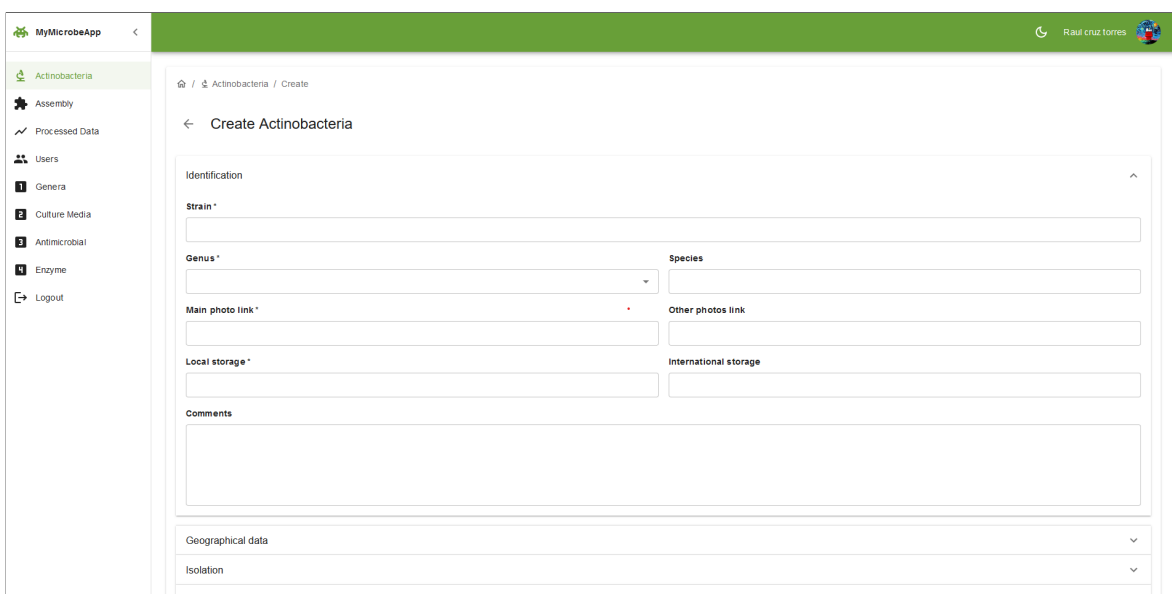


Figura 49: Interfaz Actinobacteria Create
Fuente: Elaboración propia.

MyMicrobeApp

Actinobacteria / Edit

← Edit Actinobacteria

ACTINOBACTERIA REFRESH

Identification

Strain *

Strain 2

Genera *

Frankia

Species

Species2

Main photo link *

https://upload.wikimedia.org/wikipedia/commons/thumb/4/43/Snoopy_and_Woodstock_in_the_Starry_I

Other photos link

<https://cbdal.usm.cl/>

Local storage *

Local storage 2

International storage

International storage 2

Comments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a

Geographical data

Isolation

Genomic data

Figura 50: Interfaz Actinobacteria Edit
Fuente: Elaboración propia.

MyMicrobeApp

Actinobacteria

Assembly

Processed Data

Users

Genera

Culture Media

Antimicrobial

Enzyme

Logout

Filters

Actinobacteria

Creator

FILTER

Assemblies

CREATE

COLUMNS FILTERS DENSITY EXPORT

Actinobacteria	Date	Software trimming	Software assembly	Parameters assembly	Actions
strain 4	11/04/2022	a	a	a	
Strain 2	25/04/2022	b	b	b	
Strain 2	13/10/2023	Software trimming 1	Software assembly 1	Parameters assembly 1	
Strain 3	18/10/2023	a	a	a	
strain 4	30/10/2023	c	c	c	

Rows per page: 10 1-5 of 5

Figura 51: Interfaz Assembly List
Fuente: Elaboración propia.

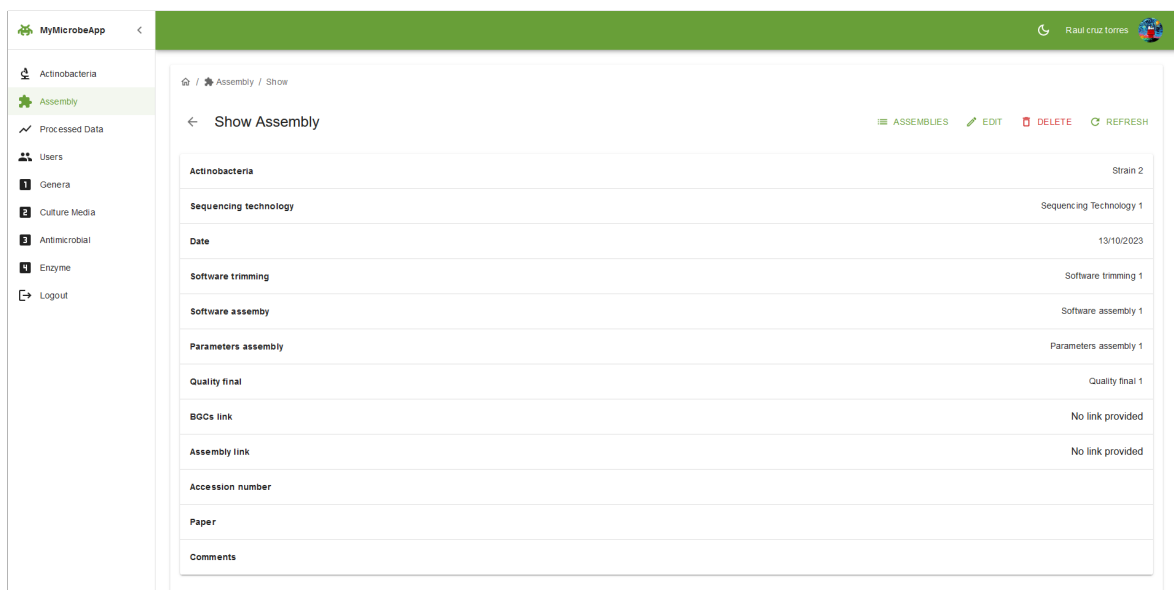


Figura 52: Interfaz Assembly Show
Fuente: Elaboración propia.

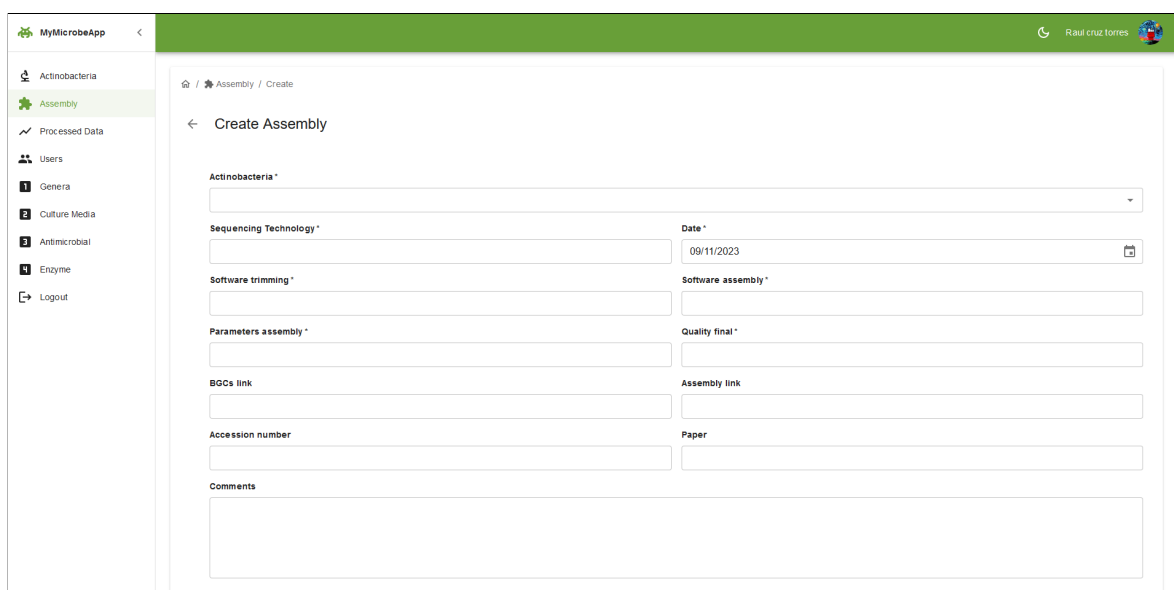


Figura 53: Interfaz Assembly Create
Fuente: Elaboración propia.

MyMicrobeApp < Raul cruz torres

Actinobacteria
Assembly
Processed Data
Users
Genera
Culture Media
Antimicrobial
Enzyme
Logout

Assembly / Edit

← Edit Assembly ASSEMBLIES REFRESH

Actinobacteria *
Strain 2

Sequencing Technology *
Sequencing Technology 1

Date *
13/10/2023

Software trimming *
Software trimming 1

Software assembly *
Software assembly 1

Parameters assembly *
Parameters assembly 1

Quality final *
Quality final 1

BGCs link

Assembly link

Accession number

Paper

Comments

Figura 54: Interfaz Assembly Edit
Fuente: Elaboración propia.

MyMicrobeApp < Raul cruz torres

Actinobacteria
Assembly
Processed Data
Users
Genera
Culture Media
Antimicrobial
Enzyme
Logout

Filters

Actinobacteria
Creator
FILTER

Processed Data CREATE

COLUMNS FILTERS DENSITY EXPORT

Actinobacteria	Data source	Equipment	File name	Actions
strain 4	ds	e	tn	

Rows per page: 10 1-1 of 1

Figura 55: Interfaz Processed Data List
Fuente: Elaboración propia.

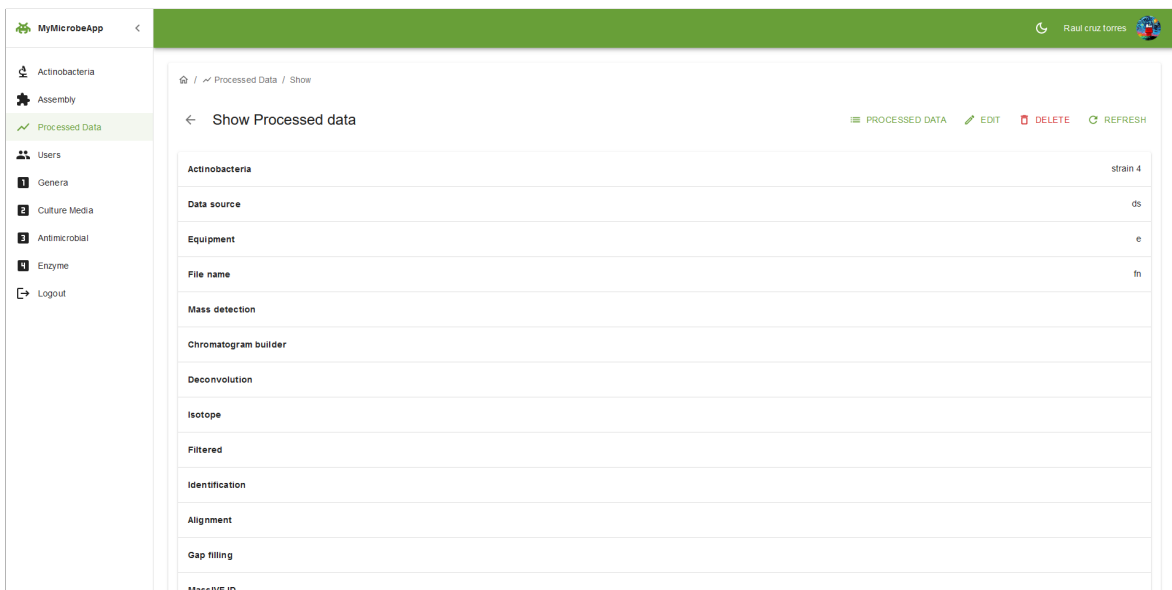


Figura 56: Interfaz Processed Data Show
Fuente: Elaboración propia.

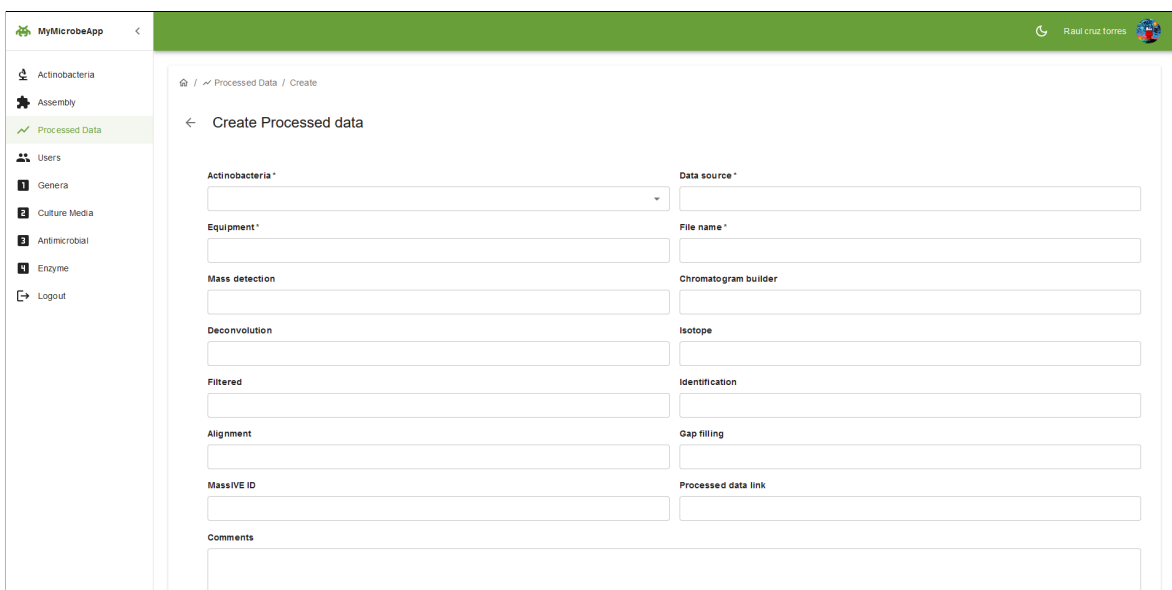


Figura 57: Interfaz Processed Data Create
Fuente: Elaboración propia.

Figura 58: Interfaz Processed Data Edit
Fuente: Elaboración propia.

Name	Email	Avatar	Role	Actions
Raul Cruz	raul.cruz@sansano.usm.cl		manager	
Raul cruz torres	cruztorresraul@gmail.com		admin	

Figura 59: Interfaz Users List
Fuente: Elaboración propia.

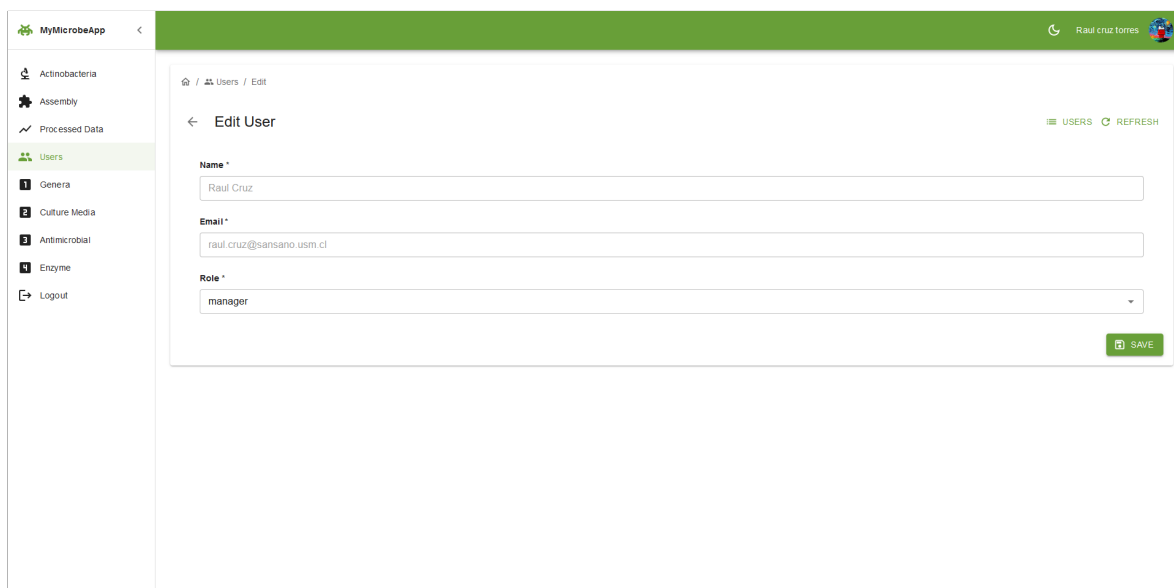


Figura 60: Interfaz Users Edit
Fuente: Elaboración propia.

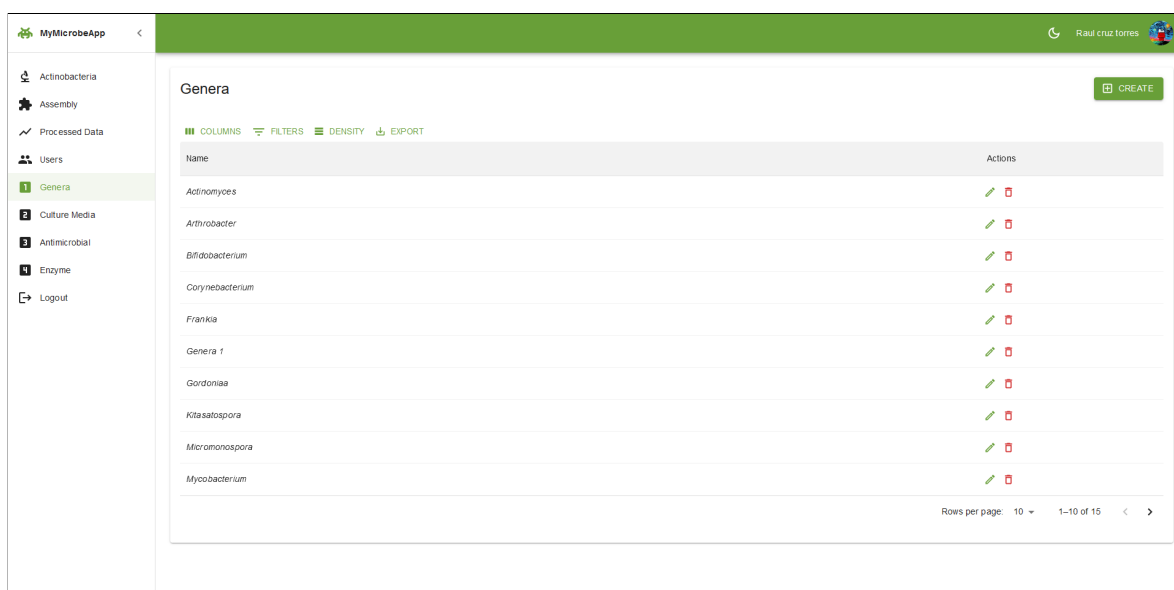


Figura 61: Interfaz Genera List
Fuente: Elaboración propia.

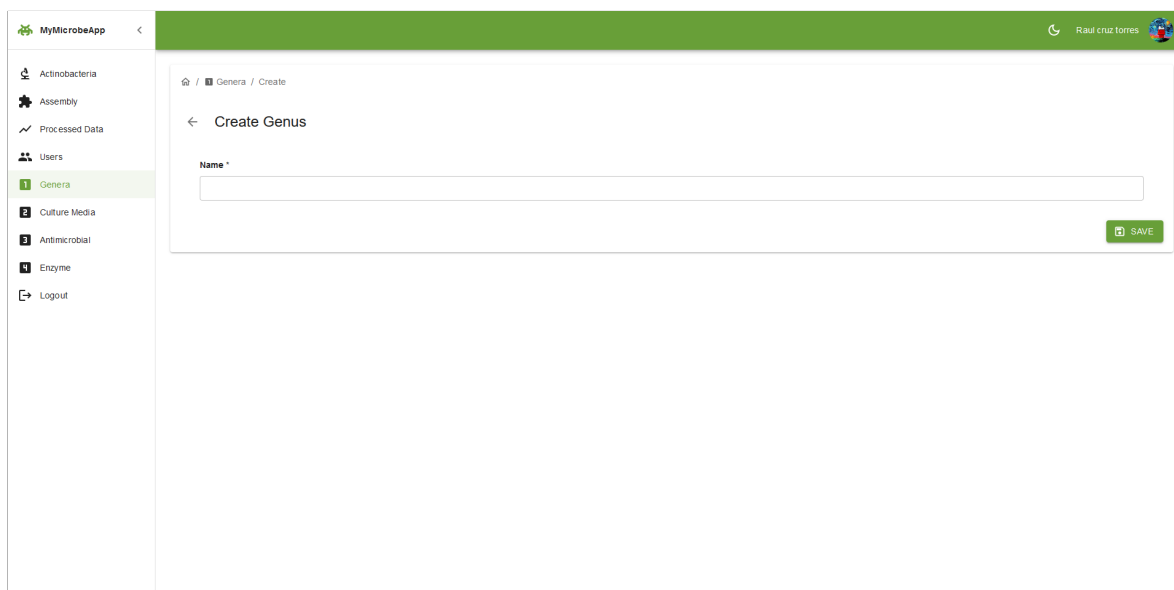


Figura 62: Interfaz Genera Create
Fuente: Elaboración propia.

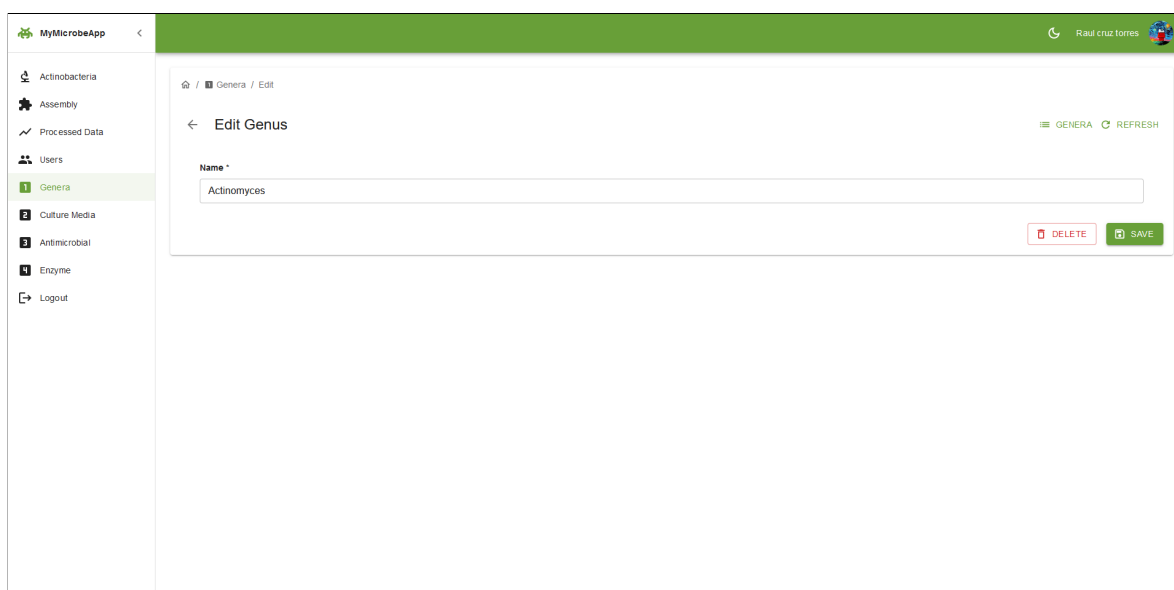


Figura 63: Interfaz Genera Edit
Fuente: Elaboración propia.

Name	Actions
Agar actinomycete isolation agar	
Agar almidón-carboximetilcelulosa	
Agar bennett	
Agar caseína	
Agar czapek-dox	
Agar glicerol-almidón	
Agar humate	
Agar inorgánico	
Agar isp (international streptomyces project)	
Agar nutrient	

Figura 64: Interfaz Culture Media List
Fuente: Elaboración propia.

← Create Culture Media

Name *

SAVE

Figura 65: Interfaz Culture Media Create
Fuente: Elaboración propia.

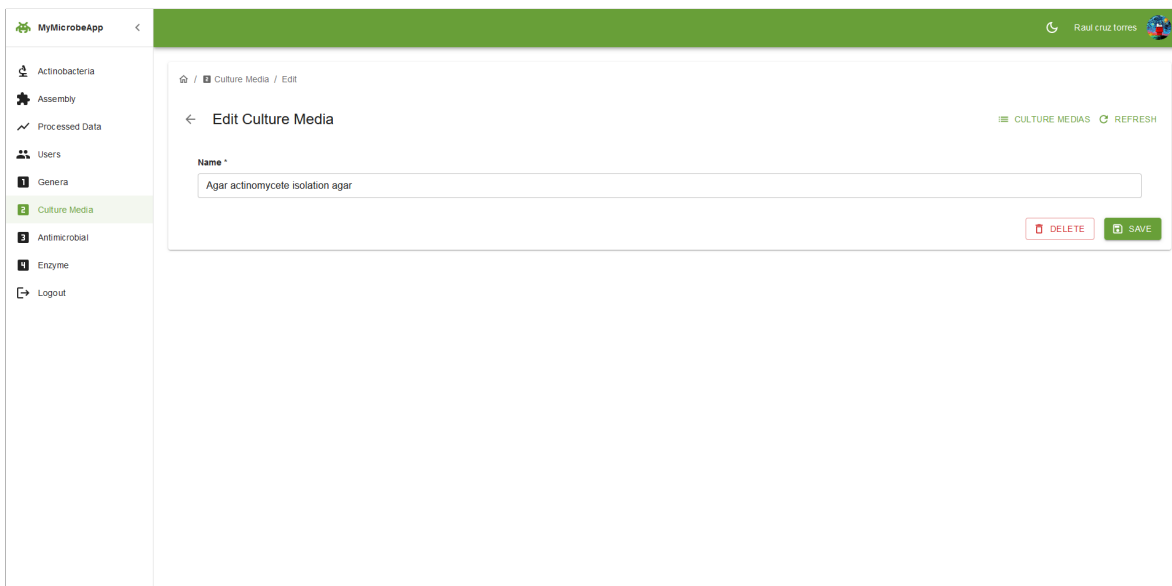


Figura 66: Interfaz Culture Media Edit
Fuente: Elaboración propia.

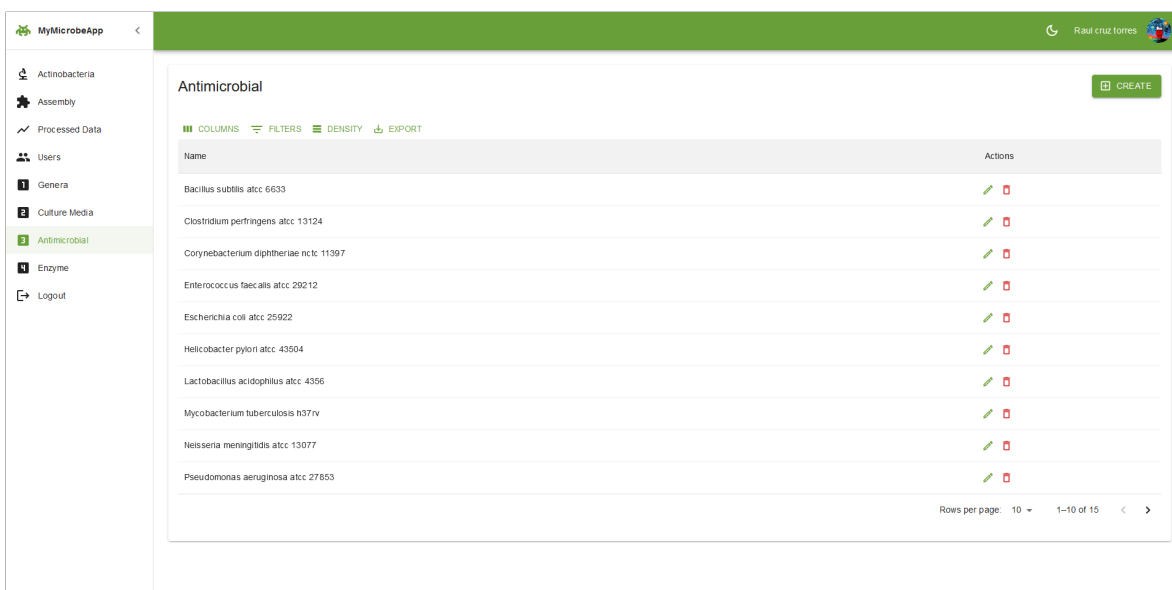


Figura 67: Interfaz Antimicrobial List
Fuente: Elaboración propia.

MyMicrobeApp

Actinobacteria
Assembly
Processed Data
Users
Genera
Culture Media
Antimicrobial
Enzyme
Logout

Antimicrobial / Create

Create Antimicrobial

Name *

SAVE

Figura 68: Interfaz Antimicrobial Create
Fuente: Elaboración propia.

MyMicrobeApp

Actinobacteria
Assembly
Processed Data
Users
Genera
Culture Media
Antimicrobial
Enzyme
Logout

Antimicrobial / Edit

Edit Antimicrobial

ANTIMICROBIALS REFRESH

Name *

Bacillus subtilis atcc 6633

DELETE SAVE

Figura 69: Interfaz Antimicrobial Edit
Fuente: Elaboración propia.

The screenshot shows the 'MyMicrobeApp' interface. On the left is a sidebar with navigation links: Actinobacteria, Assembly, Processed Data, Users, Genera, Culture Media, Antimicrobial, Enzyme (highlighted), and Logout. The main content area is titled 'Enzymes' and includes a 'CREATE' button. Below the title are tabs for COLUMNS, FILTERS, DENSITY, and EXPORT. A table lists various enzymes with their names and action icons (edit and delete). The table has 14 rows. At the bottom right, there is a pagination control showing 'Rows per page: 10' and '1-10 of 14'.

Name	Actions
Amidasas	
Amilasas	
Celulasas	
Fenotoxidasa	
Fosfatasa	
Ligninasas	
Lipasas	
Nitrogenasa	
Pectinasas	
Poli fenol oxidasa	

Figura 70: Interfaz Enzyme List
Fuente: Elaboración propia.

The screenshot shows the 'MyMicrobeApp' interface with the 'Enzyme / Create' form. The sidebar is the same as in Figure 70. The main content area has a breadcrumb 'Enzyme / Create' and a back arrow. Below is a form titled 'Create Enzyme' with a single text input field labeled 'Name *'. A 'SAVE' button is located at the bottom right of the form.

Enzyme / Create

← Create Enzyme

Name *

SAVE

Figura 71: Interfaz Enzyme Create
Fuente: Elaboración propia.

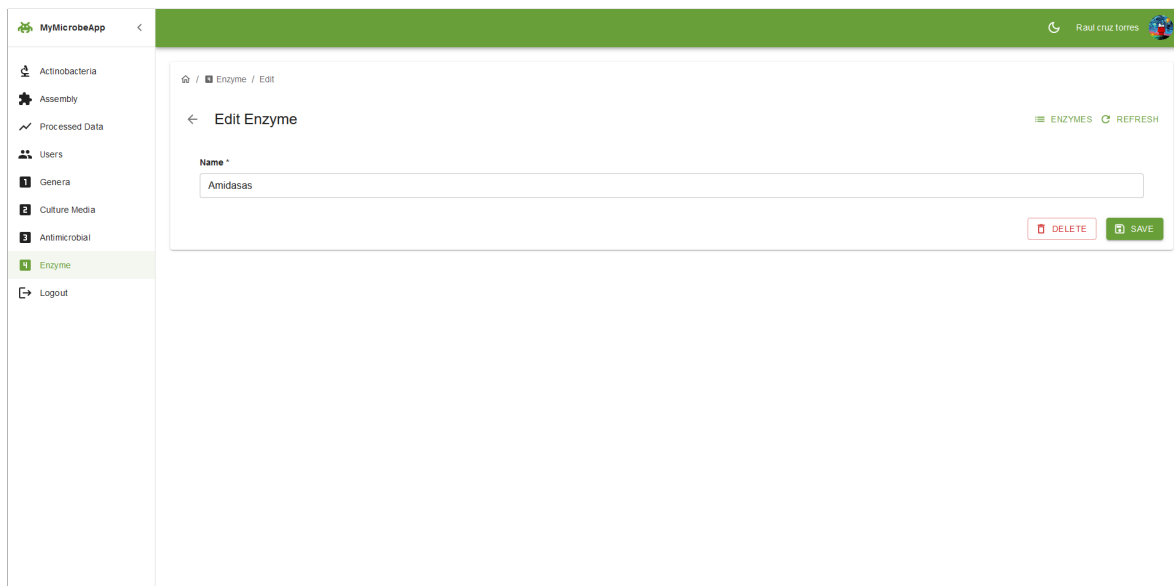


Figura 72: Interfaz Enzyme Edit
Fuente: Elaboración propia.

REFERENCIAS BIBLIOGRÁFICAS

- [Mon, 2023] (2023). mongoose. <https://mongoosejs.com/>.
- [Anandan *et al.*, 2016] Anandan, R., Dharumadurai, D., y Manogaran, G. P. (2016). An introduction to actinobacteria. En Dhanasekaran, D. y Jiang, Y., editores, *Actinobacteria*, capítulo 1. IntechOpen, Rijeka.
- [Angular, 2022] Angular (2022). <https://angular.io/guide/what-is-angular>.
- [Brajovic, 2020] Brajovic, F. (2020). Todo sobre cromatografía líquida con espectrometría de masas. <https://www.cromtek.cl/2020/07/22/todo-sobre-cromatografia-liquida-con-espectrometria-de-masas/>.
- [Brenes-Guillén, 2019] Brenes-Guillén, L. (2019). View of ensamblaje y anotación de genomas microbianos: Revista de biología tropical. <https://revistas.ucr.ac.cr/index.php/rbt/article/view/35899/36641>.
- [Buckley, 2020] Buckley, G. (2020). Prokaryotic cell - definition, examples and structure. <https://biologydictionary.net/prokaryotic-cell/>.
- [Busquets, 2021] Busquets, C. (2021). Cómo medir la usabilidad con un sus. <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>.
- [Busquets, 2022] Busquets, C. (2022). Escala de likert: Qué es y cómo usarla en investigación ux. <https://www.uifrommars.com/escala-likert-que-es-investigacion-ux/>.
- [Chen y Lee, 2018] Chen, J.-K. y Lee, W.-Z. (2018). A study of nosql database for enterprises. En *2018 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 436–440. IEEE.
- [Cherwell Labs, 2023] Cherwell Labs (2023). Growth media types. <https://www.cherwell-labs.co.uk/growth-media-types>.
- [Claverías, 2014] Claverías, F. (2014). Aislamiento de actinobacterias marinas y búsqueda de genes involucrados en la síntesis de compuestos bioactivos.
- [Clinical Info, 2022] Clinical Info (2022). Cepa. <https://clinicalinfo.hiv.gov/es/glossary/cepa>.
- [Codd, 1970] Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.
- [Contador *et al.*, 2020] Contador, C. A., Veas-Castillo, L., Tapia, E., Antipán, M., Miranda, N., Ruiz-Tagle, B., García-Araya, J., Andrews, B. A., Marin, M., y Dorador, C. (2020). Atacama database: a platform of the microbiome of the atacama desert. *Antonie Van Leeuwenhoek*, 113(2):185–195.

- [Contreras, 2021] Contreras, M. (2021). Tutorial de mongodb atlas - ¿cómo empezar? <https://www.freecodecamp.org/espanol/news/tutorial-de-mongodb-atlas-como-empezar/>.
- [Contributors, 2022] Contributors, M. (2022). What is javascript? - learn web development: Mdn. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
- [Cooke, 2022] Cooke, C. (2022). Visualize your workflow with kanban boards in jira. <https://www.oldstreetsolutions.com/visualize-workflow-kanban-jira>.
- [Coppola, 2022] Coppola, M. (2022). Frontend y backend: Qué son, en qué se diferencian y ejemplos. <https://blog.hubspot.es/website/frontend-y-backend>.
- [Corzo, 2017] Corzo, G. (2017). Programación asíncrona. <https://medium.com/laboratoria-how-to/programacion-asincrona-cea3bad7c3c6>.
- [Cumsille, 2015] Cumsille, A. (2015). Biodiversidad y actividad antibacteriana de actinobacterias aisladas desde sedimentos y esponjas marinas.
- [del Rosario Rodicio y del Carmen Mendoza, 2004] del Rosario Rodicio, M. y del Carmen Mendoza, M. (2004). Identification of bacteria through 16s rRNA sequencing: principles, methods and applications in clinical microbiology. *Enfermedades infecciosas y microbiología clínica*, 22(4):238-245.
- [Dionisi et al., 2012] Dionisi, H. M., Lozada, M., y Olivera, N. L. (2012). Bioprospection of marine microorganisms: biotechnological applications and methods. *Revista argentina de microbiología*, 44(1):49-60.
- [Fryrear, 2022] Fryrear, A. (2022). Going agile on your own. <https://www.agilesherpas.com/blog/going-agile-alone>.
- [Gill, 2022] Gill, N. S. (2022). Sql vs nosql vs newsql: The full comparison. <https://www.xenonstack.com/blog/sql-vs-nosql-vs-newsql>.
- [Gillis, 2020] Gillis, A. (2020). What is rest api (restful api)? <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>.
- [González, 2015] González, V. (2015). Aislamiento y caracterización de actinobacterias marinas y de ambientes extremos y búsqueda de su potencial rol en biotecnología.
- [González, 2021] González, V. (2021). Bioprospección de actinobacterias marinas: Búsqueda de enzimas implicadas en el mecanismo de degradación de la queratina.
- [Green, 2023] Green, E. (2023). Shotgun sequencing. <https://www.genome.gov/genetics-glossary/Shotgun-Sequencing>.

- [Hakoune, 2022] Hakoune, R. (2022). Everything you need to know about kanban project management. <https://monday.com/blog/project-management/kanban-project-managment/>.
- [Hiberus, 2023] Hiberus (2023). ¿qué es el middleware? ¿para qué sirve? <https://www.hiberus.com/crecemos-contigo/que-es-el-middleware/>.
- [IBM, 2023] IBM (2023). ¿qué es mongodb? <https://www.ibm.com/ar-es/topics/mongodb>.
- [Ivanovs, 2023] Ivanovs, A. (2023). The most popular front-end frameworks in 2023. <https://stackdiary.com/front-end-frameworks/>.
- [Kaur, 2022] Kaur, P. (2022). 10 most popular frameworks for building restful apis. <https://www.moesif.com/blog/api-product-management/api-analytics/10-Most-Popular-Frameworks-For-Building-RESTful-APIs/>.
- [KeepCoding, 2022] KeepCoding, R. (2022). Cuando usar contexto en react. <https://keepcoding.io/blog/cuando-usar-contexto-en-react/>.
- [KMS Solutions, 2022] KMS Solutions (2022). Mobile application development: A comprehensive guide. <https://blog.kms-solutions.asia/an-ultimate-guide-to-mobile-application-development>.
- [Kuddus, 2019] Kuddus, M. (2019). *Enzymes in food biotechnology: Production, applications, and future prospects*. Academic Press, an imprint of Elsevier.
- [Kuate, 2017] Kuate, V. (2017). *Medicinal spices and vegetables from Africa: therapeutic potential against metabolic, inflammatory, infectious and systemic diseases*. Academic Press.
- [LabSerGen, 2022] LabSerGen (2022). Identificación de género y especie para bacterias y arqueobacterias a partir del gen 16s y secuenciación capilar. <http://labsergen.langebio.cinvestav.mx/genomics/?services=idbacteria>.
- [Lemos, 2022] Lemos, M. (2022). Lipasa: Qué es y valores normales. <https://www.tuasaude.com/es/lipasa/>.
- [MDS, 2022] MDS (2022). Difference between ux and ui. <https://www.mastersindatascience.org/learning/difference-between-ui-and-ux/>.
- [Nebot, 2019] Nebot, M. (2019). La heurística en scrum: Deloitte españa. <https://www2.deloitte.com/es/es/pages/technology/articles/metodo-heuristica-scrum.html>.
- [Nielsen, 2005] Nielsen, J. (2005). Ten usability heuristics.

- [Olawanle, 2022] Olawanle, J. (2022). What is a framework? software frameworks definition. <https://www.freecodecamp.org/news/what-is-a-framework-software-frameworks-definition>.
- [Oracle, 2022] Oracle (2022). What is a database? <https://www.oracle.com/database/what-is-database>.
- [Pérez de Castro, 2011] Pérez de Castro, A. M. (2011). Reacción en cadena de la polimerasa (polymerase chain reaction, pcr).
- [React, 2023] React (2023). React – una biblioteca de javascript para construir interfaces de usuario. <https://es.reactjs.org/>.
- [Reimer et al., 2022] Reimer, L. C., Sardà Carbasse, J., Koblitz, J., Ebeling, C., Podstawka, A., y Overmann, J. (2022). Bac dive in 2022: the knowledge base for standardized bacterial and archaeal data. *Nucleic Acids Research*, 50(D1):D741–D746.
- [Roche, 2020] Roche, J. (2020). ¿qué es el desarrollo en espiral?: Deloitte España. <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-el-desarrollo-en-espiral.html>.
- [Rojas, 2016] Rojas, C. (2016). Estudio de actinobacterias marinas provenientes de isla de pascua como agentes productores de compuestos bioactivos y evaluación de su potencial en biodegradación de micro-plásticos.
- [Sauro, 2011] Sauro, J. (2011). Measuring usability with the system usability scale (sus). <https://measuringu.com/sus/>.
- [Shi et al., 2021] Shi, W., Sun, Q., Fan, G., Hideaki, S., Moriya, O., Itoh, T., Zhou, Y., Cai, M., Kim, S.-G., y Lee, J.-S. (2021). gctype: a high-quality type strain genome database for microbial phylogenetic and functional research. *Nucleic Acids Research*, 49(D1):D694–D705.
- [Silkalns, 2023] Silkalns, A. (2023). 12 best open-source css frameworks 2023. <https://colorlib.com/wp/free-css3-frameworks/>.
- [Singh y Singh, 2009] Singh, J. y Singh, H. (2009). Kaizen philosophy: a review of literature. *IUP journal of operations management*, 8(2):51.
- [State of JavaScript, 2022] State of JavaScript (2022). The state of js 2021. <https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/>.
- [Svelte, 2023] Svelte (2023). Svelte. <https://svelte.dev/>.
- [TypeScript, 2023] TypeScript (2023). Javascript with syntax for types. <https://www.typescriptlang.org/>.

- [Undabarrena *et al.*, 2021] Undabarrena, A., Valencia, R., Cumsille, A., Zamora-Leiva, L., Castro-Nallar, E., Barona-Gomez, F., y Cámara, B. (2021). *Rhodococcus* comparative genomics reveals a phylogenomic-dependent non-ribosomal peptide synthetase distribution: insights into biosynthetic gene cluster connection to an orphan metabolite. *Microbial genomics*, 7(7).
- [Vargas, 2022] Vargas, N. (2022). Aislamiento de actinobacterias marinas provenientes de la costa de la región de aysén y evaluación de su potencial actividad antimicrobiana.
- [Vergara, 2016] Vergara, G. (2016). Actinobacterias marinas de la isla de pascua: Efecto inhibitorio de extractos bioactivos y su rol potencial en biotecnología.
- [Viéitez, 2013] Viéitez, I. (2013). Estudio de proteínas musculares por western blot y técnicas de genética aplicadas a las enfermedades neuromusculares: Pcr, rflp y secuenciación.
- [Vue, 2023] Vue (2023). What is vue? <https://vuejs.org/guide/introduction.html#the-progressive-framework>.
- [Wikipedia contributors, 2022a] Wikipedia contributors (2022a). Archaea. <https://es.wikipedia.org/wiki/Archaea>.
- [Wikipedia contributors, 2022b] Wikipedia contributors (2022b). Arn ribosomal 16s. https://es.wikipedia.org/wiki/ARN_ribosomal_16S.