

UNIVERSIDAD DE COSTA RICA
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE0499 – Proyecto Eléctrico

**Desarrollo de un microscopio virtual web para acceder a
imágenes de patología digital de hiper ultra alta
resolución**

por

James Sibaja Granados

Ciudad Universitaria Rodrigo Facio

Febrero de 2024

Desarrollo de un microscopio virtual web para acceder a imágenes de patología digital de hiper ultra alta resolución

por

James Sibaja Granados

B77342

IE0499 – Proyecto Eléctrico

Aprobado por

Dr. rer. nat. Francisco Siles Canales

Profesor guía

Dr. Rodrigo Mora Rodríguez

Profesor lector

M.Sc. Juan Carlos Coto Ulate

Profesor lector

Febrero de 2024

Resumen

Desarrollo de un microscopio virtual web para acceder a imágenes de patología digital de hiper ultra alta resolución

por

James Sibaja Granados

Universidad de Costa Rica

Escuela de Ingeniería Eléctrica

Profesor guía: Dr. rer. nat. Francisco Siles Canales

Febrero de 2024

La microscopía virtual es una herramienta poderosa para el análisis de imágenes de alta resolución, ofreciendo una versatilidad en la exploración remota de preparados microscópicos. Este proyecto tiene como objetivo desarrollar una plataforma web que implemente funciones de microscopía, a partir de la tecnología usada en servicios de mapas web, proporcionando acceso a imágenes de ultra alta definición y herramientas de análisis.

La implementación de este proyecto implica un diseño preliminar basado en el análisis de casos de uso y la creación de wireframes. En la implementación se emplea una infraestructura de software robusta, utilizando tecnologías como Django, Celery, Daphne, Gunicorn y PostgreSQL. Se ha priorizado la usabilidad y la escalabilidad del sistema, utilizando herramientas modernas como Docker, Makefile y Git para facilitar el despliegue y la continuidad del proyecto.

El resultado es una plataforma funcional de software libre, diseñada para satisfacer las necesidades de investigación y educación. La plataforma ofrece un conjunto completo de herramientas para la microscopía virtual, facilitando la exploración, análisis y colaboración en proyectos de investigación.

Palabras claves: *microscopia virtual, plataforma web, Django, Leaflet, software libre, alta resolución.*

Acerca de IE0499 – Proyecto Eléctrico

El Proyecto Eléctrico es un curso semestral bajo la modalidad de trabajo individual supervisado, con el propósito de aplicar estrategias de diseño y análisis a un problema de temática abierta de la ingeniería eléctrica. Es un requisito de graduación para el grado de bachiller en Ingeniería Eléctrica de la Universidad de Costa Rica.

Abstract

Desarrollo de un microscopio virtual web para acceder a imágenes de patología digital de hiper ultra alta resolución

Original in Spanish. Translated as: “Title of the Project in English”

by

James Sibaja Granados

University of Costa Rica

Department of Electrical Engineering

Tutor: Dr. rer. nat. Francisco Siles Canales

February of 2024

Virtual microscopy is a powerful tool for high-resolution image analysis, offering versatility in the remote exploration of microscopic specimens. This project aims to develop a web platform that implements microscopy features, based on technology used in web mapping services, providing access to ultra-high-definition images and analysis tools.

The implementation of this project involves a preliminary design based on the analysis of use cases and the creation of wireframes. Robust software infrastructure is employed, using technologies such as Django, Celery, Daphne, Gunicorn, and PostgreSQL. Usability and scalability of the system have been prioritized, utilizing modern tools like Docker, Makefile, and Git to facilitate deployment and project continuity.

The result is a functional open-source software platform designed to meet the needs of research and education. The platform offers a comprehensive set of tools for virtual microscopy, enabling exploration, analysis, and collaboration in research projects.

Keywords: *virtual microscopy, web platform, Django, Leaflet, open-source, high resolution.*

About IE0499 – Proyecto Eléctrico (“Electrical Project”)

The “Electrical Project” is a course of supervised individual work of one semester, with the purpose of applying design and analysis strategies to a problem in an open topic in electrical engineering. It is a requisite of graduation for the Bachelor of Science in Electrical Engineering, granted by the University of Costa Rica.

Dedicado a mi familia y amigos.

Agradecimientos

Quiero expresar mi sincero agradecimiento a todas las personas que contribuyeron de alguna manera a la realización de este trabajo. En primer lugar, quiero agradecer a mi director de proyecto, el Dr. rer. nat. Francisco Siles Canales, por su guía experta, su paciencia y su constante apoyo durante todo el proceso de investigación. Sus conocimientos y orientación fueron fundamentales para el éxito de este proyecto.

Además, quiero expresar mi gratitud a mi familia por su amor incondicional, su apoyo constante y su comprensión durante los momentos difíciles. Su aliento y motivación fueron un pilar fundamental para mí a lo largo de esta travesía académica.

Índice general

Índice general	xii
Índice de figuras	xiii
Índice de tablas	xiv
Nomenclatura	xvii
1 Introducción	1
1.1. Justificación	1
1.2. Alcances	2
1.3. Objetivos	2
1.3.1. General	2
1.3.2. Específicos	2
1.4. Metodología	2
1.4.1. Investigación bibliográfica	2
1.4.2. Montar una infraestructura de software	3
1.4.3. Implementar el sitio web	3
1.4.4. Incorporar herramientas básicas relacionadas con la microscopía virtual	3
1.5. Cronograma	4
2 Teoría del trabajo escrito	5
2.1. Microscopía virtual	5
2.1.1. Microscopio virtual	5
2.1.2. Adquisición de la muestra	5
2.1.3. Proceso de ensamblaje de las placas virtuales	5
2.1.4. Visualización y navegación	6
2.1.5. Análisis de microscopios virtuales disponibles en internet	6
2.1.6. OpenSlide	9
2.2. Herramientas de diseño web	10
2.2.1. casos de uso	10
2.2.2. Wireframing	10

2.3.	Herramientas de desarrollo web	11
2.3.1.	Python y Django	11
2.3.2.	MVC y MVT	11
2.3.3.	JavaScript	11
2.3.4.	Nginx	12
2.3.5.	Docker	12
2.3.6.	Gunicorn	12
2.3.7.	Celery	12
2.4.	Sistemas de información geográfica	13
2.4.1.	Definición y aplicaciones	13
2.4.2.	Latitud y longitud	13
2.4.3.	Mercator	14
2.4.4.	Leaflet	14
3	Diseño y Especificaciones	17
3.1.	Casos de uso	17
3.2.	Wireframes	26
3.2.1.	Página de inicio	26
3.2.2.	Microscopio virtual público	28
3.2.3.	Catálogo	29
3.2.4.	Gestor de nuevas placas virtuales	29
3.2.5.	Tabla de Procesamiento de Archivos	30
3.2.6.	Inicio de sesión	31
3.2.7.	Registro de usuarios	32
3.2.8.	Gestión de Proyectos	33
3.2.9.	Detalles del Proyecto	34
3.2.10.	Visualizador de Placas Virtuales y sus Instancias	35
3.2.11.	Editor de Instancias de Láminas Virtuales	36
3.2.12.	Panel de herramientas de proyecto	37
3.2.13.	Descripción del panel de apoyo a proyectos	38
3.2.14.	Base de datos	39
4	Implementación y Resultados	43
4.1.	Implementación con el framework Django	43
4.1.1.	Gestión del catálogo y visualización de láminas virtuales	43
4.1.2.	Administración del Sitio web y perfiles de usuario	52
4.1.3.	Gestión de proyectos	56
4.1.4.	Configuración y ajustes del entorno de la plataforma	68
4.2.	Encapsulado y automatización de la plataforma	69
4.2.1.	Makefile	70
4.2.2.	Docker-compose	71
4.2.3.	Dockerfile	72

5 Conclusiones y recomendaciones	75
5.1. Conclusiones	75
5.2. Recomendaciones	76
A README	79
Bibliografía	81

Índice de figuras

1.1. Propuesta de cronograma a seguir en el proyecto	4
2.1. Ejemplo de microscopio virtual. Tomada de [22]	6
2.2. Ejemplo de mapa con Leaflet. Tomada de [15]	15
3.1. Diagrama de casos de uso. Elaboración propia	18
3.2. Wireframe: Pagina de inicio. Elaboración propia	27
3.3. Wireframe: Microscopio público. Elaboración propia	28
3.4. Wireframe: Catalogo público. Elaboración propia	29
3.5. Wireframe: Gestión archivos subidos. Elaboración propia	30
3.6. Wireframe: Gestión de procesamiento. Elaboración propia	31
3.7. Wireframe: Inicio sesión. Elaboración propia	32
3.8. Wireframe: Crear cuenta. Elaboración propia	33
3.9. Wireframe: Interfaz principal. Elaboración propia	34
3.10. Wireframe: Detalles de proyecto. Elaboración propia	35
3.11. Wireframe: Visualizador de placas. Elaboración propia	36
3.12. Wireframe: Edición de instancia. Elaboración propia	37
3.13. Wireframe: Panel de herramientas. Elaboración propia	38
3.14. Wireframe: Panel de apoyo. Elaboración propia	39
3.15. Diagrama entidad relación de la base de datos. Elaboración propia	40
4.1. Resultado del catálogo público. Elaboración propia	46
4.2. Resultado del microscopio público. Elaboración propia	47
4.3. Esquema de líneas radiales de la Tierra. Elaboración propia	49
4.4. Resultado de menú de nuevas placas. Elaboración propia	51
4.5. Resultado de menú de gestión de procesamiento de placas. Elaboración propia	52
4.6. Resultado de la página de inicio. Elaboración propia	54

4.7. Resultado de formulario para inicio de sesión. Elaboración propia	55
4.8. Resultado de formulario para crear un nuevo usuario. Elaboración propia	56
4.9. Resultado de la implementación de la interfaz administradora de perfil y proyectos. Elaboración propia	60
4.10. Resultado del área de trabajo de proyecto para la invitación de nuevos colaboradores. Elaboración propia	61
4.11. Resultado de implementación de catalogó para proyectos. Elaboración propia	62
4.12. Resultado área de trabajo para le manejo de instancias de placas virtuales. Elaboración propia	63
4.13. Resultado de la implementación de un chat para el proyecto. Elaboración propia	64
4.14. Resultado del despliegue de las invitaciones en el área de gestión de proyectos. Elaboración propia	65
4.15. Resultado del la vista previa de un proyecto. Elaboración propia	66
4.16. Resultado del área de edición de una lámina virtual de proyecto. Elaboración propia	67

Índice de tablas

2.1. Funciones y Contexto de Microscopios Virtuales	8
2.2. Funciones y Contexto de Microscopios Virtuales	9
3.1. Descripción de Actores	17
3.2. UC: Usar Catálogo	19
3.3. UC: Usar Microscopio Virtual	19
3.4. UC: Usar Interfaz de Proyecto	20
3.5. UC: Invitar Nuevos Miembros al Proyecto	21
3.6. UC: Chatear con Miembros del Proyecto	21
3.7. UC: Usar Catálogo Incrustado	22
3.8. UC: Crear Lámina Virtual	22
3.9. UC: Usar Microscopio Virtual para Proyecto	23
3.10. UC: Crear y Guardar Anotaciones en una Lámina Virtual	23
3.11. UC: Usar Interfaz de Perfil de Usuario	24
3.12. UC: Gestionar Invitación a Proyecto	24
3.13. UC: Visualizar Perfil de Proyecto	25
3.14. UC: Crear Proyecto	25
3.15. UC: Crear Elemento del Catálogo	25

4.1.	Extensiones utilizadas de Leaflet	48
4.2.	Descripción de los objetivos del Makefile.	71
4.3.	Descripción de los servicios del archivo docker-compose.	72
4.4.	Descripción de los archivos Dockerfile.	73

Nomenclatura

C	Circunferencia de la tierra
ASGI	Interfaz de puerta de enlace de servidor asíncrono (del inglés <i>Asynchronous Server Gateway Interface</i>)
CSS	Hojas de estilo en cascada (del inglés <i>Cascading Style Sheets</i>)
HTML	Lenguaje de marcado de hipertexto (del inglés <i>HyperText Markup Language</i>)
HTTP	Protocolo de Transferencia de Hipertexto (del inglés <i>Hypertext Transfer Protocol</i>)
MV	Microscopio Virtual)
MVC	Modelo-Vista-Controlador (del inglés <i>Model-View-Controller</i>)
MVT	Modelo-Vista-Plantilla (del inglés <i>Model-View-Template</i>)
UC	Caso de uso (del inglés <i>Use Case</i>)
URL	Localizador de Recursos Uniforme (del inglés <i>Uniform Resource Locator</i>)
WSGI	Interfaz de puerta de enlace del servidor web (del inglés <i>Web Server Gateway Interface</i>)

Capítulo 1

Introducción

La microscopia virtual permite generar placas histológicas virtuales, las cuales consisten en imágenes de muy alta resolución, que al ser complementadas con el software adecuado, habilita la opción de acceder a dicho material por medio de las computadoras [5]. La virtualidad en este ámbito ofrece ventajas como; el acceso a la información remotamente, versatilidad en la manipulación del material y la mayor disponibilidad del uso de técnicas de procesamiento digital de imágenes. Las ventajas proporcionadas por esta herramienta tienen aplicación en campos como la telepatología y la educación.

Costa Rica cuenta con diversos institutos dedicados a la investigación y educación, en el área de la microscopia, de los cuales podemos mencionar; el Laboratorio Institucional de Microscopía (LIM), del Tecnológico de Costa Rica (TEC) y el Centro de Investigaciones en Estructuras Microscópicas (CIEMic), de la Universidad de Costa Rica (UCR). De esta forma se infiere que existe suficiente material en las universidades públicas de Costa Rica, para generar numerosas placas virtuales y con estas desarrollar proyectos de microscopia virtual. Un ejemplo de esto se puede ver en la UCR, con el Laboratorio de investigación en reconocimiento de patrones y sistemas inteligentes (PRIS-Lab), en donde se realizan investigaciones en diferentes áreas, entre ellas la biocomputación, contando con trabajos relacionados con la segmentación y rastreo de células cancerígenas a partir de imágenes de microscopia [25] [24], por mencionar algunos ejemplos.

Es por eso que un proyecto como la elaboración de un microscopio virtual web, no solo es viable en instituciones como la Universidad de Costa Rica, sino además proporciona numerosas ventajas y nuevas oportunidades, para la investigación y la educación.

1.1. Justificación

Debido a la pandemia de COVID19 muchos experimentos de investigación y actividades docentes, entre otras han debido suspenderse. Por ello se propone la creación de una microscopio virtual utilizando tecnología similar a la utilizada por google maps, entre otras para proveer a los investigadores, docentes y estudiantes de herramientas para seguir analizando imágenes de microscopía de fondo claro o fluorescencia de manera remota.

1.2. Alcances

En este proyecto se desarrollará una plataforma web que permita el uso de un microscopio virtual, capaz de visualizar imágenes de muy alta definición, además de manipularlas con mecánicas propias de un microscopio, tales como; ampliar, reducir, moverse de forma vertical y de forma horizontal. Para lograr esto se implementará la tecnología necesaria en cuanto a uso de memoria, estructuras de datos y algoritmos para acceder y manipular dichas imágenes. La generación de imágenes para el microscopio virtual, no será parte de lo desarrollado por el proyecto, además no se implementarán funciones avanzadas relacionadas al campo de aplicación de la plataforma, pero si se incluirán herramientas sencillas, para facilitar su uso y mostrar el potencial del proyecto.

1.3. Objetivos

1.3.1. General

Desarrollar un microscopio virtual web que permita la telepatología virtual, empleando imágenes de ultra alta definición.

1.3.2. Específicos

1. Realizar una investigación bibliográfica sobre la microscopía virtual y el procesamiento digitales de imágenes de ultra alta definición.
2. Montar una infraestructura de software, para el desarrollo web y procesamiento de imágenes, que permita el desarrollo óptimo del proyecto.
3. Implementar un sitio web que permita el manejo de imágenes con resoluciones muy altas, y que cuente con dinámicas de uso similares a las empleadas en sistema de información geográfica como; Google Maps, OpenStreetMap, entre otros.
4. Incorporar herramientas básicas relacionadas con la microscopía virtual, para ampliar el potencial del sitio web.
5. Desarrollar un entorno de encapsulamiento y automatización para la plataforma, con el propósito de agilizar y simplificar su despliegue, así como facilitar su mantenimiento y escalabilidad.

1.4. Metodología

Para desarrollar los objetivos planteados de forma adecuada se procederá con la siguiente metodología:

1.4.1. Investigación bibliográfica

1. Seleccionar palabras claves relacionadas con la microscopía virtual, para emplearlas en repositorios como IEEE, Springer, entre otros.

2. Seleccionar según su título y abstracto, los artículos más pertinentes a la investigación en proceso.
3. Buscar en los artículos recientes de la lista seleccionada la información más destacada.
4. Elaborar un informe con la información obtenida.

1.4.2. Montar una infraestructura de software

1. Buscar en internet que páginas de microscopios virtuales se encuentran disponibles, para tomarlos de ejemplo.
2. Investigar sobre herramientas de software, plataformas y bibliotecas, que sean apropiadas para montar la infraestructura necesaria del proyecto, sin incurrir en la necesidad de empezar desde cero.
3. Discernir entre las herramientas investigadas, y escoger las más apropiadas para la elaboración del proyecto.
4. Montar una infraestructura de software

1.4.3. Implementar el sitio web

1. Estudiar los casos de uso, y las características ofrecidas por las herramientas seleccionadas.
2. Diseñar el concepto de la página web, a partir de la información obtenida.
3. Diseñar las bases datos y algoritmos según el concepto diseñado.
4. Realizar pruebas de implementación de las funciones principales de la página web.
5. Montar las bases de datos y demás estructuras funcionales.
6. Terminar de implementar funciones principales de la página.

1.4.4. Incorporar herramientas básicas relacionadas con la microscopía virtual

1. Implementar funciones básicas adicionales.
2. Realizar los ajustes necesarios para poner en funcionamiento el software.

1.5. Cronograma

El cronograma que se seguirá para el desarrollo del proyecto se muestra en la figura 1.1

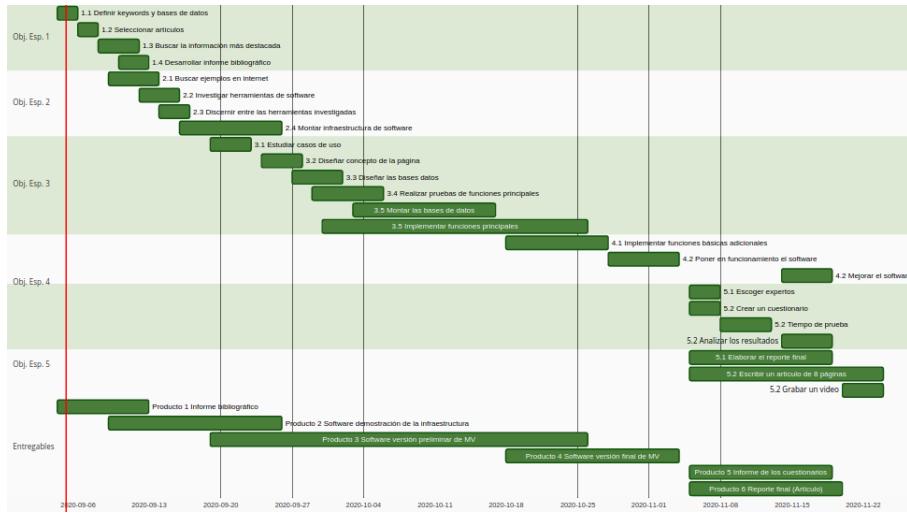


Figura 1.1: Propuesta de cronograma a seguir en el proyecto

/

Capítulo 2

Teoría del trabajo escrito

2.1. Microscopía virtual

2.1.1. Microscopio virtual

Para la implementación de un microscopio virtual es necesario considerar las implicaciones técnicas que conlleva migrar la información de un medio fisco a uno virtual, bajo este contexto de la microscopía. Un microscopio físico transforma un objeto en una imagen y amplifica sus detalles, mientras que en un microscopía virtual se incluyen nuevos procesos; como la construcción de placas virtuales, el almacenamiento de estas y el diseño de herramientas versátiles que permitan visualizar interactivamente dichas placas. [19]

2.1.2. Adquisición de la muestra

En esta etapa se emplea el microscopio físico, pero el uso que se le da es diferente al convencional, ya que la finalidad es preparar la información necesaria para montar un escenario donde posteriormente se estudiará la muestra. La información de la lamina se adquiere en forma de imágenes de gran aumento, que se toman de forma sucesiva y adyacentes entre si, hasta barrer toda la muestra. [19] Este proceso define muchas de las características finales de la placa virtual por lo que es un proceso importante y detallado pero sistemático. Para realizar esta tarea se emplea un microscopio robotizado el cual consiste en un sistema automatizado. [19]

2.1.3. Proceso de ensamblaje de las placas virtuales

De la etapa anterior se puede notar que la información de una placa consta de una serie de imágenes, pero para formar la placa virtual dicho conjunto de imágenes deben asociarse en una sola, de forma aparente.

El proceso de la construcción de imágenes de histopatología puede ser visto como armar un rompecabezas, en el cual se debe procurar el mínimo error en el ensamblaje, para este proceso se deben tomar imágenes adyacentes cuyos bordes se superponen entre el 5 % y el 10 %, del campo microscópico capturado, hasta hacer el barrido en toda muestra. [19]

Cuando termina la etapa de captura, se continúa con el montaje, donde las imágenes próximas se fusionan entre ellas, empleando técnicas de registro de imágenes, donde se superponen imágenes que fueron tomadas en diferentes momentos, por diferentes sensores o desde diferentes puntos de vista (análisis multivista), logrando obtener una vista 2D más grande o una representación 3D de la escena escaneada. [19]

2.1.4. Visualización y navegación

La microscopía virtual se emplea para brindar a los usuarios información sobre las muestras, mediante la visualización interactiva de las placas virtuales, permitiendo la consulta, interpretación y diagnóstico. Los sistemas virtuales de microscopía requieren del envío eficiente de la información, para cumplir de manera adecuada este requisito, los desarrolladores se basan principalmente en una arquitectura cliente/servidor, donde un servidor conectado a la red almacena la información de las placas virtuales, y el usuario remoto accede a la información requerida mediante una interfaz gráfica. [19]

La microscopía tiene una amplia gama de aplicaciones, y dependiendo del usuario las características requeridas pueden ser diferentes, por dicha razón el éxito de un microscopio virtual depende de la capacidad de esta plataforma para satisfacer y adaptarse a dichas necesidades. Se puede lograr una mayor versatilidad si se tiene una interfaz gráfica que facilite al usuario la visualización y exploración de la información contenida en la placa virtual. [19]

Para lograr esta etapa se emplearán algunas herramientas necesarias que permitan desarrollar apropiadamente la plataforma, aprovechando los recursos de software disponibles. En 2.1 se muestra el ejemplo de una aplicación de microscopio virtual.

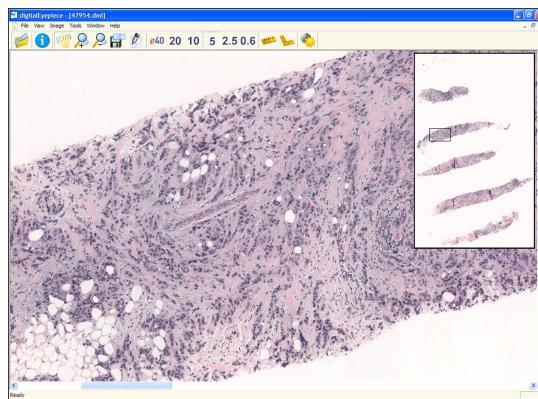


Figura 2.1: Ejemplo de microscopio virtual. Tomada de [22]

2.1.5. Análisis de microscopios virtuales disponibles en internet

Se exploraron diversas opciones de microscopios virtuales disponibles en internet. La información recopilada se sintetizó mediante la comparación de las distintas funciones de los microscopios identificados, como se detalla en la tabla 2.2. La lista de funciones utilizadas en esta comparación, junto con su contexto, se presenta en la tabla 2.1. Estas funciones seleccionadas representan una lista de las características

2.1. Microscopía virtual

7

identificadas de manera general en los microscopios analizados, proporcionando así un resumen conciso de las diferencias observadas entre ellos.

Letra	Función	Contexto
A	Zoom	Permite ampliar y reducir las imágenes para apreciar las diferentes escalas de la placa.
B	Desplazamiento	Permite navegar con libertad por las extensas imágenes mostradas.
C	Acceso sin suscripción	Permite acceder al microscopio virtual sin necesidad de registrarse en la plataforma, esta es una forma de no poner trabas a los usuarios.
D	Inglés	La plataforma está disponible en el idioma inglés, el cual es deseado por ser el idioma más difundido en el ámbito académico.
E	Español	La plataforma está disponible en el idioma español.
F	Pantalla completa	La opción de pantalla completa está disponible de forma intuitiva.
G	Captura de pantalla	La opción de tomar una captura de pantalla y descargarla desde la plataforma está disponible de forma intuitiva.
H	Minimap	Es visible un cuadro con la imagen actual de la placa en una escala menor, que funciona como guía de la ubicación actual en la imagen ampliada, lo cual es de mucha ayuda en imágenes muy grandes.
I	Vuelta al origen	La opción de volver a la vista inicial, donde la totalidad de la placa es visible, está disponible de forma intuitiva.
J	Catálogo amplio	Están disponibles más de 30 placas virtuales al público en general en un catálogo intuitivo.
K	Vista previa de placas	El catálogo muestra una imagen previa de cada placa, no solo el nombre o texto, facilitando la búsqueda
L	Opciones de búsqueda	Es posible navegar por el catálogo filtrando resultados mediante una barra de búsqueda.

Tabla 2.1 – Continuación de la página anterior

Letra	Función	Contexto
M	Medición de longitud	Entre las herramientas disponibles se encuentra la opción de realizar mediciones de longitudes en la imagen, esta es una herramienta deseable para realizar microscopía virtual.
N	Medición de área	Es posible demarcar una sección en la imagen y obtener el área real de la zona.
O	Agregar notas	Es posible agregar notas de texto en la placa virtual, para agregar información a la placa.
P	Agregar marcadores	Es posible agregar diferentes tipos de marcadores en las placas virtuales, los cuales funcionan como indicadores e información visual extra

Tabla 2.1: Funciones y Contexto de Microscopios Virtuales

Microscopio	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Observaciones
University of Michigan [27]	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	Se implementa a través de la tecnología Zoomify.
Virtuelle Mikroskopie Homberg [10]	✓	✓	✓	✗	✗	✓		✗	✓	✓	✓	✓	✓	✓	✓	✓	Tiene notas por defecto, pero posiblemente se requiere ser administrador.
Histology Guide [7]	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	Tiene notas por defecto, pero posiblemente se requiere ser administrador. La opción de pantalla completa es parcial.
Virtual Microscope [29]	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	Permite compartir la imagen en una ubicación con un link.
Microscopio Virtual Udelar [9]	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	Permite compartir la imagen en una ubicación con un link.

Tabla 2.2 – Continuación de la página anterior

Microscopio	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Observaciones
Atlas de histología vegetal y animal [17]	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓	✗	Tiene notas por defecto, pero posiblemente requiere ser administrador.
Microscopio Virtual UNAM [1]	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	
Microscopio Virtual Universidad Nacional de Río Cuarto [11]	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	Se implementa a través de la tecnología EasyStitch.
Microscopio Virtual Universidad de Granada [8]	-	-	✗	-	-	-	-	-	-	-	-	-	-	-	-	No se puede ingresar sin estar registrado.	
Virtual Microscope BioNetwork [3]	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	Es un microscopio virtual en 3D centrado más en la enseñanza del uso de un microscopio físico que en aplicaciones de telepatología

Tabla 2.2: Funciones y Contexto de Microscopios Virtuales

2.1.6. OpenSlide

En los objetivos se plantea trabajar con tecnología dedicada a la visualización de mapas, sin embargo no está diseñada para trabajar directamente con placas virtuales, para facilitar el uso de la plataforma es necesario explorar herramientas, que faciliten la compatibilidad con diferentes formatos de placas, además de habilitar funciones específicas de la patología digital que se podrían usar para mejorar el proyecto. La herramienta escogida es OpenSlide.

OpenSlide es una biblioteca de C que permite leer de forma sencilla placas virtuales, también es posible asociarlo con python y java. Esta biblioteca es de software libre y además es compatible con formatos como Aperio (.svs, .tif), DICOM (.dcm), Hamamatsu (.vms, .vmu, .ndpi), Leica (.scn), MIRAX (.mrxs), Philips (.tiff), Sakura (.svslide), Trestle (.tif), Ventana (.bif, .tif), y Formato TIFF [23]

Esta biblioteca se escogió como una de las herramientas usadas en el proyecto, porque ofrece una API para python por lo que se puede usar junto Django de forma sencilla, además ofrece compatibilidad

con un gran numero de formatos, .

2.2. Herramientas de diseño web

2.2.1. casos de uso

Una descripción robusta de requisitos facilita el desarrollo de plataformas eficaces. La implementación de un proyecto informático requiere la inversión de tiempo y recursos. Es por eso, que el propósito del proyecto debe justificar el esfuerzo del desarrollo. Por consiguiente, los criterios de diseño tienen que definirse tomando en cuenta la meta. El problema que encontramos con este planteamiento es que la transición de un objetivo a los criterios mencionados no es explícita. Por ello, es necesario derivar del objetivo principal del proyecto una lista de funciones concisas que la plataforma debe ser capaz de realizar. Estos requisitos, si son detallados de forma simple, servirán como una guía clara para el diseño, y si además son robustos garantizará que el proyecto satisfaga el objetivo inicial. Por lo tanto, es importante identificar herramientas que permitan crear dicha descripción de requisitos.

El enfoque en los casos de uso presenta ventajas frente al método convencional de captura de requisitos estrictamente declarativos. Una lista de requisitos que solo enuncia una descripción tras otra carece de las herramientas para describir pasos y secuencias de eventos. En consecuencia, esta forma de plasmar los requisitos no logran capturar la dinámica del comportamiento del sistema. Los casos de uso son una herramienta sencilla pero pradera que centra su función en describir el comportamiento del sistema [4].

Los casos de uso son narración de como los actores interactúan con el sistema para obtener algo que consideren de valor. Un actor se define como una persona o cosa fuera del sistema, pero que interactúa con el mismo. Cada actor posee un nombre y una breve descripción. Los casos de uso, a pesar de no ser funciones o características, son una técnica que permite expresar requisitos funcionales. Este enfoque es especialmente útil para describir el comportamiento de un sistema ante las acciones del usuario [4],

2.2.2. Wireframing

Para facilitar el diseño de la interfaz de usuario (UI) de la plataforma, es necesario emplear técnicas que permitan separar el concepto del diseño de la implementación, para poder centrarse en este último. Se propone el uso de wireframes para diseñar los diferentes elementos de la plataforma necesarios para satisfacer los UC.

Un wireframe es un esquema que muestra bloques de contenido y su posición en la página. Estos bocetos representan un soporte importante para el diseño de la plataforma, ya que proporcionan flexibilidad para el intercambio de ideas y pueden dar una idea del enfoque de la plataforma. Un wireframe establece únicamente la estructura del diseño y la posición de los elementos, sin incluir colores, imágenes reales ni elementos HTML [2].

2.3. Herramientas de desarrollo web

2.3.1. Python y Django

Django es un framework de desarrollo web de alto nivel en Python que fomenta el desarrollo rápido y un diseño limpio y pragmático. Fue diseñado con el propósito de resolver gran parte de las complicaciones del desarrollo web, lo que permite centrar la atención del desarrollador en su aplicación, y no en tareas comunes del desarrollo web, además es gratuito y de código abierto. [12]

Entre las ventajas de emplear este framework, podemos encontrar que Django fue diseñado para ayudar a los desarrolladores a completar aplicaciones lo más rápido posible. Django incluye muchos complementos que se pueden utilizar para manejar tareas comunes de desarrollo web, como la autenticación de usuarios, la administración de contenido, los mapas del sitio y muchas más tareas. Django se diseñó de forma robusta con respecto a la seguridad y ayuda a los desarrolladores a evitar muchos errores comunes de seguridad, como la inyección SQL, el scripting entre sitios, la falsificación de solicitudes entre sitios y el clickjacking. Su sistema de autenticación de usuarios ofrece una forma segura de gestionar cuentas de usuario y contraseñas. Además, es sumamente escalable e increíblemente versátil. [12]

Para trabajar con llamadas asíncronas en Django, se puede utilizar Daphne. Daphne es un servidor de protocolos HTTP, HTTP2 y WebSocket para ASGI y ASGI-HTTP, diseñado para brindar soporte a Django Channels [13].

2.3.2. MVC y MVT

El modelo MVC se utiliza en frameworks como Ruby on Rails, Spring (Java), Laravel (PHP) y ASP.NET (C#). Este patrón separa los datos, la lógica y la presentación de una aplicación, lo que permite un fácil razonamiento para el desarrollador. El modelo gestiona los datos y la lógica central, la vista renderiza datos desde el modelo y el controlador acepta entradas del usuario y realiza la lógica específica de la aplicación ??.

El patrón de Django se llama MVT pero consta de cuatro partes ya que incorpora la configuración de URL. El modelo gestiona los datos y la lógica central, la vista describe qué datos se envían al usuario pero no su presentación, la plantilla representa los datos como HTML, CSS y JavaScript, y la configuración de URL gestiona los componentes de expresión regular para enlazar con una vista ??.

2.3.3. JavaScript

JavaScript es un lenguaje de programación usado con frecuencia en la programación de páginas web, pero también es utilizado en entornos diferentes al navegador. Es un lenguaje de scripts que es dinámico, basado en prototipos y multiparadigma. Este lenguaje se ejecuta del lado del cliente web, es utilizado para programar el comportamiento de una página web cuando ocurre un evento. JavaScript (JS) es un potente lenguaje de scripts usado ampliamente para controlar el comportamiento de las páginas web, además es fácil de aprender [20]

2.3.4. Nginx

En busca de desarrollar una plataforma que sea funcional no solo para un entorno local de desarrollo, sino además para un entorno de producción, se buscaron herramientas que permitan que el proyecto sea escalable y fácil de configurar, por esta razón se decidió trabajar con NGINX.

NGINX es un servidor http y de proxy inverso, gratuito y de código abierto, además de un servidor MAP/POP3. Entre sus ventajas podemos encontrar que, posee un alto rendimiento, es estable, tiene un bajo consumo de recursos, un alto rendimiento y posee una variedad extensa de funciones y su configuración es sencilla [21]. Por la cual permite funcionar como un intermediario muy poderoso entre el usuario y el servidor, lo cual es un gran apoyo para cumplir con los requerimientos del proyecto.

NGINX también es una opción muy potente porque; este servidor esta diseñado para abordar el problema de gestionar a más de diez mil usuarios de forma de forma simultanea, este no depende de hilos con otros servidores, y su arquitectura es más estable porque esta basada en eventos, es asíncrona, por lo que es muy escalable y versátil para muchos tipos de proyectos [21].

Otra razón para el uso de NGINX, es que ampliamente usado en la actualidad por lo que su efectividad es comprobable. Este servidor es usado en sitios muy conocidos como Netflix, Pinteres o GitHub [21].

2.3.5. Docker

El desarrollo de aplicaciones web requiere de un entorno que puede variar considerablemente de un proyecto a otro. Los marcos, arquitecturas, lenguajes e interfaces pueden ser diferentes entre herramientas, lo que aumenta la complejidad del proceso. Docker aborda esta problemática aplicando el concepto de contenedor, proporcionando una serie de herramientas que facilitan y aceleran el flujo de trabajo [14].

Un contenedor es una unidad de software estandarizada que permite a los desarrolladores aislar la aplicación de su entorno. Docker es ampliamente utilizado por desarrolladores de todo el mundo [14].

2.3.6. Gunicorn

Para hacer que el proyecto sea más escalable, fue necesario dotar a la aplicación de una aplicación eficiente que pudiera manejar las solicitudes de los clientes y servir la aplicación web al público. Con este propósito, se decidió trabajar con Gunicorn. Gunicorn es un servidor HTTP WSGI para entornos UNIX, implementado en Python. Es conocido por su confiabilidad y facilita la integración con Django. [6]

2.3.7. Celery

La plataforma requiere de herramientas que le permitan gestionar tareas de diferentes naturalezas. Por ejemplo, en ocasiones el usuario necesita realizar una serie de acciones que pueden requerir mucho tiempo, por lo que no es aceptable que la plataforma se detenga. Con el objetivo de procesar tareas relacionadas al análisis y gestión de imágenes de microscopía de ultra alta resolución, se decidió trabajar con Celery.

Celery es una cola de tareas centrada en el procesamiento en tiempo real y en segundo plano de tareas programables. Para lograrlo, Celery se comunica a través de mensajes, normalmente utilizando

un broker para mediar entre los clientes y los workers. Es un sistema simple, sencillo y confiable, capaz de procesar grandes cantidades de mensajes y de integrarse fácilmente con Django [28].

2.4. Sistemas de información geográfica

2.4.1. Definición y aplicaciones

Se trata de una categorización de un tipo de sistema que opera con información geográfica georeferenciada o basada en la ubicación, haciendo uso de tecnología informática. Estos sistemas se dedican a llevar a cabo tareas como la adquisición, almacenamiento, gestión, análisis y presentación de resultados de datos espaciales. Además, su objetivo principal es abordar cuestiones relacionadas con el medio ambiente y el territorio. [26]

La utilización de sistemas de información geográfica (SIG) basados en herramientas informáticas encuentra su justificación en su capacidad para abordar problemas relacionados con la ubicación y la organización espacial. Podemos identificar seis categorías principales para las cuales el empleo de estos sistemas es apropiado: la búsqueda ágil y eficiente de atributos vinculados a entidades geográficas, localización con restricciones, análisis de tendencias, trazado de rutas, identificación de patrones y tendencias, así como la construcción de modelos descriptivos de la realidad que permitan su simulación. [26]

De las anteriores afirmaciones, surge la evidencia de que la tecnología de Sistemas de Información Geográfica (SIG) provee herramientas esenciales para abordar cuestiones en el ámbito de la ubicación de puntos georeferenciados. Para una comprensión más profunda de la naturaleza de estas herramientas, es necesario explorar en detalle el contexto en el que operan, el cual se refiere a la superficie terrestre. La tierra, como entidad en tres dimensiones, constituye este contexto.

En el ámbito de la modelización de recursos naturales, cobra gran importancia el establecimiento de un marco de referencia. La forma real de la Tierra, conocida como geoide, es intrincada e irregular, lo que dificulta su descripción con total precisión. Para abordar esta complejidad, se han desarrollado dos sistemas de coordenadas principales: la latitud y la longitud, y el Sistema de Coordenadas Universal Transversal de Mercator (UTM). [18]

2.4.2. Latitud y longitud

La latitud y la longitud constituyen un sistema de coordenadas que establece un marco de referencia para la superficie terrestre. Estas coordenadas se derivan a partir de un punto de referencia, que en este caso es un modelo elipsoidal que representa la forma de la Tierra. Este modelo elipsoidal se utiliza como base para facilitar cálculos relacionados con áreas y volúmenes en la superficie terrestre. El Sistema Geodésico Mundial de 1984 (WGS 84) ofrece un elipsoide que se adapta al geoide con un error mínimo en la diferencia de altitud entre ambas representaciones. Este sistema se emplea ampliamente como referencia en el funcionamiento del Sistema de Posicionamiento Global (GPS). Cabe destacar que este elipsoide es una superficie que representa el nivel medio del mar en sus diversas localidades geográficas. [18]

- Las coordenadas elipsoidales son tridimensionales y se componen de tres valores: latitud, longitud y altura elipsoidal. La longitud se define como un ángulo medido en dirección este u oeste desde el meridiano principal. La latitud, por su parte, es un ángulo medido en dirección norte o sur desde el plano ecuatorial. La altura elipsoidal se refiere a la distancia vertical medida en dirección normal respecto al elipsoide. [18]

[18]

2.4.3. Mercator

Las Coordenadas Universales Transversales de Mercator (UTM) constituyen una proyección cartográfica que se basa en un sistema de coordenadas cilíndrico. Esta proyección divide la superficie terrestre en zonas, cada una de las cuales forma un sistema cartesiano con coordenadas Este y Norte (buscar imagen y hacer referencia). Dado que la Tierra presenta una forma aproximadamente elipsoidal, cualquier intento de proyectar su superficie plana en un mapa conlleva cierta distorsión. Esta distorsión se refiere a la relación entre las dimensiones reales de la Tierra y su representación en el cilindro de la proyección. Es importante señalar que esta distorsión varía a lo largo de las distancias y, en la mayoría de los casos, no sigue una relación lineal [18]. En las ecuaciones 2.1 y 2.2 se muestra una variación de la proyección para mapas virtuales conocida como Mercator web, esta es una relación entre la longitud, la latitud y los píxeles en pantalla [16].

$$x = \frac{(lon + 180)}{360} \cdot 2^z \quad (2.1)$$

$$y = \left(1 - \frac{\ln \left(\tan \left(\frac{lat \cdot \pi}{180} \right) + \frac{1}{\cos \left(\frac{lat \cdot \pi}{180} \right)} \right)}{\pi} \right) \cdot 2^{z-1} \quad (2.2)$$

2.4.4. Leaflet

Para la manipulación de las imágenes, es necesario desarrollar un "software" capaz de ampliar, reducir y mover los recursos gráficos mostrados al usuarios, de forma cómoda e intuitiva. Es importante reconocer los recursos ya existentes que agilicen la implementación de los proyectos, en este caso se puede notar un parecido entre el funcionamiento deseado del microscopio y las aplicaciones de GIS (geographic information system). Por esta razón se buscó un 'API' o una biblioteca que proporcionara estos recursos, y que además fuese de código libre, finalmente se escogió Leaflet.

Leaflet es la biblioteca JavaScript de código abierto líder para mapas interactivos compatibles con dispositivos móviles. Con un peso de aproximadamente 39 KB de JS, tiene todas las funciones de mapeo que la mayoría de los desarrolladores necesitan. [15]

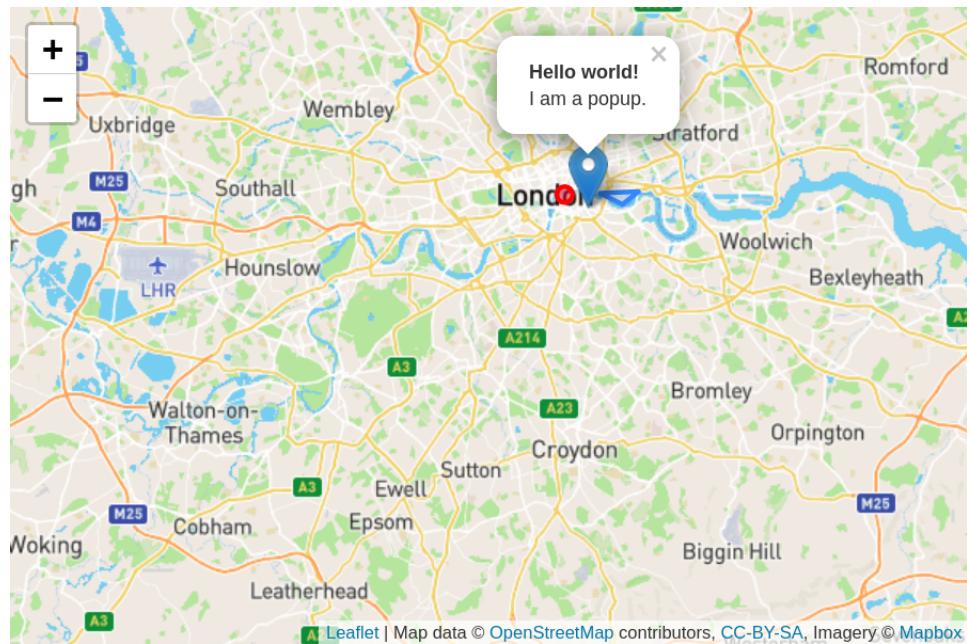


Figura 2.2: Ejemplo de mapa con Leaflet. Tomada de [15]

En la figura 3.15 se puede observar un mapa y la interfaz proporcionada por la biblioteca Leaflet. En dicha imagen se muestra un ejemplo de la implementación de una aplicación empleando dicho recurso. A partir de esta biblioteca se pueden agregar imágenes y manejarlas como los mapas mostrados, proporcionando dinámicas de navegación similares a los de programas GIS comunes, como por ejemplo Google Maps.

Capítulo 3

Diseño y Especificaciones

3.1. Casos de uso

En concordancia con la metodología del proyecto, se llevaron a cabo estudios sobre los casos de uso de la plataforma. El diseño de estos casos se fundamentó en los objetivos iniciales del proyecto, así como en un análisis comparativo de diversas plataformas disponibles para la microscopía virtual. Esta investigación se condensa en las tablas 2.1 y 2.2 del marco teórico, las cuales exponen las características principales ofrecidas por los medios disponibles en este ámbito. Con el propósito de garantizar que el proyecto represente una contribución significativa, se partió de las características descritas en estas tablas. De esta manera, se obtiene una nueva opción con características relevantes, pues han sido implementadas previamente, pero novedosas al ofrecerlas como conjunto en una sola plataforma. Es relevante destacar que la plataforma se desarrolla íntegramente en software libre y está estructurada para facilitar su despliegue, mantenimiento y escalabilidad. Esto permite que el proyecto sea aplicable en universidades, centros de investigación o en proyectos más complejos relacionados con la temática.

El diagrama de casos de uso de la plataforma se presenta en la figura 3.1, mientras que las tablas 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.13, 3.12, 3.14 y 3.15 ofrecen una descripción detallada de cada elemento de dicho diagrama. Finalmente, la tabla 3.1 proporciona descripciones de los actores involucrados en la plataforma.

Tabla 3.1: Descripción de Actores

Actor	Descripción
Usuario	Usuario de la plataforma. Tiene acceso a todas las funciones públicas de esta.
Usuario Registrado	Es un usuario con una cuenta en la plataforma. Tiene acceso a ciertas funcionalidades dependiendo de su rol. También hereda las características de un "usuario".
Administrador	Hereda las características del un "usuario registrado", además posee privilegios especiales dentro del sistema, y es responsable de administrar y configurar la plataforma.

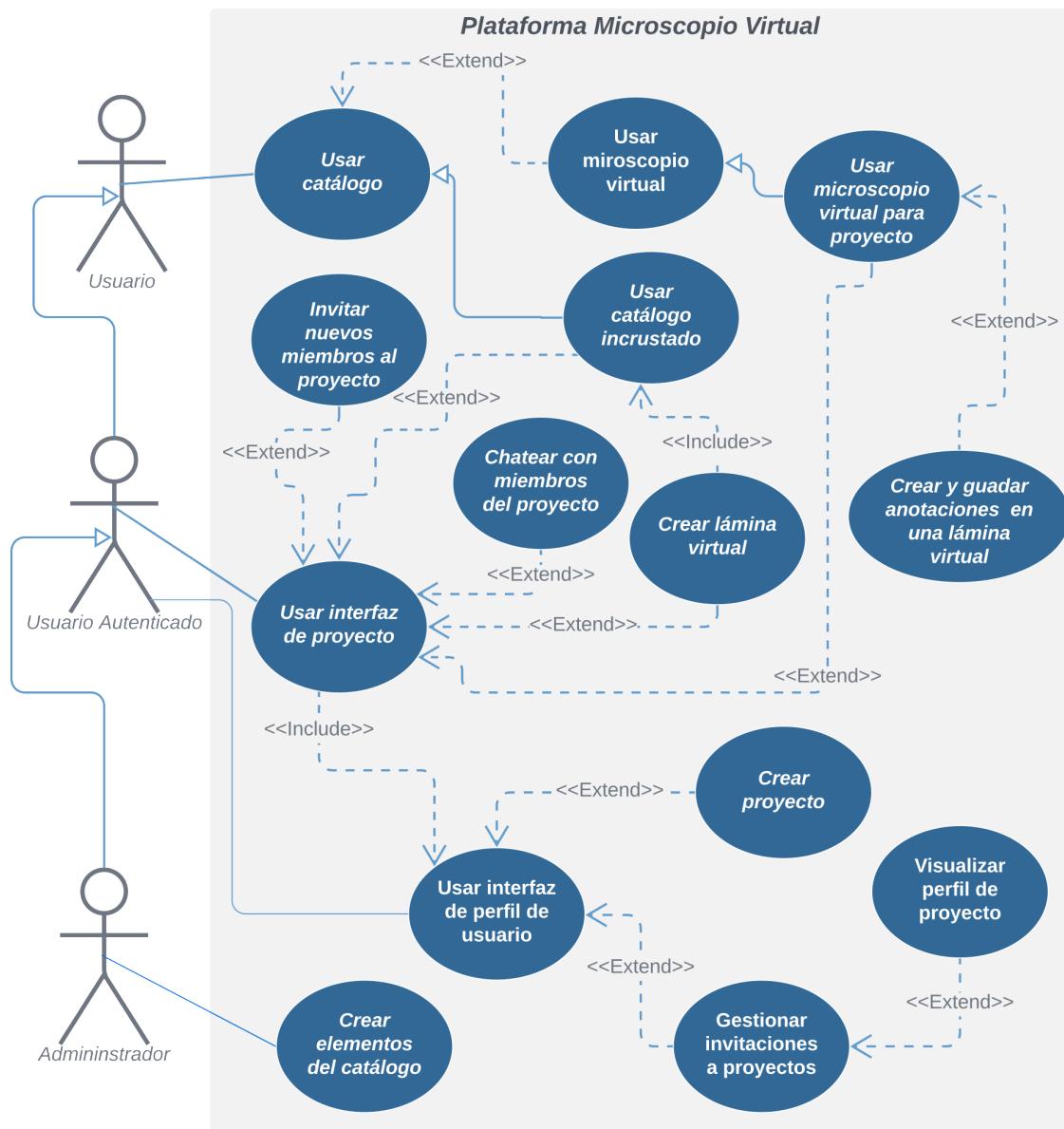


Figura 3.1: Diagrama de casos de uso. Elaboración propia

Tabla 3.2: UC: Usar Catálogo

Campo	Descripción
Caso de Uso	Usar Catálogo
Objetivo	Permitir a los usuarios explorar las diferentes láminas virtuales de la base de datos de la plataforma.
Actor	Usuario
Inclusiones	Ninguna
Precondiciones	Ninguna
Descripción	El usuario navega a través de los elementos del catálogo de placas virtuales, disponiendo de una imagen previa y una descripción por elemento. Puede filtrar resultados para una búsqueda más personalizada.
Extensiones	Usar Microscopio Virtual (Tabla 3.3): Si el usuario selecciona alguna placa del catálogo.

Tabla 3.3: UC: Usar Microscopio Virtual

Campo	Descripción
Caso de Uso	Usar Microscopio Virtual
Objetivo	Permitir a los usuarios explorar de forma detallada e interactuar con una lámina de microscopía virtual.
Actor	Usuario
Inclusiones	Ninguna
Precondiciones	Ninguna
Descripción	El usuario puede navegar a través de imágenes de ultra alta resolución, con tecnologías similares a las de un mapa web. Además, el usuario puede usar herramientas para agregar anotaciones y destacar regiones. Puede descargar capturas de pantalla, pero no puede guardar.
Extensiones	Ninguna

Tabla 3.4: UC: Usar Interfaz de Proyecto

Campo	Descripción
Caso de Uso	Usar Interfaz de Proyecto
Objetivo	Permitir a los usuarios trabajar de forma colaborativa en la interacción y el análisis de láminas virtuales.
Actor	Usuario registrado
Inclusiones	Usar Interfaz de Perfil de Usuario
Precondiciones	El usuario debe haber iniciado sesión y haber creado un proyecto previamente.
Descripción	El usuario puede interactuar y analizar un conjunto de láminas virtuales a través de instancias específicas de las mismas, asociadas a un proyecto. Los proyectos pueden asociarse a varios usuarios registrados, permitiendo el trabajo colaborativo. Además, las anotaciones realizadas a las instancias pueden ser guardadas. El usuario puede seleccionar instancias y explorarlas a través de un visualizador que se comporta como un MV público pero con funciones reducidas.
Excepciones	<ul style="list-style-type: none"> • Invitar Nuevos Miembros al Proyecto (Tabla 3.5): Si el usuario requiere más colaboradores en el proyecto. • Chatear con Miembros del Proyecto (Tabla 3.6): Si el usuario requiere comunicarse con otros miembros. • Crear Lámina Virtual (Tabla 3.8): Si el usuario requiere una instancia nueva para trabajar. • Usar MV para Proyecto (Tabla 3.7): Si el usuario quiere interactuar con una instancia haciendo uso de funciones extendidas. • Usar Catálogo Incrustado (Tabla 3.7): Si el usuario desea explorar el catálogo desde la interfaz de proyecto.

Tabla 3.5: UC: Invitar Nuevos Miembros al Proyecto

Campo	Descripción
Caso de Uso	Invitar Nuevos Miembros al Proyecto
Objetivo	Permitir a los usuarios incluir nuevos colaboradores en un proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión y haber creado un proyecto previamente.
Descripción	El usuario, tiene la capacidad de buscar y seleccionar nuevos colaboradores para unirse al proyecto. Esta función permite al usuario buscar colaboradores utilizando el nombre de usuario registrado en el sistema. Una vez seleccionado el usuario deseado, el sistema proporciona una opción para enviar una invitación al colaborador seleccionado para unirse al proyecto.
Extensiones	Ninguna

Tabla 3.6: UC: Chatear con Miembros del Proyecto

Campo	Descripción
Caso de Uso	Chatear con Miembros del Proyecto
Objetivo	Facilitar herramientas de comunicación entre los miembros de un proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión y haber creado un proyecto previamente.
Descripción	El usuario contará con un chat grupal incluido en cada proyecto que permite la comunicación en tiempo real entre los miembros.
Extensiones	Ninguna

Tabla 3.7: UC: Usar Catálogo Incrustado

Campo	Descripción
Caso de Uso	Usar Catálogo Incrustado
Objetivo	Permitir a los usuarios registrados explorar las diferentes láminas virtuales de la base de datos en el contexto de un proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión y haber creado un proyecto previamente.
Descripción	Este UC hereda sus funciones de Usar Catálogoz además permite realizar actividades dentro del marco de un proyecto, lo que significa que los usuarios pueden explorar el catálogo de láminas virtuales mientras trabajan en un proyecto específico.
Extensiones	Ninguna

Tabla 3.8: UC: Crear Lámina Virtual

Campo	Descripción
Caso de Uso	Crear Lámina Virtual
Objetivo	Permitir a los usuarios crear nuevas instancias de placas virtuales dentro del contexto de un proyecto.
Actor	Usuario registrado
Inclusiones	Usar Catálogo Incrustado (para seleccionar el elemento del catálogo del cual se creará la instancia).
Precondiciones	El usuario debe haber iniciado sesión y haber creado un proyecto previamente.
Descripción	El usuario puede crear una nueva instancia de placa virtual para un proyecto a partir de cualquier placa virtual disponible en el catálogo.
Extensiones	Ninguna

Tabla 3.9: UC: Usar Microscopio Virtual para Proyecto

Campo	Descripción
Caso de Uso	Usar Microscopio Virtual para Proyecto
Objetivo	Permitir a los usuarios usar funciones extendidas en las instancias de un proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión y haber creado un proyecto previamente.
Descripción	Hereda las funciones del UC "Usar Microscopio Virtual". Además, el usuario puede acceder a funciones extendidas, como mostrar y ocultar capas de marcaciones de una instancia o utilizar una variedad más amplia de colores.
Extensiones	Crear y guardar anotaciones en una lámina virtual (Tabla 3.10): Si el usuario desea mantener las marcaciones y anotaciones realizadas en la instancia.

Tabla 3.10: UC: Crear y Guardar Anotaciones en una Lámina Virtual

Campo	Descripción
Caso de Uso	Crear y Guardar Anotaciones en una Lámina Virtual
Objetivo	Permitir a los usuarios usar funciones extendidas en las instancias de un proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión, haber creado un proyecto y una lámina virtual (instancia) previamente.
Descripción	El usuario puede guardar cambios en las anotaciones y marcaciones de una instancia, y asociarlos a una nota que actuará como una capa en el MV.
Extensiones	Ninguna

Tabla 3.11: UC: Usar Interfaz de Perfil de Usuario

Campo	Descripción
Caso de Uso	Usar Interfaz de Perfil de Usuario
Objetivo	Permitir a los usuarios registrados gestionar tanto su perfil como los proyectos asociados a este.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión.
Descripción	El usuario puede acceder y gestionar todos los proyectos asociados a su cuenta registrada, además de poder filtrar la lista de proyectos para una búsqueda más personalizada.
Extensiones	<ul style="list-style-type: none"> • Gestionar Invitación a Proyectos (Tabla 3.12): Si el usuario recibe invitaciones para colaborar en un proyecto • Crear Proyecto (Tabla 3.14): Si el usuario requiere crear un entorno donde pueda trabajar de forma colaborativa e interactuar y analizar láminas virtuales a través de instancias del catálogo

Tabla 3.12: UC: Gestionar Invitación a Proyecto

Campo	Descripción
Caso de Uso	Gestionar Invitación a Proyecto
Objetivo	Permitir a los usuarios gestionar las invitaciones recibidas para colaborar en algún proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión.
Descripción	El usuario podrá acceder a una lista con todas las invitaciones recibidas para participar en un proyecto. Además, contará con la opción de aceptar o rechazar dicha invitación.
Extensiones	Visualizar Perfil de Proyecto (Tabla 3.13): Si el usuario desea revisar a detalle algún proyecto al que lo están invitando

Tabla 3.13: UC: Visualizar Perfil de Proyecto

Campo	Descripción
Caso de Uso	Visualizar Perfil de Proyecto
Objetivo	Permitir a un usuario el acceso a la vista previa de un proyecto al ser invitado a colaborar en él.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión y haber recibido una invitación para colaborar en algún proyecto.
Descripción	Cuando el usuario recibe la invitación del proyecto, puede revisar los detalles del mismo para evaluar si acepta o no la invitación, pero no puede editar el proyecto hasta aceptarla.
Extensiones	Ninguna

Tabla 3.14: UC: Crear Proyecto

Campo	Descripción
Caso de Uso	Crear Proyecto
Objetivo	Permitir al usuario crear un nuevo proyecto.
Actor	Usuario registrado
Inclusiones	Ninguna
Precondiciones	El usuario debe haber iniciado sesión.
Descripción	El usuario podrá crear un nuevo proyecto, seleccionando un nombre y una descripción.
Extensiones	Ninguna

Tabla 3.15: UC: Crear Elemento del Catálogo

Campo	Descripción
Caso de Uso	Crear Elemento del Catálogo
Objetivo	Permitir a los administradores incluir nuevas placas virtuales en el catálogo de la plataforma.
Actor	Administrador
Inclusiones	Ninguna
Precondiciones	El administrador debe haber iniciado sesión.
Descripción	El administrador podrá agregar nuevas placas a la plataforma de forma sencilla, y estas se agregarán automáticamente al catálogo.
Extensiones	Ninguna

3.2. Wireframes

Con base en los casos de uso planteados, se han elaborado una serie de wireframes que representan la interfaz gráfica de la plataforma. El concepto principal de esta interfaz es asegurar que cuente con las herramientas necesarias para que la mayoría de los usuarios puedan utilizarla con la menor cantidad de dificultades posible. Los siguientes wireframes son de baja fidelidad, sin embargo, muestran la estructura básica de cada sección de la página y servirán como guía para la implementación de la plataforma.

3.2.1. Página de inicio

En la figura 3.2 se muestra el wireframe de la página de inicio de la plataforma. Este diseño incluye una sección destacada, conocida como "hero", diseñada para captar la atención del usuario de manera efectiva.

El contenido de la página de inicio se organiza en dos secciones principales. La primera sección es una breve presentación de la plataforma, donde se destacan sus principales funciones y ventajas a través de texto e imágenes. La segunda es una guía de uso detallada, diseñada para resolver cualquier duda relacionada con la navegación y uso de la plataforma.

En el wireframe también se muestra la barra de menú y el pie de página, los cuales estarán presentes en la mayoría de secciones de la plataforma. En la parte superior de la página se encuentra una barra de menú que proporciona acceso rápido a diferentes secciones, incluyendo "Inicio" para volver a esta página de inicio, "Microscopio" para explorar el catálogo público de la plataforma, y "Personas" para acceder a las cuentas de los usuarios registrados. Además, en la parte inferior se encuentra el pie de página con información adicional sobre la plataforma.

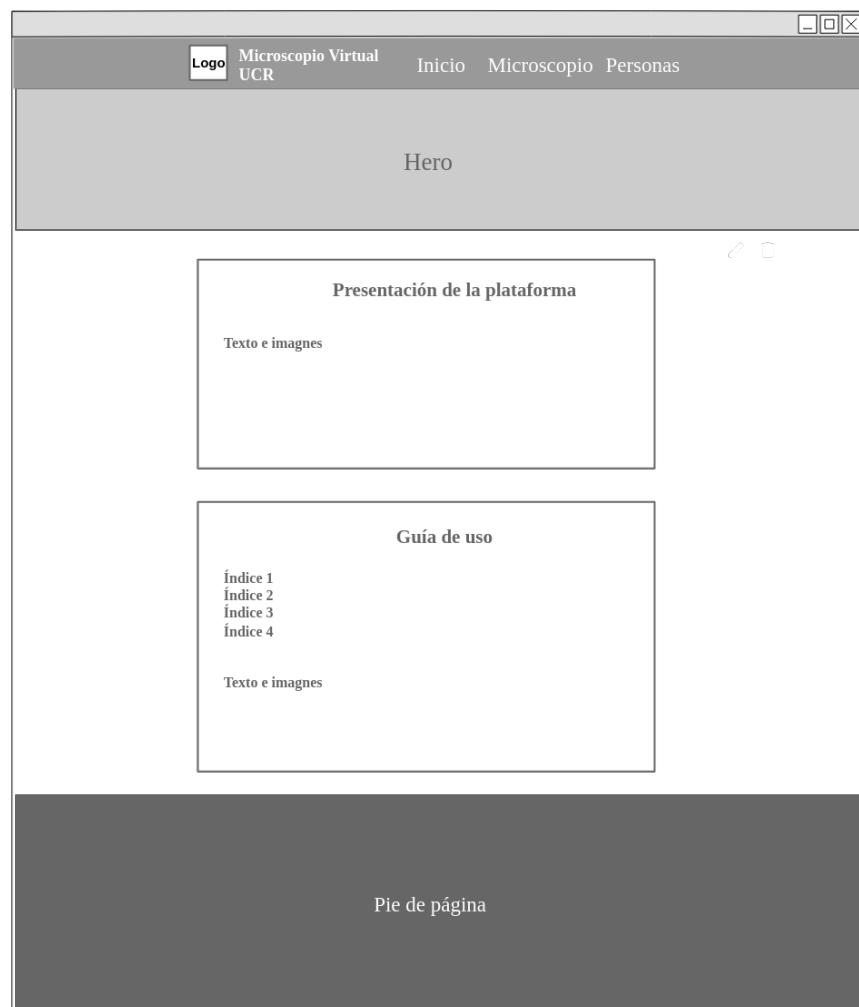


Figura 3.2: Wireframe: Pagina de inicio. Elaboración propia

3.2.2. Microscopio virtual público

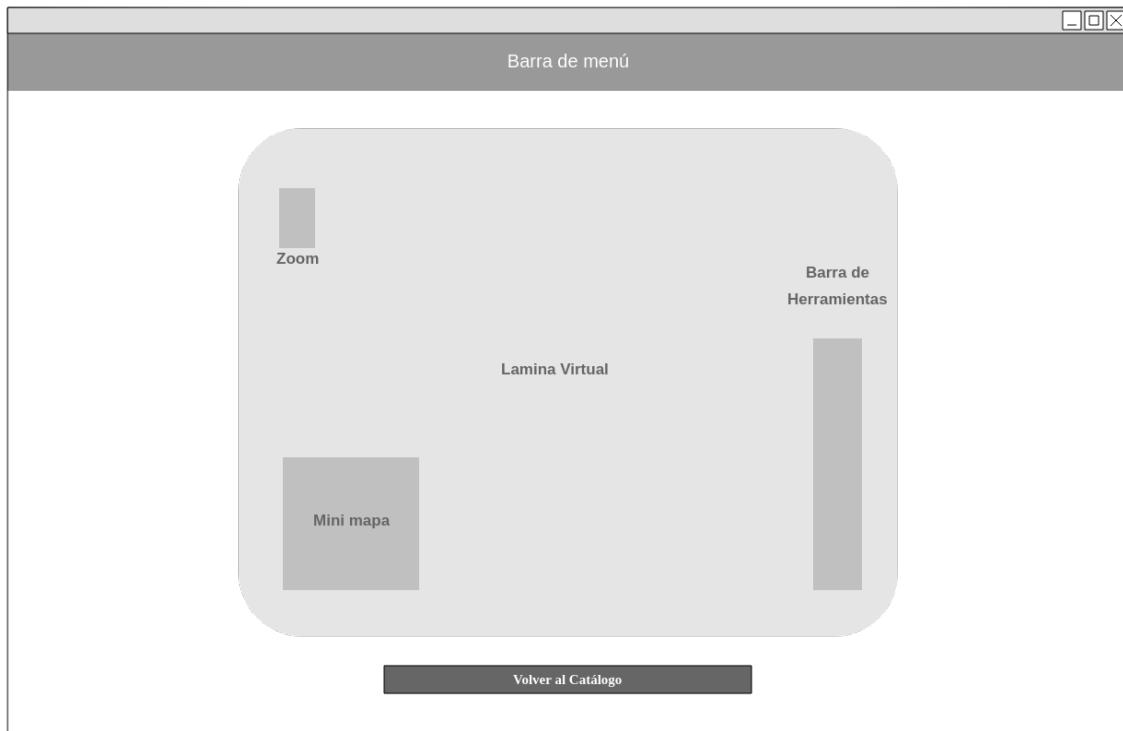


Figura 3.3: Wireframe: Microscopio público. Elaboración propia

En la figura 3.3 se presenta el wireframe correspondiente a la sección de la interfaz destinada al microscopio público de la plataforma. Este componente principal está conformado principalmente por un cuadro que exhibe la lámina virtual de microscopía mediante la tecnología de mapas web. Este cuadro, específicamente diseñado para esta plataforma, se denomina como "microscopio virtual". Ocupando la mayor parte de la pantalla, este microscopio se integra en el contexto habitual de la plataforma, acompañado de una barra de menú.

El MV está equipado con una serie de herramientas diseñadas para mejorar la experiencia del usuario al interactuar con él. Entre estas herramientas se incluye un mini mapa que proporciona una vista general del contexto de la imagen actual del microscopio dentro de la lámina virtual. Además, se dispone de un control de zoom que facilita la navegación entre las diferentes escalas de la imagen. Asimismo, se ha implementado una barra de herramientas que permite realizar mediciones y marcar áreas de interés en la lámina.

Finalmente, para garantizar una navegación fluida, se ha añadido un botón fácilmente identificable que permite regresar al catálogo principal de la plataforma.

3.2.3. Catálogo

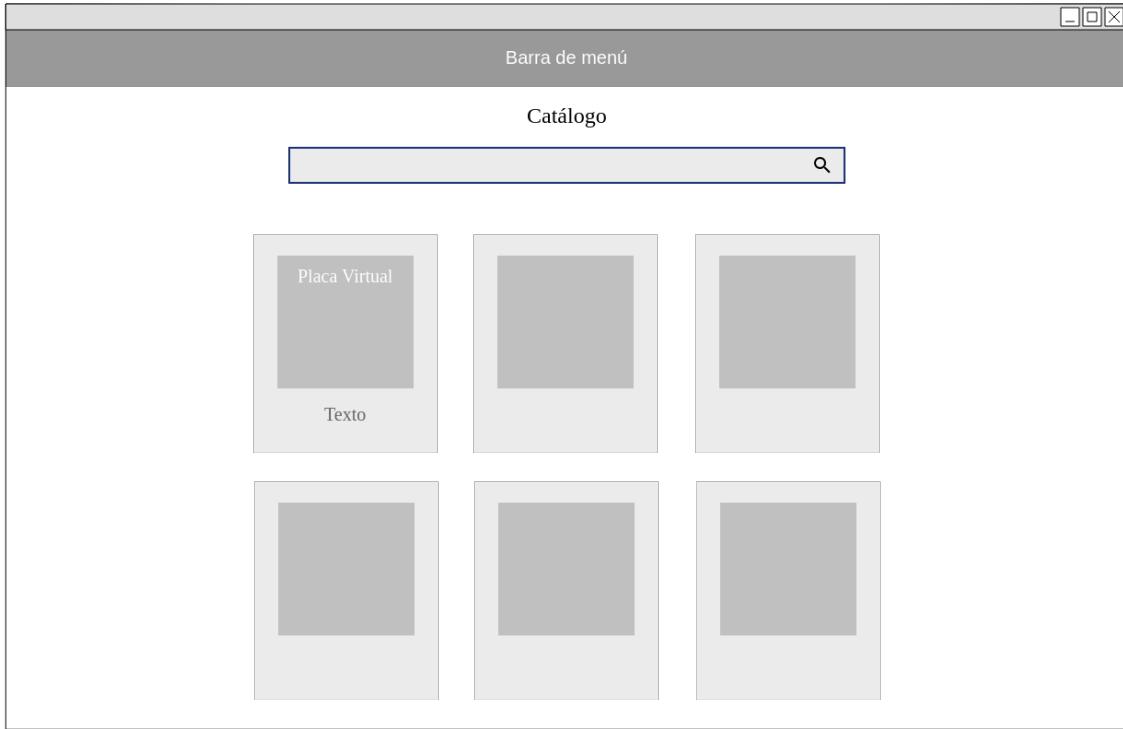


Figura 3.4: Wireframe: Catalogo público. Elaboración propia

En la figura 3.4 se muestra el wireframe correspondiente al catálogo público de la plataforma. Este diseño presenta una galería que exhibe las diversas láminas virtuales disponibles sin requerir la creación de una cuenta en la plataforma. El catálogo se caracteriza por su diseño simple, mostrando únicamente una imagen preliminar de la lámina virtual y una breve descripción de la misma. Además, el catálogo está equipado con un buscador que permite filtrar las opciones mostradas. Con la interfaz de catálogo, los usuarios pueden seleccionar fácilmente qué lámina virtual desean utilizar en el MV.

3.2.4. Gestor de nuevas placas virtuales

En la figura 3.5 se presenta el wireframe correspondiente a la interfaz encargada de gestionar la subida de nuevas placas virtuales en varios formatos, así como la opción de procesar los archivos para poder utilizar las placas en el microscopio de la plataforma. Esta interfaz ofrece un área donde los usuarios pueden cargar los archivos de dos formas: arrastrando el archivo al área designada o navegando entre los directorios para seleccionar los archivos.

El área principal de la interfaz muestra una tabla con los archivos subidos, donde cada elemento cuenta con dos opciones: procesar la placa o eliminarla, brindando así un control sobre la gestión de

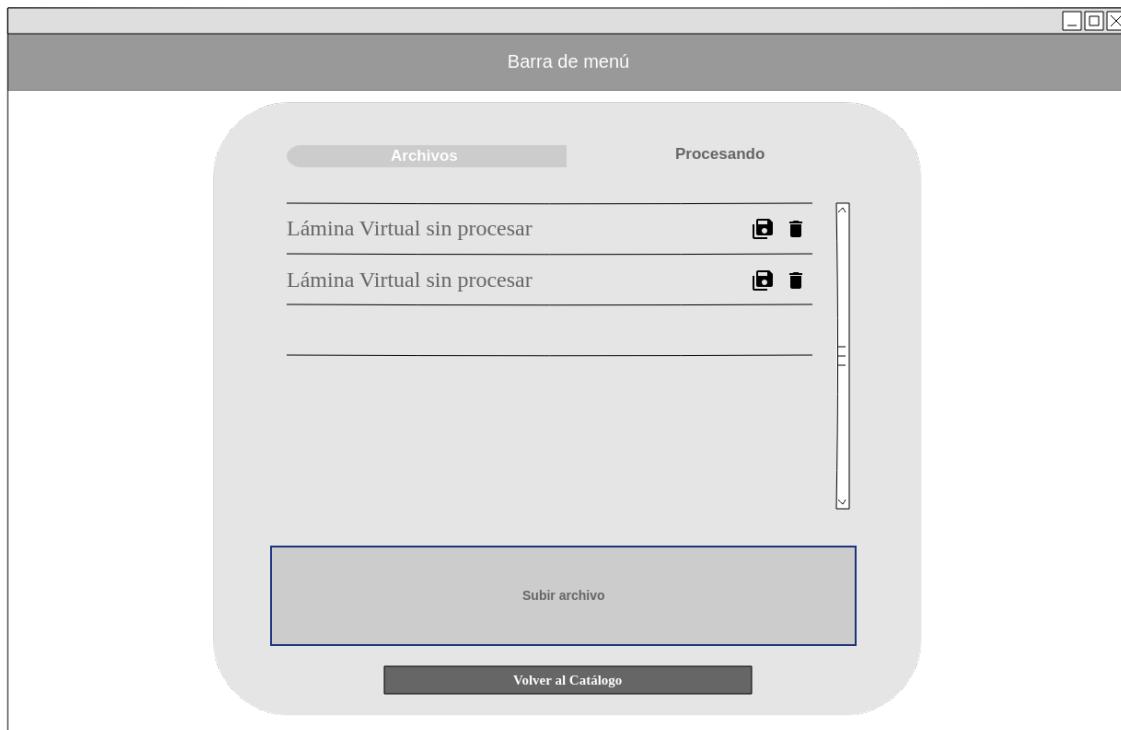


Figura 3.5: Wireframe: Gestión archivos subidos. Elaboración propia

los archivos cargados. Además, la interfaz está dividida en dos pestañas que permiten alternar entre el gestor de archivos subidos y el gestor de placas en proceso, lo que facilita la visualización y la gestión de los archivos según su estado.

Finalmente, se incluye un botón que permite a los usuarios volver al catálogo principal, proporcionando una navegación fluida dentro de la plataforma.

3.2.5. Tabla de Procesamiento de Archivos

En la figura 3.6 se muestra el wireframe correspondiente a la segunda pestaña de la interfaz encargada de gestionar la subida de nuevas placas virtuales. El área principal de esta interfaz presenta una tabla que muestra los archivos que se están procesando en ese momento para crear un nuevo elemento en el catálogo.

Al igual que en la primera pestaña, esta interfaz está dividida en dos pestañas que permiten alternar entre el gestor de archivos subidos y el gestor de placas en procesamiento, lo que facilita la gestión de los archivos según su estado.

Finalmente, se incluye un botón que permite a los usuarios volver al catálogo principal, asegurando una navegación coherente dentro de la plataforma.

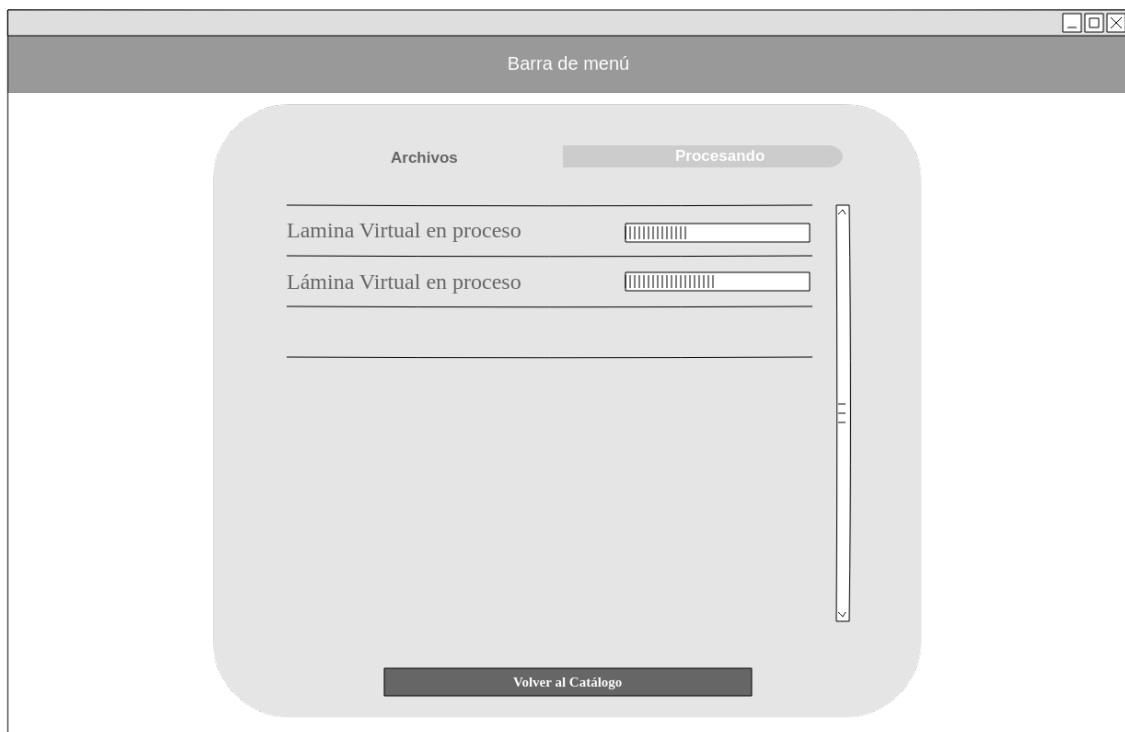


Figura 3.6: Wireframe: Gestión de procesamiento. Elaboración propia

3.2.6. Inicio de sesión

El wireframe de la interfaz para iniciar sesión de usuarios se presenta en la figura 3.7. El diseño de esta interfaz es sencillo y se compone principalmente del formulario para iniciar sesión, situado en el contexto habitual de la plataforma. El formulario solicita el nombre de usuario y contraseña, y además cuenta con dos botones: uno para iniciar sesión y otro para acceder a la sección de crear una nueva cuenta.

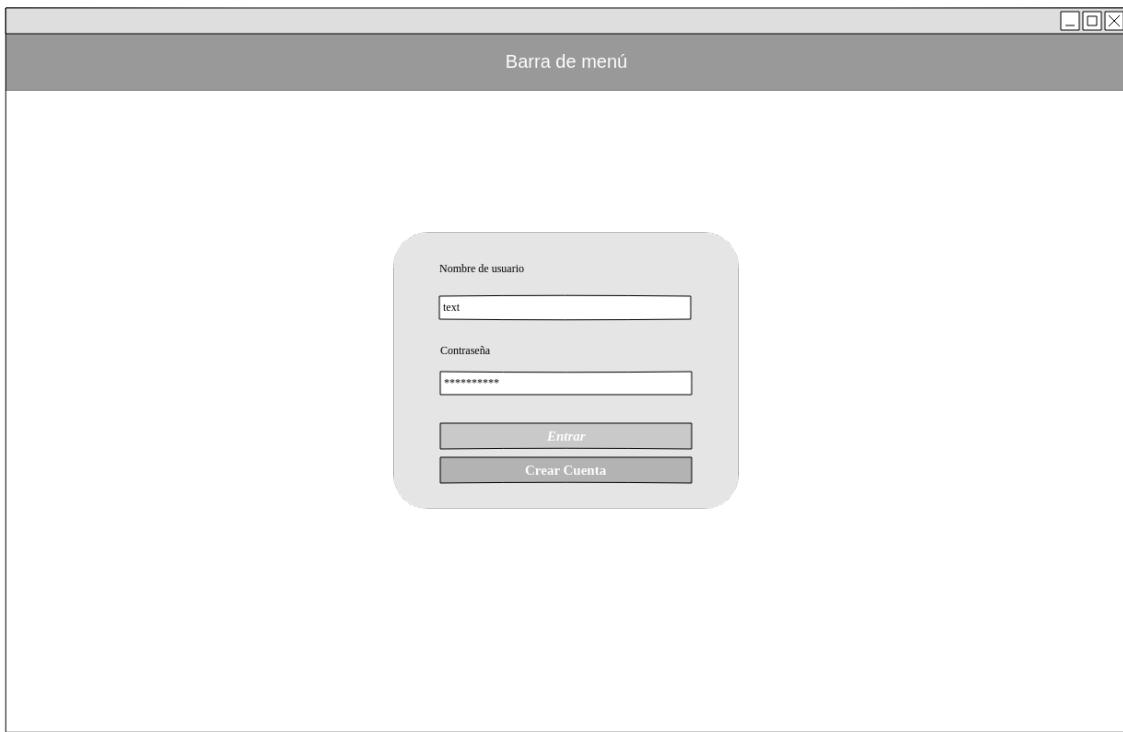


Figura 3.7: Wireframe: Inicio sesión. Elaboración propia

3.2.7. Registro de usuarios

El diseño de la interfaz para crear una nueva cuenta se detalla en la figura 3.8. Al igual que en la sección de inicio de sesión, esta interfaz presenta un diseño sencillo y consta principalmente de un formulario ubicado en el contexto habitual de la plataforma. El formulario solicita algunos datos del usuario, como el nombre de usuario o correo electrónico, y también confirma que la contraseña cumpla con el formato correcto.

Además, al igual que en la sección de inicio de sesión, esta interfaz cuenta con dos botones: uno para crear la nueva cuenta y otro para volver al formulario de inicio de sesión.

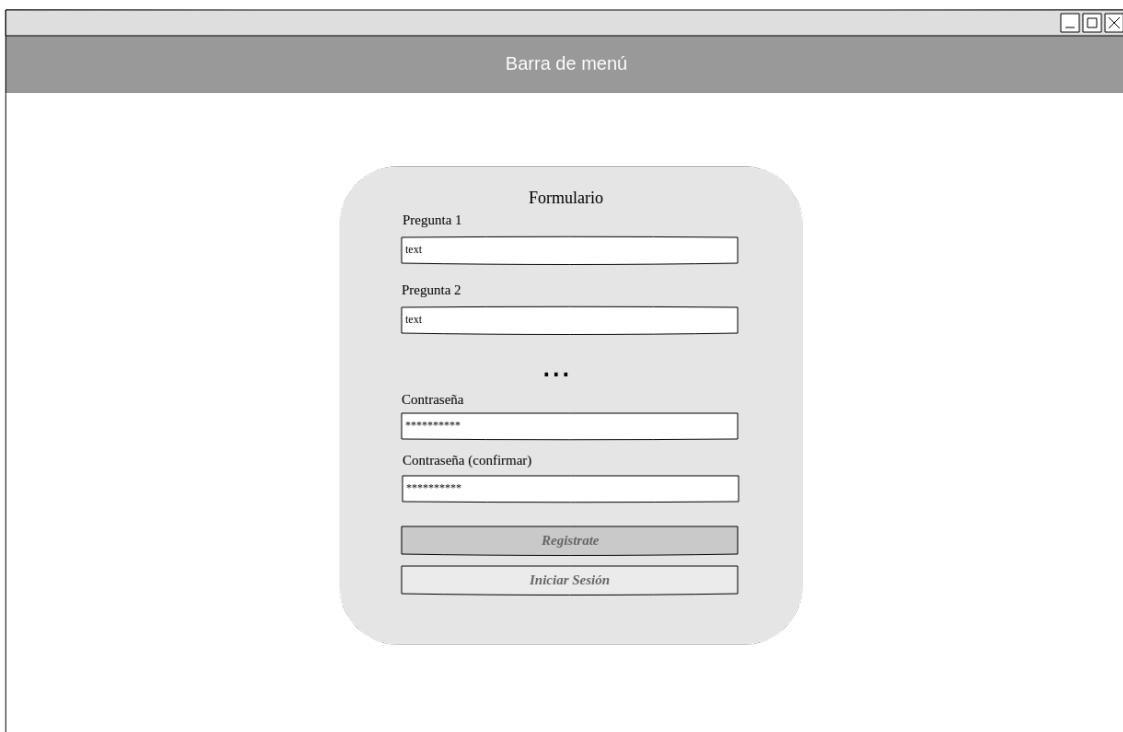


Figura 3.8: Wireframe: Crear cuenta. Elaboración propia

3.2.8. Gestión de Proyectos

El wireframe de la interfaz principal de usuarios se puede observar en la figura 3.9. Esta sección solo es accesible después de iniciar sesión y está diseñada principalmente para gestionar los diferentes proyectos del usuario. La interfaz presenta un panel izquierdo que consta de tres secciones principales. En la primera sección se muestra el perfil del usuario, junto con las opciones para cerrar sesión y editar el perfil. En la siguiente sección se encuentra una barra de búsqueda para filtrar los proyectos del usuario, y finalmente, la tercera sección está destinada a la creación de nuevos proyectos.

Además del panel mencionado, la interfaz principal cuenta con una sección donde se despliega una tabla con los proyectos del usuario. Esto permite al usuario seleccionar y acceder rápidamente al proyecto deseado. La tabla muestra el nombre del proyecto y una breve descripción. Al colocar el ratón sobre un elemento, se muestran dos opciones: para borrar o editar las características principales del proyecto.

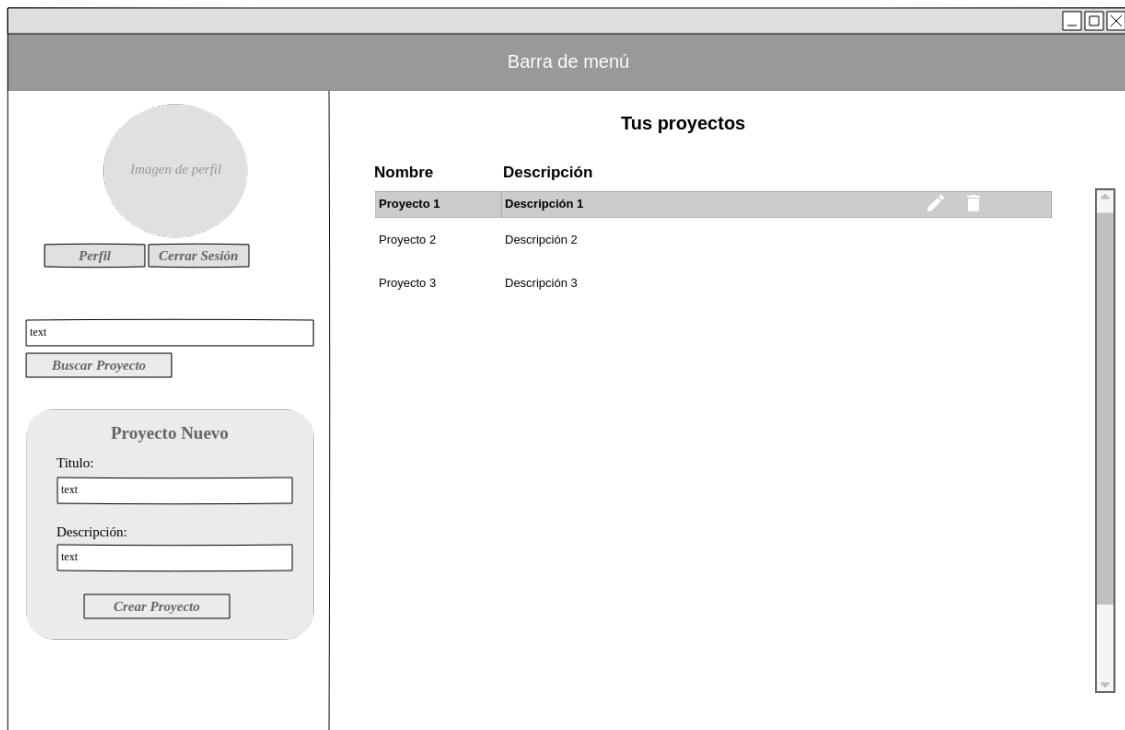


Figura 3.9: Wireframe: Interfaz principal. Elaboración propia

3.2.9. Detalles del Proyecto

En la figura 3.10 se presenta el wireframe de la sección que despliega los detalles principales del proyecto. Esta sección es accesible al seleccionar un proyecto en el gestor de proyectos y forma parte de la interfaz de trabajo de cada proyecto. En esta sección se muestra un panel principal con información clave del proyecto, como su nombre, propietario, descripción, el número de instancias de placas virtuales con las que se está trabajando y quienes son los colaboradores del proyecto.

Además, la sección cuenta con un panel de herramientas ubicado a la derecha de la pantalla. Este panel permite al usuario navegar por diferentes menús que ofrecen opciones para trabajar en el proyecto, como la gestión de archivos, la realización de análisis o la colaboración con otros usuarios.

Finalmente, se incluye un botón para volver al gestor de proyectos, proporcionando una navegación coherente dentro de la plataforma.



Figura 3.10: Wireframe: Detalles de proyecto. Elaboración propia

3.2.10. Visualizador de Placas Virtuales y sus Instancias

En la figura 3.11 se presenta el wireframe de la sección destinada a la visualización tanto de las placas virtuales del catálogo como de las instancias de estas creadas para cada proyecto. Esta área forma parte integrante de la interfaz de trabajo de cada proyecto y se puede acceder a ella al seleccionar la opción de Catálogo.^º "Láminas virtuales."^{en} en el panel de herramientas.

Aquí, el MV ocupa un lugar central, permitiendo a los usuarios examinar minuciosamente las imágenes de las láminas virtuales, que constituyen los elementos fundamentales de cada proyecto. Además, esta sección cuenta con un panel de herramientas que ofrece diversas opciones para interactuar con las imágenes. En la parte inferior de la pantalla, un panel de apoyo proporciona detalles adicionales sobre las imágenes visualizadas y ofrece diversas herramientas, para trabajar con las instancias. Por último, se incluye un botón para regresar al gestor de proyectos, lo que facilita la navegación dentro de la plataforma.



Figura 3.11: Wireframe: Visualizador de placas. Elaboración propia

3.2.11. Editor de Instancias de Láminas Virtuales

En la figura 3.12 se aprecia el diseño de la interfaz dedicada a la edición de cada instancia de placa virtual, donde los usuarios pueden agregar notas y marcaciones de diversos tipos. Esta función es accesible desde el visualizador de instancias.

El diseño incluye un MV que ocupa la mayoría de la pantalla y es similar al microscopio público descrito anteriormente, pero con algunas funciones adicionales. Desde esta visualización del microscopio, los usuarios pueden agregar marcaciones en forma de capas, las cuales se asociarán a notas.

En el wireframe, se observa que en la esquina superior izquierda se despliega una lista de notas, encabezada por una barra con el nombre de la instancia de lámina virtual. Esto permite a los usuarios realizar sus anotaciones, activar y desactivar capas de marcaciones, además de asociar ubicaciones a comentarios. Los cambios se guardan en la instancia y pueden ser vistos y editados por los colaboradores del proyecto asociado.

Finalmente, se incluye un botón para regresar a la interfaz de trabajo del proyecto, lo que facilita la navegación dentro de la plataforma.

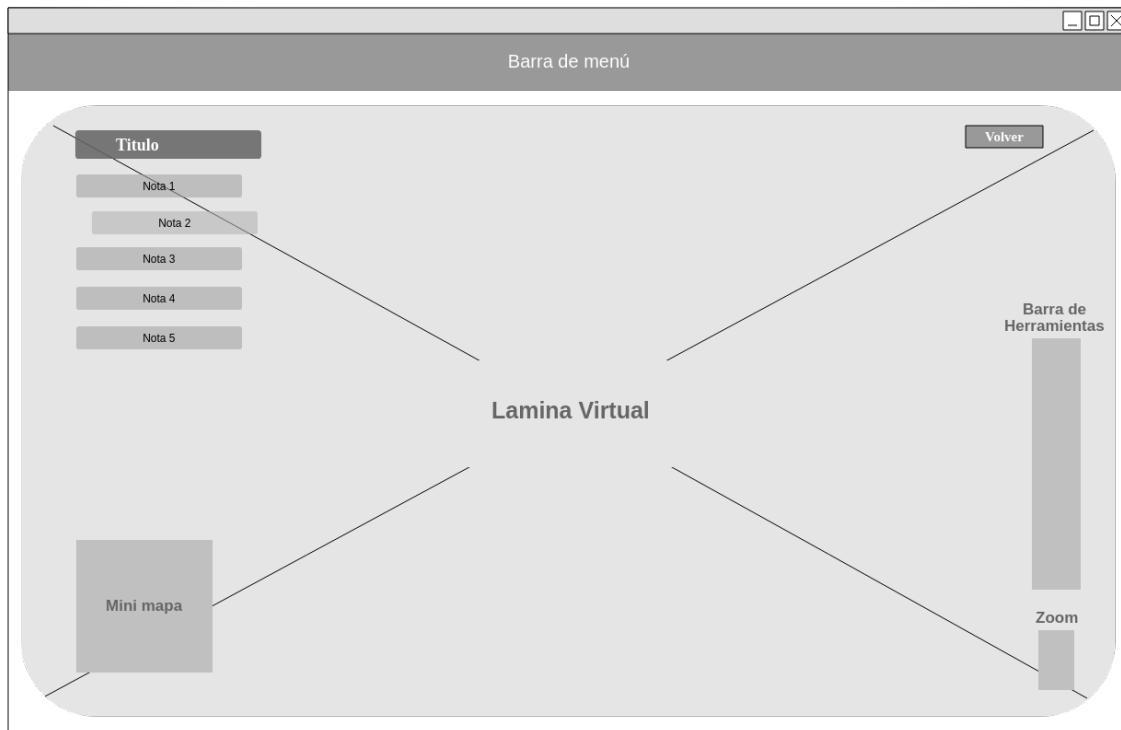


Figura 3.12: Wireframe: Edición de instancia. Elaboración propia

3.2.12. Panel de herramientas de proyecto

En el wireframe de la figura 3.13 se muestra el diseño del panel de herramientas, que ofrece cuatro menús diferentes. En la parte superior del panel, se observa una barra con el título del menú seleccionado. Al hacer clic en esta barra, se despliegan cuatro opciones para cambiar el tipo de menú y el título correspondiente.

Justo debajo del título desplegado en el menú, se encuentra una barra de búsqueda que permite filtrar los elementos desplegados en cada sección. A continuación, se describen los cuatro menús ofrecidos:

1. **Menú de láminas virtuales:** En este menú se muestran todas las instancias de placas asociadas al proyecto. Cada elemento de la lista muestra el nombre y un botón para abrir el visualizador del área de trabajo con la instancia seleccionada.
2. **Menú de catálogo:** Este menú muestra todos los elementos del catálogo disponibles para crear instancias. Cada placa virtual muestra una imagen de previsualización y su título. Al seleccionar algún elemento, se abre el visualizador del área de trabajo y se habilita la opción de crear una instancia.

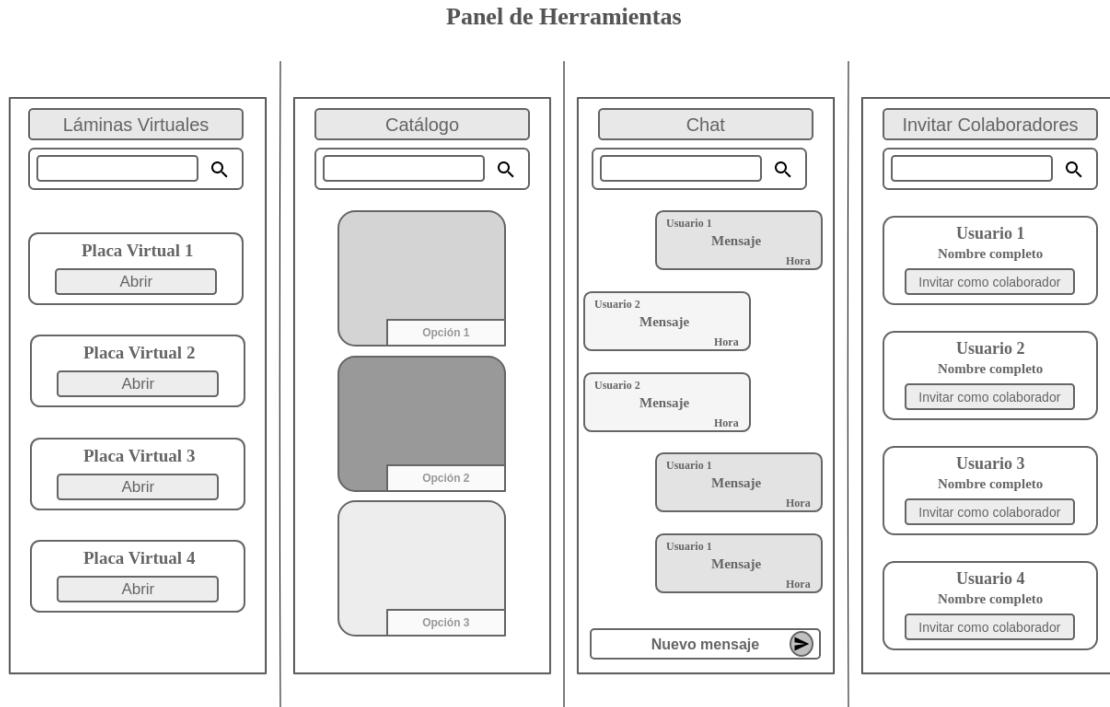


Figura 3.13: Wireframe: Panel de herramientas. Elaboración propia

3. **Chat en tiempo real:** Este menú consiste en un chat sencillo para la comunicación entre los colaboradores. Los mensajes se despliegan en orden cronológico y el menú cuenta con una barra para escribir y enviar nuevos mensajes.
4. **Menú de búsqueda de colaboradores:** Este menú está dedicado a buscar nuevos colaboradores para el proyecto. Se puede escribir el nombre o nombre de usuario en la barra de búsqueda, y se mostrarán las opciones para enviar una invitación al proyecto.

3.2.13. Descripción del panel de apoyo a proyectos

La plataforma cuenta con un visualizador que permite explorar tanto las placas virtuales como las instancias creadas en el proyecto. Para ambos casos, se dispone de un panel de apoyo. En la figura 3.14 se muestra el wireframe con el panel de apoyo en ambos casos.

Cuando se visualiza una instancia, se despliega el nombre y una descripción, además de cuatro botones. El primero permite volver a la página de detalles del proyecto, el segundo es para abrir la instancia en la interfaz de edición, donde se pueden agregar anotaciones y marcaciones. El tercer botón permite editar los detalles principales de la instancia, como el nombre y la descripción, mientras que el último es para borrar la instancia.

En el caso de visualizar un elemento del catálogo, se muestra de igual forma el nombre y la descripción, pero las opciones cambian a solo dos. La primera es para volver a la página de detalles del proyecto y la segunda para crear una nueva instancia de la placa virtual.

Panel de Apoyo de la Interfaz de Proyectos

The wireframe illustrates the 'Panel de Apoyo de la Interfaz de Proyectos' (Support Panel for the Project Interface) in two main sections:

- Para Láminas Virtuales (Top Section):**
 - Header: 'Nombre de la instancia de la placa virtual'
 - Text input field: 'Descripción'
 - Action buttons on the right: 'Volver al perfil de proyecto', 'Abrir', 'Editar', and 'Eliminar'.
- Para Elementos del Catálogo (Bottom Section):**
 - Header: 'Nombre del elemento del catálogo'
 - Text input field: 'Descripción'
 - Action buttons on the right: 'Volver al perfil de proyecto' and 'Crear instancia del elemento del catálogo'.

Figura 3.14: Wireframe: Panel de apoyo. Elaboración propia

3.2.14. Base de datos

Una base de datos con un diseño robusto es fundamental para el desarrollo de una plataforma web. El proyecto requiere almacenar, organizar y gestionar información, por lo que es necesario contar con los recursos necesarios para realizar estas tareas. En el caso de las bases de datos relacionales, la forma en que están definidas las tablas y sus relaciones tiene influencia directa sobre el comportamiento de la plataforma y sus capacidades.

Con la definición de los casos de uso se evidencian las interacciones de los usuarios con la plataforma. De esta forma, se pueden identificar las entidades o tablas de la base de datos y cómo se relacionan entre sí.

A partir de la definición de los wireframes se pueden identificar los requisitos específicos de la plataforma, lo que permite definir los campos necesarios de las tablas.

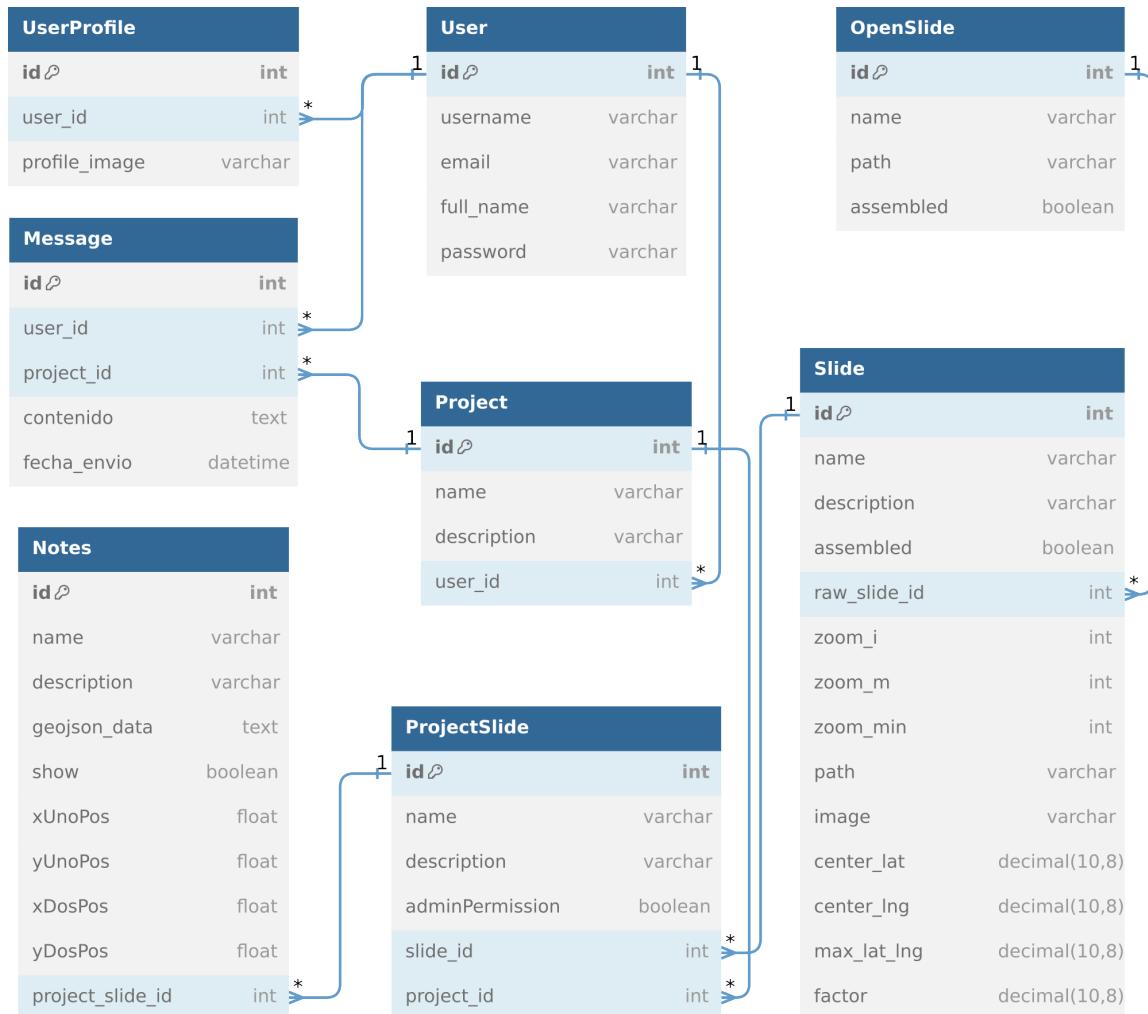


Figura 3.15: Diagrama entidad relación de la base de datos. Elaboración propia

- **User:** Esta plataforma tiene como función principal interactuar con los usuarios, identificados como usuarios registrados y administradores según se describe en la tabla 3.1. Ambos tipos de usuarios requieren una cuenta con datos importantes, lo cual es gestionado por la tabla **User-Profile**.
- **UserProfile:** El wireframe de la figura 3.9 muestra que los usuarios cuentan con un perfil que incluye una imagen. Para dar soporte a esta información personalizada, se utiliza la tabla **User-Profile**, directamente relacionada con la tabla **User**.
- **Project:** Los usuarios registrados pueden crear y utilizar proyectos nuevos, según se describe en los casos de uso "Usar Interfaz de Proyecto"(tabla 3.4) y "Usar Interfaz de Perfil de Usuario"(tabla 3.11). Un proyecto relaciona un grupo de usuarios con un conjunto de implementaciones de placas virtuales, y cuenta con un nombre y una descripción, como se muestra en el wireframe Detalles de Proyecto (figura 3.10).
- **Slide:** Un slide o placa virtual es un elemento que se despliega en el MV y el catálogo, según se describe en los casos de uso de las tablas 3.3, 3.7, 3.9 y 3.2. Esta tabla almacena información necesaria para el despliegue de imágenes de microscopía, como el zoom máximo y mínimo, la dirección al directorio raíz de las imágenes de la placa, datos para centrar y limitar la visualización, así como información para realizar mediciones.
- **OpenSlide:** Los usuarios pueden crear nuevas placas virtuales a partir de preparados con formatos específicos de microscopios, como se describe en el UC de la tabla 3.8. La tabla **OpenSlide** gestiona la información de estos archivos para procesarlos y crear nuevos elementos del catálogo. Registra información como la dirección del archivo y si ya fue procesada, y se gestiona desde la interfaz de usuario, como se muestra en los wireframes de las figuras 3.5 y 3.6.
- **ProjectSlide:** Para realizar anotaciones personalizadas se requiere una nueva tabla que gestione implementaciones de las placas virtuales y las vincule a proyectos, como se describe en el UC de la tabla 3.8. Estos archivos se pueden utilizar desde la interfaz de proyecto, como se observa en los wireframes 3.11 y 3.12.
- **Notes:** Las notas son anotaciones y figuras que se vinculan a una implementación, como se describe en el UC 3.10. La tabla **Notes** almacena principalmente información en forma de un GeoJSON, una estructura comúnmente utilizada en mapas web. Las notas se despliegan en la interfaz de edición, como se muestra en el wireframe de la figura 3.12.
- **Message:** Los casos de uso incluyen un chat en tiempo real, como se describe en la tabla 3.6. La tabla **Message** gestiona la información básica para la implementación de esta función, y se visualiza en el panel de herramientas, como se muestra en el wireframe de la figura 3.13.

Capítulo 4

Implementación y Resultados

4.1. Implementación con el framework Django

Para la implementación de la plataforma se ha empleado el framework Django, el cual sigue el patrón Modelo-Vista-Plantilla (MVT por sus siglas en inglés), ambos descritos en el marco teórico. Se escogió este framework por su facilidad de uso, versatilidad y amplias capacidades. Los proyectos en Django se componen de una o varias aplicaciones, cada una de las cuales contiene archivos como `models.py`, `urls.py`, `views.py`, `forms.py`, entre otros. Esta plataforma cuenta con cuatro aplicaciones, que se describen en las siguientes secciones:

1. Gestión del catálogo y visualización de láminas virtuales: Implementación de la aplicación **microscope**.
2. Administración del sitio web y perfiles de usuario: Implementación de la aplicación **website_management**.
3. Gestión de proyectos: Implementación de la aplicación **projects**.
4. Configuración y ajustes del entorno de la plataforma: **virtual_microscope**.

En cada uno se detallará cómo se implementó y el resultado, además de cómo se integraron herramientas como Daphne, Celery, Gunicorn y Leaflet.

4.1.1. Gestión del catálogo y visualización de láminas virtuales

La implementación de la aplicación de Django con el nombre **microscope** se encarga de gestionar todo lo relacionado con el catálogo, las placas virtuales, el funcionamiento del microscopio y la gestión de los archivos subidos a la plataforma para crear nuevas placas.

Implementación del archivo `models.py` de la aplicación

El archivo ‘models.py’ se crea a partir del diseño de las tablas de la base de datos correspondientes a **OpenSlide** y **Slide**. En el código 4.1 se representan cada tabla con una clase y sus atributos son los campos, donde se detalla el tipo de dato, su extensión y otras configuraciones. También se detallan dos funciones, una para el nombre y otra para la URL.

```

1 import os
2 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'virtual_microscope.settings')
3 from django.db import models
4 from django.contrib.auth.models import User
5 from django.urls import reverse
6
7 # Create your models here.
8 class OpenSlide(models.Model):
9     name = models.CharField(unique=True, max_length=50)
10    path = models.CharField(max_length=50, null=True)
11    assembled = models.BooleanField(default=False)
12
13    def __str__(self):
14        return self.name
15
16    def get_absolute_url(self):
17        return reverse('open-micro-slide', args=[str(self.id)])
18
19
20 class Slide(models.Model):
21    name = models.CharField(unique=True, max_length=50)
22    description = models.CharField(max_length=500)
23    assembled = models.BooleanField(default=False)
24    rawSlide = models.ForeignKey(OpenSlide, on_delete=models.CASCADE, null=True)
25    zoomI = models.IntegerField(null=True)
26    zoomM = models.IntegerField(null=True)
27    zoomMin = models.IntegerField(null=True)
28    path = models.CharField(max_length=50, null=True)
29    image = models.ImageField(upload_to='slides', null=True, )
30    centerLat = models.DecimalField(max_digits=10, decimal_places=8, null=True)
31    centerLng = models.DecimalField(max_digits=10, decimal_places=8, null=True)
32    maxLatLng = models.DecimalField(max_digits=10, decimal_places=8, null=True)
33    factor = models.DecimalField(max_digits=10, decimal_places=8, null=True)
34    error = models.BooleanField(default=False, null=True)
35
36    def __str__(self):
37        return self.name
38
39    def get_absolute_url(self):
40        return reverse('micro-slide', args=[str(self.id)])

```

Listing 4.1: Código del archivo `models` de `microscope`

Implementación del archivo urls.py de la aplicación

Para definir el archivo urls.py, se asocian URLs a los elementos de la interfaz diseñados en los wireframes presentados en las imágenes 3.3, 3.4, 3.5 y 3.6, además se asignaron urls a las diferentes solicitudes de la página hacia el servidor. El código 4.2 muestra la implementación de este archivo. Es importante destacar que las URLs asociadas al microscopio y catálogos públicos son accesibles para todos los usuarios, mientras que las vinculadas a editar el catálogo solo están disponibles si se inicia sesión como administrador.

```

1 from django.urls import path, re_path
2 from django.conf import settings
3 from django.conf.urls.static import static
4 from microscope.views import catalogo, micro, delete
5 from . import views
6 from django.contrib.auth.decorators import user_passes_test
7 from django.contrib.auth.models import User
8 from django.http import HttpResponseRedirect
9
10 def admin_required(view_func):
11     def _wrapped_view(request, *args, **kwargs):
12         if not request.user.is_superuser:
13             return HttpResponseRedirect("Acceso denegado. Debes ser
14             administrador para acceder a esta pagina.")
15         return view_func(request, *args, **kwargs)
16     return _wrapped_view
17
18
19 urlpatterns = [
20     path('catalogo/', catalogo, name = "Catalogo"),
21     path('micro/<pk>/', micro.as_view(), name='micro-slide'),
22     path('upload_file/', admin_required(views.upload_file), name='upload_file'),
23     path('processing/', admin_required(views.processing), name='processing'),
24     path('delete/<int:slide_id>', admin_required(delete))
25 ]

```

Listing 4.2: Código del archivo urls de microscope

Implementación de la vista y plantilla para el catálogo

En la figura 4.1 se muestra el catálogo público de la plataforma, junto con cinco etiquetas que resaltan los principales elementos de la interfaz gráfica. Para acceder a esta sección, es necesario seleccionar la opción **Microscopio** en la barra de menú, la cual está disponible para cualquier usuario sin necesidad de registrarse en la plataforma. Este diseño sigue las pautas propuestas en el wireframe de la figura 3.4.A continuación, se enumeran los elementos principales destacados en la imagen:

- Título y barra de búsqueda:** Esta opción permite filtrar la lista de elementos mostrados en el catálogo. Se implementaron filtros en la vista correspondiente del archivo views.py de la aplicación. La barra de búsqueda actúa como un formulario, donde las palabras indicadas son los

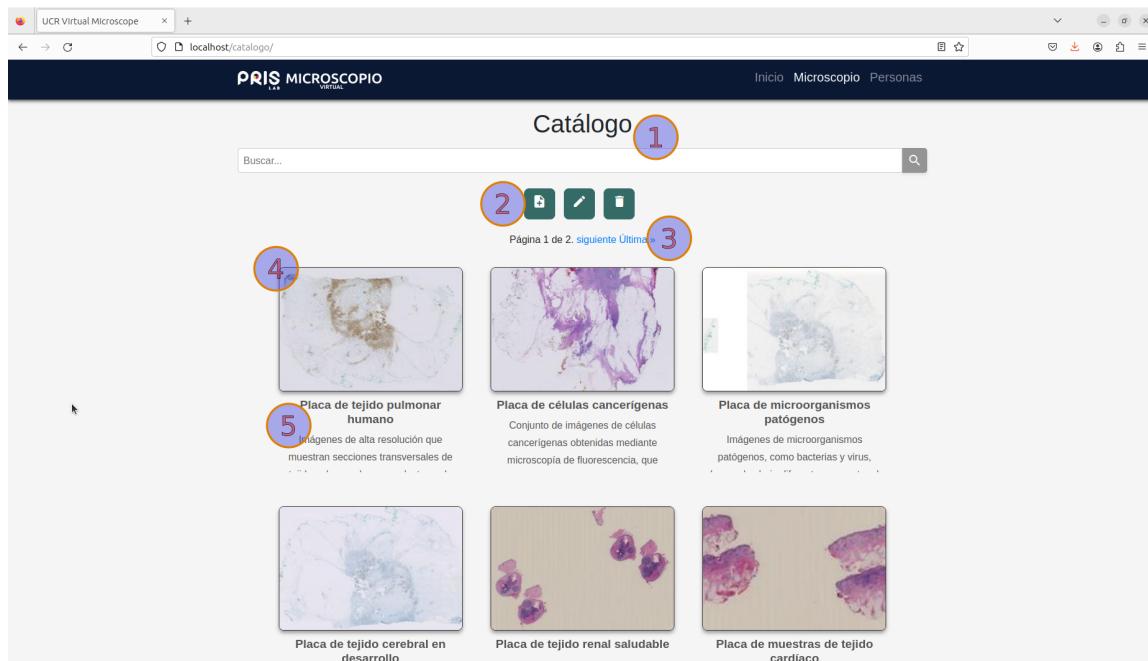


Figura 4.1: Resultado del catálogo público. Elaboración propia

terminos de búsqueda que se aplicarán tanto en los nombres de las placas como en sus descripciones.

2. **Menú de edición del catálogo:** Dispone de tres opciones para manipular los elementos mostrados. Solo está disponible si un usuario inicia sesión como administrador. La primera opción redirige a los usuarios a un menú que gestiona el cargado y procesamiento de nuevas láminas virtuales, descrito en la sección correspondiente. La segunda opción permite la edición del título y descripción de las placas, mediante un formulario habilitado al seleccionar alguna opción del catálogo. Finalmente, la opción de eliminar elementos permite borrar los preparados virtuales, aprovechando una vista y una URL como se muestra en el código.
3. **Páginas:** Esta sección facilita la navegación por las páginas del catálogo. Para crear las páginas, se divide en secciones la lista de elementos filtrados, mediante una función en la vista correspondiente.
4. **Catálogo:** Es la sección principal que despliega una lista con la vista previa de los elementos del catálogo. Se agregaron los estilos necesarios en la plantilla para lograr un diseño interactivo, y se gestiona la base de datos mediante las vistas.
5. **Título y descripción:** Proporciona información de cada placa virtual, obtenida de la base de datos manipulada por las vistas.

Implementación de la vista y plantilla para el microscopio público

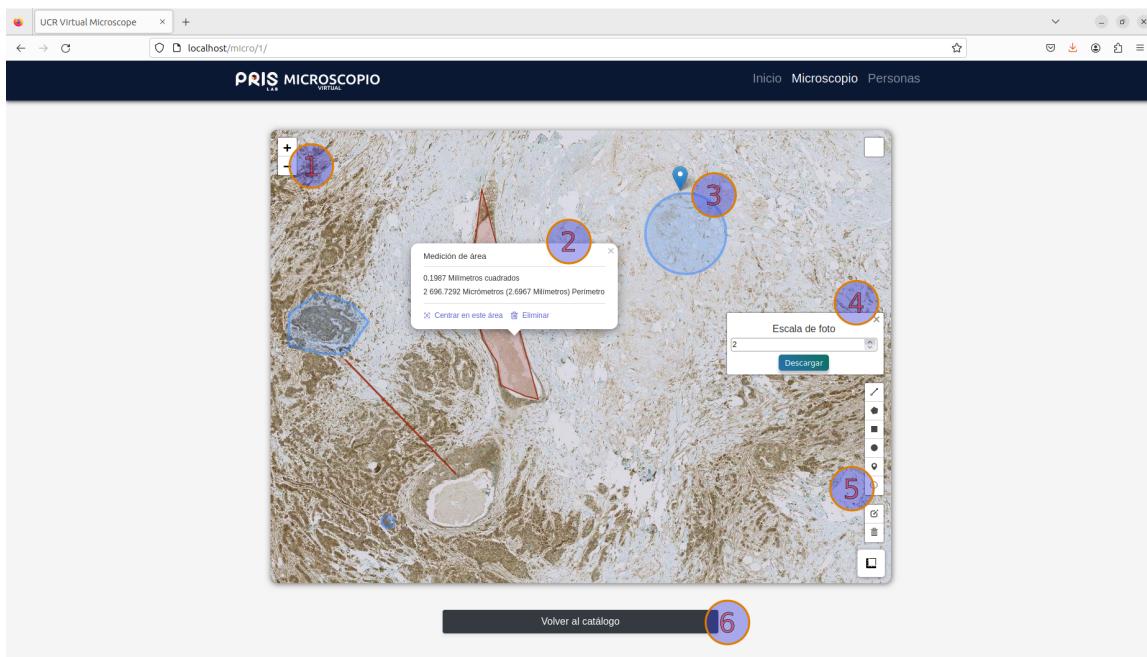


Figura 4.2: Resultado del microscopio público. Elaboración propia

En la figura 4.2 se muestra el microscopio público de la plataforma, junto con seis etiquetas que resaltan los principales elementos de la interfaz gráfica. Para acceder a esta sección, es necesario seleccionar la opción **Microscopio** en la barra de menú, y escoger un elemento del catálogo, la cual está disponible para cualquier usuario sin necesidad de registrarse en la plataforma. Este diseño sigue las pautas propuestas en el wireframe de la figura 3.3.

Para la implementación del microscopio, se empleó la biblioteca de código abierto Leaflet en JavaScript. Esta biblioteca se utiliza para desplegar mapas interactivos en páginas web, como se describe en el marco teórico. Para adaptarla al uso de un preparado de microscopía web en lugar del mapa, se utilizó una extensión de la biblioteca que permite emplear teselas organizadas en árboles de escritorio. Esto permite representar imágenes de ultra alta resolución en diferentes niveles de zoom. Esta implementación se encuentra en una de las plantillas de la aplicación de microscopio, que despliega su correspondiente vista. Además, para agregar más funcionalidades a la aplicación, en esta implementación del microscopio también se emplearon extensiones dedicadas a desplegar herramientas de dibujo, gestionar la pantalla completa, realizar mediciones, entre otras. Las diferentes extensiones utilizadas se detallan en la tabla 4.1.

Extensión	Descripción
leaflet	Hoja de estilos y archivos JavaScript de Leaflet.
leaflet-draw	Hoja de estilos CSS y archivos JavaScript de Leaflet Draw, utilizada para dibujar en el mapa.
leaflet-measure	Hoja de estilos CSS y archivos JavaScript para la extensión Leaflet Measure, utilizada para medir distancias y áreas en el mapa.
control-credits	Hoja de estilos CSS y archivos JavaScript para el control de créditos de Leaflet.
control-minimap	Hoja de estilos CSS y archivos JavaScript para el control Mini-Map de Leaflet.
Control.FullScreen	Hoja de estilos CSS y archivos JavaScript para el control FullScreen de Leaflet.
Leaflet.BigImage	Hoja de estilos CSS y archivos JavaScript para el control BigImage de Leaflet.

Tabla 4.1: Extensiones utilizadas de Leaflet

1. **Control del zoom:** Herramienta para navegar por los niveles de zoom de las placas virtuales, esta función se incluye por defecto en Leaflet.
2. **Información de medición:** Una de las funciones distintivas en una plataforma para gestionar placas virtuales de microscopía es la capacidad de realizar mediciones de manera precisa. La plataforma habilita una función de dibujo que despliega una ventana con información sobre la longitud y el área de la figura descrita por el dibujo. Para lograrlo, es crucial considerar la proyección de Mercator web, que distorsiona las medidas debido a la alta escala de la representación de la superficie terrestre y que es una superficie esférica proyectada en un cilindro. Con el fin de garantizar mediciones precisas, se emplea una extensión de Leaflet y se configura una nueva unidad para adaptar las escalas. Esto permite que medidas de varios kilómetros se ajusten, según los meta datos de la placa, a unos pocos milímetros. Además, para mitigar el impacto de la distorsión, se tiene en cuenta que a menor escala y cerca del ecuador, la distorsión es menor. Por lo tanto, se calcula el nivel mínimo de zoom necesario para que el error causado por la distorsión sea menor que un píxel en la resolución total de la placa, haciéndolo despreciable, como se observa en la ecuación 4.3.
3. **Marcaciones personalizadas:** El usuario puede destacar regiones de interés en las placas virtuales con diversas opciones de herramientas de dibujo.
4. **Control de descarga de imágenes:** Una de las extensiones instaladas permite descargar imágenes de lo que se visualiza en el microscopio en conjunto de todas las anotaciones que están desplegadas en este. Esta opción permite escoger el nivel de zoom de la captura, pero manteniendo la resolución del zoom actual.

5. **Barra de herramientas:** Esta es la barra de herramientas de dibujo que habilita una de las extensiones de Leaflet, permite dibujar líneas y polígonos libres, círculos, cuadrados, marcas y marcas circulares, además de una herramienta de edición de estos.
6. **Botón de retorno:** Permite volver al catálogo.

Implementación de la vista y plantilla para la subir archivos de placas virtuales a la plataforma

Para mejorar la versatilidad de la plataforma, se habilita una opción para agregar nuevas placas virtuales, pero solo se permite a los administradores del sistema. Esta funcionalidad simplifica las tareas de mantenimiento de la plataforma, ya que desplegar las placas virtuales en el MV de la página requiere crear una estructura de directorios con teselas por nivel de zoom, un proceso que puede llevar mucho tiempo debido al gran tamaño de los archivos. La plataforma es capaz de realizar esta tarea de forma automática, permitiendo que se realice de manera asíncrona y simultánea sin interrumpir el funcionamiento de la página. De esta manera, el administrador puede subir varias placas e iniciar el proceso de creación de una gran lista de preparados para el microscopio, y el servidor se encargará de procesarlos sin problemas. En la Figura 4.4 se muestra el menú de carga de los archivos, la lista de archivos subidos y algunas opciones, junto con cinco etiquetas que resaltan elementos importantes de la interfaz. Esta plantilla es accesible después de seleccionar la opción de editar catálogo en el menú que se habilita al iniciar sesión como administrador.

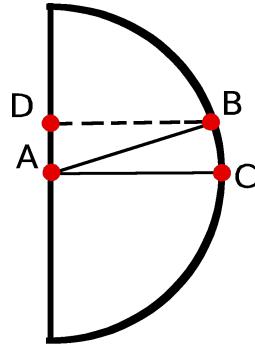


Figura 4.3: Esquema de líneas radiales de la Tierra. Elaboración propia

La plataforma carga la lámina virtual utilizando una vista que agrega la tarea a la cola de Celery y la ejecuta mediante un programa implementado en un archivo denominado task.py. El algoritmo de este programa consiste en recorrer los diferentes niveles de la lámina virtual, guardando las imágenes con nombres basados en el número de tesela en la fila, y almacenándolas en una carpeta según el número de fila. Posteriormente, todas las filas se guardan en una carpeta con un nombre determinado por el nivel de zoom. Es importante tener en cuenta que la escala puede variar de un nivel de zoom a otro, por lo que se redefine el nivel de zoom cero como el máximo nivel en el que toda la preparación se puede representar en una sola tesela de 256 por 256 píxeles. Este cálculo se ilustra en la ecuación 4.2. Además, para evitar que la distorsión causada por la proyección Mercator afecte a las mediciones, es

necesario calcular un nivel mínimo de zoom en el cual esta distorsión sea despreciable. Es decir, cuando la biblioteca Leaflet interpreta un zoom cero, una tesela representa la totalidad de la superficie terrestre, por lo que es preferible trabajar en latitudes cercanas al ecuador.

$$\text{minZoom} = \lceil \log_2 \left(\frac{90}{90 - \arcsin \left(1 - \frac{1}{\max(\text{max_side})} \right)} \right) \rceil \quad (4.1)$$

$$\text{base_level} = \lceil \log_2 \left(\frac{\max(\text{slide_width}, \text{slide_height})}{\text{tile_size}} \right) \rceil \quad (4.2)$$

$$\text{factor} = \frac{(\text{tile_size} \times 2^{\text{minZoom} + \text{base_level}})}{C} \times \frac{\text{mpp_x} + \text{mpp_y}}{2} \quad (4.3)$$

Donde:

- **base_level**: La cantidad de niveles de zoom entre el máximo nivel de zoom en el que la totalidad de la placa está contenida en una sola tesela y el nivel máximo de zoom disponible en la placade la placa.
- **minZoom**: El nivel de zoom mínimo en el que la distorsión por la proyección Mercator es despreciable.
- **mpp_x** y **mpp_y**: Los micrómetros por píxel en las direcciones horizontal y vertical, respectivamente.
- **tile_size**: El tamaño en píxeles de un lado de tesela.
- **slide_width** y **slide_height**: Las dimensiones totales en píxeles de la placa virtual.
- **factor**: El factor de conversión entre los metros proporcionados por las herramientas de dibujo y el tamaño real en el contexto de la lámina.

El cálculo del nivel de zoom mínimo en el cual una tesela será considerada como base implica determinar un nivel donde la diferencia de medida entre los extremos sea inferior al tamaño de un píxel, proporcionando información aceptable con un error menor al dado por la resolución de la imagen. En la figura 4.3 se muestra un diagrama de la Tierra. Si asumimos que en niveles muy cercanos al ecuador la distorsión en la coordenada x es mayor que la distorsión en la coordenada y, podemos observar que, dado dos paralelos, uno en el ecuador y otro que pase por el punto B, el error sería la diferencia entre las longitudes de los paralelos. Para ello, es necesario conocer la longitud en el ecuador (40075000 m) y la longitud en el nuevo paralelo. La longitud del nuevo paralelo se puede calcular con trigonometría básica teniendo en cuenta que el nuevo radio de la circunferencia del paralelo buscado es el marcado con la línea punteada DB. El resultado final de la fórmula se muestra en la ecuación 4.1. Finalmente, para calcular el factor de escala entre la unidad de medida utilizada por la herramienta y la medida real, se debe calcular el factor de píxeles por metro utilizando los datos obtenidos, y multiplicarlo por el factor de micrómetros por píxel que se obtiene de los meta datos. Este proceso se detalla en la ecuación 4.3.

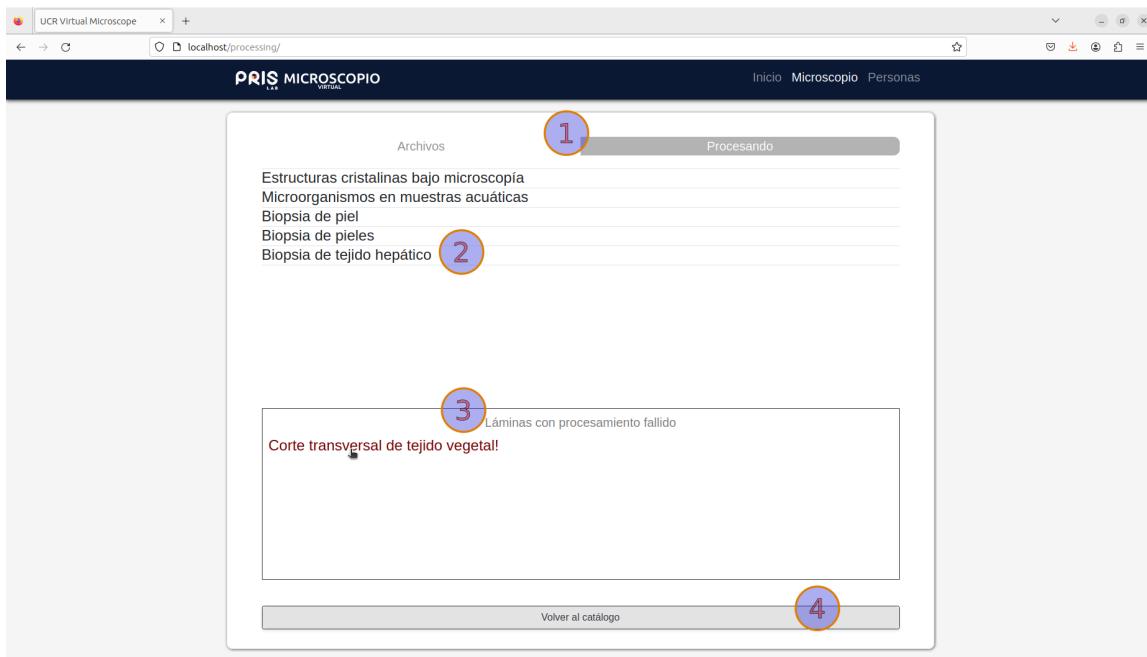


Figura 4.4: Resultado de menú de nuevas placas. Elaboración propia

1. **Barra de selección:** Este elemento consiste en una barra y un menú que permite alternar entre la lista que gestiona las placas virtuales subidas y otra lista que muestra las láminas en proceso de procesamiento.
2. **Lista de archivos subidos:** Aquí se muestran las instancias de OpenSlide que aún no han iniciado el procesamiento. Para esto, la plantilla utiliza un filtro implementado en su vista correspondiente que proporciona esta lista.
3. **Opciones para crear una nueva placa y eliminar archivo:** Estas opciones permiten eliminar el archivo subido o iniciar el proceso de creación del nuevo elemento para el catálogo.
4. **Zona para subir nuevos archivos:** Se utiliza la extensión de JavaScript Dropzone para mostrar un panel donde se pueden cargar archivos y agregarlos a la lista.
5. **Botón de retorno:** Este botón permite regresar al catálogo.

Implementación de la vista y plantilla para gestionar el procesamiento de placas virtuales nuevas

En el caso de la lista de placas virtuales en proceso, es importante poder distinguir entre aquellas que aún están en proceso y las que han presentado errores. Esta interfaz muestra estas dos listas. Una vez que una placa ha terminado de ser procesada, se agregará automáticamente al catálogo para su uso.

En la Figura 4.5 se muestra el menú de carga de los archivos, la lista de archivos subidos y algunas opciones, junto con cuatro etiquetas que resaltan elementos importantes de la interfaz.

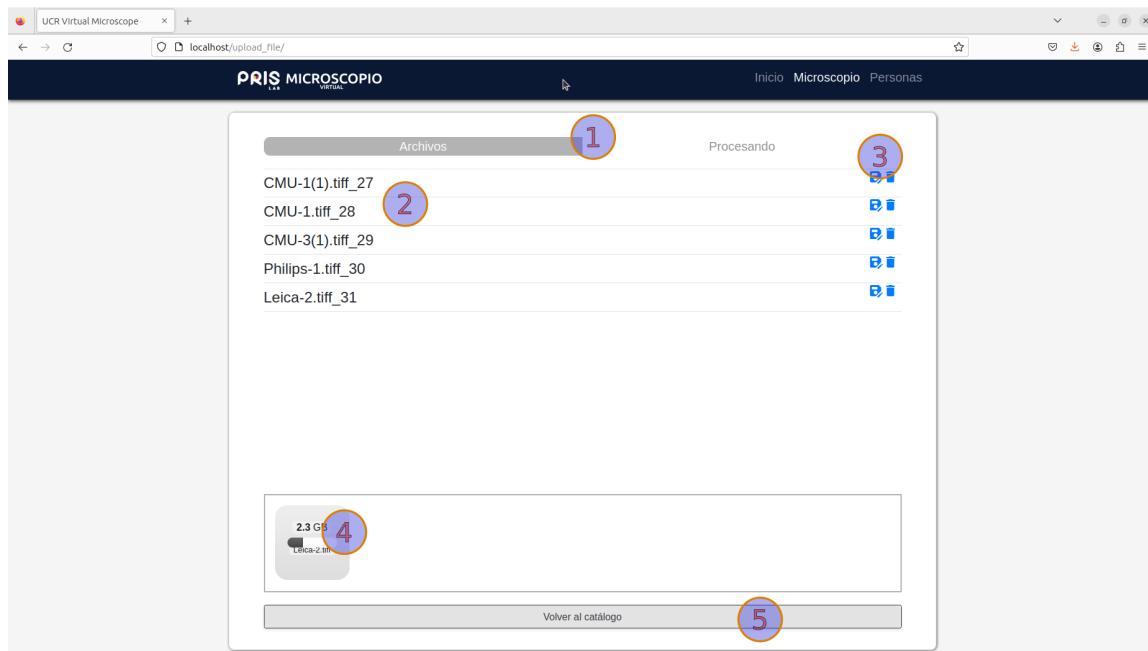


Figura 4.5: Resultado de menú de gestión de procesamiento de placas. Elaboración propia

1. **Barra de selección:** Misma función que en el menú de carga de archivos.
2. **Lista de placas virtuales en procesamiento:** Despliega todas las placas que con las que el sistema está trabajando para crear un nuevo elemento del catálogo.
3. **Lista de placas virtuales con procesamiento fallido:** Muestra los archivos subidos que no eran placas virtuales, y habilita su fácil eliminación.
4. **Botón de retorno:** Botón para volver al catálogo.

4.1.2. Administración del Sitio web y perfiles de usuario

La implementación de la aplicación de Django con el nombre **website_management** se encarga de gestionar todo lo relacionado con los usuarios, sus perfiles, la gestión de la plantilla genérica y la plantilla de la página de Inicio.

Implementación del archivo **models.py** de la aplicación

El archivo **models.py** se crea a partir del diseño de la tabla de la base de datos correspondiente a **User-Profile** y su interacción con **User**. En el código 4.3, se representa la tabla con una clase y sus atributos

son los campos, donde se detalla el tipo de dato, su extensión y otras configuraciones. También se detallan dos funciones con un decorador, los cuales permiten vincular la tabla por defecto de User a UserProfile.

```

1 from django.contrib.auth.models import User
2 from django.db import models
3 from django.db.models.signals import post_save
4 from django.dispatch import receiver
5
6 class UserProfile(models.Model):
7     user = models.OneToOneField(User, on_delete=models.CASCADE)
8     profile_image = models.ImageField(upload_to='profile_images/', default='
profile_images/default_profile_image.png')
9
10    def __str__(self):
11        return self.user.username
12
13 @receiver(post_save, sender=User)
14 def create_user_profile(sender, instance, created, **kwargs):
15     if created:
16         UserProfile.objects.create(user=instance)
17
18 @receiver(post_save, sender=User)
19 def save_user_profile(sender, instance, **kwargs):
20     instance.userprofile.save()

```

Listing 4.3: Código del archivo models.py de website_management

Implementación del archivo urls.py de la aplicación

Para definir el archivo urls.py, se han asociado URLs a los elementos de la interfaz diseñados en los wireframes presentados en las imágenes 3.7, 3.8 y 3.2. Además, se han asignado URLs a las diferentes solicitudes de la página hacia el servidor, relacionadas con la gestión de usuarios. El código 4.4 muestra la implementación de este archivo.

```

1 from django.urls import path
2 from django.conf import settings
3 from django.conf.urls.static import static
4 from django.contrib.auth.decorators import login_required
5 from website_management.views import home, Login, logoutUsuario, SignUp,
edit_profile, edit_password
6
7 urlpatterns = [
8     path('', home, name = "Inicio"),
9     path('accounts/login/', Login.as_view(), name = "Login"),
10    path('logout/', login_required(logoutUsuario), name = "Logout"),
11    path('signup/', SignUp.as_view(), name = "SignUp"),
12    path('edit_profile/', edit_profile, name='edit_profile'),
13    path('edit_password/', edit_password, name='password'),
14 ]

```

```
15
16 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Listing 4.4: Código del archivo urls.py de website_management

Implementación de la vista y plantilla para la página de inicio

Se desarrollaron las plantillas genérica y de la página de inicio para representar la primera interacción del usuario con la plataforma web. Estas plantillas proporcionan los elementos necesarios para guiar al usuario en su experiencia inicial. En la figura 4.6 se muestra el resultado de la implementación. El diseño incluye una barra de menú con tres elementos principales: **Inicio**, que permite regresar a la visualización inicial; **Microscopio**, que proporciona acceso al catálogo público; y **Iniciar sesión**, que permite a los usuarios iniciar sesión en la plataforma. La página de inicio presenta inicialmente un área destacada para dar la bienvenida al usuario, seguido de un video que introduce las diferentes funciones de la plataforma.



Figura 4.6: Resultado de la página de inicio. Elaboración propia

Implementación de la vista y plantilla para iniciar sesión

Se implementó una vista y plantilla dedicada al inicio de sesión de los usuarios utilizando las herramientas ofrecidas por el framework Django y el modelo por defecto de User. En la figura 4.7 se muestra el resultado y se destacan cuatro elementos importantes.

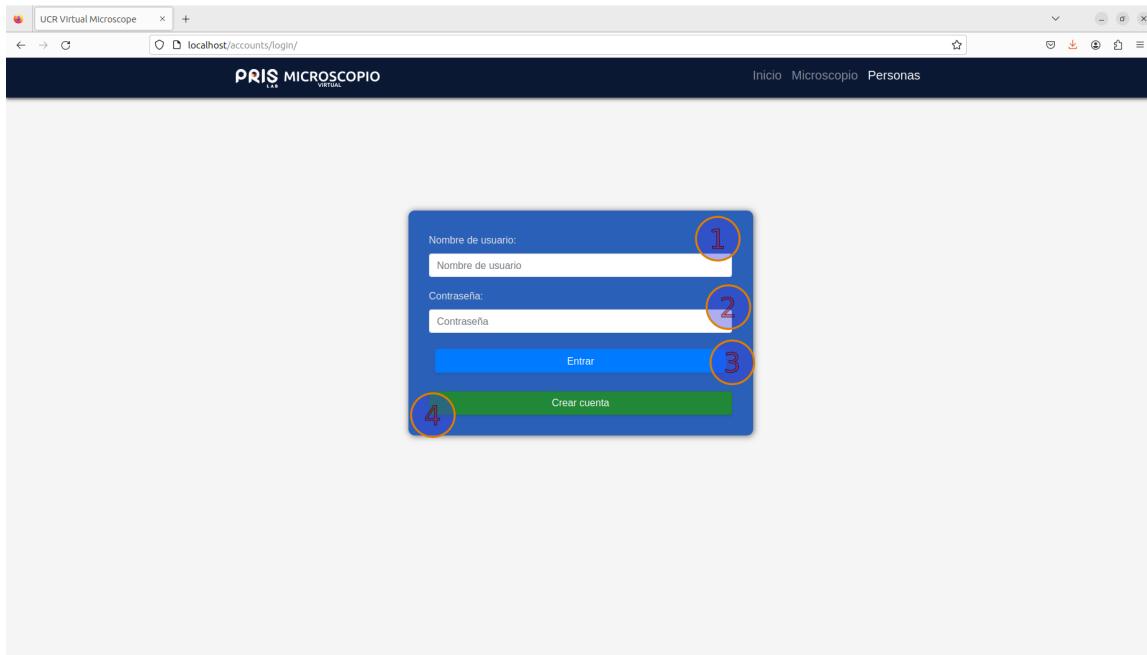


Figura 4.7: Resultado de formulario para inicio de sesión. Elaboración propia

1. **Campo de nombre:** En esta región del formulario se escribe el nombre de usuario.
2. **Campo de contraseña:** En esta región del formulario se escribe la contraseña.
3. **Botón de inicio de sesión:** Inicia sesión con los datos del formulario, o indica si hay un error con las credenciales.
4. **Botón de crear cuenta:** Redirige al usuario a un formulario de nueva cuenta.

Implementación de la vista y plantilla para crear nuevos usuarios

Se implementó una vista y plantilla dedicada a crear la cuenta de un nuevo usuarios utilizando las herramientas ofrecidas por el framework Django y el modelo por defecto de User, además se crea una instancia del modelo UserProfil y se asocia a User. En la figura 4.8 se muestra el resultado y se destacan cuatro elementos importantes.

- 1.
2. **Formulario de datos:** En esta región del formulario se escribe el nombre de usuario, el nombre completo y el correo.
3. **Campo de contraseña:** En esta región del formulario se escribe la contraseña, y una confirmación de esta.

The screenshot shows a web page titled "PRIS MICROSCOPIO VIRTUAL". At the top, there is a navigation bar with links for "Inicio", "Microscopio", and "Personas". Below the navigation bar is a search bar with placeholder text "Nombre:" and "Apellido:". Further down, there are fields for "Dirección de correo electrónico:", "Nombre de usuario:" (with a note: "Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @/./-/_.") and "Contraseña:". Below the password field is a list of validation rules: "Su contraseña no puede asemejarse tanto a su información personal.", "Su contraseña debe contener al menos 8 caracteres.", "Su contraseña no puede ser una clave utilizada comúnmente.", and "Su contraseña no puede ser completamente numérica.". There is also a "Contraseña (confirmación):" field with a note: "Para verificar, introduzca la misma contraseña anterior." At the bottom of the form are two buttons: a green "Regístrate" button and a blue "Iniciar Sesión" button.

Figura 4.8: Resultado de formulario para crear un nuevo usuario. Elaboración propia

4. **Botón de crear cuenta:** Crea una cuenta con los datos del formulario, o indica si hay un error con la información.
5. **Botón de volver al inicio de sesión:** Redirige al usuario a un formulario de iniciar sesión.

4.1.3. Gestión de proyectos

La implementación de la aplicación de Django con el nombre **projects** se encarga de gestionar todo lo relacionado con los proyectos y las instancias de las placas virtuales.

Implementación del archivo `models.py` de la aplicación

El archivo `models.py` se crea a partir del diseño de las tablas de la base de datos correspondientes a **Project**, **ProjectSlides**, **Message** y **Notes**. En el código 4.5, se representa la tabla con una clase y sus atributos son los campos, donde se detalla el tipo de dato, su extensión y otras configuraciones. También se detallan funciones para retornar su nombre o una URL.

```

1 import os
2 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'virtual_microscope.settings')
3 from microscope.models import Slide
4
5 from django.contrib.postgres.fields import ArrayField
6 from django.urls import reverse

```

```
7 from jsonfield import JSONField
8 from django.db import models
9 from django.contrib.auth.models import User
10
11 # Create your models here.
12 class Project(models.Model):
13     name = models.CharField(max_length=25)
14     description = models.CharField(max_length=500)
15
16     user = models.ForeignKey(User, on_delete=models.SET_NULL, null=True)
17     adminUsers = models.ManyToManyField(User, related_name='adminUsers')
18     sharedUsers = models.ManyToManyField(User, related_name='sharedUsers')
19     invitedUsers = models.ManyToManyField(User, related_name='invitedUsers')
20
21     def __str__(self):
22         return self.name
23
24     def get_absolute_url(self):
25         return reverse('project-detail', args=[str(self.id), str(0), str(0)])
26
27 class ProjectSlide(models.Model):
28     name = models.CharField(max_length=25)
29     description = models.CharField(max_length=500)
30     adminPermission = models.BooleanField(default=False)
31
32
33     slide = models.ForeignKey(Slide, on_delete=models.SET_NULL, null=True)
34     project = models.ForeignKey(Project, on_delete=models.SET_NULL, null=True)
35
36     def __str__(self):
37         return self.name
38
39     def get_absolute_url(self):
40         return reverse('project-slide-detail', args=[str(self.id)])
41
42
43 class Notes(models.Model):
44     name = models.CharField(max_length=50)
45
46     description = models.TextField(max_length=500)
47     geojson_data = models.TextField(null=True, blank=True)
48     show = models.BooleanField()
49
50     xUnoPos = models.FloatField()
51     yUnoPos = models.FloatField()
52     xDosPos = models.FloatField()
53     yDosPos = models.FloatField()
54
55     project = models.ForeignKey(ProjectSlide, on_delete=models.CASCADE, null=True)
56
```

```

57     def __str__(self):
58         return self.name
59
60     def get_absolute_url(self):
61         return reverse('note-detail', args=[str(self.id), str(self.project.id)])
62
63 class Message(models.Model):
64     user = models.ForeignKey(User, on_delete=models.CASCADE)
65     project = models.ForeignKey(Project, on_delete=models.CASCADE)
66     contenido = models.TextField()
67     fecha_envio = models.DateTimeField(auto_now_add=True)
68
69     def __str__(self):
70         return f'{self.user}: {self.contenido}'

```

Listing 4.5: Código del archivo models de projects

Implementación del archivo urls.py de la aplicación

Para definir el archivo urls.py y asociar las URLs con las vistas del proyecto, se partió de los wireframes presentados en las figuras 3.14, 3.13, 3.10, 3.9, 3.11 y 3.12, así como en los requerimientos y solicitudes de la página hacia el servidor. El código 4.6 muestra la implementación de este archivo. Estas URLs están asociadas al manejo de los proyectos, y para acceder a ellos es necesario iniciar sesión en la plataforma.

```

1 from django.urls import path, re_path
2 from django.contrib.auth.decorators import login_required
3 from django.conf import settings
4 from django.conf.urls.static import static
5 from projects.views import *
6
7 urlpatterns = [
8     path('project/', login_required(projects.as_view()), name='project-list'),
9     path('project2/<int:slideId>', login_required(projects.as_view()), name='
10         project-list-dos'),
11     path('projectDetail/<int:pk>/<int:option>/<int:map>', login_required(
12         projectDetail.as_view()), name='project-detail'),
13     path('projectProfileDetail/<int:pk>/<int:option>/<int:map>', login_required(
14         projectProfileDetail.as_view()), name='project-profile-detail'),
15     path('projectSlideDetail/<int:pk>/', login_required(projectSlideDetail.
16         as_view()), name='project-slide-detail'),
17     path('noteDetail/<int:pk>/<int:project_id>', login_required(noteDetail.
18         as_view()), name='note-detail'),
19     path('deleteNota/<int:notas_id>', deleteNota),
20     path('deleteProject/<int:project_id>', deleteProject),
21     path('deleteProjectSlide/<int:project_id>', deleteProjectSlide),
22     path('showNote/<int:note_id>/<int:project_id>', showNote),
23     path('newProjectUser/<int:project_id>/<int:projectUser_id>', newProjectUser)
24     ,
25     path('deleteProjectUser/<int:project_id>/<int:projectUser_id>', deleteProjectUser),
26 ]

```

```

20     path('newProjectInvitedUser/<int:project_id>/<int:projectUser_id>',  

21         newProjectInvitedUser),  

22     path('deleteProjectInvitedUser/<int:project_id>/<int:projectUser_id>',  

23         deleteProjectInvitedUser),  

24     path('obtener_lista/', obtener_lista, name='obtener_lista'),  

25     path('datos_actualizados_catalogo/<int:project>/<str:buscar>',  

26         datos_actualizados_catalogo, name='datos_actualizados_catalogo'),  

27     path('datos_actualizados_placas/<int:project>/<str:buscar>',  

28         datos_actualizados_placas, name='datos_actualizados_placas'),  

29     path('datos_actualizados_colaboradores/<int:project>/<str:buscar>',  

30         datos_actualizados_colaboradores, name='datos_actualizados_colaboradores'),  

31     path('datos_actualizados_chat/<int:project>/<str:buscar>',  

32         datos_actualizados_chat, name='datos_actualizados_chat'),  

33 ]

```

Listing 4.6: Código del archivo urls.py de projects

Implementación de la vista y plantilla para administrar proyectos

La figura 4.9 muestra la interfaz principal de la sección de personas de la plataforma, junto con siete etiquetas que resaltan los principales elementos del diseño. Para acceder a esta sección, es necesario seleccionar la opción **Personas** en la barra de menú e iniciar sesión. Este diseño sigue las pautas propuestas en el wireframe de la figura 3.9.

1. **Panel de gestión de proyecto:** Este panel muestra diferentes opciones para manipular el perfil de usuario, crear y gestionar los proyectos.
2. **Perfil de usuario:** Muestra información básica del usuario, como una imagen de perfil y el nombre completo.
3. **Opciones de sesión y perfil:** Este menú dispone de dos botones, uno para editar la información del usuario y otro para cerrar sesión.
4. **Barra de búsqueda:** Permite filtrar los proyectos mostrados, apoyándose en un formulario simple y la vista correspondiente.
5. **Nuevo proyecto:** Es un formulario sencillo que crea nuevos elementos de proyecto.
6. **Lista de proyectos:** Es la sección principal que despliega una lista con los proyectos del usuario. Se agregaron los estilos necesarios en la plantilla para lograr un diseño interactivo, y se gestiona la base de datos mediante las vistas.
7. **Opciones de proyecto:** Pequeño menú para editar datos generales del proyecto o eliminarlo.

Figura 4.9: Resultado de la implementación de la interfaz administradora de perfil y proyectos. Elaboración propia

Implementación de la vista y plantilla para invitar colaboradores

Al seleccionar una opción de la lista de proyectos, se accede a un área de trabajo como la diseñada en el wireframe de la figura 3.10. En la figura 4.10, se muestra una configuración de esta área que se habilita al seleccionar la opción de invitar colaborador del panel de herramientas derecho, descrito en el wireframe de la figura 3.13. Se destacan cuatro elementos principales de la interfaz gráfica:

- 1. Datos del proyecto:** Se muestran los datos principales del proyecto, como quién es el administrador, el propietario de este, los colaboradores, la cantidad de láminas virtuales, el nombre del proyecto y una descripción.
- 2. Título del panel de herramientas:** Cumple dos funciones: indica cuál menú de opciones está activo y permite cambiar entre menús. El panel de herramientas se puede actualizar sin recargar toda la página para una gestión más fluida. Se implementó en la plantilla del proyecto utilizando archivos de JavaScript, estilos CSS y HTML. Además, se agregaron las URLs necesarias para realizar las peticiones al servidor, como se muestra en el código 4.6.
- 3. Barra de búsqueda:** Permite buscar por nombre y nombre de usuario para enviar una invitación para colaborar en el proyecto.

4. **Lista de usuarios:** Muestra una lista de usuarios buscados por nombre y permite la opción de enviar una invitación. La vista que maneja esta plantilla se encarga de suministrar la lista según el filtro proporcionado.

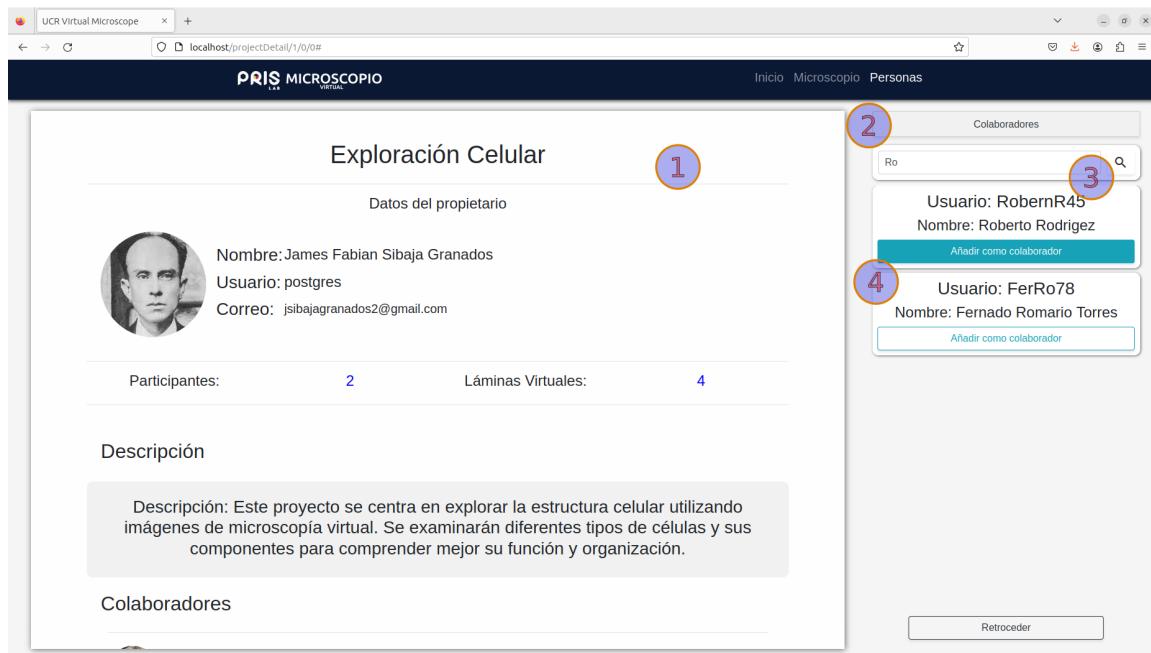


Figura 4.10: Resultado del área de trabajo de proyecto para la invitación de nuevos colaboradores. Elaboración propia

Implementación de la vista y plantilla para catálogo de proyecto

En el área de trabajo diseñada en el wireframe de la figura 3.10, se requiere la opción de crear nuevas instancias personalizables de láminas virtuales. Para esto, es necesario el uso de un catálogo incrustado en esta interfaz. En la figura 4.11, se muestra el catálogo en el panel de herramientas derecho, descrito en el wireframe de la figura 3.13. Además, se muestran otros elementos para facilitar la tarea de crear nuevas placas. Se destacan cinco elementos principales de la interfaz gráfica:

- Visualizador de microscopio:** En el área principal de la interfaz se muestra un microscopio para visualizar los elementos del catálogo. Este hereda las funciones de la plantilla del catálogo público principal.
- Datos de la lámina virtual:** Muestra el nombre y una descripción de las placas del catálogo. Esta información la proporciona la vista a partir del ID del modelo de Slide.

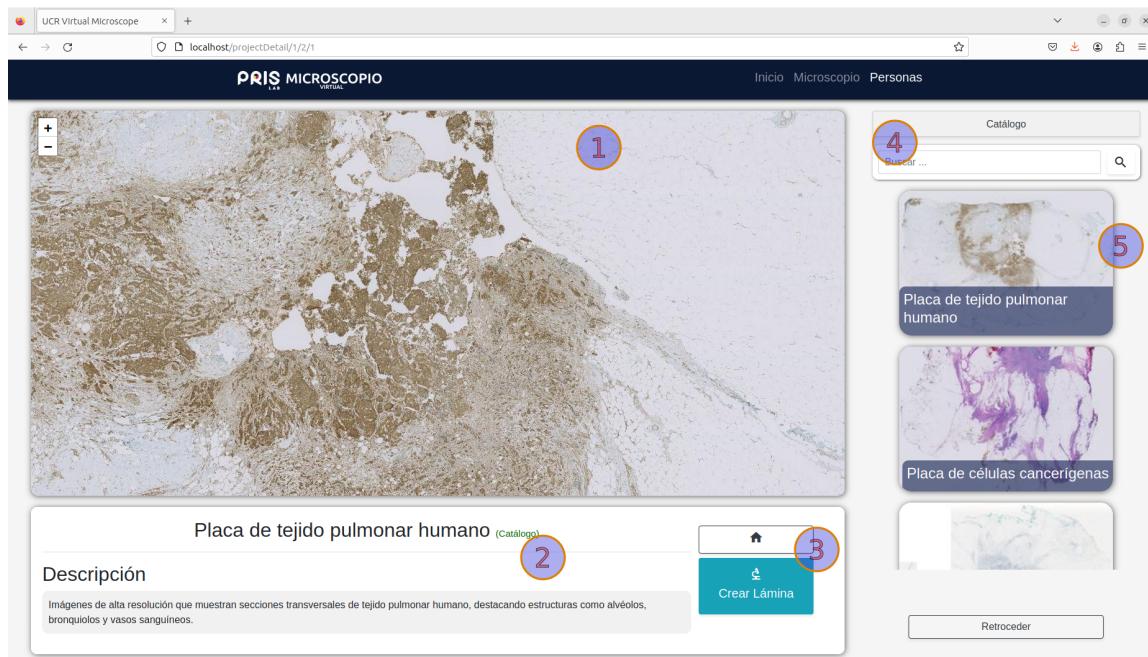


Figura 4.11: Resultado de implementación de catalogó para proyectos. Elaboración propia

3. **Opciones de placa:** Dos opciones, una para volver a la página de datos del proyecto y otra para crear una nueva instancia a partir de la placa seleccionada.
4. **Barra de búsqueda:** Permite filtrar resultados en el catálogo.
5. **Catálogo incrustado:** Lista con los elementos del catálogo que se pueden seleccionar para crear instancias, hereda sus funciones del catálogo público principal.

Implementación de la vista y plantilla para gestión de placas en proyecto

En el área de trabajo diseñada en el wireframe de la figura 3.10, se requiere la opción de administrar las instancias personalizables de láminas virtuales. En la figura 4.12, se muestra una lista con las instancias de placas virtuales, ubicado en el panel de herramientas derecho, descrito en el wireframe de la figura 3.13. Además, se muestran otros elementos para facilitar el manejo de estas instancias. Se destacan cinco elementos principales de la interfaz gráfica:

1. **Visualizador de microscopio:** En el área principal de la interfaz se muestra un microscopio para visualizar la placa personalizada seleccionada.
2. **Datos de la lámina virtual:** Muestra el nombre y una descripción de la placa virtual. Esta información la proporciona la vista a partir del ID del modelo ProjectSlide.

3. **Opciones de edición de lámina virtual:** Cuatro opciones: una para volver a la página de datos del proyecto, la segunda para abrir la instancia e iniciar la personalización, la tercera para editar los datos generales de la placa y la última para eliminarla.
4. **Barra de búsqueda:** Permite filtrar la lista de instancias de láminas virtuales guardadas.
5. **Lista de láminas virtuales:** Despliega todas las láminas virtuales guardadas en el proyecto y habilita la selección para editar tanto en el panel principal como en el de apoyo descritos anteriormente.

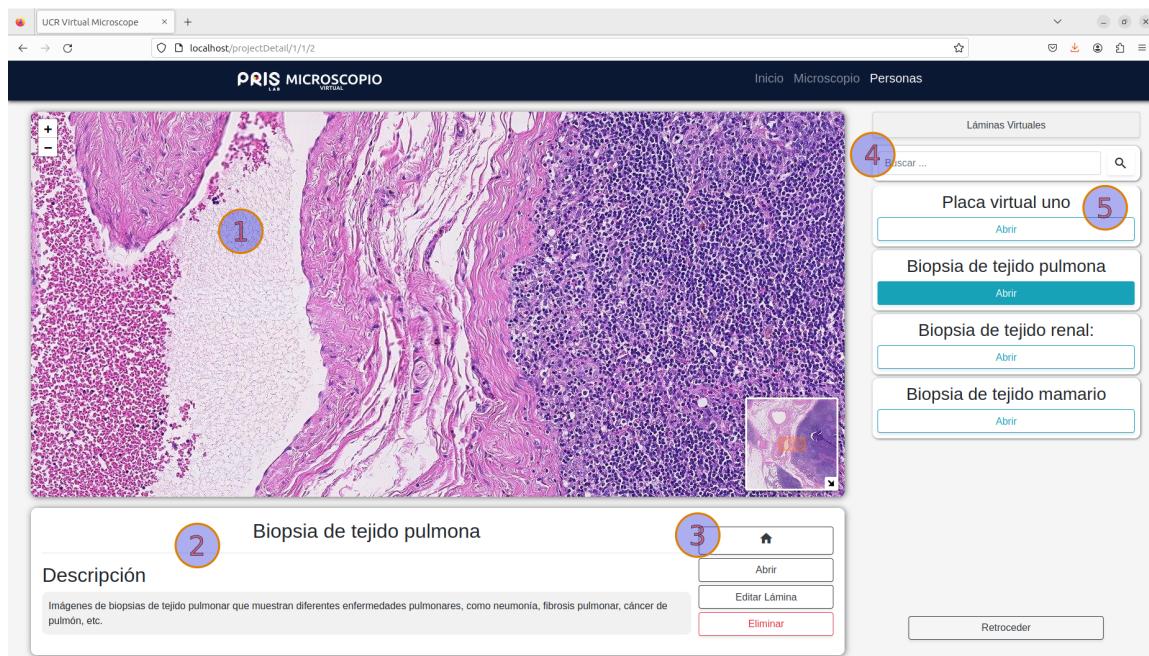


Figura 4.12: Resultado área de trabajo para le manejo de instancias de placas virtuales. Elaboración propia

Implementación de la vista y plantilla para chat de proyecto

Una de las opciones que ofrece el área de trabajo para proyectos es un chat en tiempo real para facilitar la comunicación entre los colaboradores. Este chat se despliega en el panel de herramientas derecho, descrito en el wireframe de la figura 3.13. En la figura 4.13 se muestra el chat en funcionamiento. Para la implementación, se hizo uso de WebSockets en la plantilla, la biblioteca Django Channels para dar soporte a los WebSockets y la comunicación asíncrona, el broker Redis para facilitar la comunicación entre los procesos y actuar como intermediario entre los diferentes consumidores de canales en tiempo real, y Daphne como servidor HTTP y WebSocket para proporcionar una interfaz ASGI para Django Channels. Se destacan cinco elementos de la interfaz gráfica.

Aquí tienes una versión corregida del texto:

- Datos de colaboradores:** La lista muestra los colaboradores del proyecto que pueden participar en el chat.
- Título del panel de herramientas:** Muestra el título del menú que se usa en el panel de herramientas, en este caso, chat. Permite cambiar de menú.
- Barra de búsqueda:** Permite filtrar los mensajes del chat para buscar mensajes anteriores de forma sencilla.
- Chat de proyecto:** Se despliegan las instancias del modelo Message asociadas al proyecto. La organización de los mensajes, así como la información que se muestra, cumple con las características básicas de un servicio de mensajería actual.
- Nuevo mensaje:** En esta área se escribe un nuevo mensaje y, con el botón de enviar, se muestra en tiempo real a todos los miembros la información.

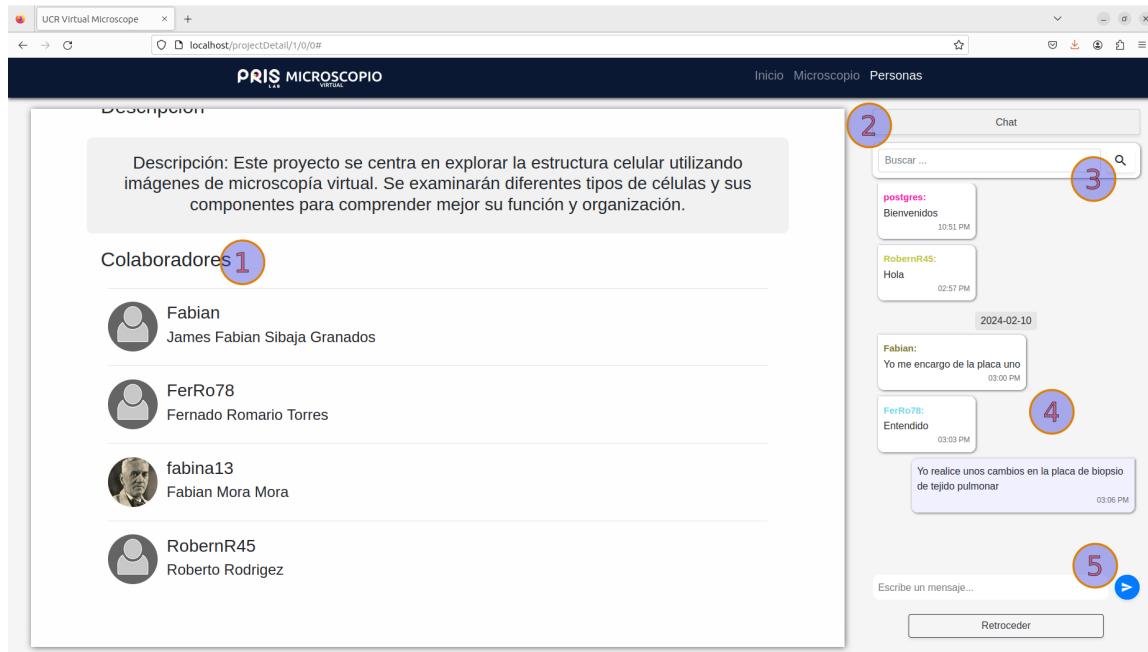


Figura 4.13: Resultado de la implementación de un chat para el proyecto. Elaboración propia

Implementación de la vista y plantilla para gestión de invitaciones

La plataforma permite enviar solicitudes para colaborar en los proyectos, por lo que es necesario administrar estas invitaciones y contar con una interfaz para esto. Para ello se incluyó una sección en el el

área de gestión de proyectos. El resultado de esta implementación se muestra en la figura 4.14, en esta figura se destacan tres elementos de la interfaz. Este funcionamiento se describe en el UC descrito en la tabla 3.12.

1. **Panel de apoyo:** Descrito anteriormente, permite filtrar las solicitudes.
2. **Lista de invitaciones:** Se muestra una lista con las invitaciones recibidas, esta lista se ubica en el mismo área de las lista de proyectos pero con la indicación de que es una invitación y se marca con signo de exclamación para destacarlo, también es importante mencionar que siempre se desplegará de primero en la lista. Al seleccionar la invitación se puede acceder a una vista previa del proyecto, para obtener la información necesaria antes de aceptar o rechazar la invitación.
3. **Opciones de invitación:** Permite aceptar o rechazar la invitación.

Figura 4.14: Resultado del despliegue de las invitaciones en el área de gestión de proyectos. Elaboración propia

Implementación de la vista y plantilla para vista previa de proyecto

Al seleccionar una invitación y haberla aceptado o rechazado, se accede a una vista previa del proyecto donde se puede visualizar la información de este, pero no editarla. Este comportamiento se describe en el UC detallado en la tabla 3.13. En la figura 4.15 se muestra el resultado de la implementación de esta sección, y se destacan tres elementos principales.

1. **Área de datos:** Muestra la información general del proyecto. Si se selecciona una placa virtual, esta sección se intercambia por un MV que permite ver las placas.
2. **Título de la barra:** A diferencia del área de trabajo para colaboradores, este título no se puede cambiar y solo permite acceder a la lista de láminas virtuales instaladas en el proyecto.
3. **Lista de láminas virtuales:** Muestra el conjunto de láminas personalizadas disponibles en el proyecto. Al seleccionar una opción, se puede visualizar el progreso de la instancia.

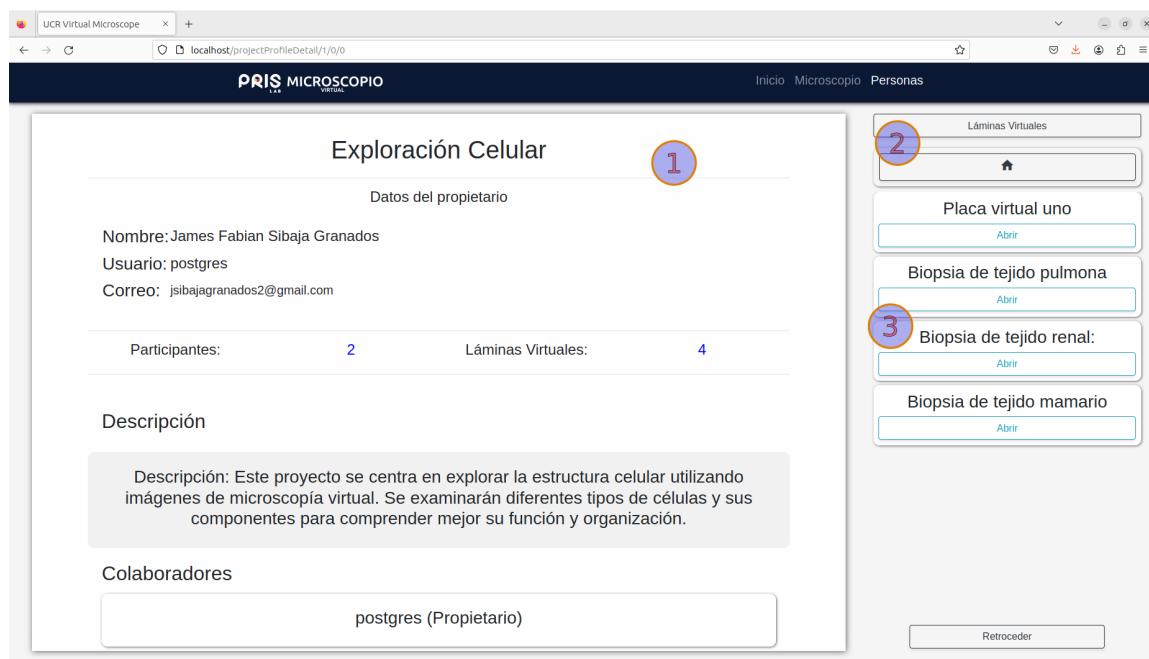


Figura 4.15: Resultado del la vista previa de un proyecto. Elaboración propia

Implementación de la vista y plantilla para edición de placas en proyectos

La principal idea detrás de implementar la gestión de cuentas de usuario y proyectos es poder seleccionar placas de interés, agruparlas, agregar marcaciones y anotaciones sobre observaciones, y compartir estos datos con otros usuarios. Las láminas del catálogo público son de uso general y no se pueden editar, por lo que es crucial contar con el modelo OpenSlide, que es el elemento central de los proyectos. Además, es necesario disponer de un entorno gráfico que permita editar las placas, crear anotaciones y guardar los cambios.

En la figura 4.16 se muestra la interfaz de edición de las placas virtuales, resaltando siete elementos principales:

1. **Mini mapa:** Esta herramienta, comúnmente utilizada en sistemas de mapas web y microscopía virtual, permite visualizar el contexto de la imagen mostrada, facilitando la navegación por la placa. Se implementa con una extensión de Leaflet y se hereda del microscopio principal.
2. **Menú de notas:** Muestra todas las notas asociadas a la placa virtual. A cada nota se le pueden asociar elementos de dibujo y se puede mostrar u ocultar las notas trabajando como capas. Las notas son elementos del modelo Notes y se utilizan para guardar marcaciones e información adicional sobre la placa, incluyendo posiciones.
3. **Opciones adicionales:** Se habilitan dos opciones adicionales a las mostradas en el microscopio público: la personalización del color para las anotaciones y el botón "Guardar", que permite guardar los cambios realizados en las notas.
4. **Barra de herramientas:** Proporciona opciones de dibujo y se hereda del microscopio público.
5. **Información de mediciones:** Cuadro que muestra la longitud o área de la región seleccionada. Se implementa con una extensión de Leaflet y hereda sus funciones del microscopio público.
6. **Elementos agregados:** Estos elementos se utilizan para marcar y señalar regiones de las placas y se almacenan en forma de GeoJSON.
7. **Botón de retorno:** Facilita la navegación por el preparado virtual.

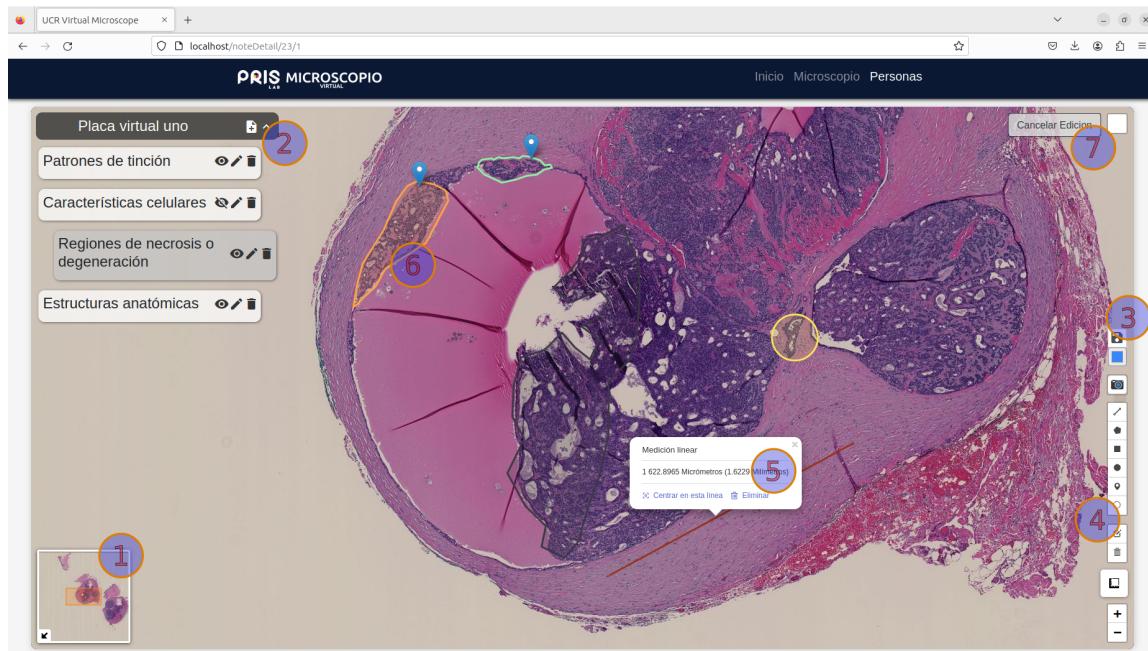


Figura 4.16: Resultado del área de edición de una lámina virtual de proyecto. Elaboración propia

4.1.4. Configuración y ajustes del entorno de la plataforma

La implementación de la aplicación de Django con el nombre **virtual_microscope** se encarga de gestionar todo lo relacionado con la configuración general del proyecto, ya que es la aplicación principal por defecto de la plataforma. Entre los principales archivos que administra se encuentra el archivo `settings.py`, que es importante porque configura el proyecto. En este archivo, podemos encontrar secciones importantes como la configuración de la base de datos, como se observa en el código 4.7, y la inclusión de las demás aplicaciones, como se observa en el código 4.8, entre otras configuraciones importantes. También administra los archivos `wsgi.py`, `asgi.py` y `celery.py`, que cumplen funciones importantes como exponer la aplicación Django al servidor web, comunicación asíncrona en tiempo real y ejecución de tareas en segundo plano, respectivamente.

```

1 DATABASES = {
2     'default': {
3         'ENGINE': 'django.db.backends.postgresql_psycopg2',
4         'NAME': 'microscopio',
5         'USER': 'postgres',
6         'PASSWORD': 'microVirtual',
7         'HOST': 'db',
8         'PORT': '5432',
9         'OPTIONS': {
10             'options': '-c timezone=UTC',
11         },
12     },
13 }
```

Listing 4.7: Código para configurar la base de datos en `setting.py`

```

1 INSTALLED_APPS = [
2     'microscope',
3     'website_management',
4     'projects',
5     'django.contrib.admin',
6     'django.contrib.auth',
7     'django.contrib.contenttypes',
8     'django.contrib.sessions',
9     'django.contrib.messages',
10    'channels',
11    'daphne',
12    'django.contrib.staticfiles',
13    'django_celery_results',
14 ]
```

Listing 4.8: Código para incluir las aplicaciones en `setting.py`

Implementación del archivo `urls.py` de la aplicación

En este directorio no se encuentra un archivo `models.py` ni plantillas, pero sí un archivo `urls.py`, el cual corresponde al archivo que configura las URLs de forma general. En el código 4.9 se muestra la implementación.

```

1 from django.contrib import admin
2 from django.urls import path, include
3 from django.conf.urls.static import static
4 from django.conf import settings
5
6 urlpatterns = [
7     path('admin/', admin.site.urls),
8     path('', include("website_management.urls")),
9     path('', include("microscope.urls")),
10    path('', include("projects.urls")),
11 ]
12
13 if settings.DEBUG:
14     urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
15     urlpatterns += static(settings.STATIC_URL, document_root=settings.
STATIC_ROOT)

```

Listing 4.9: Código del archivo urls.py de virtual_microscope

4.2. Encapsulado y automatización de la plataforma

La implementación del proyecto se optimizó para garantizar un diseño eficiente y consistente tanto en el entorno de desarrollo como en el de producción, además de simplificar el despliegue de la plataforma. Esto se logró mediante el encapsulamiento y la automatización del proyecto, aprovechando herramientas como makefile, git, dockerfile y docker-compose.

Para garantizar la escalabilidad del diseño, se integró Nginx al proyecto. Este servidor web de código abierto ofrece una serie de beneficios que se detallan en el marco teórico del proyecto. La configuración específica de Nginx se presenta en el código 4.10, que muestra el contenido del archivo nginx.conf.

Git se emplea en el proyecto como sistema de control de versiones, facilitando la gestión del código fuente de la aplicación, el seguimiento de cambios y la colaboración de forma efectiva. Tanto el repositorio del código fuente como los requisitos de la plataforma se detallan en el README adjunto en el apéndice A.

En conjunto, estas herramientas y configuraciones permiten una gestión eficiente del proyecto, facilitando su desarrollo, despliegue y mantenimiento a lo largo del ciclo de vida de la aplicación web. El despliegue del sistema encapsulado y automatizado se probó en el sistema operativo Ubuntu 23.04, con un resultado positivo. Todas las imágenes de la interfaz mostradas en el trabajo son ejemplos de la plataforma operando desde el sistema de contenedores después de ser ejecutados con el **Makefile**.

```

1 upstream django_app {
2     server gunicorn:8765;
3 }
4
5 upstream daphne_app {
6     server daphne:8089;
7 }
8
9 client_max_body_size 30G;

```

```

10 proxy_read_timeout 600s;
11
12 server {
13     listen 0.0.0.0:80;
14     server_name _;
15
16     location / {
17         proxy_pass http://django_app;
18         proxy_http_version 1.1;
19         proxy_set_header Upgrade $http_upgrade;
20         proxy_set_header Connection $http_upgrade;
21         proxy_set_header Host $host;
22         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
23         proxy_redirect off;
24     }
25
26     location /ws/ {
27         proxy_pass http://daphne_app;
28         proxy_http_version 1.1;
29         proxy_set_header Upgrade $http_upgrade;
30         proxy_set_header Connection "upgrade";
31         proxy_set_header Host $host;
32     }
33
34     location /static/ {
35         alias /app/static/;
36     }
37
38     location /media/ {
39         alias /app/media/;
40     }
41 }
```

Listing 4.10: Código del archivo docekr-compose.yml

4.2.1. Makefile

Se implementó un archivo Makefile para automatizar las tareas del proyecto, como la construcción de la aplicación, el despliegue y las tareas comunes de desarrollo. La tabla 4.2 detalla este archivo.

Objetivo	Descripción
setup	Configura el entorno de desarrollo. Instala las dependencias necesarias, crea un entorno virtual, construye los contenedores de Docker y realiza migraciones en la base de datos.
generate-docs	Genera la documentación del proyecto en formato HTML y la sirve en un servidor local. También puede generar un archivo PDF si se ejecuta después del objetivo <i>generate-pdf</i> .
generate-pdf	Genera un archivo PDF de la documentación del proyecto a partir de la versión HTML previamente generada.
run	Inicia el entorno de desarrollo. Levanta los contenedores de Docker y realiza migraciones en la base de datos.
stop	Detiene todos los contenedores de Docker relacionados con el proyecto.

Tabla 4.2: Descripción de los objetivos del Makefile.

4.2.2. Docker-compose

Con el objetivo de garantizar la escalabilidad, así como la portabilidad y el aislamiento, se implementaron contenedores Docker. La definición de Docker y sus funciones se detallan en la sección del marco teórico. Esto asegura un desarrollo robusto.

El archivo de configuración docker-compose.yml gestiona la configuración e intercomunicación de los contenedores Docker que componen la plataforma, incluidos los de la base de datos, el servidor web y otros servicios. Esta configuración simplifica el proceso de despliegue.

Servicio	Descripción
db	Contenedor de la base de datos PostgreSQL. Se utiliza la imagen ‘postgres:latest’ y se configura la base de datos, el usuario y la contraseña. Se expone en el puerto 5432 y se monta un volumen para persistir los datos.
daphne	Contenedor para el servidor Daphne. Construido a partir de un Dockerfile personalizado, expone el puerto 8089 y depende de los servicios db, redis y gunicorn.
gunicorn	Contenedor para el servidor Gunicorn. Construido a partir de un Dockerfile personalizado, expone el puerto 8765 y depende de los servicios db y redis.
celery	Contenedor para Celery. Construido a partir de un Dockerfile personalizado, depende de los servicios redis y gunicorn y utiliza un archivo de entorno ‘.env’.
redis	Contenedor para la base de datos Redis. Utiliza la imagen ‘redis:latest’ y expone el puerto 6379.
documentation	Contenedor para la generación de documentación. Construido a partir de un Dockerfile personalizado, ejecuta el comando ‘make -C docs html’ para generar la documentación en formato HTML.
nginx	Contenedor para el servidor Nginx. Utiliza la imagen ‘nginx:latest’, expone el puerto 80 y depende de los servicios daphne y gunicorn. Se monta un archivo de configuración ‘nginx.conf’ y el directorio de la aplicación.

Tabla 4.3: Descripción de los servicios del archivo docker-compose.

4.2.3. Dockerfile

El archivo de configuración Dockerfile se utilizó para definir la configuración de la aplicación. Permite encapsular todas las dependencias necesarias para el contenedor Docker, garantizando un entorno consistente y reproducible.

Archivo Dockerfile	Descripción
Dockerfile.daphne	Este archivo Dockerfile define un contenedor para el servidor Daphne. Utiliza una imagen base de Python 3.10.12 e instala las dependencias necesarias. Configura el directorio de trabajo, copia los archivos de la aplicación y ejecuta comandos para instalar las dependencias de Python, configurar las variables de entorno para Django y configurar Nginx. Finalmente, establece el comando para iniciar el servidor Django con Daphne y Nginx.
Dockerfile.celery	Este archivo Dockerfile define un contenedor para Celery. Utiliza una imagen base de Python 3.10.12 e instala las dependencias necesarias. Configura el directorio de trabajo, copia los archivos de la aplicación y ejecuta comandos para instalar las dependencias de Python y configurar las variables de entorno para Django. Establece el comando para iniciar el servidor Celery.
Dockerfile.gunicorn	Este archivo Dockerfile define un contenedor para el servidor Gunicorn. Utiliza una imagen base de Python 3.10.12 e instala las dependencias necesarias. Configura el directorio de trabajo, copia los archivos de la aplicación y ejecuta comandos para instalar las dependencias de Python, configurar las variables de entorno para Django y configurar Nginx. Finalmente, establece el comando para iniciar el servidor Django con Gunicorn y Nginx.
Dockerfile.docs	Este archivo Dockerfile define un contenedor para el servicio de documentación. Utiliza una imagen base de Python 3.10.12 e instala las dependencias necesarias para generar la documentación. Configura el directorio de trabajo, copia los archivos de la aplicación y ejecuta comandos para instalar las dependencias de Python y las dependencias específicas para la generación de documentación. Establece el comando para generar la documentación en formato HTML.

Tabla 4.4: Descripción de los archivos Dockerfile.

Capítulo 5

Conclusiones y recomendaciones

El informe debe terminarse con la enumeración de las principales conclusiones derivados del trabajo realizado. En particular, debe verificarse el cumplimiento de los objetivos planteados para el mismo.

5.1. Conclusiones

1. Se ha logrado implementar una plataforma funcional para la microscopía virtual web. Esta plataforma está equipada con las herramientas necesarias para realizar tareas relacionadas con la telepatología, cumpliendo así con todos los objetivos establecidos para el proyecto.
2. La revisión bibliográfica realizada permitió establecer una sólida base de conocimiento sobre la microscopía virtual, así como comprender el funcionamiento de la tecnología de mapas web y las herramientas para el desarrollo web. A partir de esta información, se identificaron los requisitos básicos que deben cumplir los microscopios virtuales, así como las herramientas de los sistemas de información geográfica que pueden satisfacer dichos requisitos. Además, esta revisión facilitó el discernimiento de las herramientas de diseño web que se emplearían en la plataforma.
3. La selección cuidadosa de componentes y herramientas de desarrollo ha sido fundamental para el éxito de la plataforma web de microscopía virtual. Tecnologías como Django, Celery, Gunicorn, Daphne y Nginx proporcionan una base sólida y flexible, garantizando un rendimiento óptimo y una alta disponibilidad del sistema. La elección de PostgreSQL y Redis como gestores de bases de datos asegura la eficiencia y escalabilidad de los datos. Además, las bibliotecas OpenSlide y Leaflet añaden funcionalidades clave para una experiencia de usuario completa. En conjunto, estos elementos conforman una infraestructura robusta para el desarrollo y funcionamiento de la plataforma.
4. Se ha conseguido desarrollar una herramienta web capaz de manejar imágenes de altas resoluciones mediante el uso de teselas, una técnica comúnmente empleada en servicios de mapas web como Google Maps y OpenStreetMap. Esta estrategia permitió trabajar de manera eficiente con imágenes de microscopía, ofreciendo distintos niveles de zoom y una interfaz intuitiva y familiar para los usuarios. Al aprovechar las bibliotecas destinadas al manejo de mapas en imágenes de microscopía, se optimizaron los recursos y se facilitó el desarrollo de la plataforma.

5. El proyecto ha logrado integrar herramientas comúnmente asociadas a sistemas de información geográfica en láminas de microscopía virtual. Se han incorporado funciones para resaltar áreas de interés en la imagen, trazar trayectorias, crear figuras geométricas y etiquetas, almacenándolas en formato GeoJSON. La capacidad de almacenar conjuntos de estos elementos como campos en una base de datos permite crear instancias de placas virtuales con marcaciones personalizadas, guardar el progreso en estas y desplegarlas en forma de capas superpuestas. Además, se logró añadir herramientas de medición de longitud y área, así como la posibilidad de guardar posiciones como coordenadas geográficas.
6. El proyecto ha sido desarrollado completamente utilizando software libre y está diseñado para ser escalable y de fácil descarga. A través del uso de herramientas como Git, Docker, Makefile y Docker Compose, se ha encapsulado y automatizado la plataforma. Esto se ha hecho con el propósito de facilitar su uso y promover su empleo como base para proyectos más complejos. Este enfoque constituye uno de los principales aspectos en los que se espera que la plataforma se destaque.

5.2. Recomendaciones

1. El proyecto se desarrolló intentando seguir los mejores estándares de orden y recomendaciones para la documentación. Sin embargo, al tratarse de la implementación realizada por una sola persona y ser su primer trabajo de este tipo, es posible que existan inconsistencias en la forma en que se organizó, la legibilidad o la implementación de la documentación. Se recomienda realizar una revisión diligente del proyecto para asegurar que el código sea legible y esté bien estructurado. Además, la documentación de la plataforma se realizó en Sphinx, por lo que también se sugiere realizar una revisión exhaustiva de esta documentación. Esto facilitaría el mantenimiento del código y garantizaría su continuidad.
2. Se recomienda llevar a cabo una validación exhaustiva de la plataforma en el servidor encargado de desplegarla, así como realizar una serie de pruebas con diferentes usuarios que gestionen grandes cantidades de datos. Esto permitiría detectar y depurar posibles errores de la plataforma antes de abrirla al público en general.
3. Es aconsejable crear un entorno más fluido que facilite el trabajo colaborativo. Por ejemplo, en el área de trabajo de proyecto, se ha implementado un cargado dinámico o en tiempo real del panel de herramientas derecho, lo que permite actualizarlo automáticamente sin necesidad de recargar toda la página. Esto se logra utilizando tecnologías como Ajax o WebSockets, como en el caso del chat de grupo. Sería beneficioso implementar esta funcionalidad en la interfaz de edición de la placa virtual para mejorar la colaboración entre usuarios.
4. Se sugiere utilizar la plataforma implementada para explorar enfoques avanzados con el objetivo de lograr las tareas de microscopía virtual de manera más efectiva. Se podrían explorar opciones avanzadas de software dedicado y aprovechar los recursos de la página desarrollada. Entre las principales ventajas que ofrecería esta recomendación está la posibilidad de trabajar con bases

de datos de gran tamaño sin necesidad de descargar láminas virtuales y utilizar recursos del servidor. Además, se podría trabajar de forma colaborativa de manera efectiva. Por ejemplo, se podría implementar una función similar a Google Colab, donde se habilite en el área de trabajo de proyectos la opción de subir scripts de Python que implementen algoritmos de microscopía en grandes conjuntos de datos o para entrenar modelos de inteligencia artificial. La plataforma cuenta con las herramientas necesarias para una implementación básica de este tipo, pero se deberían evaluar los recursos disponibles y la seguridad de la plataforma antes de llevarlo a cabo.

Apéndice A

README

Microscopio-Virtual-PRIS-Lab-B77342

Este es el repositorio virtual del microscopio virtual del PRIS-Lab de la Escuela de Ingeniería Eléctrica de la UCR.

Requisitos previos

Para desplegar esta plataforma en tu servidor local, asegúrate de tener instalados los siguientes programas:

- makefile
- git
- Docker
- Docker Compose

Instalación

Para comenzar, clona este repositorio utilizando el siguiente comando:

```
1 git clone https://github.com/JamesSibaja/Microscopio-Virtual-PRIS-Lab-B77342.git
```

Luego, dirígete al directorio principal del repositorio:

```
1 cd Microscopio-Virtual-PRIS-Lab-B77342
```

Despliegue en servidor local

Para desplegar la plataforma en tu servidor local, ejecuta el siguiente comando:

```
1 sudo make setup
```

Despliegue en servidor

Si deseas desplegar la plataforma en un servidor, primero asegúrate de editar el archivo nginx.conf según tus necesidades. Luego, ejecuta el siguiente comando:

```
1 sudo make setup
```

Información adicional

Este sistema ha sido probado en Ubuntu 23.04.

Bibliografía

- [1] Alayola A.; Ustarroz M.; Martínez J.; Pérez F.; Ramírez I.; Rodríguez J.; Camacho J.; Ustarroz M.; Hans E.; Arana A.; Muñoz J.; Hernández R.; Mendoza A. Microscopio virtual facultad de medicina, unam. <https://sistemasib.facmed.unam.mx/microscopio/>, 2023.
- [2] J. Beaird and George J. *The Principles of Beautiful Web Design*. SitePoint, 2014.
- [3] BioNetwork. Virtual microscope. <https://www.ncbi.nlm.nih.gov/iet/microscope/>, 2023.
- [4] K. Bittner and I. Spence. *Use Case Modeling*. Pearson Education, Inc., USA, 2003.
- [5] Figueroa C.; Martínez B.; Bosco C. *Microscopia Virtual: Tecnología al Servicio de la Enseñanza de la Histología/Embriología en Cursos de Ciencias de la Salud*. University of Chile, Viña del Mar, Chile, 2015.
- [6] Benoit Chesneau. Gunicorn. <https://gunicorn.org/>, 2023.
- [7] R. Clark, T.; Sorenson. Histology guide. <https://histologyguide.com/>, 2023.
- [8] Departamento de Histología de la Facultad de Medicina Universidad de Granada. Microscopio virtual. <https://histologiaugr.es/microscopio-virtual/>, 2023.
- [9] Facultad de Medicina Udelar. Microscopio virtual. <https://www.microscopiovirtual.net/>, 2023.
- [10] Anatomisches Institut Homburg (Universität des Saarlandes). Virtuelle mikroskopie homburg. <https://mikroskopie-uds.de/>, 2023.
- [11] Universidad Nacional de Río Cuarto Facultad de Agronomía y Veterinaria. Microscopio virtual. <http://www.ayv.unrc.edu.ar/microscopiovirtual/>, 2023.
- [12] Django Software Foundation and individual contributors. Django the web framework for perfectionists with deadlines. <https://www.djangoproject.com/>, 2023.
- [13] Django Software Foundation and individual contributors. How to use django with daphne. <https://docs.djangoproject.com/en/5.0/howto/deployment/asgi/daphne/>, 2023.
- [14] Docker Inc. Docker. <https://www.docker.com/why-docker/>, 2023.

- [15] ladimir Agafonkin. Leaflet, an open-source javascript library for mobile-friendly interactive maps, 2019.
- [16] ladimir Agafonkin. Leaflet, an open-source javascript library for mobile-friendly interactive maps, 2019.
- [17] Megías M.; Molist P.; Pombal M. Atlas de histología vegetal y animal. <http://mmegias.webs.uvigo.es/inicio.html>, 2023.
- [18] J. Manchuk. *Conversion of Latitude and Longitude to UTM Coordinates*. <https://citeseerx.ist.psu.edu/document?repid=rep1type=pdfdoi=f0180ea6f8fab77b53381c7cff3c610f7a9df64b>, 2009.
- [19] E Marín, D.; Romero. *Sistemas de microscopía virtual: análisis y perspectivas*. Instituto Nacional de Salud, Bogotá, Colombia, 2011,.
- [20] Mozilla and individual contributors. ¿qué es javascript? https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript, 2020.
- [21] F5 NGINX. Nginx wiki. <https://sistemasib.facmed.unam.mx/microscopio/>, 2023.
- [22] National Institute of Biomedical Imaging and Bioengineering (NIBIB). Resultados de biopsia mas rápidos con un nuevo panel de microscopios. <https://www.nibib.nih.gov/espanol/ciencia-highlights/resultados-de-biopsia-mas-r%C3%A1pidos-con-un-nuevo-panel-de-microscopios>, 2006.
- [23] Group of M. Satyanarayanan (Satya) in the Carnegie Mellon University School of Computer Science. Openslide. <https://openslide.org/>, 2023.
- [24] J. Peralta Sáenz. *Diseño, implementación y validación de un algoritmo para la segmentación de células cancerígenas a partir de imágenes de microscopía fluorescente*. Universidad de Costa Rica, Costa Rica, 2018.
- [25] L. Quinde Cobos. *Diseño, implementación y validación un algoritmo de rastreo de células cancerígenas basado en HMM's a partir de imágenes de microscopía de campo claro*. Universidad de Costa Rica, Costa Rica, 2018.
- [26] J. M. Santos. *Sistemas de Información Geográfica*. Universidad Nacional de Educación a distacia, 2da. edition, 2020.
- [27] Michigan MultiMedia Health Information Technology Services. Michigan histology and virtual microscopy learning resources. <https://histology.medicine.umich.edu/>, 2023.
- [28] Ask Solem. Celery - distributed task queue. <https://docs.celeryq.dev/en/stable/>, 2023.
- [29] The Open University. Virtual microscope. <https://www.virtualmicroscope.org/>, 2023.