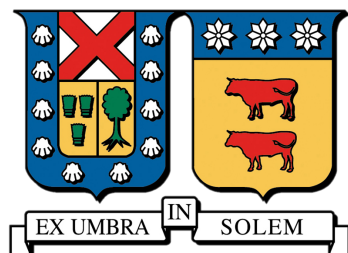


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

VALPARAÍSO - CHILE



**FUNGIGT: FUNGI GENOMIC TOOLS
PLATAFORMA DE INTEGRACIÓN DE
HERRAMIENTAS GENÓMICAS PARA EL
ANÁLISIS DE HONGOS**

FARID AGUSTÍN DÍAZ CID

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL TELEMÁTICO

PROFESOR GUÍA: PATRICIO OLIVARES RONCAGLIOLO

PROFESOR CORREFERENTE: NICOLÁS JARA CARVALLO

JULIO - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: FungiGT, Plataforma de integración de herramientas genómicas para el análisis de hongos

Nombre del candidato(a): Farid Agustín Díaz Çid

Carrera / Grado: Ingeniería Civil Telemática

Campus: Casa Central Valparaíso ; **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, PATRICIO ANDRÉS OLIVARES RONCAGLIOLO, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 25/07/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 25/07/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

“El amor por todas las criaturas vivientes es el más noble atributo de la humanidad”

-Charles Darwin (1809-1882)

Agradecimientos

Deseo expresar mi más profundo y sincero agradecimiento a todas las personas e instituciones que han hecho posible la realización de este proyecto.

En primer lugar, agradezco a Esli Lobaina, Investigador del Centro de Biotecnología Daniel Alkalay Lowitt, cuya visión y propuesta inicial fueron fundamentales para el desarrollo de este proyecto. Su liderazgo y experiencia en el campo de la biotecnología han sido una fuente constante de inspiración y guía.

Extiendo mi gratitud a Marcelo García y a su profesor guía, Nicolás Torres, así como a Fabián Godoy y a su profesora guía, Patricia Morales. Su colaboración activa y su disposición para compartir conocimientos técnicos y académicos han enriquecido significativamente el contenido y la calidad de este informe. Su apoyo constante y sus valiosas aportaciones han sido esenciales para superar los desafíos encontrados a lo largo del desarrollo del proyecto.

Un agradecimiento especial a mi profesor guía, Patricio Olivares, cuya orientación experta, paciencia y compromiso han sido pilares fundamentales en la estructuración y ejecución de este trabajo. Su dedicación para fomentar un ambiente de aprendizaje colaborativo y su interés genuino por el éxito de sus estudiantes han sido invaluable para el progreso de esta investigación.

Este esfuerzo colaborativo ha sido posible gracias a la participación de los estudiantes de Ingeniería Civil Telemática que formaron parte del Programa de Memorias Multidisciplinarias. Su compromiso, dedicación y entusiasmo han sido esenciales para avanzar en el campo de la bioinformática. La sinergia entre los diferentes perfiles académicos y técnicos ha permitido la creación de una herramienta integral, diseñada para servir de puente a nuevos investigadores que deseen realizar análisis de manera cómoda y eficiente. El trabajo en equipo y la disposición para colaborar en distintas áreas del conocimiento han demostrado ser clave para alcanzar los objetivos propuestos.

Asimismo, agradezco a todos los docentes y personal administrativo de la Ingeniería Civil Telemática por su apoyo constante y por proporcionar los recursos necesarios para el desarrollo de este proyecto. Su respaldo ha facilitado un entorno propicio para la investigación

y la innovación.

Finalmente, quiero agradecer a mis compañeros y amigos por su apoyo moral, su camaradería y por compartir este camino académico. Sus palabras de aliento y su disposición para colaborar han hecho de esta experiencia algo memorable y enriquecedor.

A todos ustedes, gracias por su invaluable apoyo y colaboración. Sin su contribución, este proyecto no habría sido posible.

Abstract

La plataforma FungiGT es una solución integral y optimizada para el análisis genómico y filogenético de hongos, diseñada para superar las limitaciones de las herramientas bioinformáticas tradicionales. Este proyecto integra herramientas avanzadas como BinDash, eggNOG-mapper y Braker3 en un entorno web unificado, permitiendo realizar tareas clave como la anotación genómica, el control de calidad y el análisis funcional de manera eficiente. Gracias a la implementación de tecnologías modernas como contenedores Docker, microservicios y arquitecturas escalables, FungiTG optimiza el procesamiento de grandes conjuntos de datos genómicos y reduce significativamente los tiempos de análisis: de dos días a solo 18 horas.

La plataforma prioriza la experiencia del usuario mediante visualizaciones interactivas, como árboles filogenéticos, mapas de calor y diagramas de vías metabólicas, que facilitan la interpretación de resultados complejos. Además, su diseño modular y escalable permite futuras expansiones e integraciones, adaptándose a las necesidades cambiantes de la investigación.

FungiGT representa una contribución significativa al campo de la genómica de hongos, ofreciendo una herramienta accesible, robusta y eficiente para investigadores de todo el mundo, al tiempo que mejora la productividad y precisión en el análisis genómico.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Problema por resolver	2
1.3. Objetivos	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
1.4. Estructura general del documento	4
2. Estado del Arte	7
2.1. Soluciones existentes en la industria	7
2.2. Workflows en genómica y enfoque en un grupo específico	11
3. Arquitectura propuesta	13
3.1. Arquitectura del sistema	13
3.1.1. Diagrama de Contexto	14
3.1.2. Diagrama de Arquitectura	15
3.1.3. Evolución de la Arquitectura	17
3.1.4. Dockerización	19

4. Diseño Propuesto	25
4.1. Descripción General de Módulos	25
4.1.1. Diseño de Módulos del Sistema	28
4.2. Diseño de Interfaz	41
5. Resultados	44
5.1. Caso de Uso: Adquisición + Sistema de archivos	44
5.1.1. Objetivo del módulo	44
5.1.2. Caso de uso M1+M9	44
5.1.3. Diagrama de caso de uso M1+M9	45
5.1.4. Evidencia funcional	46
5.1.5. Validación del caso de uso M1+M9	48
5.2. Caso de Uso: Calidad y Sistema de Archivos	48
5.2.1. Objetivo del módulo	48
5.2.2. Caso de uso M2+M9	48
5.2.3. Diagrama de caso de uso M2+M9	49
5.2.4. Evidencia funcional	50
5.2.5. Validación del caso de uso M2+M9	53
5.3. Caso de Uso: Anotación y Sistema de Archivos	54
5.3.1. Objetivo del módulo	54
5.3.2. Caso de uso M3+M9	54
5.3.3. Diagrama de caso de uso M3+M9	55
5.3.4. Evidencia funcional	56
5.3.5. Validación del caso de uso M3+M9	58

5.4.	Caso de Uso: Análisis Funcional y Sistema de Archivos	59
5.4.1.	Objetivo del módulo	59
5.4.2.	Caso de uso M4+M9	59
5.4.3.	Diagrama de caso de uso M4+M9	60
5.4.4.	Evidencia funcional	61
5.4.5.	Validación del caso de uso M4+M9	64
5.5.	Caso de Uso: Análisis filogenético y Sistema de Archivos	65
5.5.1.	Objetivo del módulo	65
5.5.2.	Caso de uso M5+M9	65
5.5.3.	Diagrama de caso de uso M5+M9	66
5.5.4.	Evidencia funcional	67
5.5.5.	Validación del caso de uso M5+M9	69
5.6.	Caso de Uso: Visualización y Sistema de Archivos	70
5.6.1.	Objetivo del módulo	70
5.6.2.	Caso de uso M8+M9	70
5.6.3.	Diagrama de caso de uso M8+M9	71
5.6.4.	Evidencia funcional	71
5.6.5.	Validación del caso de uso M8+M9	73
5.7.	Caso de Uso: Visualización, Sistema de Archivos e Interpretación Inteligente	74
5.7.1.	Objetivo del módulo	74
5.7.2.	Caso de uso M8+M9+M10	74
5.7.3.	Diagrama de caso de uso M8+M9+M10	75
5.7.4.	Evidencia funcional	76

5.7.5.	Validación del caso de uso M8+M9+M10	83
5.7.6.	Frontend	83
6.	Validación y verificación	91
6.1.	Verificación Técnica	91
6.1.1.	Estrategia de Testing	91
6.1.2.	Testing de Infraestructura Docker	92
6.1.3.	Cálculo de la tasa de disponibilidad	94
6.1.4.	Testing de Integración de Servicios	96
6.1.5.	Testing Funcional por Módulos	98
6.1.6.	Resultados de la Verificación	100
6.2.	Validación con el Cliente	100
6.2.1.	Evolución de Requisitos Basada en Feedback	101
6.2.2.	Decisiones Arquitectónicas Validadas	102
6.2.3.	Validación de Usabilidad	102
6.2.4.	Aceptación del Cliente	104
6.3.	Análisis Crítico y Gestión de Riesgos	105
6.3.1.	Riesgos Identificados y Mitigados	105
6.3.2.	Lecciones Aprendidas	106
7.	Conclusiones	107
Anexo A.	Anexos: Códigos y planteamientos iniciales	109
A.1.	Planteamientos iniciales	109
A.2.	Código y estructura	111

A.2.1. Estructura de Directorios Principal	111
A.2.2. Módulos de Microservicios	113
A.2.3. Infraestructura y Scripts de Gestión	115

Índice de figuras

3.1. Diagrama de contexto del sistema	15
3.2. Diagrama de arquitectura del sistema	18
4.1. Comandos disponibles de NCBI Datatsets CLI	29
4.2. Workflow de checkM	31
4.3. Proceso de selección de genes marcadores	32
4.4. Workflow de Braker3	34
4.5. Workflow de Egglog Mapper	36
4.6. Ejemplo de matriz de distancias genómicas generada por BinDash para especies fúngicas.	38
4.7. Modelo de navegación del sistema	42
5.1. Caso de uso del módulo de Adquisición	45
5.2. Vista del módulo de adquisición con el resultado de la descarga de Trichoderma virens Gv29-8	46
5.3. Vista del módulo de adquisición en el proceso de extracción del .zip que contiene el archivo .fna	47
5.4. Vista del módulo de sistema de archivos en donde se evidencia la existencia del archivo .fna para realizar anotaciones	47

5.5. Caso de uso del módulo de Calidad y Sistema de Archivos	49
5.6. Interfaz del módulo de calidad, mostrando la configuración principal y selección de rutas de entrada y salida.	50
5.7. Vista del módulo de calidad donde se configuran opciones avanzadas, revisión del comando y ejecución de CheckM.	51
5.8. Vista del módulo de calidad tras la finalización del proceso de CheckM. . .	52
5.9. Vista del sistema de archivos revisando los archivos generados por CheckM. .	52
5.10. Visualización de un reporte .tsv generado por CheckM mediante el módulo de sistema de archivos.	53
5.11. Caso de uso del módulo de Anotación y Sistema de Archivos.	55
5.12. Interfaz del módulo de anotación mostrando la selección de especie, archivo genómico y rutas de input/output.	56
5.13. Vista del módulo de anotación, configuración de opciones avanzadas y ejecución de BRAKER3.	57
5.14. Vista del módulo tras la finalización exitosa del proceso de BRAKER3. . .	57
5.15. Visualización, a través del sistema de archivos, de los archivos generados por BRAKER3.	58
5.16. Caso de uso del módulo de Análisis Funcional y Sistema de Archivos. . .	60
5.17. Interfaz del módulo de análisis funcional definiendo archivos de entrada y salida.	61
5.18. Vista del módulo de análisis funcional configurando los parámetros de ejecución de eggNOG-mapper.	62
5.19. Vista del módulo de análisis funcional durante la ejecución del análisis. . .	63
5.20. Vista del módulo de análisis funcional tras la finalización del proceso. . .	63
5.21. Visualización y exploración de los archivos generados por eggNOG-mapper mediante el sistema de archivos de la plataforma.	64

5.22. Caso de uso del módulo de Análisis Filogenético y Sistema de Archivos. . .	66
5.23. Interfaz del módulo de análisis filogenético, con el modo de operación seleccionado y los directorios de entrada y prefijos configurados.	67
5.24. Vista del módulo de análisis filogenético durante la ejecución de BinDash. . .	68
5.25. Vista del módulo de análisis filogenético tras la finalización exitosa del proceso de BinDash.	68
5.26. Visualización y exploración de los archivos generados por BinDash mediante el sistema de archivos de la plataforma.	69
5.27. Caso de uso del módulo de Visualización y Sistema de Archivos.	71
5.28. Interfaz del módulo de visualización de resultados.	71
5.29. Vista del módulo de visualización seleccionando el visualizador específico para las salidas de eggNOG-mapper.	72
5.30. Integración del módulo de visualización con el sistema de archivos para cargar archivos de anotaciones.	72
5.31. Utilización de la herramienta de visualización para explorar gráficos, interactuar con ellos y alternar entre las diferentes visualizaciones disponibles para los archivos de anotaciones.	73
5.32. Caso de uso integrado de Visualización, Sistema de Archivos y Dr. Fungito. . .	75
5.33. Interfaz del módulo de visualización con un gráfico del top 15 de queries con más hits, generado a partir de la salida de EggNOG-mapper.	76
5.34. Interfaz del módulo de visualización con un gráfico del top 15 de targets, generado a partir de la salida de hits de EggNOG-mapper.	77
5.35. Visualización enviada a Dr. Fungito para el análisis inteligente del gráfico de las 15 queries con más hits.	78
5.36. Visualización enviada a Dr. Fungito para el análisis inteligente del gráfico de los 15 targets más frecuentes.	79

5.37. Generación de reporte inteligente por Dr. Fungito, mostrado en la carpeta data/reports, disponible para descarga.	80
5.38. Parte del reporte PDF generado para ambos gráficos; en este caso, correspondiente al top 15 de queries con más hits.	81
5.39. Parte del reporte PDF generado para ambos gráficos; en este caso, correspondiente al top 15 de targets.	82
5.40. Landing page	83
5.41. Login	84
5.42. Módulo de adquisición	85
5.43. Módulo de anotación (Calidad y Anotación disponibles)	86
5.44. Módulo de análisis (Análisis funcional y filogenético disponibles)	87
5.45. Visualizador - Selección de archivo de entrada	88
5.46. Visualizador - Mapa de calor	89
6.1. Ejemplo representativo de healthcheck exitoso en el contenedor Docker de CheckM.	93
6.2. Pantalla de login de la plataforma FungiGT.	97
A.1. Workflow propuesto inicialmente	111

Índice de cuadros

2.1. Comparativa de diferentes herramientas y aproximaciones en análisis genómico	10
4.1. Descripción de módulos	28
5.1. Eventos del prototipo	90
6.1. Cálculo ilustrativo de tasa de disponibilidad por servicio con referencia a casos de uso validados en el Capítulo 5.	95
6.2. Criterios de verificación funcional por módulo.	99
6.3. Comparación validada por el cliente entre el flujo tradicional y la plataforma FungiGT.	103

Capítulo 1

Introducción

1.1. Contexto

La genómica funcional y filogenética de hongos es un campo que ha experimentado un crecimiento significativo en los últimos años debido a la importancia biológica, ecológica y económica de estos organismos. Recientes revisiones han profundizado en la evolución de la arquitectura genómica fúngica [18], los avances en taxonomía y filogenia [17] y el desarrollo de plataformas de genómica funcional para hongos [16], subrayando la relevancia de estos organismos como modelos para estudiar diversidad y adaptación. La filogenia permite comprender las relaciones evolutivas entre especies, proporcionando conocimientos fundamentales sobre la evolución, biodiversidad y potencial biotecnológico de los hongos. Sin embargo, realizar análisis filogenéticos y funcionales exhaustivos implica enfrentar desafíos técnicos considerables.

En este contexto, los investigadores enfrentan obstáculos recurrentes, tales como la dificultad para gestionar grandes volúmenes de datos, la necesidad de conocimientos avanzados para ejecutar análisis complejos, y la falta de interfaces accesibles que faciliten la interpretación visual y práctica de los resultados. Estas limitaciones generan una barrera de entrada considerable, especialmente para usuarios con experiencia limitada en bioinformática, lo que impide aprovechar plenamente el potencial que ofrecen estos análisis en diversas apli-

caciones, desde la agricultura hasta la biomedicina.

Para superar estas falencias, se propuso el desarrollo de una plataforma web integral denominada FungiGT, cuyo propósito es facilitar y optimizar los procesos asociados al análisis genómico funcional y filogenético de hongos. FungiGT integra herramientas avanzadas como BlastN, eggNOG-mapper, Braker3 y CheckM en un entorno único y amigable para el usuario, que simplifica tareas fundamentales tales como la anotación de secuencias genómicas, el control de calidad y la interpretación funcional y evolutiva.

La plataforma incorpora además visualizaciones interactivas tales como gráficos filogenéticos, mapas de calor y diagramas de vías metabólicas, elementos clave que permiten a los investigadores interpretar y comunicar resultados complejos de forma clara y efectiva. De este modo, FungiGT no solo representa un avance significativo en la eficiencia y precisión de los análisis genómicos, sino que también promueve una mayor democratización del conocimiento, permitiendo que investigadores con distintos niveles de experiencia puedan realizar análisis avanzados de manera sencilla e intuitiva.

Finalmente, cabe mencionar que este proyecto fue desarrollado en el contexto del Programa de Memorias Multidisciplinarias (PMM), donde distintos integrantes del equipo abordaron áreas específicas del proyecto, como el análisis funcional y la filogenia, mientras que la contribución principal descrita en esta memoria se concentró en el diseño, desarrollo e integración técnica de la plataforma web propuesta.

1.2. Problema por resolver

El problema central que aborda este proyecto no es la mera ausencia de software, sino el tiempo y la cantidad de trabajo manual que hoy exige completar un flujo de análisis genómico y filogenético en hongos. Aunque existen múltiples herramientas especializadas, su ejecución aislada casi siempre desde la línea de comandos obliga a concatenar de forma artesanal la limpieza de datos, los alineamientos, la inferencia filogenética y la anotación funcional. Este proceso fragmentado eleva los tiempos de cómputo, multiplica los pasos repetitivos y aumenta la probabilidad de errores humanos, generando cuellos de botella que

ralentizan la generación de hipótesis y encarecen los proyectos, especialmente en laboratorios con recursos limitados.

Para mitigar estas ineficiencias, proponemos una plataforma integrada y automatizada que encadene las etapas críticas desde la descarga de genomas hasta la visualización de árboles y la interpretación funcional, disminuyendo drásticamente las horas de trabajo, optimizando recursos computacionales y permitiendo que los investigadores concentren sus esfuerzos en la interpretación biológica de los resultados.

1.3. Objetivos

1.3.1. Objetivo General

El objetivo general de este proyecto es desarrollar una plataforma web integral, modular y eficiente que permita a los investigadores realizar análisis genómicos y filogenéticos de hongos de manera accesible, precisa y optimizada. La solución busca unificar herramientas bioinformáticas avanzadas para la anotación, control de calidad y análisis funcional y filogenético de genomas, ofreciendo una experiencia de usuario intuitiva a través de una interfaz visual moderna. Además, la plataforma se centrará en garantizar un rendimiento robusto y escalable, capaz de manejar grandes volúmenes de datos genómicos y adaptarse a las necesidades específicas de los usuarios. Este objetivo responde a la necesidad de superar las limitaciones actuales en el campo, proporcionando una herramienta que facilite y acelere los procesos de investigación en genómica funcional y evolutiva de hongos.

1.3.2. Objetivos Específicos

1. Realizar un análisis exhaustivo de las necesidades actuales en el campo de la genómica funcional en hongos para identificar y definir claramente los desafíos y limitaciones de los métodos existentes. Este objetivo incluye el desarrollo de una comprensión profunda del flujo de trabajo de análisis genómico, identificando oportunidades para

mejorar la eficiencia y precisión a través de automatización.

2. Diseñar y desarrollar múltiples prototipos iterativos de la plataforma, cada uno diseñado para probar las especificaciones recogidas durante la fase de definición de la problemática. Este proceso incluirá la configuración y personalización de las plataformas para satisfacer las demandas específicas de análisis genómico, permitiendo ajustes basados en retroalimentación continua de los usuarios finales.
3. Desarrollar una interfaz de usuario limpia, intuitiva y fácil de usar que facilite a los científicos y biólogos la carga, manejo y visualización de grandes volúmenes de datos genómicos. La interfaz también incluirá funcionalidades avanzadas de visualización de resultados que permitan interpretaciones claras y efectivas, apoyando tanto análisis cualitativo como cuantitativo.
4. Integrar y configurar un conjunto de herramientas bioinformáticas avanzadas dentro de la plataforma para apoyar análisis filogenético y funcional complejos. Esto incluye seleccionar herramientas que puedan manejar secuencias de datos de alta complejidad y dar resultados precisos, garantizando que la integración sea fluida y que funcionen eficientemente.
5. Realizar una serie de pruebas exhaustivas en todas las etapas del desarrollo para verificar la funcionalidad completa de la plataforma, la seguridad de los datos manejados y la usabilidad de la interfaz. Esto incluirá pruebas de estrés, pruebas de seguridad para proteger contra vulnerabilidades de datos, y pruebas de usabilidad para asegurar que la plataforma sea accesible y eficiente para todos los usuarios finales.

1.4. Estructura general del documento

Este documento está dividido en siete capítulos:

Capítulo 1: Introducción En este capítulo se presenta una descripción general del proyecto, destacando la importancia del análisis genómico y filogenético de hongos, así como los

principales desafíos que enfrentan los investigadores en este ámbito. Además, se plantean el problema a resolver, los objetivos generales y específicos del proyecto, y se justifica la necesidad de una solución innovadora como la propuesta.

Capítulo 2: Estado del Arte Se analiza la literatura y las herramientas existentes relacionadas con la genómica funcional y filogenética, incluyendo plataformas como Galaxy, herramientas especializadas como BlastN, eggNOG-mapper y AntiSMASH, y su impacto en la investigación actual. También se evalúan sus limitaciones, destacando cómo estas influyen en la necesidad de soluciones más integradas y eficientes.

Capítulo 3: Propuesta de Solución En este capítulo se detalla la propuesta del proyecto FungiGT como una plataforma web unificada, modular y escalable, diseñada para superar las deficiencias de las herramientas actuales. Se describen los principios fundamentales de su diseño, los módulos principales y los beneficios de integrar herramientas avanzadas de análisis genómico en un solo entorno accesible.

Capítulo 4: Diseño del Sistema Se presenta el diseño arquitectónico de la plataforma, incluyendo los módulos principales, el flujo de trabajo, las tecnologías utilizadas y las estrategias para garantizar la escalabilidad, modularidad y robustez del sistema. Se incluyen diagramas y descripciones detalladas de la integración de herramientas bioinformáticas como BlastN, Braker3 y CheckM.

Capítulo 5: Resultados Este capítulo expone los resultados obtenidos durante el desarrollo y pruebas del prototipo, destacando la funcionalidad de los módulos implementados, el rendimiento del sistema en escenarios de prueba y los análisis realizados con datos genómicos reales. Se presentan visualizaciones y reportes generados como parte del flujo de trabajo.

Capítulo 6: Validación Se describen las pruebas realizadas para validar la funcionalidad, precisión y eficiencia de la plataforma. Se incluyen evaluaciones de calidad de los resultados, retroalimentación de usuarios potenciales y comparaciones con herramientas existentes para demostrar la viabilidad y ventajas del sistema.

Capítulo 7: Conclusiones y Trabajo Futuro En el capítulo final, se sintetizan los principales logros del proyecto, se reflexiona sobre las limitaciones encontradas y se proponen posibles

mejoras y extensiones futuras para la plataforma FungiGT. Se resalta el impacto potencial de la herramienta en la investigación genómica de hongos y otros campos relacionados.

Capítulo 2

Estado del Arte

2.1. Soluciones existentes en la industria

El ecosistema bioinformático abarca desde plataformas integrales hasta utilidades altamente especializadas. Entre las primeras destaca **Galaxy** [4](<https://galaxyproject.org/>), que popularizó los entornos web para construir *workflows* reproducibles. Su fortaleza radica en la orquestación gráfica de cientos de herramientas; sin embargo, al procesar volúmenes masivos de datos fúngicos su rendimiento se resiente, lo que obliga a complementar la plataforma con soluciones de mayor eficiencia computacional.

Las necesidades de anotación funcional se satisfacen habitualmente mediante **eggNOG-mapper v2** [14](<http://eggnog-mapper.embl.de>), que asigna ortología y funciones a escala genómica, y **antiSMASH**(<https://antismash.secondarymetabolites.org>), referencia en la detección de *biosynthetic gene clusters*. Para delimitar grupos ortólogos entre eucariotas, **OrthoMCL** [3](<http://www.cbil.upenn.edu/gene-family>) continúa siendo un método clásico, mientras que la clasificación ecológica de comunidades fúngicas se apoya en **FUNGuild** [2](<https://github.com/UMNFUN/FUNGuild>). En metagenómica, **MetaPhlAn** [5](<http://huttenhower.sph.harvard.edu/metaphlan>) sobresale por su perfilado ultrarrápido mediante genes marcadores clado-específicos.

La predicción génica ha evolucionado desde enfoques *ab initio*, representados por

AUGUSTUS [6](<https://bioinf.uni-greifswald.de/augustus>), hacia tuberías híbridas que incorporan evidencia transcriptómica y proteica, como **BRAKER3** [13](<https://github.com/Gaius-Augustus/BRAKER>). En paralelo, el control de calidad de ensamblajes se confía a **CheckM** [11](<https://ecogenomics.github.io/CheckM/>), basado en genes marcadores de linaje, mientras que la descarga masiva de genomas y metadatos se ha simplificado con **NCBI Datasets** [12](<https://www.ncbi.nlm.nih.gov/datasets/>). Para comparar cientos de genomas en segundos, **BinDash/Mashtree** [15](<https://github.com/lskatz/mashtree>) emplea bocetos min-hash y proporciona distancias confiables con recursos computacionales mínimos.

Los hongos constituyen un caso de estudio singular. El análisis comparativo de *Trichoderma* reveló que el micoparasitismo es su estilo de vida ancestral [1], hallazgo reforzado por la secuenciación de *T. virens*, que evidenció su adaptación a la rizosfera [7]. En el ámbito taxonómico, la caracterización de quinasas MAPKKK en trigo empleó **BLASTP** para dilucidar redes de interacción [8], y la identificación molecular de *Penicillium* en quesos demostró la utilidad de marcadores ADN multigénicos [9].

La frontera más reciente la marca **GenePT** [10](<https://github.com/yiqunchen/GenePT>), un modelo fundacional que traslada las incrustaciones de ChatGPT al espacio genómico y celular, logrando un rendimiento superior en tareas de predicción funcional y regulación génica.

En conjunto, estas herramientas ofrecen capacidades potentes pero fragmentadas; la integración suele implicar conversiones de formato, scripts ad-hoc y una curva de aprendizaje elevada. El proyecto **FungiGT** nace para unificar este ecosistema: combina la flexibilidad de Galaxy con la eficiencia de contenedores Docker, incorpora módulos de anotación, calidad y comparativa genómica optimizados para hongos, y proporciona una interfaz web que prioriza la visualización interactiva y la escalabilidad horizontal, garantizando así reproducibilidad y rapidez en estudios de genómica fúngica de gran escala.

El cuadro 2.1 muestra una comparativa detallada de diferentes herramientas y enfoques en el análisis genómico, destacando tanto sus funcionalidades clave como sus principales limitaciones. Esta comparativa es especialmente relevante, ya que permite identificar claramente

las ventajas y deficiencias de las soluciones actualmente disponibles en la bioinformática aplicada a la genómica y filogenia. Se observa que herramientas como **Galaxy**, aunque ampliamente utilizadas debido a su flexibilidad y facilidad para diseñar workflows personalizados, presentan limitaciones importantes relacionadas con el rendimiento y escalabilidad en grandes volúmenes de datos. Otras soluciones especializadas como **OrthoMCL** y **MetaPhlAn** destacan por funcionalidades específicas, como la identificación precisa de grupos ortólogos y el uso de marcadores taxonómicos clade-específicos respectivamente, aunque su uso aislado incrementa la complejidad operativa.

Además, algunas herramientas especializadas, como **Trichoderma Genome Analysis** y **Trichoderma virens Analysis**, están limitadas a aplicaciones o especies específicas, reduciendo su aplicabilidad general. Por otro lado, herramientas como **AUGUSTUS**, aunque ofrecen una predicción precisa de genes, requieren ensamblados previos completos, lo que puede no ser factible en todos los casos.

En conjunto, esta tabla resalta la necesidad de contar con una plataforma integral, modular y escalable como la propuesta en este proyecto (FungiGT), capaz de integrar múltiples herramientas especializadas y superar estas limitaciones, simplificando así significativamente los procesos asociados al análisis genómico y filogenético para hongos.

Ref	Año	Aplicación Genómica	Herramienta	Funcionalidades Clave	Limitaciones Identificadas
[1]	2011	Genómica comparativa	Trichoderma Genome Analysis	Análisis evolutivo, enfoque en Trichoderma	Específico a Trichoderma
[2]	2015	Anotación funcional	FUNGuild	Asignación ecológica a guilds fúngicos	Limitado a datos de comunidades
[3]	2003	Filogenia y ortólogos	OrthoMCL	Identificación de clusters ortólogos	Requiere integración manual compleja
[4]	2022	Análisis integrados	Galaxy	Workflows personalizables	Bajo rendimiento con datasets grandes
[5]	2012	Perfilado taxonómico	MetaPhlAn	Uso de marcadores clade-específicos	Solo se puede usar en bacterias y arqueas
[6]	2019	Predicción génica	AUGUSTUS	Predicción precisa de genes	Limitado a genomas ensamblados
[7]	2019	Adaptación evolutiva	Trichoderma virens Analysis	Análisis asociado a estilos de vida de plantas	Enfoque específico a T. virens
[8]	2019	Identificación de genoma de familia quinasas MAPKKK	BlastP	Predicción de redes de interacción	Se utilizó una validación experimental de 14 genes
[9]	2019	Identificación molecular	Penicillium DNA Markers	Identificación mediante marcadores de ADN	Limitado a especies de queso
[10]	2024	Modelos genómicos	GenePT	Modelo fundacional para genes y células	Requiere acceso a ChatGPT
[11]	2015	Identificación de genes marcadores	CheckM	Evaluar calidad de genomas	Se ejecuta vía comando, sin interfaz
[12]	2024	Exploración de datos genómicos	NCBI Datasets	Descarga de datos a gran escala	Se ejecuta vía comando, sin interfaz
[13]	2024	Anotación de genes	Braker3	Predicción génica automática	Se ejecuta vía comando, sin interfaz
[14]	2021	Anotación funcional	EggNog-Mapper	Asignación funcional automática	Se ejecuta vía comando, sin interfaz
[15]	2019	Filogenia, distancia genómica	Bindash	Análisis comparativo genómico	Se ejecuta vía comandos, sin interfaz

Cuadro 2.1: Comparativa de diferentes herramientas y aproximaciones en análisis genómico

2.2. Workflows en genómica y enfoque en un grupo específico

En genómica, los **workflows** son secuencias estructuradas de procesos bioinformáticos diseñados para analizar, transformar y visualizar datos genómicos de manera sistemática. Estos flujos de trabajo abarcan múltiples etapas, desde la adquisición de datos hasta la obtención de resultados interpretables. Los componentes principales de un workflow genómico incluyen:

1. **Descarga de datos genómicos:** La recopilación de secuencias genómicas desde bases de datos públicas, como NCBI o ENA, en formatos estandarizados (por ejemplo, FASTA, GBK). Esta etapa garantiza que los datos sean accesibles y adecuados para análisis posteriores.
2. **Análisis inicial y control de calidad:** Se evalúa la calidad de los datos genómicos para detectar errores de secuenciación, contaminaciones o ensamblajes incompletos. Herramientas como *CheckM* o *FastQC* son comunes en esta etapa.
3. **Anotación:** Identificación de genes, ARN, promotores y elementos regulatorios en las secuencias genómicas. Esta fase permite caracterizar las funciones biológicas y estructurales de los componentes genéticos, utilizando herramientas como *Prokka* o *Braker3*.
4. **Análisis funcional y filogenético:** Incluye la asignación de funciones a los genes identificados mediante bases de datos como KEGG o Gene Ontology, así como la construcción de árboles filogenéticos para determinar relaciones evolutivas.
5. **Transformación y visualización de datos:** Generación de gráficos de filogenia, mapas de calor, diagramas de vías metabólicas y otras visualizaciones interactivas para interpretar los resultados de manera comprensible y accionable.

Dado que los datos genómicos abarcan una diversidad inmensa de organismos y tipos de análisis, es crucial enfocar el trabajo en un grupo específico al desarrollar workflows. En este caso, los hongos representan un grupo ideal debido a su relevancia ecológica, médica y biotecnológica. Centrar los esfuerzos en hongos permite diseñar workflows especializados

que aborden sus características únicas, como la complejidad de sus genomas, su diversidad genética y las aplicaciones prácticas asociadas (como la producción de compuestos bioactivos o el control biológico).

Este enfoque también facilita optimizar los recursos y garantizar resultados de alta calidad. Por ejemplo, priorizar un grupo permite ajustar las herramientas de análisis y visualización a las necesidades específicas del organismo, mejorar la reproducibilidad de los análisis y generar conocimiento más profundo sobre los patrones evolutivos y funcionales de los hongos. Además, este enfoque modularizable establece una base sólida que puede extenderse a otros grupos en el futuro, manteniendo la flexibilidad y escalabilidad del sistema.

Al partir por hongos, el proyecto puede desarrollar un workflow completo y bien diseñado, que integre herramientas desde la descarga de datos hasta su análisis y visualización. Esto no solo mejora la precisión y eficiencia de los análisis, sino que también responde a necesidades críticas en la investigación de este grupo de organismos.

Capítulo 3

Arquitectura propuesta

3.1. Arquitectura del sistema

Para comprender el funcionamiento del sistema propuesto conviene repasar, al menos de forma sintética, algunos pilares de la genómica. Un **nucleótido** es la unidad estructural del ADN y del ARN; está compuesto por una base nitrogenada (adenina, timina, citosina o guanina en el ADN), un azúcar (desoxirribosa) y un grupo fosfato. La cadena de **ADN** formada por millones de nucleótidos almacena la información hereditaria. Dentro de esa cadena, un **gen** es un tramo que dirige la síntesis de un producto funcional—en la mayoría de los casos, una **proteína**, es decir, un polímero de aminoácidos cuya conformación tridimensional determina la mayor parte de las funciones celulares. El ADN se empaqueta, junto con proteínas histónicas, en estructuras denominadas **cromosomas**, lo que garantiza su estabilidad y su correcta segregación durante la división celular.

El flujo genómico general suele dividirse en tres grandes fases: *pre-secuenciación* (manipulación y preparación de la muestra), *secuenciación* propiamente tal (obtención de lecturas crudas) y *pos-secuenciación*. Empresas de distintos países lideran el proceso de secuenciación: **Illumina** (<https://www.illumina.com>) y **Pacific Biosciences** (<https://www.pacb.com>) (Estados Unidos), **Oxford Nanopore Technologies** (<https://nanoporetech.com>) (Reino Unido), **BGI Genomics** (<https://www.bgi.com/global>)

(China) y **Macrogen** (<https://www.macrogen.com>) (Corea del Sur) proporcionan tecnologías de lectura corta, larga o ultralarga capaces de generar volúmenes masivos de datos para todo tipo de organismos. Una vez validados, muchos de esos genomas se depositan en repositorios públicos como el *National Center for Biotechnology Information (NCBI)* (<https://www.ncbi.nlm.nih.gov>).

Nuestra arquitectura se sitúa exclusivamente en la fase *pos-secuenciación*. El sistema descarga los genomas ya secuenciados desde la base de datos del NCBI, los almacena en un *bucket* o servidor FTP con trazabilidad completa y, a continuación, ejecuta de forma secuencial los módulos de **control de calidad, anotación y análisis funcional y filogenético**. Los resultados se integran en análisis comparativos y se visualizan mediante gráficos interactivos que permiten a los investigadores interpretar con agilidad la información biológica. De este modo, la infraestructura abstrae la complejidad computacional y orienta el esfuerzo hacia la generación de conocimiento a partir de los datos.

3.1.1. Diagrama de Contexto

En el siguiente diagrama podemos observar que el Investigador/Biólogo es el encargado de cargar los archivos genómicos al Sistema de Análisis Genómico. Este sistema obtiene datos de bases de datos genómicas externas y envía los datos al servidor para realizar el procesamiento y análisis genómico. Después, el servidor devuelve los datos procesados al sistema, que da los resultados del análisis en formato CSV, gráficos y diagramas al Investigador/Biólogo.

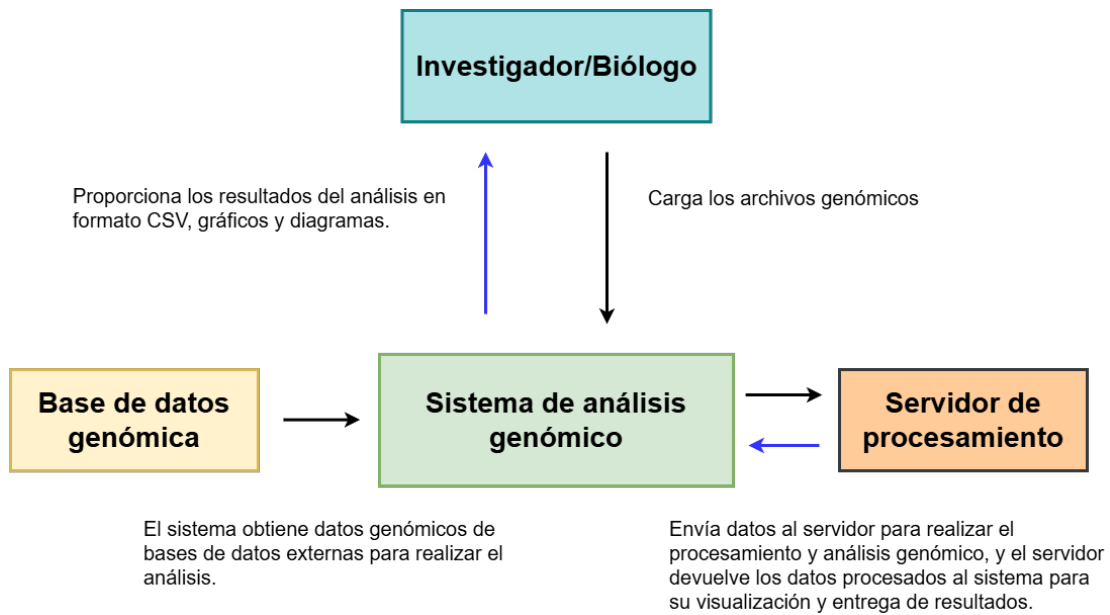


Figura 3.1: Diagrama de contexto del sistema

La Figura 3.1 ilustra el diagrama de contexto del sistema, mostrando claramente las interacciones principales entre los componentes que participan en el flujo del análisis genómico propuesto. El diagrama destaca el rol del investigador o biólogo como el usuario principal, quien carga los archivos genómicos al sistema de análisis genómico. Este sistema se comunica directamente con una base de datos genómica externa para obtener datos adicionales necesarios para realizar el análisis. A su vez, el sistema envía estos datos al servidor de procesamiento, donde se llevan a cabo los análisis filogenéticos y funcionales específicos. Finalmente, el servidor devuelve los resultados procesados al sistema, que genera reportes detallados y gráficos visuales en formatos accesibles (CSV, diagramas, gráficos) para facilitar la interpretación por parte del usuario. Esta representación permite entender fácilmente la dinámica y flujo de información entre cada uno de los componentes, enfatizando la simplicidad y claridad del proceso desde la perspectiva del usuario final.

3.1.2. Diagrama de Arquitectura

La arquitectura del sistema desarrollado para FungiGT se fundamenta en un enfoque modular y escalable, integrando herramientas de análisis genómico y filogenético dentro de un

flujo de trabajo robusto y reproducible. Los usuarios pueden cargar archivos genómicos en formatos como `.fna`, `.gff`, `.gtf`, `.bed`, `.vcf` y `.bam` a través de una interfaz web responsiva construida con **EJS** y **Tailwind CSS**, que facilita la gestión de archivos y el acceso a las funcionalidades clave de la plataforma.

A diferencia de arquitecturas tradicionales que centralizan la información en una base de datos relacional, FungiGT implementa un sistema de archivos centralizado y gestionado desde la propia plataforma. Este file system actúa como repositorio principal para las entradas y salidas de cada una de las herramientas integradas, permitiendo que los distintos módulos accedan, procesen y compartan datos de manera eficiente y transparente para el usuario. Así, se facilita la trazabilidad de los archivos y se optimiza el flujo de trabajo, ya que cada herramienta puede consumir directamente los resultados generados por las etapas previas.

En el backend, se emplean tecnologías modernas como **Node.js** y **Express.js** para implementar microservicios que gestionan las distintas etapas del flujo de trabajo. Cada microservicio se ejecuta en un contenedor **Docker**, lo que garantiza entornos consistentes y aislados, evitando conflictos de dependencias y facilitando tanto la portabilidad como la escalabilidad del sistema. Esta arquitectura basada en contenedores permite la interoperabilidad entre diferentes sistemas operativos (Windows, Linux, MacOS), ya que Docker abstrae las particularidades del sistema anfitrión y posibilita el despliegue uniforme de la plataforma en cualquier entorno moderno.

Además, la arquitectura propuesta sienta las bases para un futuro escalamiento mediante herramientas como **Kubernetes**, que permitirían balancear cargas y paralelizar operaciones de análisis a gran escala, aunque este aspecto no fue abordado en profundidad en el presente informe.

El diseño final prioriza la eficiencia, la reproducibilidad y la facilidad de uso, permitiendo a los investigadores obtener resultados precisos y gestionar grandes volúmenes de datos de manera eficiente y accesible desde cualquier sistema operativo moderno.

Para un detalle de la evolución y los cambios arquitectónicos realizados durante el desarrollo, véase la siguiente subsección.

3.1.3. Evolución de la Arquitectura

El desarrollo de FungiGT fue un proceso iterativo en el que la arquitectura del sistema se fue ajustando en función de la experiencia práctica, los desafíos técnicos y la retroalimentación del cliente. A continuación, se resumen los principales hitos y decisiones que marcaron la evolución arquitectónica del proyecto:

- **Gestión de datos mediante file system:** Inicialmente, se planteó la creación de una base de datos propia para almacenar los archivos genómicos generados y procesados por la plataforma. Sin embargo, tras analizar alternativas y observar experiencias de usuario en soluciones como Google Colab, se optó por implementar un sistema de archivos centralizado, similar a un bucket, accesible y gestionable directamente desde la plataforma. Esta aproximación resultó ser más inmersiva y útil para el usuario, permitiendo que todas las herramientas integradas pudieran acceder y compartir archivos de entrada y salida de manera directa y eficiente. Además, facilitó la interoperabilidad entre módulos y simplificó la trazabilidad de los datos a lo largo del flujo de trabajo.
- **Interoperabilidad y portabilidad:** Inicialmente, se exploró la interoperabilidad entre **WSL** (Windows Subsystem for Linux) y **PowerShell** para facilitar la ejecución de scripts y comandos en entornos mixtos. Sin embargo, la adopción de **Docker** y la arquitectura de microservicios resolvió de manera más eficiente y robusta la integración entre sistemas operativos, simplificando la experiencia de instalación y operación para los usuarios finales.
- **Cambio de herramientas de análisis filogenético:** En las primeras etapas del desarrollo, se consideró el uso de **BlastN** para el análisis filogenético. No obstante, tras evaluar las necesidades de eficiencia y escalabilidad, se optó por **BinDash**, una herramienta más adecuada para el procesamiento de grandes volúmenes de datos genómicos.
- **Visualización de resultados:** Se evaluó la integración de soluciones como **Lida** e

IGV para la visualización de datos genómicos. Finalmente, se decidió prescindir de estas herramientas y enfocar los esfuerzos en el desarrollo de un servidor principal en **Node.js** y un servidor auxiliar en **Flask** (Python). Este último se encarga de ejecutar scripts de análisis y procesar resultados, los cuales son posteriormente integrados y visualizados en la plataforma web a través del servidor Node.js, asegurando una experiencia de usuario fluida y centralizada.

- **Escalabilidad futura:** Aunque no se implementó en esta versión, la arquitectura basada en contenedores deja abierta la posibilidad de incorporar **Kubernetes** para el balanceo de carga y la paralelización de operaciones, permitiendo así el escalamiento horizontal de la plataforma en escenarios de alta demanda.

Esta evolución arquitectónica refleja la capacidad de adaptación del equipo de desarrollo y la orientación constante hacia la eficiencia, la usabilidad y la satisfacción de las necesidades reales del usuario final.

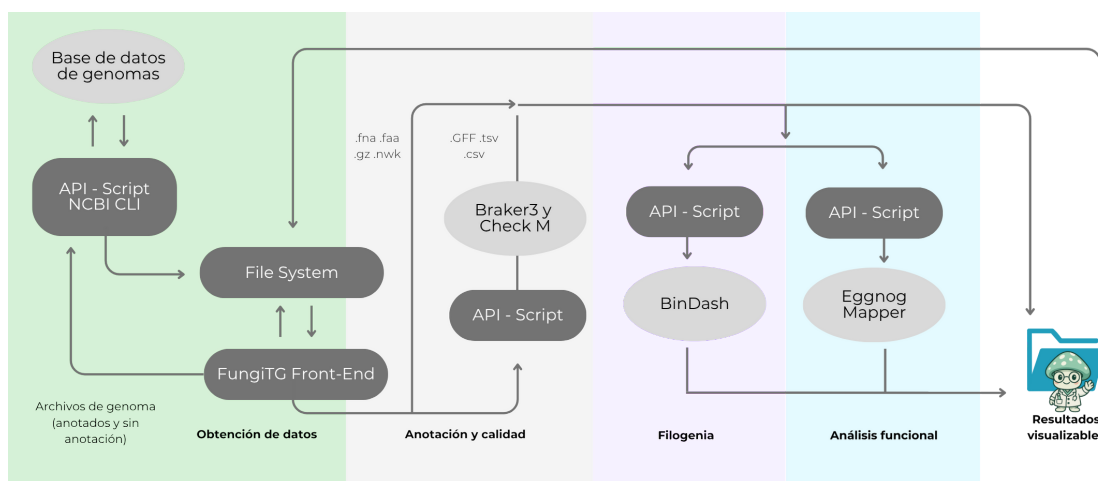


Figura 3.2: Diagrama de arquitectura del sistema

La Figura 3.2 representa el diagrama de arquitectura del sistema FungiGT, mostrando de manera estructurada los módulos principales y el flujo de trabajo integral propuesto. La arquitectura está claramente dividida en etapas clave del análisis genómico: obtención de datos, anotación y control de calidad, análisis filogenético, análisis funcional y visualización final de resultados.

Inicialmente, los datos genómicos se obtienen a través de scripts automatizados mediante la API de NCBI CLI y son almacenados en un directorio de datos interno, accesible a través del frontend del sistema. Posteriormente, estos datos pasan por procesos críticos como la anotación de genes y evaluación de calidad, utilizando herramientas especializadas como Braker3 y CheckM. Seguidamente, se realizan análisis filogenéticos utilizando BinDash y análisis funcionales mediante Egnog Mapper, ambos integrados mediante APIs y scripts personalizados que garantizan la automatización y eficiencia del proceso. Finalmente, los resultados generados en las etapas anteriores se presentan al usuario mediante un servidor Flask en python el cual integra distintos scripts para visualizar los resultados de BinDash y Egnog Mapper.

Este esquema arquitectónico resalta cómo cada módulo interactúa de forma coordinada y eficiente, asegurando una integración fluida desde la obtención inicial de datos hasta la visualización final, simplificando significativamente el trabajo de los investigadores en genómica funcional y filogenia de hongos.

3.1.4. Dockerización

3.1.4.1. Arquitectura de Microservicios

FungiGT implementa una arquitectura de microservicios completamente dockerizada, donde cada componente funcional opera como un servicio independiente y especializado:

- **MongoDB** (Puerto 27017): Base de datos NoSQL para gestión de usuarios y metadatos del sistema
- **Frontend** (Puerto 4005): Interfaz web responsiva desarrollada en Node.js con Express y EJS
- **Autenticación** (Puerto 4001): Servicio especializado de autenticación con JWT para control de acceso seguro

- **Gestión de archivos** (Puerto 4002): Manejo centralizado del file system para archivos genómicos y resultados
- **Servidor de visualización** (Puerto 4003): Servidor Flask (Python) para renderizado de gráficos y visualizaciones interactivas
- **Control de calidad** (Puerto 4004): Análisis de integridad genómica mediante CheckM integrado
- **Adquisición de datos** (Puerto 4006): Descarga automatizada de genomas desde NCBI utilizando NCBI Datasets CLI
- **Análisis filogenético** (Puerto 4007): Servicio BinDash para análisis de distancias genómicas y filogenia
- **Servidor BRAKER3** (Puerto 3004): API personalizada para anotación genómica automática
- **Análisis funcional** (Puerto 3002): Servicio eggNOG-mapper para anotación funcional de proteínas
- **Agente Dr. Fungito** (Puerto 4009): Asistente de IA para análisis inteligente y generación de reportes

3.1.4.2. Script de Instalación Automatizada

FungiGT incluye un script principal de instalación automatizada (`setup_fungigt.py`) que simplifica significativamente el proceso de despliegue y configuración del sistema completo. Este script maneja automáticamente la verificación de prerequisites, la configuración del entorno, la construcción de imágenes Docker y el inicio de servicios.

3.1.4.2.1. Opciones de Configuración: El script ofrece múltiples flags para diferentes escenarios de instalación:

- **--restart:** Reinicia servicios existentes, útil para actualizaciones o reconfiguración
- **--clean:** Realiza una limpieza completa del sistema antes de la instalación, eliminando contenedores, volúmenes e imágenes previas
- **--skip-build:** Omite la construcción de imágenes Docker, ideal para reinstalaciones rápidas cuando las imágenes ya existen
- **--start-services:** Inicia automáticamente todos los servicios después de la construcción (por defecto, solo prepara el sistema)
- **--prepare-only:** Ejecuta únicamente la preparación de archivos y construcción de imágenes, sin iniciar servicios

3.1.4.2.2. Detección Automática del Sistema: El script detecta automáticamente el sistema operativo (Windows, Linux, MacOS) y configura las variables de entorno específicas, incluyendo la gestión de permisos de Docker en sistemas Linux y la configuración de usuarios y grupos apropiados.

3.1.4.3. Gestión de Bases de Datos Especializadas

Una característica importante de FungiGT es la gestión flexible de bases de datos bioinformáticas de gran tamaño, particularmente para eggNOG-mapper:

3.1.4.3.1. Base de Datos eggNOG (Gestión Manual): El sistema incluye un script especializado (`eggnog_db_manager.py`) para la gestión de la base de datos eggNOG, que puede superar los 40 GB de tamaño. Esta base de datos es esencial para realizar anotaciones funcionales completas, pero su descarga es opcional y controlada manualmente para permitir flexibilidad en diferentes escenarios de despliegue.

- **Gestión manual:** El administrador decide cuándo descargar la base de datos mediante el script especializado

- **Uso individual vs. en la nube:** Para instalaciones individuales, el usuario puede optar por descargar según necesidad; para despliegues en la nube, la base de datos se mantiene pre-descargada
- **Verificación de integridad:** El script valida la completitud y correcta descarga de todos los archivos requeridos
- **Gestión de almacenamiento:** Verificación automática del espacio disponible y limpieza de archivos temporales
- **Reanudación de descargas:** Capacidad de reanudar descargas interrumpidas para evitar pérdida de progreso

3.1.4.4. Contenedores Especializados

La plataforma integra tanto contenedores personalizados desarrollados específicamente para FungiGT como imágenes oficiales de herramientas bioinformáticas reconocidas:

3.1.4.4.1. Contenedores Personalizados Desarrollados por el Equipo:

- **Servicios principales:** Frontend, autenticación, gestión de archivos y visualización con configuraciones optimizadas para el flujo de trabajo genómico
- **APIs especializadas:** Servidores Node.js para BRAKER3 y eggNOG-mapper que encapsulan herramientas complejas en interfaces REST
- **Análisis filogenético (BinDash):** Contenedor desarrollado internamente debido a limitaciones de las opciones disponibles en DockerHub
- **Agente de IA:** Integración de capacidades de análisis inteligente mediante Anthropic Claude API

3.1.4.4.2. Contenedores de Herramientas Oficiales:

- **BRAKER3**: `teambraker/braker3:latest` para anotación automática de genes en genomas fúngicos
- **CheckM**: `nanozoo/checkm:latest` para evaluación de completitud y contaminación genómica
- **eggNOG-mapper**: `nanozoo/eggno-mapper:2.1.9--4f2b6c0` para anotación funcional ortóloga

3.1.4.5. Gestión de Volúmenes y Persistencia

La arquitectura implementa una estrategia de volúmenes que garantiza la persistencia y compartición eficiente de datos:

- **Volumen de datos**: `./data` montado en todos los servicios de análisis, funcionando como file system centralizado
- **Persistencia MongoDB**: Volumen dedicado `mongodb_data` para garantizar la integridad de los datos de usuario
- **Volúmenes de código**: Montaje de directorios de desarrollo para facilitar el desarrollo iterativo
- **Socket Docker**: Acceso controlado a `/var/run/docker.sock` para servicios que requieren orquestación de contenedores adicionales

3.1.4.6. Red de Comunicación y Monitoreo

Todos los servicios operan dentro de la red interna `fungigt-network`, garantizando comunicación segura y eficiente entre microservicios. Cada servicio implementa healthchecks automáticos para monitoreo continuo del estado operacional, con configuraciones específicas de intervalos, timeouts y reintentos según la criticidad de cada componente.

3.1.4.7. Perfiles de Despliegue

La arquitectura soporta múltiples perfiles de despliegue para adaptarse a diferentes escenarios de uso:

- **Perfil por defecto:** Servicios principales para operación estándar de la plataforma
- **Perfil "tools":** Contenedores especializados (BRAKER3, CheckM) para análisis avanzados y operaciones manuales

3.1.4.8. Justificación Técnica y Científica

La implementación de contenedores Docker en FungiGT responde a la complejidad inherente del análisis genómico de hongos, que requiere la orquestación de múltiples herramientas bioinformáticas especializadas, cada una con requisitos específicos de dependencias y configuración. La plataforma integra tecnologías heterogéneas —desde Node.js para interfaces web hasta herramientas nativas de C++ como BRAKER3— que tradicionalmente presentan conflictos de dependencias y problemas de compatibilidad entre sistemas operativos.

Los contenedores proporcionan encapsulación completa de cada componente con sus dependencias exactas, garantizando reproducibilidad científica esencial para la validación de resultados en investigación genómica. La arquitectura de microservicios permite escalabilidad granular: módulos computacionalmente intensivos como la anotación genómica pueden ejecutarse en recursos dedicados con mayor capacidad de procesamiento, mientras que servicios de interfaz mantienen baja latencia para una experiencia de usuario óptima.

Esta flexibilidad arquitectónica es crucial para adaptarse desde estaciones de trabajo individuales hasta infraestructuras de computación de alto rendimiento, facilitando la adopción de FungiGT en diversos entornos académicos y de investigación.

Capítulo 4

Diseño Propuesto

4.1. Descripción General de Módulos

El diseño del sistema propuesto para la plataforma FungiGT se basa en decisiones estratégicas orientadas a proporcionar una estructura modular y escalable, garantizando la claridad, facilidad de uso y eficiencia operativa. Este capítulo profundiza en las razones fundamentales que justifican cada elección de diseño, explicando no solo cómo se implementó cada módulo, sino también por qué se tomó cada decisión específica.

Inicialmente, el diseño modular del sistema surgió como respuesta a la necesidad de estructurar adecuadamente un proyecto que combina la bioinformática con la ingeniería telemática. El desafío principal radicaba en la integración de conocimientos bioinformáticos complejos, introducidos por el Centro de Biotecnología Daniel Alkalay Lowitt (CB-DAL)(<https://cbdal.usm.cl/>), cliente del proyecto, con las capacidades técnicas propias del área telemática. Esta situación demandó un esfuerzo significativo en el aprendizaje y manejo de conceptos bioinformáticos específicos, tales como anotación de genes, análisis funcional y filogenético, así como métricas de calidad específicas en genómica.

Durante las primeras etapas del proyecto, se llevaron a cabo múltiples reuniones con el cliente, Esli Lobaina, para identificar y comprender sus requerimientos específicos, aclarar términos técnicos y definir claramente los objetivos. En este proceso, quedó en evidencia

que no existía previamente una herramienta integral capaz de automatizar y gestionar todo el flujo de análisis genómico requerido por el cliente. Así, surgió la necesidad imperiosa de desarrollar FungiGT como una solución única e integral.

Una de las dificultades más relevantes que enfrentó el equipo durante el desarrollo fue la limitación en la disponibilidad práctica de infraestructura dedicada exclusivamente para pruebas de sistemas bioinformáticos en etapa de creación. Si bien el Departamento de Electrónica dispone de salas con computadores que podrían eventualmente utilizarse para el despliegue de la plataforma, el proyecto se desarrolló principalmente en entornos locales, distribuidos entre los equipos personales de los integrantes del equipo y un equipo del CBDAL, lo cual fue suficiente considerando que el sistema está diseñado para funcionar inicialmente en una máquina individual.

Este enfoque permitió validar la funcionalidad general del sistema en condiciones controladas, priorizando la portabilidad y simplicidad en el despliegue. Por esta razón, se optó por utilizar Docker como tecnología base, ya que facilita el empaquetado de herramientas bioinformáticas y sus dependencias en entornos aislados, reproducibles y fácilmente trasladables entre sistemas.

No obstante, cabe señalar que aún no se ha realizado un despliegue en un entorno de producción con uso intensivo de datos, por ejemplo, en un estudio específico sobre una especie de hongo como *Trichoderma*, que involucre el análisis simultáneo de una gran cantidad de genomas. Si existiera la posibilidad de trabajar con un conjunto distribuido de equipos —como un clúster interno— se podría extender esta arquitectura utilizando orquestadores como Kubernetes para implementar balanceadores de carga y paralelizar procesos entre contenedores. Si bien esta implementación aún no se ha abordado por las condiciones actuales del proyecto, representa una proyección natural para trabajos futuros, especialmente en contextos de mayor escala de uso y colaboración entre laboratorios. A continuación, se presentan detalladamente las decisiones de diseño y justificación teórica para cada módulo del sistema propuesto, asegurando su comprensión por parte de lectores tanto del área telemática como bioinformática:

El cuadro 4.1 presenta una descripción clara y organizada de los módulos principales del

sistema propuesto en la plataforma FungiGT. Cada módulo está detallado con un propósito específico y claramente vinculado a las secciones donde se explican de manera más exhaustiva. El módulo de Control de Calidad (M1) se encarga de verificar la integridad de los archivos genómicos y generar reportes necesarios para garantizar la validez de los análisis posteriores. El módulo de Anotación (M3) facilita la identificación precisa de genes y elementos regulatorios dentro de las secuencias genómicas. Por otro lado, el módulo de Análisis Funcional (M4) asigna funciones específicas a estos genes, lo que permite una comprensión más profunda del rol biológico que desempeñan. El módulo de Análisis Filogenético (M5) construye árboles filogenéticos para explorar y entender relaciones evolutivas importantes entre organismos. Además, la plataforma integra herramientas avanzadas especializadas para análisis funcional y filogenético mediante los módulos M5 y M6, utilizando herramientas como OrthoVenn y AntiSMASH, así como métodos específicos para genomas completos y MLSA. Para gestionar de manera eficiente los datos que fluyen entre cada herramienta, el módulo de Sistema de Archivos (M9) actúa como un gestor centralizado de entrada y salida, asegurando que los resultados de un proceso se encaminen correctamente como insumos del siguiente. El módulo de Visualización de Datos (M8) genera representaciones gráficas a partir de los resultados obtenidos en los módulos de Análisis Filogenético (M5) y Análisis Funcional (M4), facilitando la interpretación de información compleja. Finalmente, el módulo Dr Fungito (M10), un agente de IA capaz de interpretar las imágenes producidas por M8, transforma estas visualizaciones en reportes inteligentes para el investigador. Esta tabla resalta la estructura modular y claramente definida del sistema, que permite una interacción intuitiva y eficiente entre los distintos componentes.

Módulo	Propósito	Sección
Adquisición (M1)	Descarga de archivos genómicos	4.1.1.1
Control de Calidad (M2)	Verifica la integridad de los archivos genómicos y genera reportes de calidad.	4.1.1.2
Anotación (M3)	Realiza la anotación de genes, ARN y elementos reguladores en las secuencias.	4.1.1.3
Análisis Funcional (M4)	Determina las funciones de los genes y categoriza según su rol biológico.	4.1.1.4
Análisis Filogenético (M5)	Construye árboles filogenéticos para estudiar relaciones evolutivas.	4.1.1.6
Integración de herramientas de análisis funcional (M5)	Integra herramientas como OrthoVenn y AntiSMASH para un análisis completo.	4.1.1.7
Integración de herramientas de análisis filogenético (M6)	Implementa herramientas para análisis de genoma completo o MLSA.	4.1.1.8
Visualización de datos (M8)	Proporciona gráficos y visualizaciones para interpretar resultados genómicos.	4.1.1.8
Sistema de archivos (M9)	Integra un explorador de archivos en la interfaz para la gestión de datos genómicos	4.1.1.9
Dr. Fungito (M10)	Integra un asistente inteligente en la interfaz para la generación de reportes	4.1.1.10

Cuadro 4.1: Descripción de módulos

4.1.1. Diseño de Módulos del Sistema

En esta sección se presenta el diseño de cada módulo del sistema.

4.1.1.1. Adquisición (M1)

El primer paso fundamental para realizar un análisis genómico es la adquisición de secuencias genómicas completas y de calidad. Para ello, se requiere obtener genomas previamente secuenciados desde bases de datos públicas que garanticen información confiable y actualizada. En el contexto de este proyecto, se optó por utilizar la herramienta NCBI Datasets

en su versión de línea de comandos (CLI), la cual permite automatizar la descarga de genomas de manera eficiente mediante el uso de números de acceso (accession numbers).

Los números de acceso funcionan como identificadores únicos asignados a cada genoma dentro de la base de datos del NCBI, lo que facilita localizar y descargar específicamente los genomas de interés sin ambigüedad. Esta estrategia no solo reduce la posibilidad de error humano al seleccionar los archivos, sino que también permite replicar el proceso en distintos entornos de manera consistente.

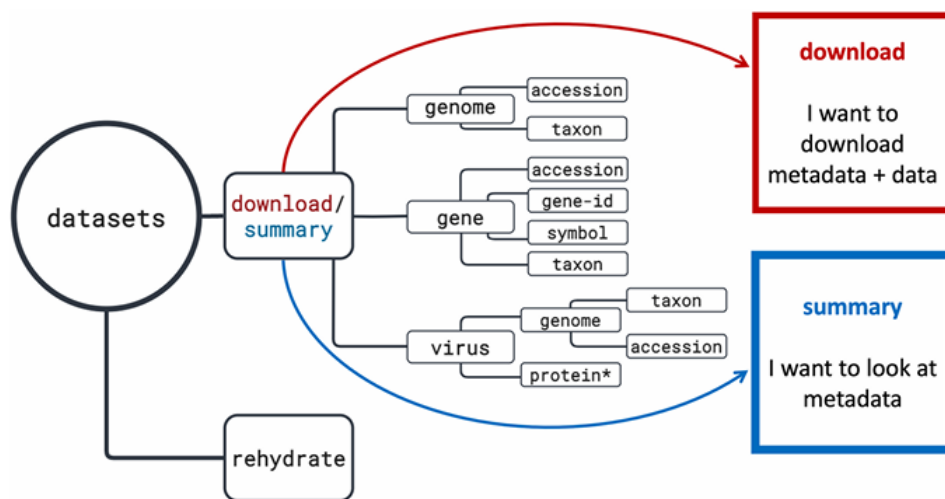


Figura 4.1: Comandos disponibles de NCBI Datasets CLI

El uso de NCBI Datasets CLI permite no solo obtener la secuencia genómica, sino también acceder de forma sencilla y estructurada a los archivos asociados con información esencial para el análisis, como metadatos taxonómicos, versiones de ensamblaje y diferentes formatos de archivo (FASTA, GenBank). Como se muestra en la Figura 4.1, el comando principal datasets ofrece dos subcomandos clave:

- **download:** permite descargar tanto los datos genómicos como todos los metadatos asociados. Por ejemplo, se pueden obtener secuencias completas junto a la información taxonómica y de ensamblaje de un organismo específico.

- **summary:** proporciona únicamente un resumen de los metadatos, ideal para inspeccionar información antes de realizar una descarga masiva.

Por ejemplo, para descargar el genoma de un hongo específico del género *Trichoderma*, se puede ejecutar el siguiente comando:

```
datasets download genome taxon "Trichoderma virens"
```

Este comando descargará el genoma y todos los archivos asociados (incluyendo archivos FASTA y GenBank, y un archivo con metadatos detallados) directamente desde la base de datos de NCBI, asegurando así que los datos de entrada para el análisis sean válidos, completos y fácilmente trazables hasta su origen.

Esta funcionalidad resulta fundamental para la reproducibilidad y confiabilidad de los análisis realizados en la plataforma, sentando una base sólida para todos los módulos siguientes del sistema, la referencia a esta imagen y a la herramienta como tal se encuentra en [12].

4.1.1.2. Control de Calidad (M2)

La necesidad de evaluar la integridad y calidad de los archivos genómicos surge debido a que cualquier error o dato incompleto en las secuencias puede afectar drásticamente los análisis posteriores. Este módulo implementa métricas críticas en genómica, como N50 (indicador clave que mide la continuidad de los ensamblajes), número y longitud media de contigs. Estos parámetros permiten al usuario verificar rápidamente si los datos obtenidos cumplen con los estándares mínimos para realizar análisis fiables.

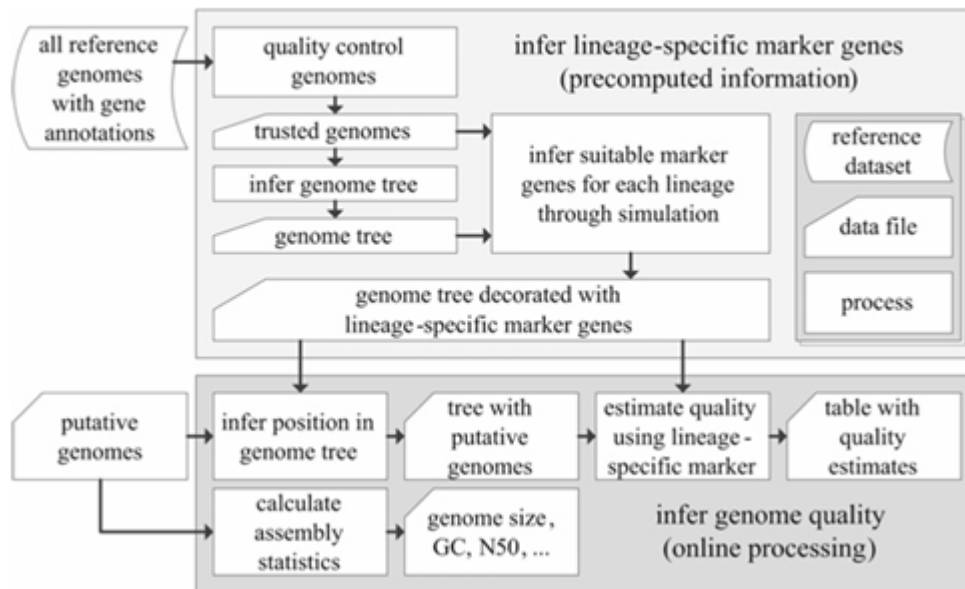


Figura 4.2: Workflow de checkM

La referencia esta imagen se puede encontrar en [11] y esta presenta el flujo de trabajo general para la inferencia de la calidad genómica utilizando genes marcadores específicos de linaje, un enfoque implementado por herramientas como CheckM. El diagrama se divide en dos procesos principales: la generación de información precomputada y el procesamiento en línea de genomas putativos.

En la parte superior del diagrama se observa cómo, a partir de todos los genomas de referencia con anotaciones génicas, se seleccionan genomas confiables que pasan por un control de calidad. A continuación, estos genomas son utilizados para inferir un árbol filogenético de referencia. Mediante simulaciones, se identifican genes marcadores apropiados para cada linaje evolutivo, decorando así el árbol filogenético con estos genes específicos. Toda esta información precomputada se almacena en conjuntos de datos y archivos de referencia que serán usados en etapas posteriores.

En la parte inferior del diagrama se muestra el procesamiento en línea, donde genomas putativos (nuevos ensamblajes) se evalúan. Primero, se calcula su posición dentro del árbol filogenético de referencia y se obtienen estadísticas de ensamblaje como tamaño del genoma,

porcentaje de GC, y N50. Posteriormente, el genoma se incorpora al árbol junto con otros genomas putativos y se estima su calidad basándose en la presencia y características de los genes marcadores específicos de linaje. El resultado final es una tabla con estimaciones de calidad para cada genoma evaluado.

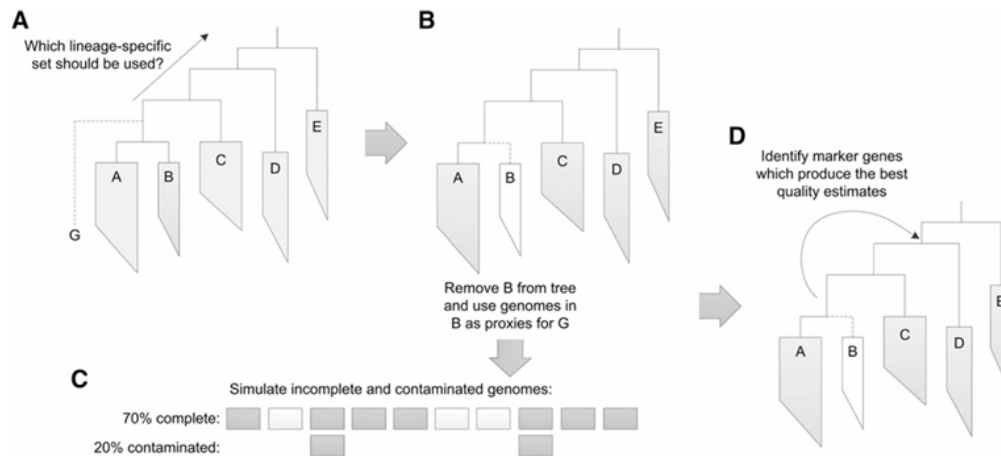


Figura 4.3: Proceso de selección de genes marcadores

La Figura 4.3 ilustra el proceso de selección y validación de genes marcadores específicos de linaje para la estimación de calidad genómica, utilizando simulaciones filogenéticas. El flujo está dividido en varios pasos secuenciales:

- **A:** Se parte de un árbol filogenético que agrupa diferentes genomas de referencia (A, B, C, D, E) y un genoma objetivo (G) cuya calidad se desea evaluar. El objetivo inicial es determinar cuál conjunto de genes marcadores específicos de linaje es el más apropiado para usar en la evaluación de G.
- **B:** Para simular la evaluación realista de genomas no caracterizados, se elimina temporalmente el genoma B del árbol y se utilizan los genomas de su mismo grupo filogenético como referencias proxy para el genoma objetivo G. Este procedimiento permite probar el sistema en ausencia de información directa sobre G.
- **C:** Se generan simulaciones de genomas con diferentes niveles de integridad y contaminación, por ejemplo, genomas con 70 % de completitud o con 20 % de contaminación.

ción. Estas simulaciones permiten calibrar y validar la precisión de los estimadores de calidad.

- **D:** Finalmente, se identifican aquellos genes marcadores que generan las mejores estimaciones de calidad, evaluando su desempeño en las simulaciones anteriores y seleccionando el conjunto óptimo para su uso en la evaluación real de genomas.

Este proceso es fundamental para optimizar la precisión en la estimación automatizada de la calidad genómica, ya que permite seleccionar de manera informada los genes marcadores más confiables para cada linaje evolutivo, la referencia esta imagen se puede encontrar en [11].

4.1.1.3. Anotación (M3)

La anotación es un proceso fundamental en la genómica, ya que permite identificar elementos específicos dentro del genoma, como genes y regiones reguladoras. Las herramientas elegidas para este módulo (Braker3, RAST y AUGUSTUS) fueron seleccionadas por su reconocida precisión y eficacia en la identificación automática de genes, ARN ribosomales y ARN de transferencia, esenciales para el análisis funcional posterior.

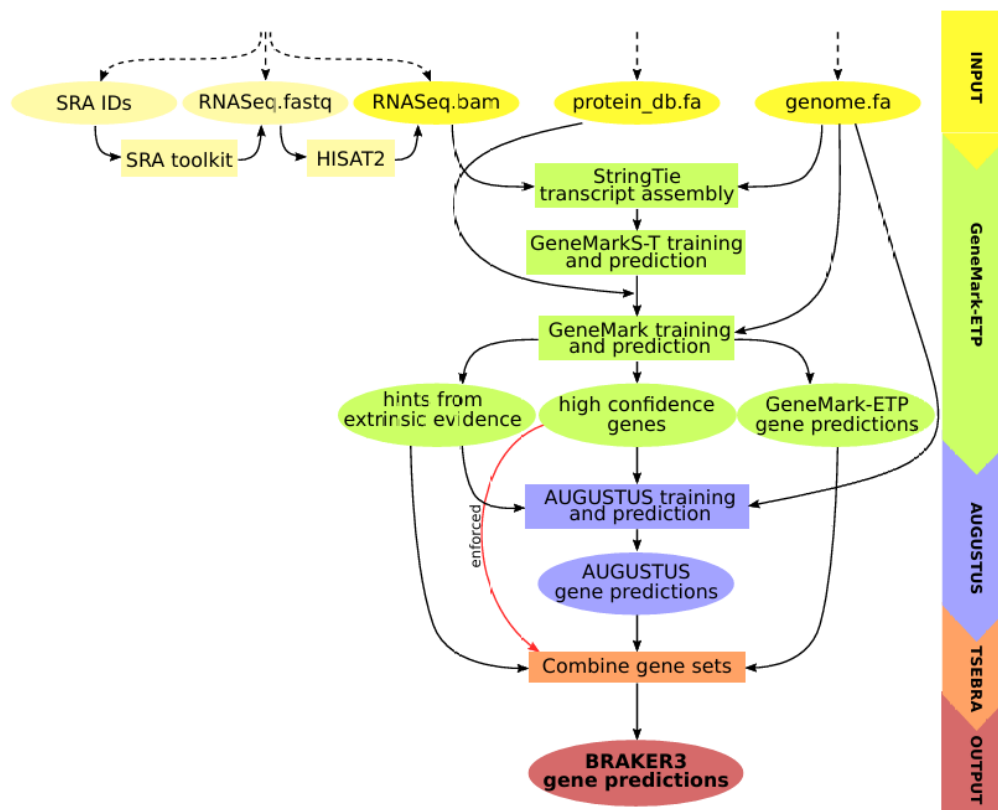


Figura 4.4: Workflow de Braker3

La Figura 4.4 muestra el flujo de trabajo de anotación genómica automatizada realizado por BRAKER3, integrando evidencia transcriptómica y proteica para lograr predicciones de genes de alta calidad.

El proceso inicia con diferentes fuentes de entrada (*input*), que incluyen archivos de secuencias genómicas (genome.fa), bases de datos de proteínas (protein_db.fa), y datos de ARN-seq ya sea en formato fastq, bam o mediante identificadores SRA. Las secuencias de ARN-seq pueden ser procesadas utilizando herramientas como SRA toolkit o alineadas al genoma con HISAT2.

Posteriormente, la herramienta StringTie se emplea para ensamblar transcritos a partir de los datos de ARN-seq. Estos transcritos y los datos de proteínas sirven como base para el entrenamiento y predicción inicial de genes mediante GeneMarkS-T y GeneMark-ETP. Se generan así un conjunto de genes de alta confianza y un set de predicciones genéticas

usando diferentes tipos de evidencia (*hints from extrinsic evidence*).

La siguiente etapa corresponde a la integración de los resultados en el módulo AUGUSTUS, que realiza un entrenamiento específico con los datos procesados y produce predicciones finales de genes. Finalmente, la herramienta TSEBRA combina los conjuntos de genes generados por GeneMark y AUGUSTUS, para entregar el resultado final de anotaciones genéticas integradas.

Este workflow referente a [13] demuestra cómo BRAKER3 automatiza y unifica todas las etapas necesarias para obtener anotaciones precisas, integrando información de múltiples fuentes y optimizando cada fase del proceso.

4.1.1.4. Análisis Funcional (M4)

Este módulo proporciona información crítica sobre el rol biológico de los genes anotados, utilizando bases de datos estandarizadas como KEGG, GO y COG. Se seleccionaron herramientas como OrthoVenn y BlastKOALA por su capacidad para categorizar claramente funciones génicas y facilitar análisis comparativos, fundamentales para estudios biológicos detallados.

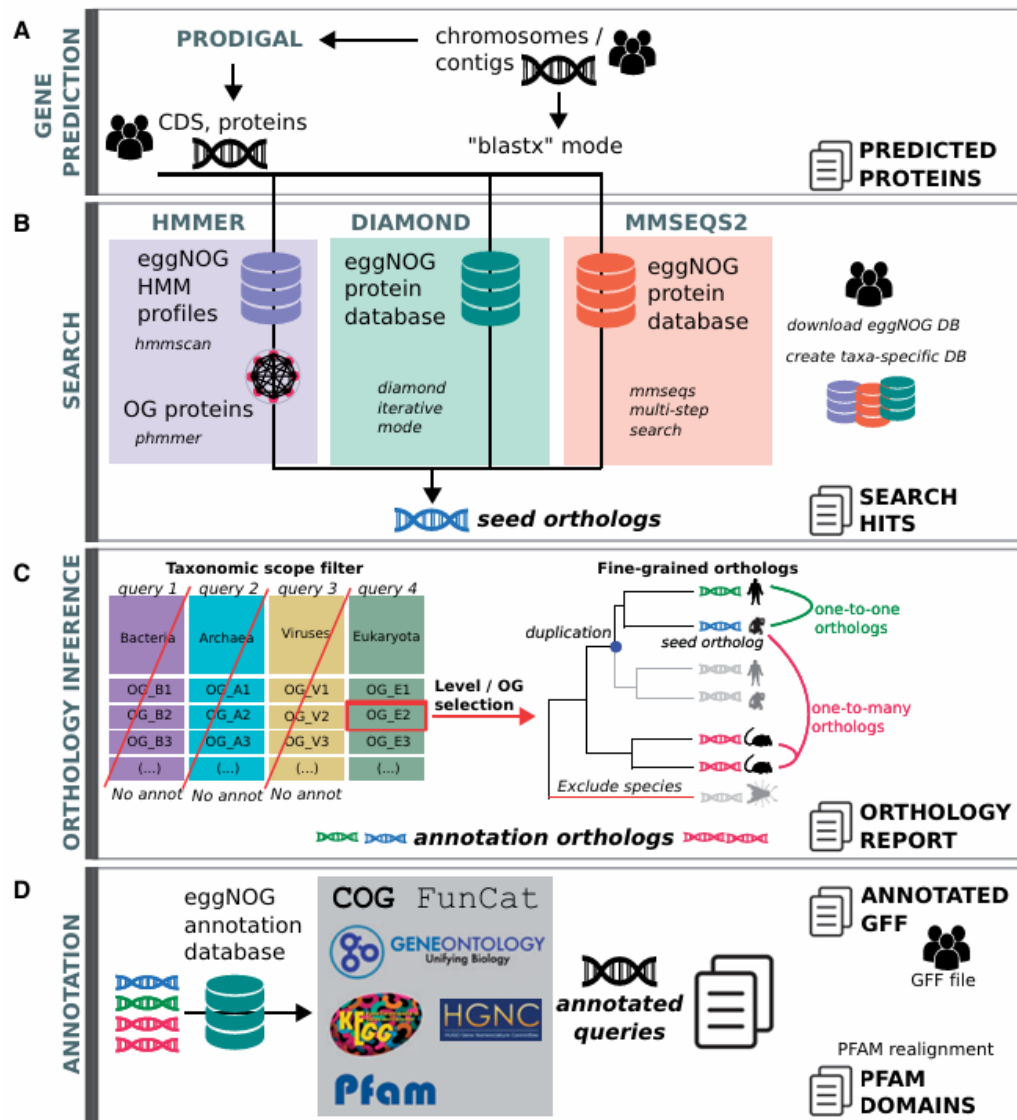


Figura 4.5: Workflow de Egnog Mapper

La Figura 4.5 representa el flujo de trabajo completo de anotación funcional automatizada utilizando *eggNOG-mapper*, integrando predicción de genes, búsqueda de ortólogos, inferencia de ortología y generación de anotaciones.

- **A. Predicción génica:** El proceso comienza con la identificación de genes codificantes (CDS) y proteínas a partir de genomas completos (cromosomas o contigs), utilizando herramientas como *Prodigal* o mediante el modo *blastx*. El resultado

es un conjunto de proteínas predichas listas para su análisis posterior.

- **B. Búsqueda de ortólogos:** Las proteínas predichas se comparan contra bases de datos de proteínas y perfiles HMM de eggNOG utilizando herramientas como HMMER, DIAMOND y MMseqs2. Estos métodos identifican proteínas homólogas en bases de datos bien curadas, generando una lista de ortólogos semilla (seed orthologs) para cada consulta.
- **C. Inferencia de ortología:** El sistema aplica filtros taxonómicos para seleccionar el grupo ortólogo (OG) más adecuado, considerando el alcance taxonómico de cada consulta (bacterias, arqueas, virus, eucariotas). Posteriormente, se lleva a cabo una inferencia filogenética fina que distingue entre ortólogos uno-a-uno (one-to-one) y ortólogos uno-a-muchos (one-to-many), descartando especies irrelevantes y duplicaciones, para así seleccionar los mejores ortólogos anotados.
- **D. Anotación funcional:** Finalmente, los genes y proteínas consultados son anotados mediante la base de datos de eggNOG, asociando información funcional proveniente de múltiples bases de datos reconocidas como COG, FunCat, Gene Ontology, KEGG, HGNC y Pfam. El resultado incluye archivos anotados en formato GFF y dominios PFAM, proporcionando una descripción exhaustiva y funcional de los elementos genéticos identificados.

Este flujo de trabajo destaca cómo *eggNOG-mapper*[14] automatiza desde la predicción inicial de proteínas hasta la obtención de anotaciones funcionales detalladas, facilitando estudios de genómica comparativa y análisis funcional en grandes conjuntos de datos genómicos.

4.1.1.5. Análisis Filogenético (M5)

El módulo de análisis filogenético fue desarrollado con el objetivo de comprender las relaciones evolutivas entre especies de hongos a partir de la comparación de sus genomas completos. Para este propósito, se implementó la herramienta BinDash [15], que permite

calcular de manera eficiente y precisa la distancia genética entre múltiples genomas, incluso en computadores personales de recursos limitados.

A diferencia de métodos tradicionales basados en alineamiento de genes específicos, BinDash emplea técnicas basadas en el cálculo de distancias utilizando sketches de k-mers (fragmentos de longitud fija) para estimar rápidamente la similitud global entre genomas completos. Este enfoque es especialmente útil en genómica comparativa y filogenética de hongos, ya que evita la necesidad de alineamientos exhaustivos y permite analizar grandes conjuntos de datos de manera escalable.

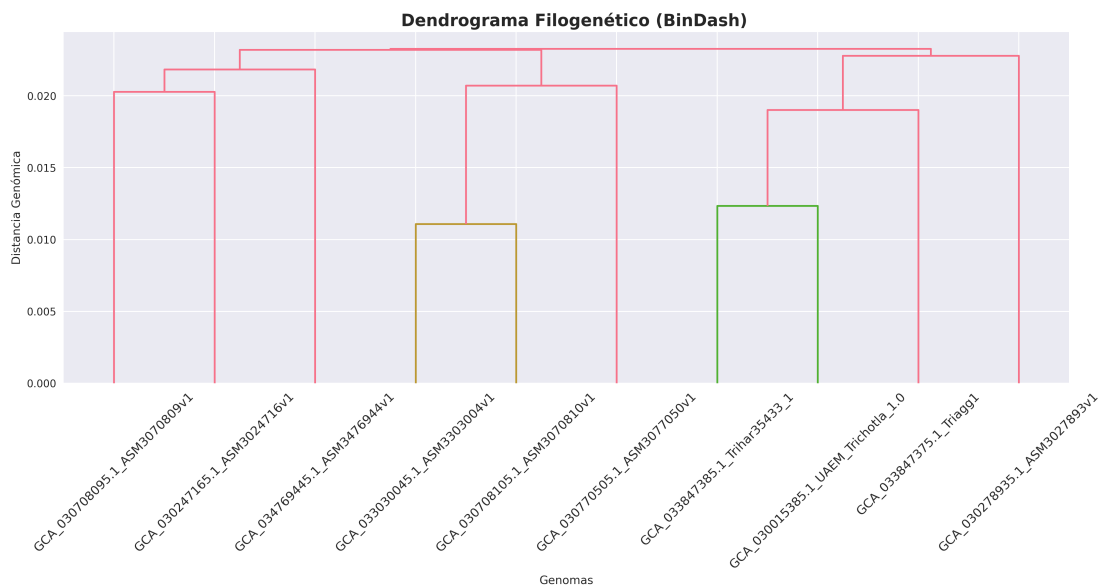


Figura 4.6: Ejemplo de matriz de distancias genómicas generada por BinDash para especies fúngicas.

En la Figura 4.6 se muestra una matriz de distancias genómicas producida por BinDash a partir de un conjunto de genomas de diferentes especies de hongos. Cada celda de la matriz representa el grado de similitud (o diferencia) entre pares de genomas, calculado a partir del análisis de sus k-mers compartidos.

El uso de BinDash en la plataforma FungiGT resulta altamente relevante porque permite construir matrices de distancias y dendrogramas filogenéticos de forma rápida, robusta y reproducible, facilitando la exploración de relaciones evolutivas, la identificación de agru-

pamientos taxonómicos y la comparación directa de genomas completos sin depender exclusivamente de genes marcadores individuales. Esta metodología amplía las capacidades analíticas de la plataforma y garantiza resultados confiables y escalables en el análisis filogenético de hongos.

4.1.1.6. Integración de herramientas de análisis funcional (M6)

Este módulo fue diseñado para facilitar la integración y automatización de análisis funcionales avanzados dentro de la plataforma FungiGT. Tras evaluar diversas herramientas bioinformáticas, se optó finalmente por integrar *eggNOG-mapper* (eggNOGer) como solución principal, dada su robustez y capacidad para realizar anotaciones funcionales y asignación de ortología a escala genómica. Si bien se exploraron otras opciones como OrthoVenn y AntiSMASH, la estructura modular del sistema permitió separar este componente para gestionar su despliegue y operación de manera independiente.

La integración se realizó a través de un contenedor Docker dedicado, cuyo `Dockerfile` instala únicamente los componentes necesarios de *eggNOG-mapper*. No obstante, la base de datos utilizada por *eggNOG-mapper* (aproximadamente 50 GB) no se incluye directamente en el contenedor para evitar aumentar excesivamente el tiempo y tamaño de instalación del proyecto completo. En su lugar, se proporciona un script opcional para que el usuario descargue la base de datos según necesidad. Esto asegura que el sistema sea más ligero y flexible, permitiendo personalizar la instalación en función de los recursos disponibles y de los requerimientos de cada usuario.

4.1.1.7. Integración de herramientas de análisis filogenético (M7)

Este módulo fue implementado para automatizar y simplificar los análisis filogenéticos a gran escala en la plataforma. Inicialmente se consideraron herramientas como *pyANI* y *BLASTn* para la comparación de genomas, pero finalmente se seleccionó *BinDash* debido a su capacidad para generar matrices de distancias y dendrogramas filogenéticos de forma rápida y eficiente, comparando todos los genomas entre sí sin requerir una base de datos

externa.

La integración de BinDash en FungiGT se realizó mediante un contenedor Docker dedicado, que expone su propio puerto y se comunica con el resto del sistema de manera transparente para el usuario.

4.1.1.8. Visualización de datos (M8)

Este módulo es esencial porque facilita la interpretación rápida y efectiva de resultados complejos mediante gráficos interactivos (mapas de calor, diagramas de vías metabólicas). Se seleccionaron específicamente herramientas gráficas que simplifican considerablemente la visualización, permitiendo al usuario tomar decisiones informadas basadas en los análisis realizados.

4.1.1.9. Sistema de archivos (M9)

Adicionalmente, se introdujo un módulo de gestión de archivos con el objetivo claro de mejorar significativamente la experiencia del usuario. Este módulo permite una navegación sencilla dentro de la plataforma web para visualizar archivos de entrada y salida, evitando la necesidad de acceder manualmente al explorador del sistema operativo. Esta decisión mejora notablemente la usabilidad y comodidad para los usuarios finales.

4.1.1.10. Dr. Fungito (M10)

El módulo Dr. Fungito constituye un agente de inteligencia artificial especializado en genómica de hongos, diseñado para asistir a los usuarios investigadores tanto en consultas sobre el funcionamiento general de la plataforma FungiGT como en la interpretación inteligente de los gráficos generados por el módulo de visualización (M8). El sistema utiliza la API de Anthropic con el modelo Claude Sonnet 4 para análisis multimodal, permitiendo procesar tanto texto como imágenes de manera integrada.

Cuando los usuarios generan visualizaciones en los módulos de análisis, estas imágenes se transforman automáticamente a formato base64 mediante funciones JavaScript que capturan el contenido del canvas de renderizado y lo codifican en texto ASCII para su transmisión eficiente al servidor de Dr. Fungito. El agente puede interpretar matrices de distancias genómicas de BinDash, mapas de calor de anotaciones funcionales de eggNOG-mapper, y otros tipos de gráficos científicos, proporcionando análisis biológicos específicos, identificación de patrones relevantes, y recomendaciones para análisis adicionales.

Una funcionalidad destacada es la generación automática de reportes inteligentes en formato PDF que integran las interpretaciones de múltiples imágenes analizadas. Actualmente, el sistema de reportes está optimizado específicamente para análisis de BinDash y eggNOG-mapper, incluyendo las imágenes originales junto con sus interpretaciones detalladas. Adicionalmente, Dr. Fungito mantiene memoria contextual mediante MongoDB, permitiendo consultas sobre el uso de la plataforma, navegación entre módulos, y proporcionando continuidad en las conversaciones para investigaciones longitudinales complejas.

4.2. Diseño de Interfaz

La interfaz propuesta tiene como objetivo central facilitar la interacción intuitiva del usuario con los distintos módulos y funcionalidades del sistema. En etapas iniciales del proyecto, se planteó la posibilidad de desarrollar una interfaz inteligente basada en un modelo avanzado de lenguaje natural (LLM), que pudiera interpretar consultas específicas de genómica en lenguaje natural y traducirlas automáticamente en comandos ejecutados sobre una consola o terminal. El propósito de esta idea inicial era permitir al usuario realizar análisis y visualizar resultados de manera fluida y automatizada, usando un chatbot similar a lo que posteriormente se conocería como Model Context Protocol (MCP).

Sin embargo, esta propuesta inicial resultó demasiado ambiciosa dada la complejidad técnica, el entrenamiento específico requerido y las limitaciones tecnológicas existentes durante el desarrollo del proyecto en 2024. Por esta razón, se decidió optar por una solución más práctica y realista, enfocándose en una interfaz gráfica tradicional y estructurada, similar a

plataformas bioinformáticas ya establecidas, como Galaxy. Esta decisión permitió al equipo avanzar con mayor eficacia, garantizando resultados concretos y funcionales.

Actualmente, la plataforma FungiGT expone claramente sus funcionalidades mediante endpoints REST, lo que facilita considerablemente la futura implementación de un sistema MCP. Aunque dicha implementación no es abordada directamente en este proyecto, la arquitectura modular y la disponibilidad de estos endpoints proporcionan una base sólida para futuras integraciones inteligentes, potenciando así la evolución tecnológica de la plataforma.

El diseño final consideró especialmente la facilidad de uso, accesibilidad y claridad para el usuario final, aprovechando tecnologías modernas y gráficas interactivas. Asimismo, se tuvo en cuenta desde el inicio la alta demanda computacional inherente a las herramientas bioinformáticas integradas, debido a algoritmos específicos como búsquedas por alineamiento de secuencias (BlastN), predicción probabilística de genes (AUGUSTUS, Braker3) o categorización funcional avanzada (Eggnog Mapper). Para abordar estos desafíos técnicos, se emplearon contenedores Docker que aseguran una eficiente gestión de recursos, estabilidad del entorno y escalabilidad operativa.

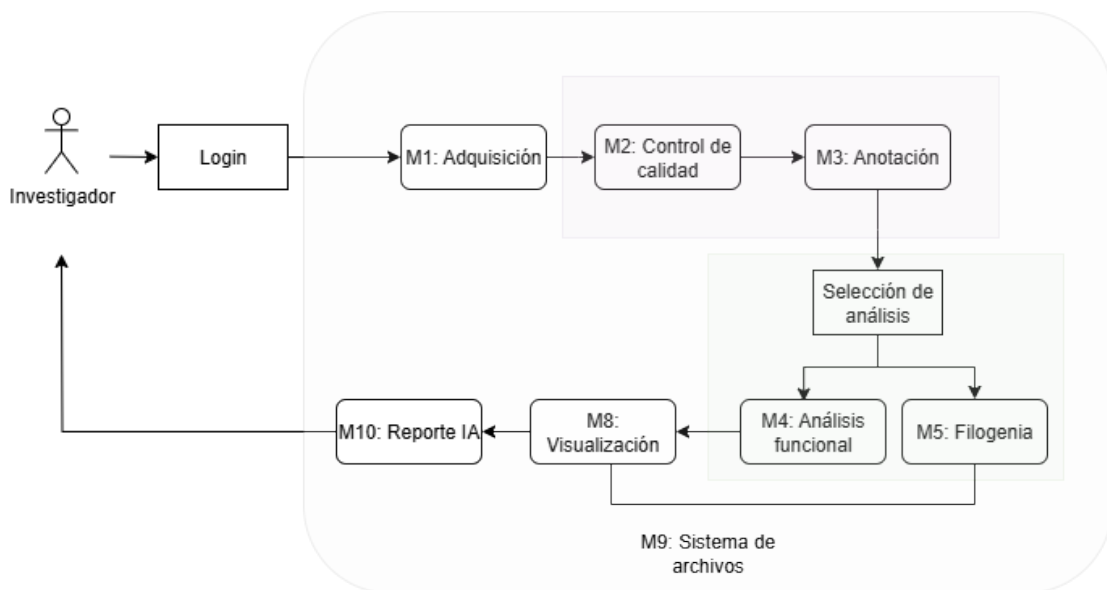


Figura 4.7: Modelo de navegación del sistema

En la figura 4.7 se presenta el modelo de navegación de la plataforma FungiGT. Si bien el

flujo general coincide con el planteado en el diagrama de arquitectura de la figura 3.2 , el objetivo en este caso es representar de forma explícita cómo interactúa el usuario con cada uno de los módulos del sistema.

Los módulos funcionales están representados mediante cajas de bordes curvos, cada una con una breve descripción de su propósito. Por otro lado, las cajas de bordes rectos indican las distintas vistas o interfaces gráficas que componen la plataforma. Cabe destacar que algunas vistas agrupan múltiples herramientas, lo cual se refleja en el modelo.

Por ejemplo, en la vista correspondiente al módulo de análisis funcional, se integran tanto *eggNOG-mapper* como *BlastN*. En este contexto, el usuario debe seleccionar el tipo de análisis que desea realizar, lo que a su vez determina el tipo de archivo de salida que se generará. Este modelo permite visualizar claramente cómo se estructura el flujo de trabajo desde el punto de vista de la experiencia del usuario.

Capítulo 5

Resultados

En esta sección se presentan los diferentes casos de uso correspondientes a cada uno de los módulos principales de la aplicación, abarcando desde el proceso de instalación y la utilización de la plataforma, hasta la integración de contenedores Docker. Cada caso de uso es validado detalladamente, incluyendo la descripción de su flujo de operación, precondiciones, actores involucrados y postcondiciones esperadas. Además, se proporciona una justificación argumentada para cada caso de uso específico, demostrando la correcta implementación y funcionamiento de las funcionalidades clave del sistema. mismo validando argumentativamente el caso de uso específico

5.1. Caso de Uso: Adquisición + Sistema de archivos

5.1.1. Objetivo del módulo

Ese modulo es importante ya que nos permite descargar correctamente el genoma de nuestro interes teniendo su numero de accesion correspondiente a la base de datos de NCBI.

5.1.2. Caso de uso M1+M9

- **Actor:** Usuario investigador

- **Precondición:** El usuario ha ingresado correctamente a la plataforma.
- **Flujo principal:**
 1. El usuario ingresa al módulo de adquisición.
 2. Selecciona el número de accesion.
 3. Hace clic en "Descargar genoma".
 4. El sistema devuelve el archivo descargado en el directorio data visible en el sistema de archivos.
- **Postcondición:** Se muestran mensajes de validación de la descarga.

5.1.3. Diagrama de caso de uso M1+M9

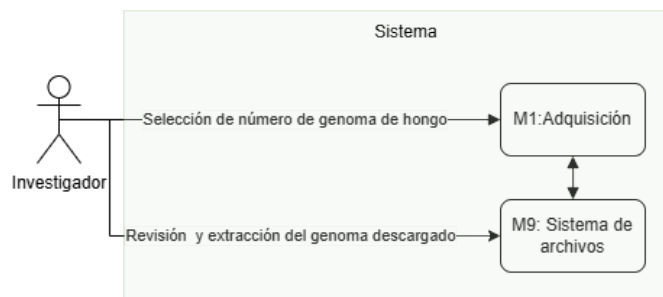


Figura 5.1: Caso de uso del módulo de Adquisición

5.1.4. Evidencia funcional

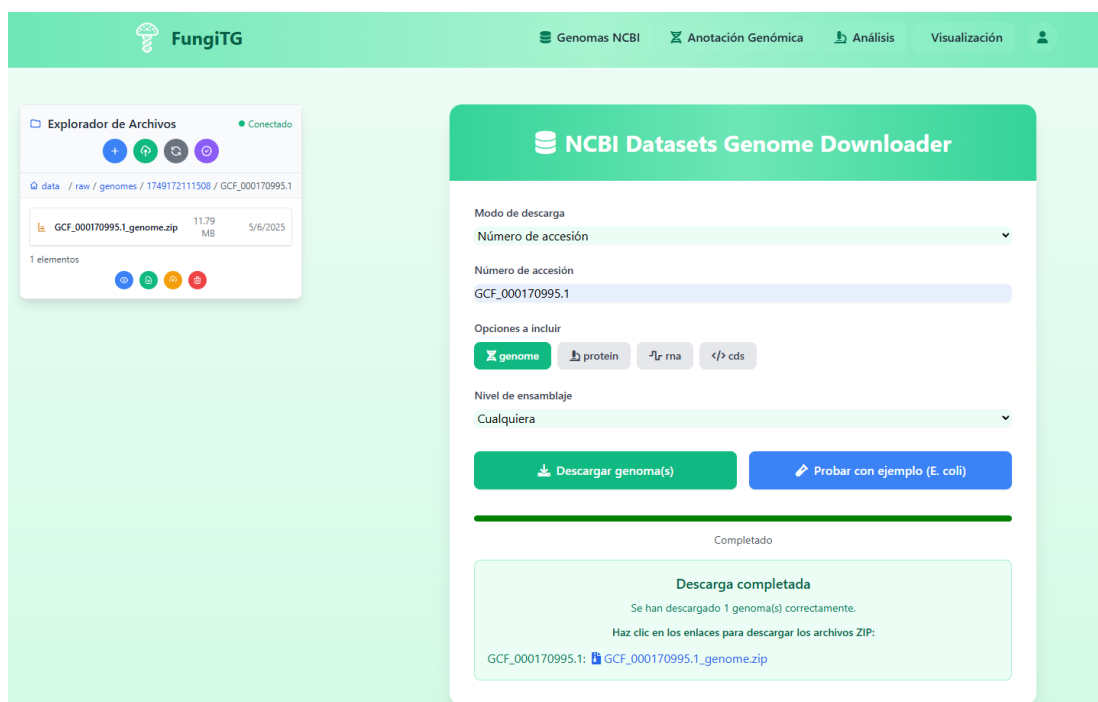


Figura 5.2: Vista del módulo de adquisición con el resultado de la descarga de *Trichoderma virens* Gv29-8

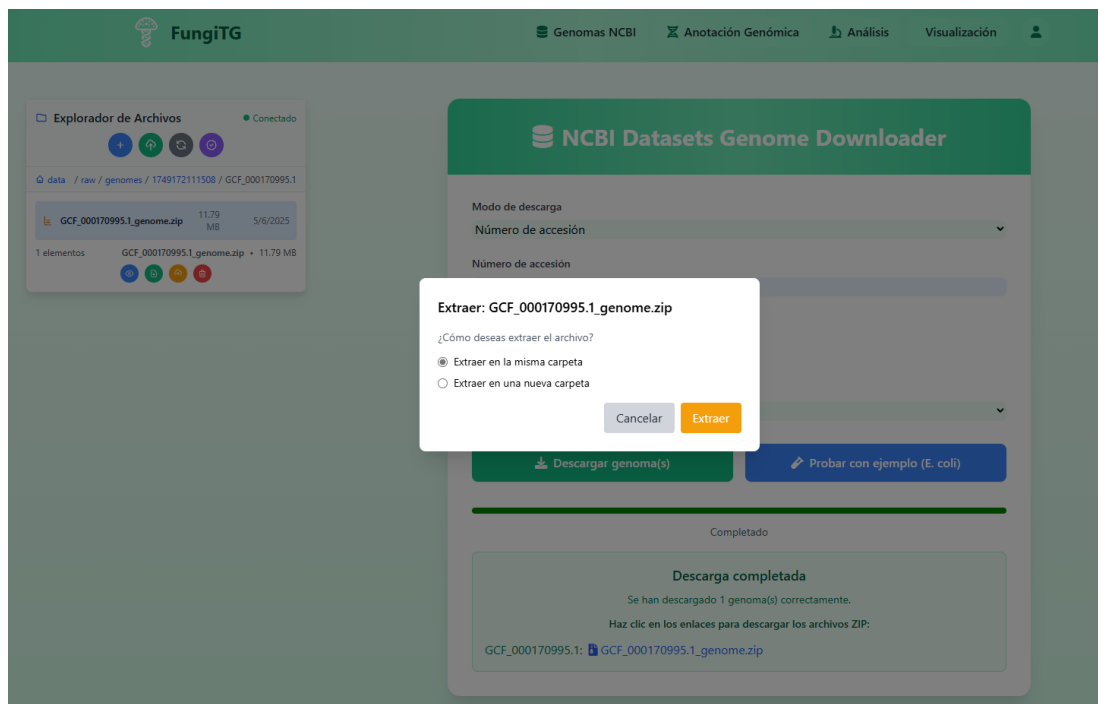


Figura 5.3: Vista del módulo de adquisición en el proceso de extracción del .zip que contiene el archivo .fna



Figura 5.4: Vista del módulo de sistema de archivos en donde se evidencia la existencia del archivo .fna para realizar anotaciones

5.1.5. Validación del caso de uso M1+M9

Las Figuras 5.2, 5.3 y 5.4 ilustran la ejecución exitosa y la validación del flujo correspondiente a este caso de uso específico. En este escenario, se verifica que la plataforma FungiGT permite acceder y descargar correctamente cualquier genoma de hongo previamente secuenciado y disponible en la base de datos de NCBI. Asimismo, se demuestra que la descarga y extracción del archivo genómico pueden realizarse directamente desde la propia plataforma, facilitando el acceso y la gestión de los datos para el usuario final.

5.2. Caso de Uso: Calidad y Sistema de Archivos

5.2.1. Objetivo del módulo

Este módulo es fundamental porque permite calcular métricas clave para evaluar la calidad de los genomas antes de proceder con las siguientes etapas del análisis. Garantizar la integridad y confiabilidad de los datos de entrada es esencial para obtener resultados válidos en la anotación y análisis funcional o filogenético.

5.2.2. Caso de uso M2+M9

- **Actor:** Usuario investigador
- **Precondición:** El usuario ha accedido correctamente a la plataforma y dispone de un archivo genómico descargado.
- **Flujo principal:**
 1. El usuario accede al módulo de Calidad.
 2. Selecciona la carpeta de entrada donde se encuentra el archivo del genoma en formato .fna.

3. Configura los parámetros para el análisis de calidad, como el tipo de workflow, el número de hilos y otras opciones avanzadas necesarias.
 4. Ejecuta el proceso de control de calidad mediante CheckM.
 5. El sistema procesa la información y devuelve los reportes detallados sobre la calidad del genoma.
- **Postcondición:** El usuario visualiza los registros y reportes generados, pudiendo así validar la calidad del genoma antes de proceder con la anotación.

5.2.3. Diagrama de caso de uso M2+M9

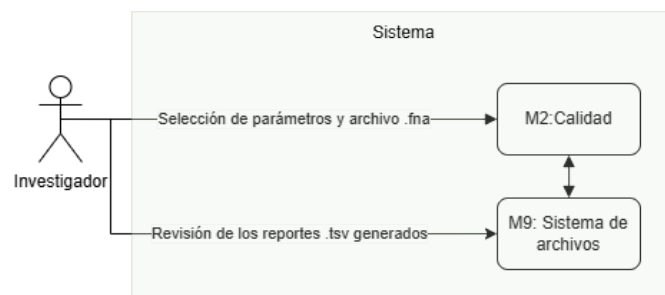


Figura 5.5: Caso de uso del módulo de Calidad y Sistema de Archivos

5.2.4. Evidencia funcional

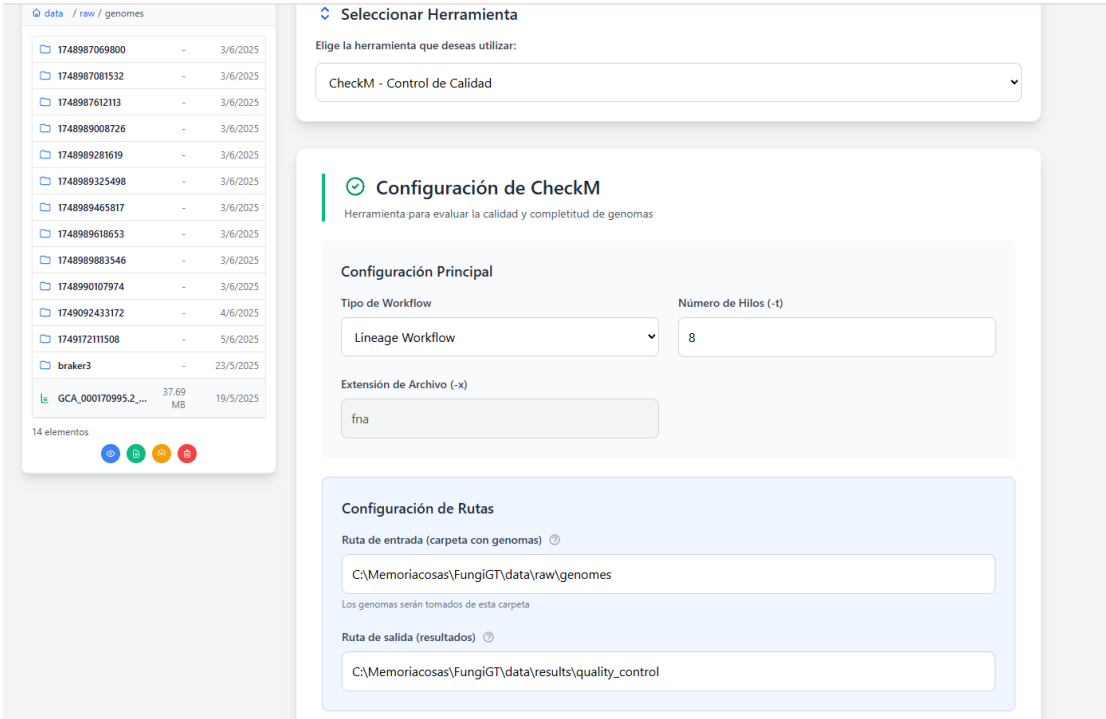


Figura 5.6: Interfaz del módulo de calidad, mostrando la configuración principal y selección de rutas de entrada y salida.

Opciones Avanzadas

☒ --reduced_tree
 ☐ --quiet
 ☐ --tab_table

Subir Archivos

Subir archivos .fna a la carpeta de entrada

Elegir archivos

No se eligió ningún archivo

Subir Archivos

Mostrar Comando

Ejecutar CheckM

Comando Docker:

```

docker run --rm -v "C:\Memoriacosas\FungiGT\data\raw\genomes":/input -v
"C:\Memoriacosas\FungiGT\data\results\quality_control":/output nanozoo/checkm:latest checkm lineage_wf -t 8
-x fna --reduced_tree /input /output
          
```

Estado de Anotaciones y Análisis de Calidad

TIPO DE ANÁLISIS	ESTADO	PROGRESO
Control de Calidad (CheckM)	Ejecutando CheckM	10% <pre> ***** Iniciando análisis: Colocando genomas en árbol de referencia Identificando genes marcadores en genomas ***** Iniciando análisis: Colocando genomas en árbol de referencia Identificando genes marcadores en genomas </pre>

Figura 5.7: Vista del módulo de calidad donde se configuran opciones avanzadas, revisión del comando y ejecución de CheckM.

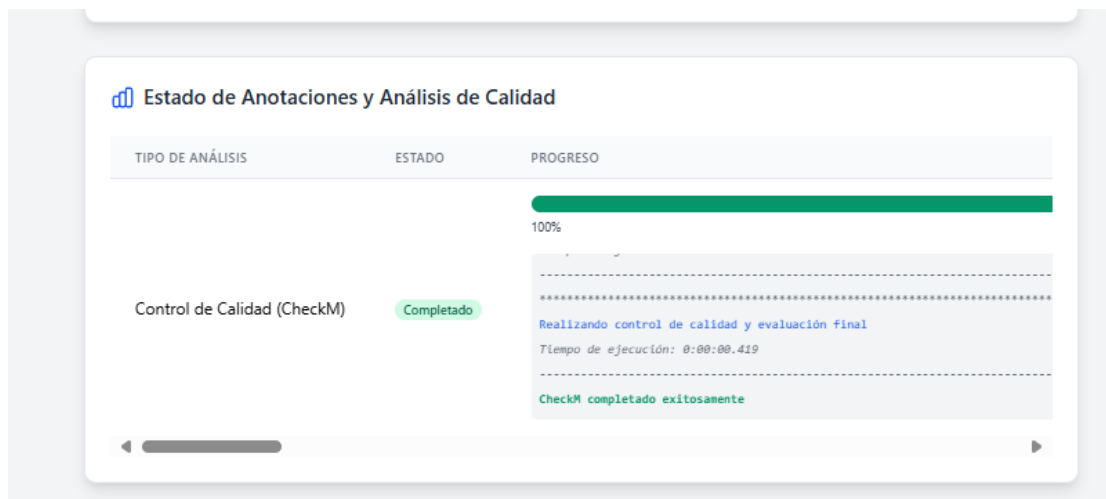


Figura 5.8: Vista del módulo de calidad tras la finalización del proceso de CheckM.

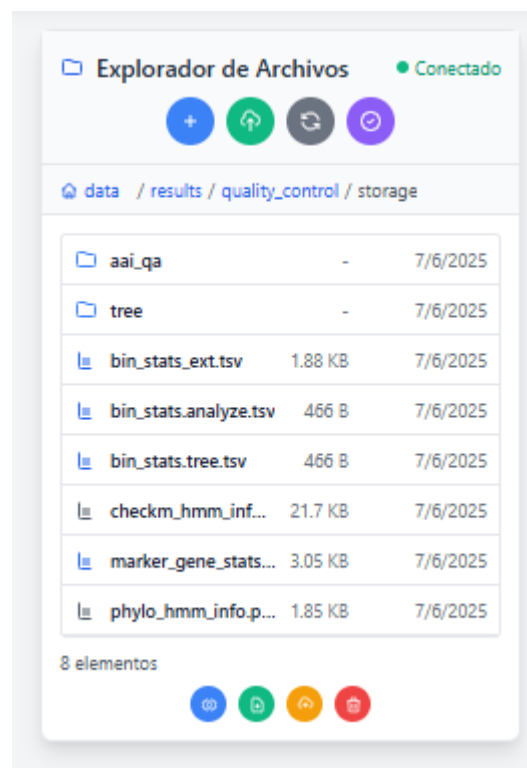


Figura 5.9: Vista del sistema de archivos revisando los archivos generados por CheckM.

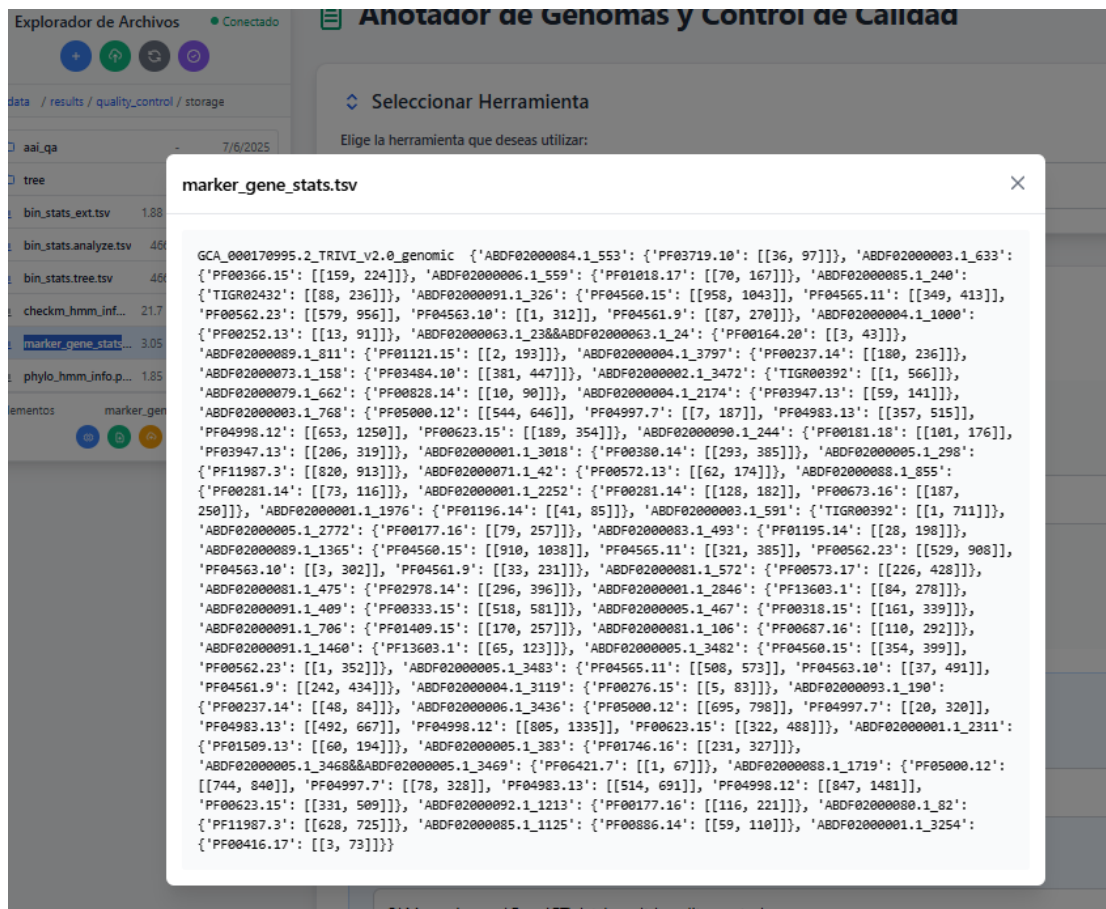


Figura 5.10: Visualización de un reporte .tsv generado por CheckM mediante el módulo de sistema de archivos.

5.2.5. Validación del caso de uso M2+M9

Las Figuras 5.6, 5.7, 5.8, 5.9 y 5.10 validan este caso de uso, ya que demuestran que la plataforma FungiGT fue capaz de ejecutar el proceso de control de calidad con la herramienta CheckM de forma exitosa. Además, se evidencia la utilidad del módulo de sistema de archivos para revisar y gestionar los archivos generados durante este proceso, facilitando la trazabilidad y el manejo de los resultados.

5.3. Caso de Uso: Anotación y Sistema de Archivos

5.3.1. Objetivo del módulo

Este módulo es esencial dentro del flujo de trabajo genómico, ya que permite realizar la anotación estructural del genoma previamente procesado. La anotación consiste en identificar, delimitar y clasificar regiones funcionales dentro de la secuencia genómica, tales como genes, exones, intrones, ARN y otras características relevantes. El principal producto de este proceso es un archivo en formato `.gtf`, el cual contiene la información detallada de las posiciones y características de cada uno de los elementos anotados en el genoma. Este archivo resulta fundamental para los análisis posteriores, tales como estudios funcionales, filogenéticos, de expresión génica, y para la visualización de la arquitectura génica en plataformas especializadas.

5.3.2. Caso de uso M3+M9

- **Actor:** Usuario investigador
- **Precondición:** El usuario ha accedido correctamente a la plataforma y ha verificado previamente la calidad del genoma a ser anotado.
- **Flujo principal:**
 1. El usuario accede al módulo de Anotación.
 2. Selecciona la carpeta de entrada donde se encuentra el archivo del genoma en formato `.fna`.
 3. Configura los parámetros necesarios para la anotación, especificando las rutas de los archivos de entrada y salida, así como las opciones avanzadas disponibles (por ejemplo, `softmasking`).
 4. Ejecuta el proceso de anotación a través del botón correspondiente, iniciando la ejecución de BRAKER3.

5. El sistema procesa la información, mostrando los reportes de ejecución y generando el archivo de genoma anotado en formato .gtf.

- **Postcondición:** El usuario visualiza los registros y reportes generados, valida la correcta generación de la anotación y puede utilizar el archivo resultante para análisis funcionales o filogenéticos posteriores.

5.3.3. Diagrama de caso de uso M3+M9

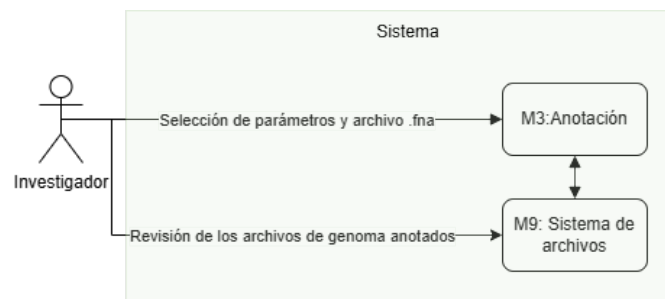
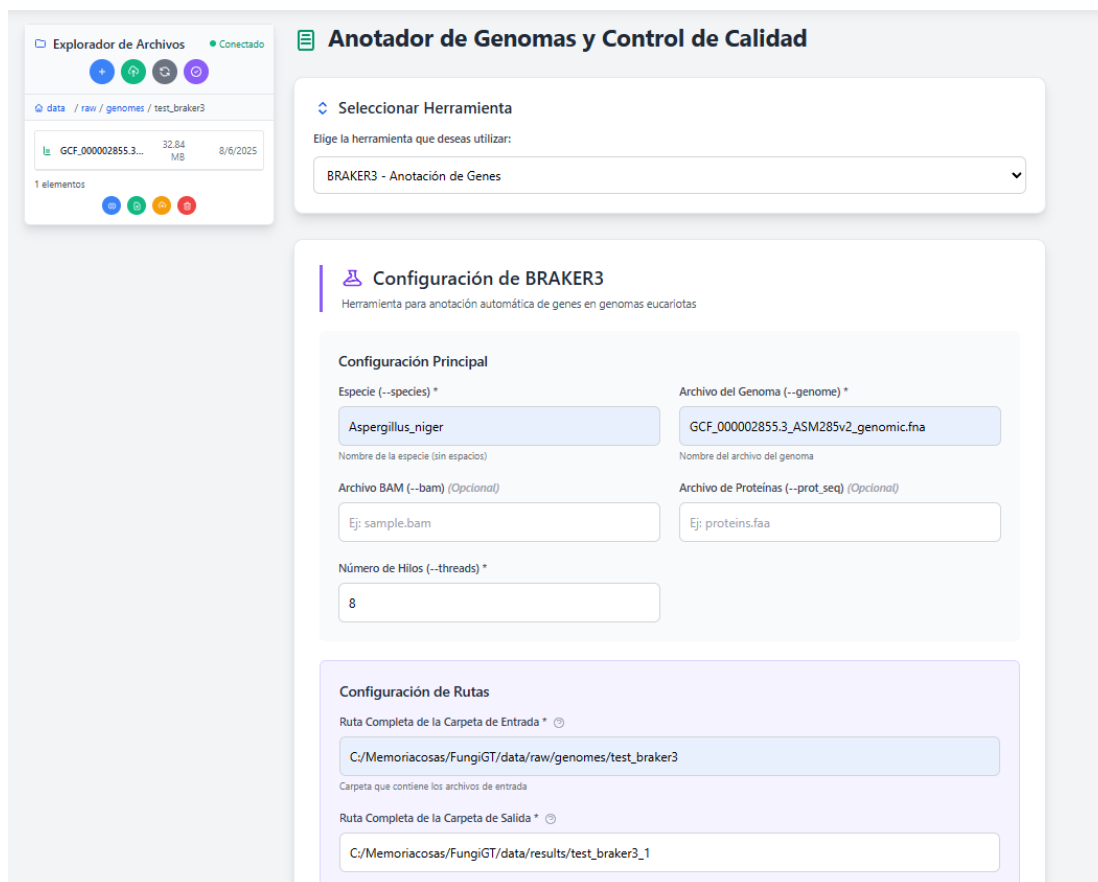


Figura 5.11: Caso de uso del módulo de Anotación y Sistema de Archivos.

5.3.4. Evidencia funcional



Explorador de Archivos Conectado

data / raw / genomes / test_braker3

GCF_000002855.3... 32.84 MB 8/6/2025

1 elementos

Anotador de Genomas y Control de Calidad

Selección de Herramienta

Elige la herramienta que deseas utilizar:

BRAKER3 - Anotación de Genes

Configuración de BRAKER3

Herramienta para anotación automática de genes en genomas eucariotas

Configuración Principal

Especie (--species) *

Aspergillus_niger

Nombre de la especie (sin espacios)

Archivo del Genoma (--genome) *

GCF_000002855.3_ASM285v2_genomic.fna

Nombre del archivo del genoma

Archivo BAM (--bam) (Opcional)

Ej: sample.bam

Archivo de Proteínas (--prot_seq) (Opcional)

Ej: proteins.faa

Número de Hilos (--threads) *

8

Configuración de Rutas

Ruta Completa de la Carpeta de Entrada *

C:/Memoriasos/FungiGT/data/raw/genomes/test_braker3

Carpeta que contiene los archivos de entrada

Ruta Completa de la Carpeta de Salida *

C:/Memoriasos/FungiGT/data/results/test_braker3_1

Figura 5.12: Interfaz del módulo de anotación mostrando la selección de especie, archivo genómico y rutas de input/output.

Opciones Avanzadas

☒ --softmasking
 ☐ --gff3
 ☐ --fungus

☐ --keepCrf
 ☐ --makehub
 ☐ --useexisting

Subir Archivos

Subir archivos a la carpeta de entrada

No se eligió ningún archivo

Mostrar Comando

Ejecutar BRAKER3

Comando Docker:

```

docker run --rm \
-v "C:/Memoriacosas/FungiGT/data/raw/genomes/test_braker3:/input" \
-v "C:/Memoriacosas/FungiGT/data/results/test_braker3_1:/output" \
teambaker/braker3:latest \
braker.pl \
--species=Aspergillus_niger \
--genome=/input/GCF_000002855.3_ASM285v2_genomic.fna \
--threads=8 \
--softmasking \
--workingdir=/output
          
```

Figura 5.13: Vista del módulo de anotación, configuración de opciones avanzadas y ejecución de BRAKER3.

Anotación (BRAKER3)

Completado

100%

```

***** # WARNING: Detected whitespace in fasta header of file /input/GCF_
# Sun Jun 8 06:23:56 2025: Log information is stored in file /output/braker..
***** # WARNING: Detected whitespace in fasta header of file /input/GCF_
# Sun Jun 8 06:23:56 2025: Log information is stored in file /output/braker..
***** # WARNING: Detected whitespace in fasta header of file /input/GCF_
BRAKER3 completado exitosamente
          
```

Figura 5.14: Vista del módulo tras la finalización exitosa del proceso de BRAKER3.

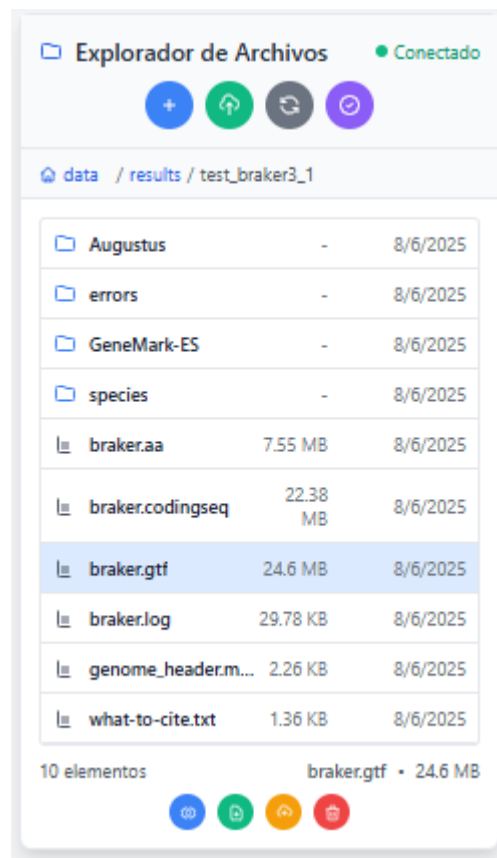


Figura 5.15: Visualización, a través del sistema de archivos, de los archivos generados por BRAKER3.

5.3.5. Validación del caso de uso M3+M9

Las Figuras 5.12, 5.13, 5.14 y 5.15 validan el funcionamiento del módulo, demostrando la correcta selección y procesamiento del archivo genómico, la generación del archivo anotado y la gestión eficiente de los archivos mediante el navegador de archivos integrado en la plataforma.

5.4. Caso de Uso: Análisis Funcional y Sistema de Archivos

5.4.1. Objetivo del módulo

El objetivo principal de este módulo es realizar análisis funcionales exhaustivos a partir de las secuencias de proteínas derivadas del genoma anotado. Utilizando eggNOG-mapper, el sistema permite identificar ortólogos, asignar funciones génicas y categorizar proteínas en familias y dominios, tales como PFAM, COG y anotaciones de bases de datos funcionales reconocidas. Los resultados generados son fundamentales para interpretar la función biológica de los genes, establecer comparaciones funcionales entre organismos y facilitar estudios de biología molecular, biotecnología y evolución.

5.4.2. Caso de uso M4+M9

- **Actor:** Usuario investigador
- **Precondición:** El usuario ha accedido correctamente a la plataforma y ha verificado la calidad y anotación de los genomas a analizar.
- **Flujo principal:**
 1. El usuario accede al módulo de Análisis Funcional.
 2. Selecciona eggNOG-mapper de la lista de herramientas disponibles.
 3. Especifica el archivo de entrada en formato `.faa` y define el nombre y ruta para los archivos de salida generados por el análisis (por ejemplo, `.emapper.hits`, `.emapper.orthologs`, `.emapper.annotations`).
 4. Configura parámetros clave del análisis, como el alcance taxonómico, el número de CPUs, criterios de búsqueda (modo, e-value, porcentaje de identidad), y opciones adicionales para personalizar la salida y la ejecución.
 5. Inicia la ejecución del análisis funcional.

6. El sistema procesa los datos y genera los archivos de resultados, incluyendo información sobre ortología, anotaciones funcionales y hits de búsqueda.

- **Postcondición:** El usuario puede acceder al módulo de visualización para explorar los distintos gráficos y reportes generados a partir de los archivos de resultados de eggNOG-mapper.

5.4.3. Diagrama de caso de uso M4+M9

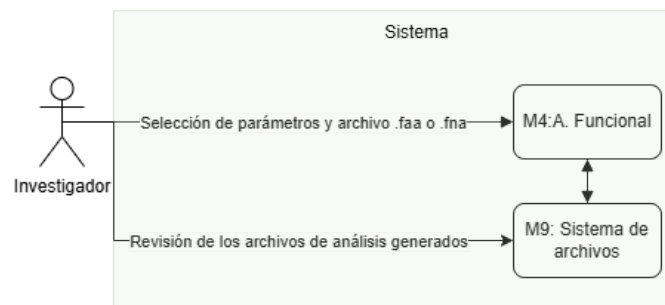


Figura 5.16: Caso de uso del módulo de Análisis Funcional y Sistema de Archivos.

5.4.4. Evidencia funcional

 **Analizador Funcional de Genomas**

⚡ Seleccionar Herramienta de Análisis

Elige la herramienta que deseas utilizar:

eggNOG-mapper - Anotación Funcional

Configuración de eggNOG-mapper

Herramienta para anotación funcional de proteínas y genes

📖 Información sobre Base de Datos

eggNOG-mapper utiliza **una sola base de datos principal** (~50 GB) que contiene información de múltiples dominios taxonómicos. El **alcance taxonómico** determina qué nivel de la jerarquía usar para la anotación:

- 🍄 **Fungi (4751)**: Específico para hongos
- 🌿 **Eukaryota (2759)**: Recomendado para hongos (más cobertura)
- 🌐 **Cellular organisms (1)**: Máxima cobertura taxonómica

Datos de Entrada

Ruta al Archivo de Entrada (-i) ⓘ [Ver archivos disponibles](#)

/data/genomes/GCA_000988865.1.faa

Ruta absoluta al archivo FASTA de proteínas

Nombre de Salida (-o) ⓘ

resultado_2406

Nombre base para los archivos de salida (sin extensión)

Figura 5.17: Interfaz del módulo de análisis funcional definiendo archivos de entrada y salida.

Datos de Entrada

Ruta al Archivo de Entrada (-i)
[Ver archivos disponibles](#)

Ruta absoluta al archivo FASTA de proteínas

Nombre de Salida (-o)

Nombre base para los archivos de salida (sin extensión)

Configuración Básica

Alcance Taxonómico (--tax_scope)

Número de CPUs (--cpu)

☐ **Mostrar Opciones Avanzadas**

Opciones de Búsqueda

Modo de Búsqueda (-m)

E-value (--evalue)

Porcentaje de Identidad (--pident)

Opciones de Salida y Ejecución

☐ Sin comentarios (--no_file_comments)

☐ Reanudar (--resume)

☐ Sobrescribir (--override)

Figura 5.18: Vista del módulo de análisis funcional configurando los parámetros de ejecución de eggNOG-mapper.

Base de Datos eggNOG

Disponible (47.18 GB)

VERIFICAR ESTADO

Mostrar Comando

Ejecutar eggNOG-mapper

Estado de Análisis

HERRAMIENTA	ARCHIVO	ESTADO	PROGRESO	TIEMPO	ACCIONES
eggNOG-mapper	GCA_000988865.1.faa	En progreso	5%	4s	Cancelar

Figura 5.19: Vista del módulo de análisis funcional durante la ejecución del análisis.

Estado de Análisis

HERRAMIENTA	ARCHIVO	ESTADO	PROGRESO	TIEMPO	ACCIONES
eggNOG-mapper	GCA_000988865.1.faa	Completado	100%	32m 55s	Ver resultados Eliminar

Figura 5.20: Vista del módulo de análisis funcional tras la finalización del proceso.

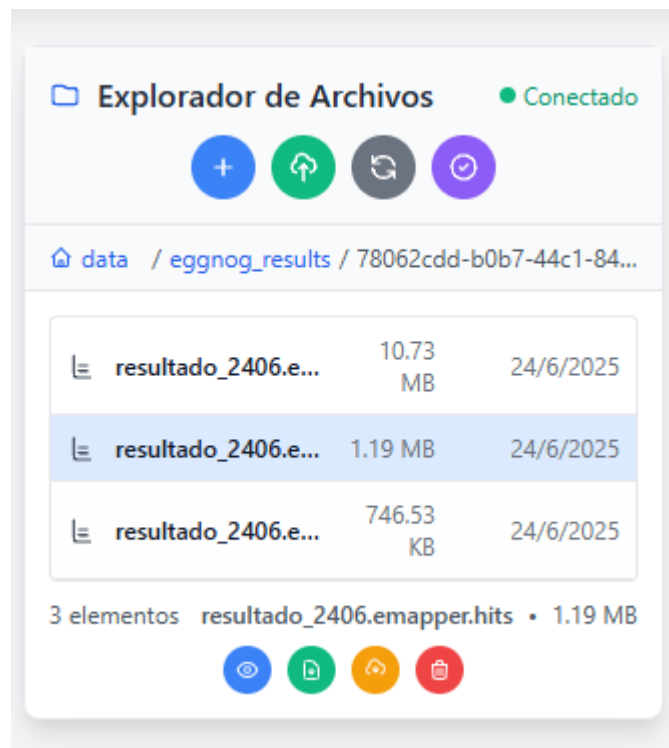


Figura 5.21: Visualización y exploración de los archivos generados por eggNOG-mapper mediante el sistema de archivos de la plataforma.

5.4.5. Validación del caso de uso M4+M9

Las Figuras 5.17, 5.18, 5.19, 5.20 y 5.21 validan este caso de uso, ya que demuestran que la plataforma FungiGT permite definir correctamente los archivos de entrada y salida, configurar y ejecutar el análisis funcional con eggNOG-mapper, y explorar los resultados generados de manera eficiente mediante el sistema de archivos integrado. Esto garantiza la correcta implementación del flujo de análisis funcional y su integración con los módulos de visualización y manejo de archivos.

5.5. Caso de Uso: Análisis filogenético y Sistema de Archivos

5.5.1. Objetivo del módulo

El objetivo principal de este módulo es permitir la comparación filogenética de múltiples genomas de hongos a través del cálculo eficiente de distancias genómicas, utilizando la herramienta BinDash. Gracias a su enfoque basado en sketches de k-mers, BinDash facilita la estimación rápida y precisa de similitud genética entre genomas, permitiendo explorar relaciones evolutivas, construir dendrogramas y realizar análisis comparativos de manera escalable y reproducible. Los resultados generados por este módulo son fundamentales para estudios filogenéticos, delimitación de especies y análisis de diversidad genómica.

5.5.2. Caso de uso M5+M9

- **Actor:** Usuario investigador
- **Precondición:** El usuario ha accedido correctamente a la plataforma y ha verificado la calidad y anotación de los genomas a analizar.
- **Flujo principal:**
 1. El usuario accede al módulo de Análisis Filogenético.
 2. Selecciona el modo de operación adecuado (por ejemplo, análisis de genomas completos o proteínas).
 3. Especifica la carpeta de entrada donde se encuentran los archivos genómicos en formato .fna o .faa (BinDash permite trabajar tanto con genomas anotados como no anotados).
 4. Configura los parámetros necesarios, incluyendo el directorio de entrada, el patrón de archivos, el prefijo de salida y las opciones avanzadas como el tamaño de k-mer.
 5. Inicia la ejecución del análisis filogenético mediante BinDash.

6. El sistema procesa los datos y genera los archivos de resultados, como la matriz de distancias (.txt) y el sketch binario (sketch.dat).

- **Postcondición:** El usuario puede acceder al módulo de visualización para explorar las matrices de distancias y dendrogramas generados, o examinar los archivos mediante el sistema de archivos integrado.

5.5.3. Diagrama de caso de uso M5+M9

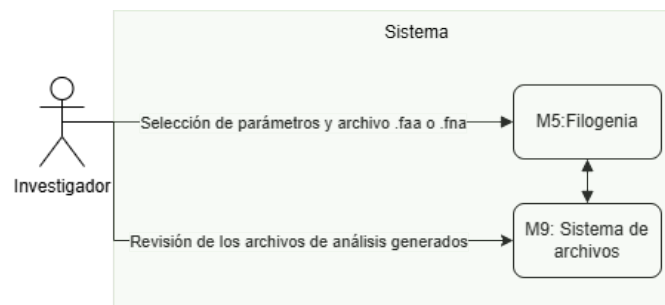


Figura 5.22: Caso de uso del módulo de Análisis Filogenético y Sistema de Archivos.

5.5.4. Evidencia funcional

The screenshot displays the FungiGT web application interface. The top navigation bar includes links for Genomas NCBI, Anotación Genómica, Análisis, Visualización, and a user profile icon. The main content area is divided into two panels. The left panel, titled 'Explorador de Archivos', shows a file explorer for the directory '/data/genomes/thricoderma' containing 10 files. The right panel, titled 'Analizador Funcional de Genomas', contains several configuration sections: 'Seleccionar Herramienta de Análisis' with a dropdown menu set to 'BinDash - Comparación de Genomas y Cálculo de ANI'; 'Configuración de BinDash' with a '¿Qué es BinDash?' section explaining its features; 'Modo de Operación' with a dropdown menu set to 'Completo - Sketch + Dist en un solo paso'; 'Configuración Completa (Sketch + Dist)' with input fields for 'Directorio de Genomas' (set to '/data/genomes/thricoderma'), 'Patrón de Archivos' (set to '*.fna'), and 'Prefijo de Salida' (set to 'test_analysis_thricoderma'); a 'Mostrar Parámetros Avanzados' checkbox; and 'Interpretación de Resultados' with a 'Formato de salida' section explaining the output format. At the bottom, there are two buttons: 'Mostrar Comando' and 'Ejecutar BinDash'.

Explorador de Archivos • Conectado

/data/genomes/thricoderma

Nombre	Tamaño	Fecha
GCA_030015385.1...	38.49 MB	13/6/2025
GCA_030247165.1...	35.24 MB	13/6/2025
GCA_030278935.1...	37.87 MB	13/6/2025
GCA_030708095.1...	35.16 MB	13/6/2025
GCA_030708105.1...	31.33 MB	13/6/2025
GCA_030770505.1...	30.51 MB	13/6/2025
GCA_033030045.1...	31.32 MB	13/6/2025
GCA_033847375.1...	37.55 MB	13/6/2025
GCA_033847385.1...	37.08 MB	13/6/2025
GCA_034769445.1...	35.63 MB	13/6/2025

10 elementos

Analizador Funcional de Genomas

Seleccionar Herramienta de Análisis

Elige la herramienta que deseas utilizar:

BinDash - Comparación de Genomas y Cálculo de ANI

Configuración de BinDash

Herramienta ultrarrápida para comparación de genomas y cálculo de ANI (Average Nucleotide Identity)

¿Qué es BinDash?

- **Comparación ultrarrápida:** Compara genomas completos en segundos
- **Cálculo de ANI:** Estima Average Nucleotide Identity con alta precisión
- **Eficiente en memoria:** Maneja terabytes de datos en laptops típicas
- **Formatos soportados:** FASTA, FASTQ (comprimidos o no)
- **Algoritmo avanzado:** Binwise Densefied MinHash
- **Casos de uso:** Genomas, metagenomas, pangenomas

Modo de Operación

Seleccionar operación a realizar:

Completo - Sketch + Dist en un solo paso

Configuración Completa (Sketch + Dist)

Directorio de Genomas: /data/genomes/thricoderma

Patrón de Archivos: *.fna

Prefijo de Salida: test_analysis_thricoderma

☐ Mostrar Parámetros Avanzados

Interpretación de Resultados

Formato de salida: Cada línea contiene 5 campos separados por tabulaciones:

- **Genoma consulta (Q):** Nombre del genoma de consulta
- **Genoma objetivo (T):** Nombre del genoma objetivo
- **Distancia de mutación:** Entre Q y T (1-ANI)
- **Valor p:** Significancia estadística de la distancia
- **Índice de Jaccard:** Similitud entre Q y T

Nota: Para genomas con ANI > 99.5%, se recomienda usar sketch size ≥ 188

Mostrar Comando

Ejecutar BinDash

Figura 5.23: Interfaz del módulo de análisis filogenético, con el modo de operación seleccionado y los directorios de entrada y prefijos configurados.

Mostrar Comando
Ejecutar BinDash

Comando BinDash:

```
# Paso 1: Crear lista de genomas
ls /data/genomes/thricoderma*.fna > test_analysis_thricoderma_genomes.txt

# Paso 2: Crear sketch
bindash sketch \
--listfname="test_analysis_thricoderma_genomes.txt" \
--outfname="test_analysis_thricoderma.sketch"

# Paso 3: Calcular distancias (all-vs-all)
bindash dist \
"test_analysis_thricoderma.sketch" \
--outfname="test_analysis_thricoderma_distances.txt"
```

Estado de Análisis Funcional

TIPO DE ANÁLISIS	ESTADO	PROGRESO	ACCIONES
BinDash	Creando sketch	40%	

Figura 5.24: Vista del módulo de análisis filogenético durante la ejecución de BinDash.

Estado de Análisis Funcional

TIPO DE ANÁLISIS	ESTADO	PROGRESO	ACCIONES
BinDash	Completado	100%	Ver resultados Descargar

Figura 5.25: Vista del módulo de análisis filogenético tras la finalización exitosa del proceso de BinDash.

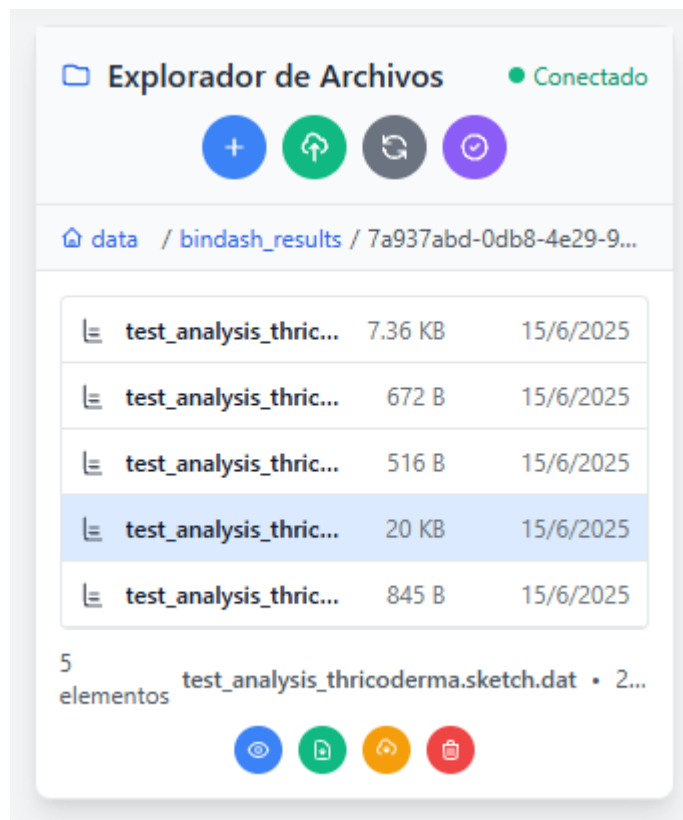


Figura 5.26: Visualización y exploración de los archivos generados por BinDash mediante el sistema de archivos de la plataforma.

5.5.5. Validación del caso de uso M5+M9

Las Figuras 5.23, 5.24, 5.25 y 5.26 validan la correcta implementación de este caso de uso, demostrando que la plataforma permite configurar, ejecutar y visualizar exitosamente el análisis filogenético de genomas de hongos mediante BinDash, así como gestionar los archivos generados para su posterior interpretación y análisis comparativo.

5.6. Caso de Uso: Visualización y Sistema de Archivos

5.6.1. Objetivo del módulo

El objetivo de este módulo es proporcionar una visualización adecuada y comprensible de las salidas generadas por las distintas herramientas integradas en el flujo de trabajo de FungiGT. Esto permite al usuario explorar, interpretar y presentar de manera efectiva los resultados obtenidos a lo largo del análisis genómico.

5.6.2. Caso de uso M8+M9

- **Actor:** Usuario investigador
- **Precondición:** El usuario ha procesado datos genómicos con alguna de las herramientas disponibles; en este caso, se ejemplifica con la visualización de anotaciones generadas por eggNOG-mapper.
- **Flujo principal:**
 1. El usuario accede al módulo de Visualización.
 2. Selecciona el visualizador correspondiente al análisis realizado.
 3. Selecciona el archivo en formato `.emapper.annotations` u otro formato soportado por el visualizador específico.
 4. Procesa el archivo para su visualización.
 5. Interactúa con las herramientas de visualización, como realizar zoom, explorar diferentes gráficos disponibles y descargar el archivo visual generado.
- **Postcondición:** El usuario puede generar un reporte del análisis visualizado, por ejemplo, en formato PDF, utilizando el asistente Dr. Fungito.

5.6.3. Diagrama de caso de uso M8+M9

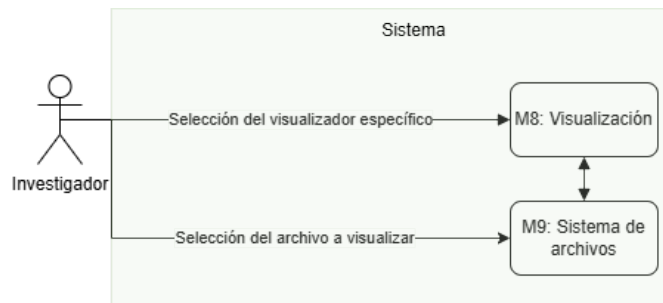


Figura 5.27: Caso de uso del módulo de Visualización y Sistema de Archivos.

5.6.4. Evidencia funcional



Figura 5.28: Interfaz del módulo de visualización de resultados.

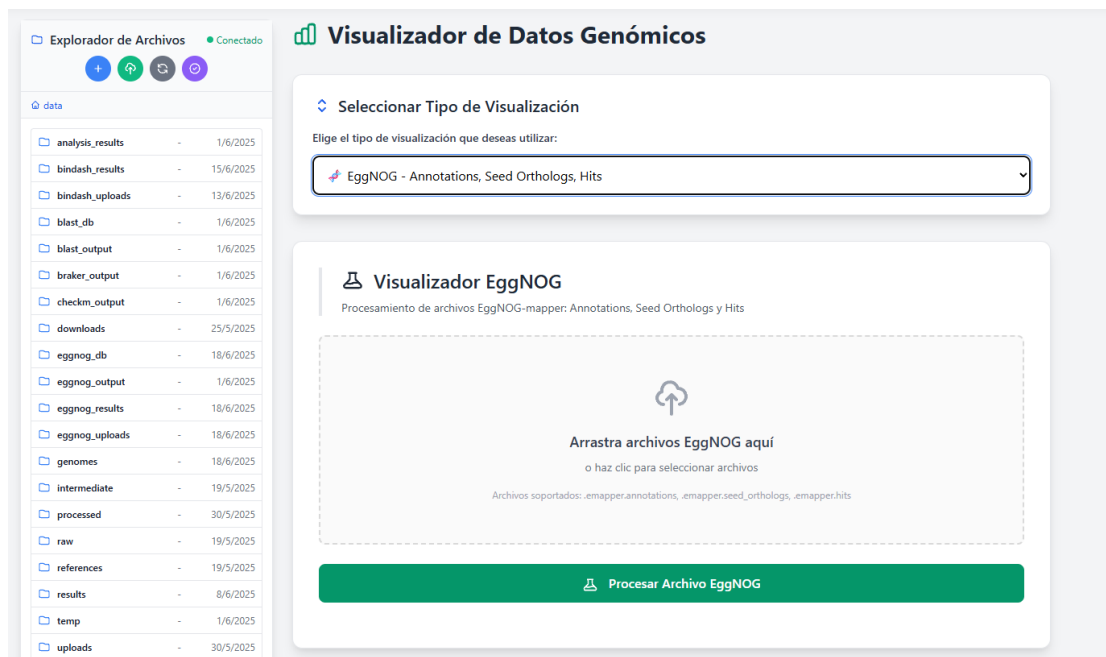


Figura 5.29: Vista del módulo de visualización seleccionando el visualizador específico para las salidas de eggNOG-mapper.

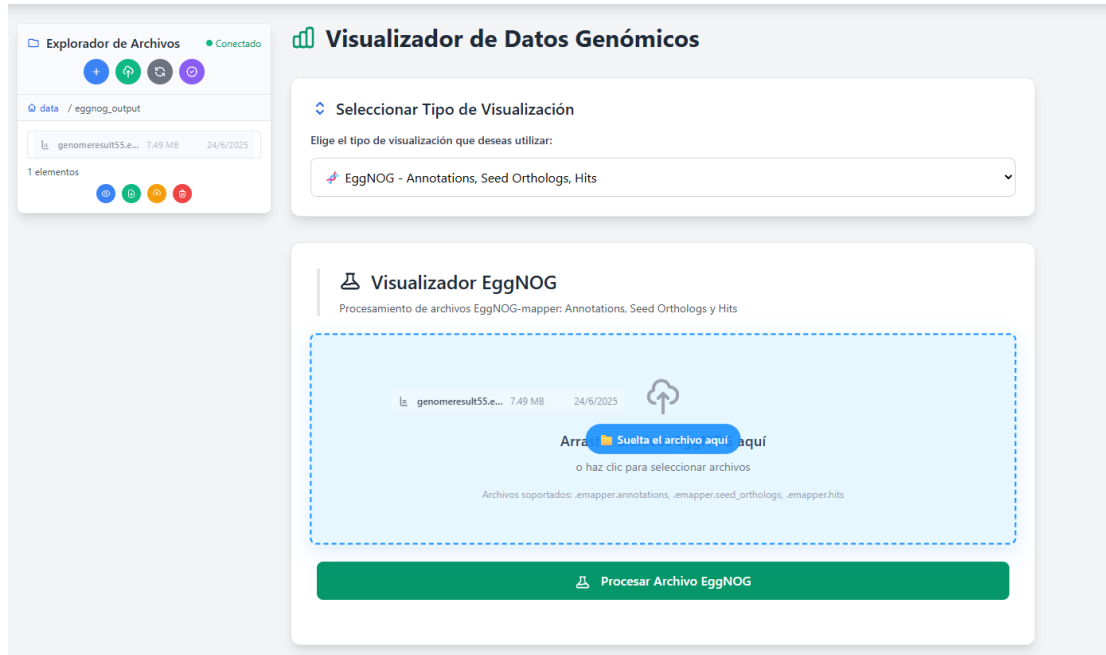


Figura 5.30: Integración del módulo de visualización con el sistema de archivos para cargar archivos de anotaciones.

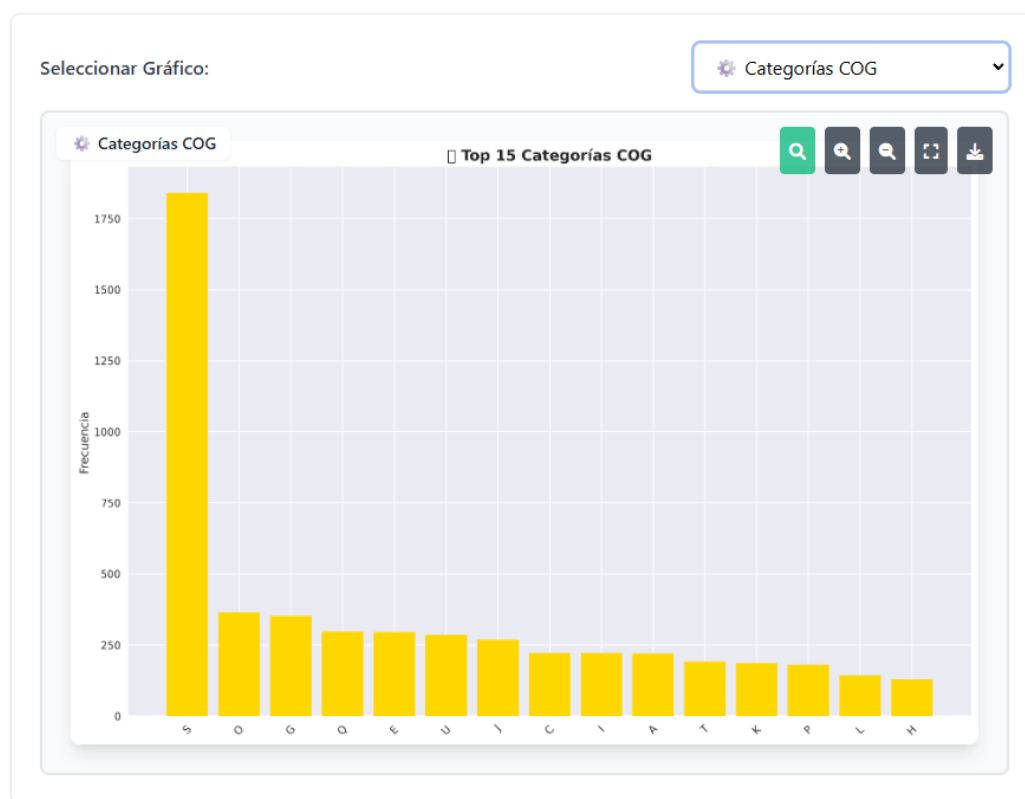


Figura 5.31: Utilización de la herramienta de visualización para explorar gráficos, interactuar con ellos y alternar entre las diferentes visualizaciones disponibles para los archivos de anotaciones.

5.6.5. Validación del caso de uso M8+M9

Las Figuras 5.28, 5.29, 5.30 y 5.31 validan este caso de uso, ya que demuestran que la plataforma FungiGT permite cargar correctamente archivos generados por herramientas como eggNOG-mapper, seleccionar el visualizador apropiado, interactuar con las visualizaciones disponibles y generar reportes de manera eficiente. De este modo, se garantiza que el módulo de visualización y sistema de archivos cumple su función de facilitar la exploración e interpretación de los resultados genómicos obtenidos.

5.7. Caso de Uso: Visualización, Sistema de Archivos e Interpretación Inteligente

5.7.1. Objetivo del módulo

El objetivo de este módulo integrado es proporcionar una experiencia completa de análisis genómico que abarca desde la visualización interactiva de resultados hasta la interpretación automatizada mediante inteligencia artificial. Esta integración permite al usuario no solo explorar y gestionar los datos generados, sino también obtener interpretaciones biológicas especializadas y reportes inteligentes que faciliten la toma de decisiones en investigación genómica de hongos.

5.7.2. Caso de uso M8+M9+M10

- **Actor:** Usuario investigador
- **Precondición:** El usuario ha ejecutado análisis genómicos (BinDash, eggNOG-mapper, CheckM u otros) y dispone de archivos de resultados para visualizar e interpretar.
- **Flujo principal:**
 1. El usuario accede al módulo de Visualización (M8) y selecciona el visualizador correspondiente al tipo de análisis realizado.
 2. Utiliza el Sistema de Archivos (M9) para navegar y cargar los archivos de resultados apropiados (matrices de distancias, anotaciones funcionales, etc.).
 3. Procesa y visualiza los datos mediante gráficos interactivos (mapas de calor, dendrogramas, gráficos de barras).
 4. Interactúa con las visualizaciones generadas, aplicando zoom, filtros y explorando diferentes representaciones gráficas.

5. Envía las visualizaciones generadas a Dr. Fungito (M10) mediante el botón Enviar a Dr. Fungito.
 6. Dr. Fungito analiza automáticamente las imágenes y proporciona interpretaciones biológicas especializadas.
 7. El usuario puede realizar consultas adicionales sobre la interpretación o solicitar la generación de reportes inteligentes en formato PDF.
 8. Los reportes generados se almacenan automáticamente en el Sistema de Archivos para fácil acceso posterior.
- **Postcondición:** El usuario obtiene visualizaciones comprensibles, interpretaciones biológicas automatizadas y reportes profesionales que integran toda la información analizada, facilitando la comunicación de resultados y la toma de decisiones científicas.

5.7.3. Diagrama de caso de uso M8+M9+M10

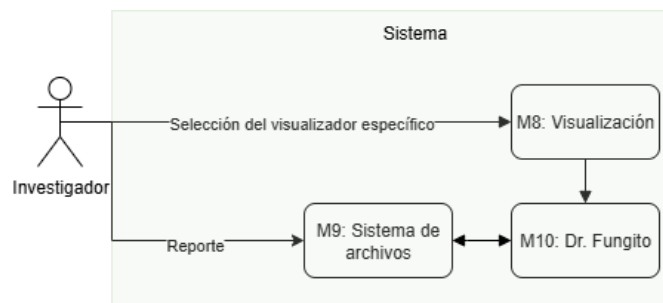


Figura 5.32: Caso de uso integrado de Visualización, Sistema de Archivos y Dr. Fungito.

5.7.4. Evidencia funcional

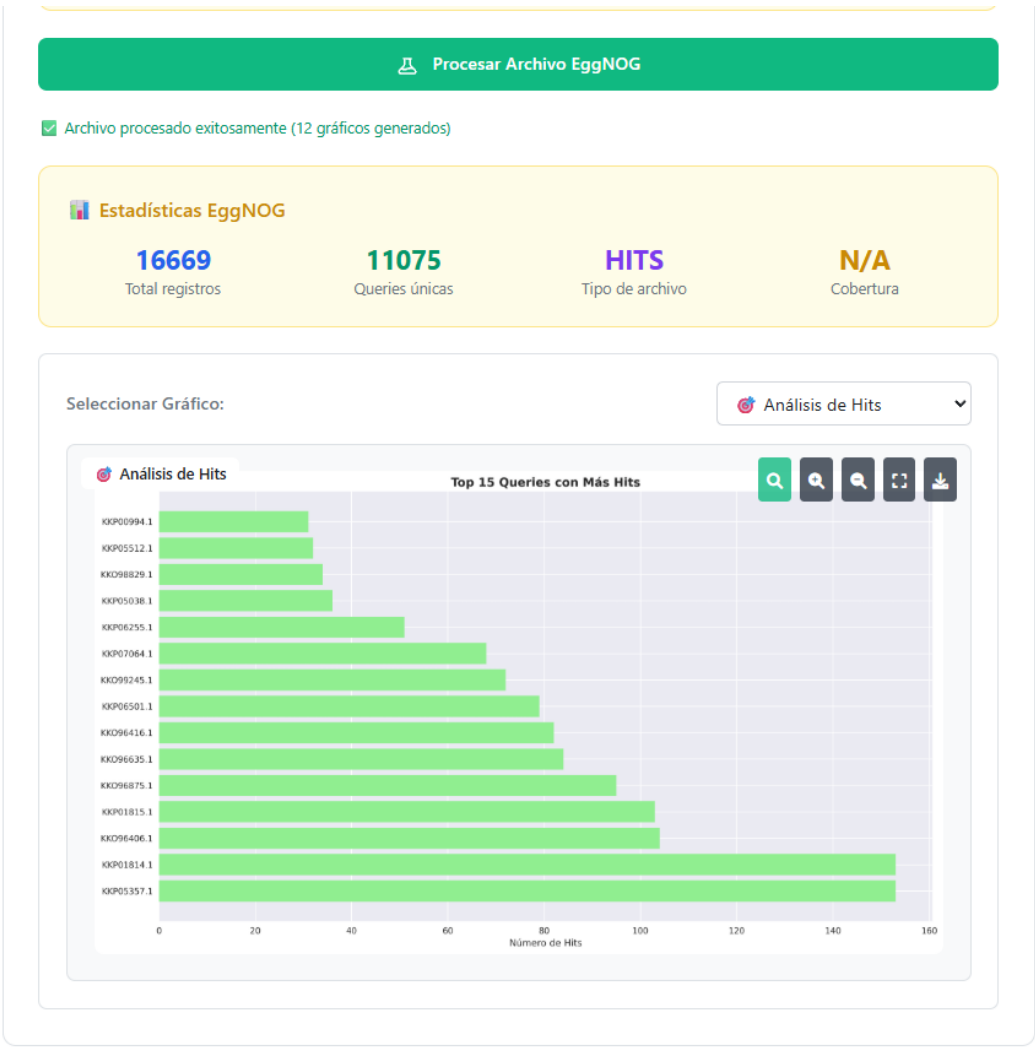


Figura 5.33: Interfaz del módulo de visualización con un gráfico del top 15 de queries con más hits, generado a partir de la salida de EggNOG-mapper.

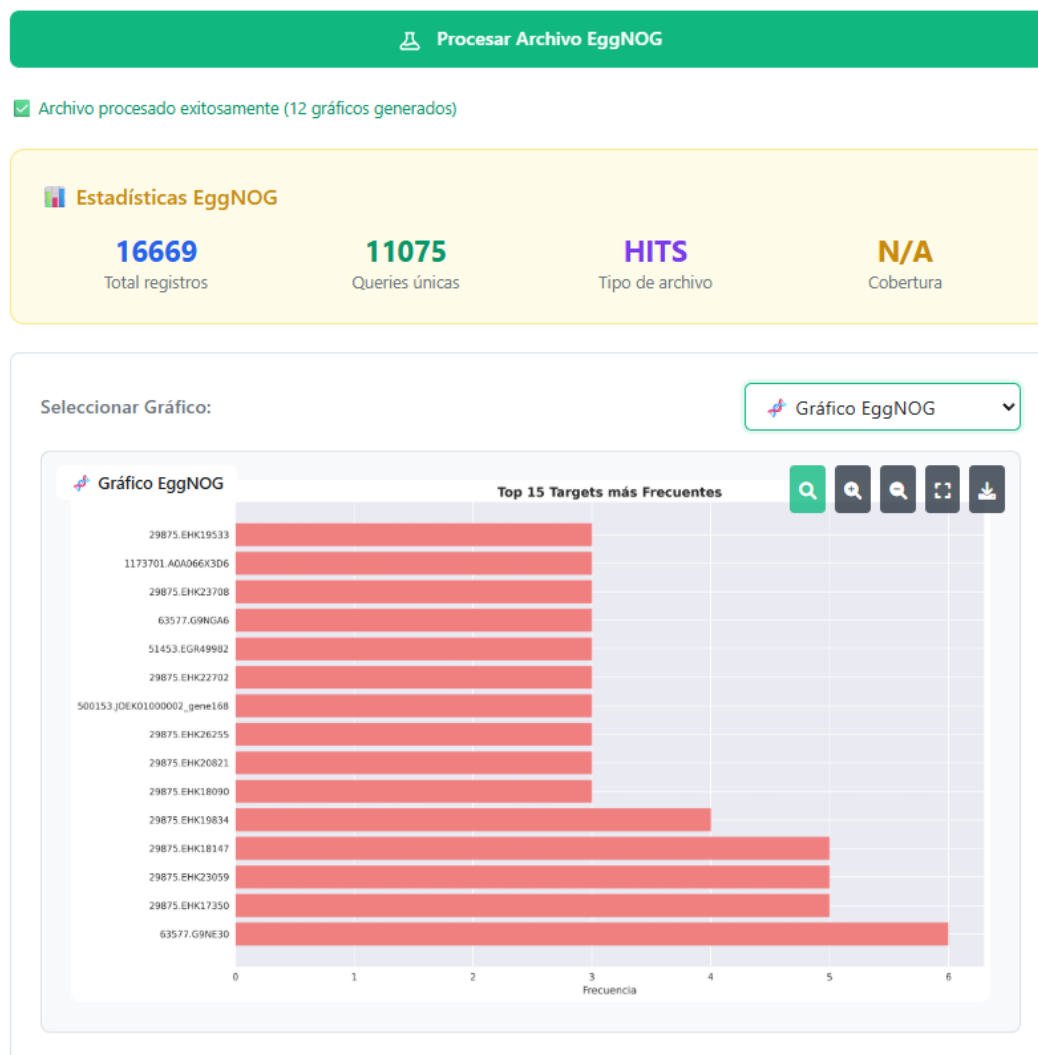


Figura 5.34: Interfaz del módulo de visualización con un gráfico del top 15 de targets, generado a partir de la salida de hits de EggNOG-mapper.

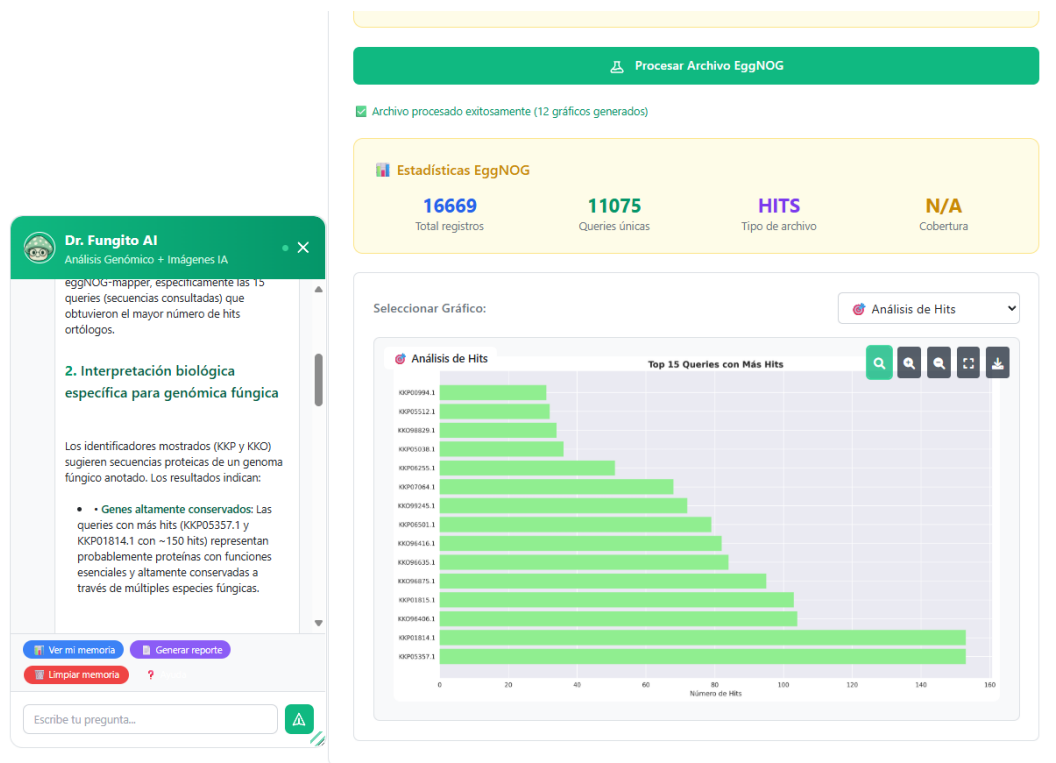
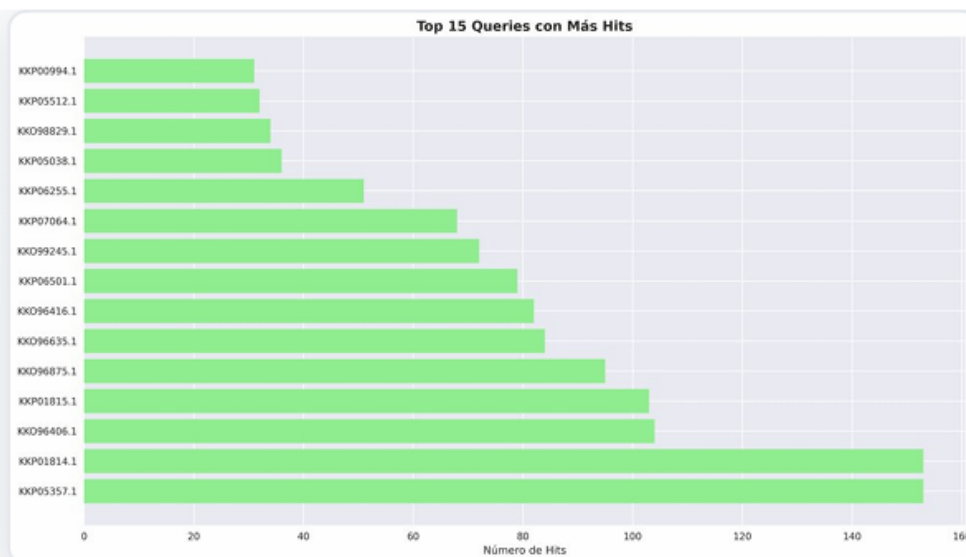


Figura 5.35: Visualización enviada a Dr. Fungito para el análisis inteligente del gráfico de las 15 queries con más hits.



Figura 5.37: Generación de reporte inteligente por Dr. Fungito, mostrado en la carpeta data/reports, disponible para descarga.



Análisis:

Análisis de Resultados eggNOG - Top 15 Queries con Más Hits

1. Tipo de análisis detectado

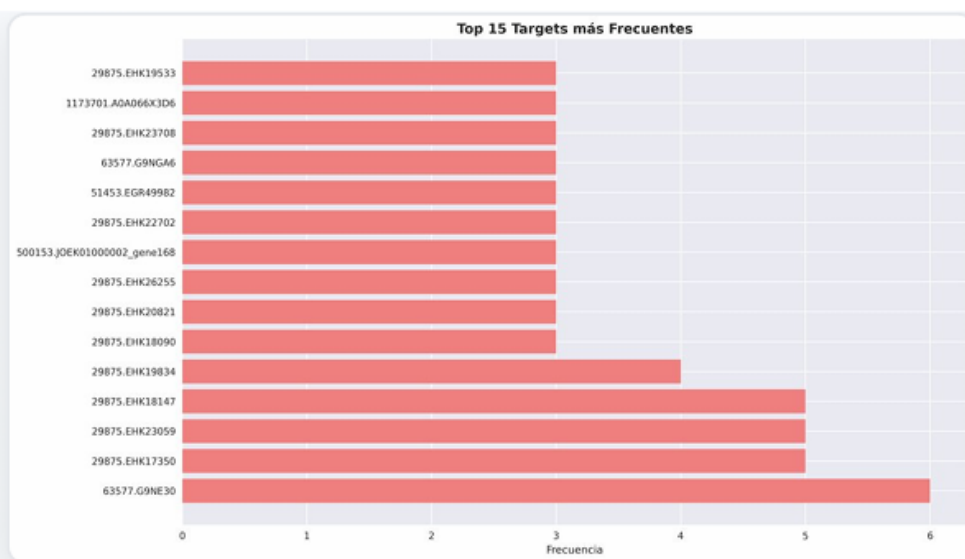
Gráfico de barras horizontales mostrando los resultados de anotación funcional mediante eggNOG-mapper, específicamente las 15 queries (secuencias consultadas) que obtuvieron el mayor número de hits ortólogos.

2. Interpretación biológica específica para genómica fúngica

Los identificadores mostrados (KKP y KKO) sugieren secuencias proteicas de un genoma fúngico anotado. Los resultados indican:

- **Genes altamente conservados:** Las queries con más hits (KKP05357.1 y KKP01814.1 con ~150 hits) representan probablemente proteínas con funciones esenciales y altamente conservadas a través de múltiples especies fúngicas.
- **Familias génicas expandidas:** El alto número de ortólogos sugiere que

Figura 5.38: Parte del reporte PDF generado para ambos gráficos; en este caso, correspondiente al top 15 de queries con más hits.



Análisis:

Análisis de Targets Genómicos Fúngicos - eggNOG

1. Tipo de análisis detectado

Gráfico de barras horizontales que muestra la frecuencia de los 15 targets (genes/proteínas) más abundantes identificados mediante anotación funcional con eggNOG-mapper.

2. Interpretación biológica específica

Este análisis representa la **abundancia relativa de familias de genes ortólogos** en el genoma fúngico analizado. Los identificadores mostrados corresponden a:

- **Códigos eggNOG** (ej: 29875.EHK19533, 63577.GQNE30)
- **Grupos de genes ortólogos** que comparten función evolutiva
- **Familias proteicas conservadas** en hongos

La distribución sugiere un **core genómico** con funciones esenciales altamente

Figura 5.39: Parte del reporte PDF generado para ambos gráficos; en este caso, correspondiente al top 15 de targets.

5.7.5. Validación del caso de uso M8+M9+M10

Las Figuras 5.33, 5.34, 5.35, 5.36, 5.37, 5.38 y 5.39 validan la implementación exitosa de este caso de uso integrado. La evidencia demuestra que la plataforma FungiGT permite una experiencia completa desde la carga y visualización de datos genómicos hasta la interpretación automatizada mediante inteligencia artificial. El flujo integrado garantiza que los usuarios pueden obtener no solo representaciones gráficas de sus datos, sino también interpretaciones biológicas especializadas y reportes profesionales, mejorando significativamente la capacidad de análisis y comunicación de resultados en investigación genómica de hongos.

5.7.6. Frontend

En esta sección se presenta el flujo de navegación de la aplicación FungiGT para el análisis genómico. Se muestran las principales interfaces que guían al usuario desde el acceso inicial hasta la visualización de los resultados, ilustrando la experiencia de usuario y la organización funcional del frontend. Las capturas reflejan las pantallas clave involucradas en cada etapa del proceso, facilitando la comprensión integral del recorrido dentro de la plataforma.



Figura 5.40: Landing page



Bienvenido a FungiGT

Plataforma de análisis genómico de hongos

Correo electrónico

 admin2@fungigt.com

Contraseña

☐ Recordarme [¿Olvidaste tu contraseña?](#)

 Iniciar sesión

O continúa con

 Google  Microsoft

Al iniciar sesión, aceptas nuestros [Términos de Servicio](#) y [Política de Privacidad](#)

[¿No tienes una cuenta? Regístrate aquí](#)

Figura 5.41: Login

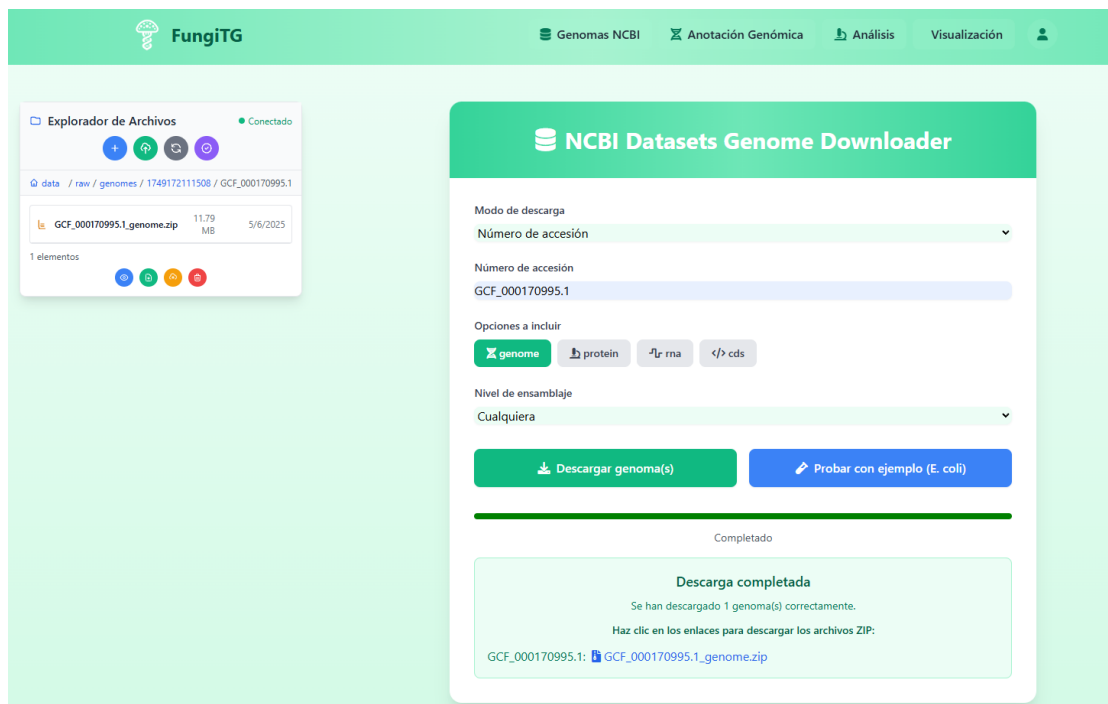


Figura 5.42: Módulo de adquisición

Genomas NCBI
Anotación Genómica
Análisis
Visualización

Explorador de Archivos

Conectado

data

analysis_results	-	1/6/2025
blastash_results	-	15/6/2025
blastash_uploads	-	13/6/2025
blast_db	-	1/6/2025
blast_output	-	1/6/2025
braker_output	-	1/6/2025
checkm_output	-	1/6/2025
downloads	-	25/5/2025
eggnog_db	-	18/6/2025
eggnog_output	-	24/6/2025
eggnog_results	-	24/6/2025
eggnog_uploads	-	18/6/2025
genomes	-	18/6/2025
intermediate	-	19/5/2025
processed	-	30/5/2025
raw	-	19/5/2025
references	-	19/5/2025
results	-	8/6/2025
temp	-	1/6/2025
uploads	-	30/5/2025
visualizations	-	2/5/2025
README.txt	253 B	25/5/2025

22 elementos

Anotador de Genomas y Control de Calidad

Seleccionar Herramienta

Elige la herramienta que deseas utilizar:

CheckM - Control de Calidad

Configuración de CheckM

Herramienta para evaluar la calidad y completitud de genomas

Configuración Principal

Tipo de Workflow

Lineage Workflow

Número de Hilos (-t)

8

Extensión de Archivo (-x)

fna

Configuración de Rutas

Ruta de entrada (carpeta con genomas)

C:\Memoriacosas\FungiGT\data\raw\genomes

Los genomas serán tomados de esta carpeta

Ruta de salida (resultados)

C:\Memoriacosas\FungiGT\data\results\quality_control

Opciones Avanzadas

☐ --reduced_tree

☐ --quiet

☐ --tab_table

Subir Archivos

Subir archivos .fna a la carpeta de entrada

Elegir archivos No se eligió ningún archivo

Subir Archivos

Mostrar Comando

Ejecutar CheckM

Estado de Anotaciones y Análisis de Calidad

TIPO DE ANÁLISIS	ESTADO	PROGRESO	ACCIONES
------------------	--------	----------	----------

© 2025 FungiGT. Todos los derechos reservados.

© 2025 FungiGT. Todos los derechos reservados.

Plataforma de análisis genómico para hongos
Documentación
Soporte
Acerca de

Figura 5.43: Módulo de anotación (Calidad y Anotación disponibles)

86

Genomas NCBI

Anotación Genómica

Análisis

Visualización

Explorador de Archivos

Conectado

data

analysis_results

1/6/2025

blastdash_results

15/6/2025

blastdash_uploads

13/6/2025

blast_db

1/6/2025

blast_output

1/6/2025

braker_output

1/6/2025

checkm_output

1/6/2025

downloads

25/5/2025

eggnog_db

18/6/2025

eggnog_output

24/6/2025

eggnog_results

24/6/2025

eggnog_uploads

18/6/2025

genomes

18/6/2025

intermediate

19/5/2025

processed

30/5/2025

raw

19/5/2025

references

19/5/2025

results

8/6/2025

temp

1/6/2025

uploads

30/5/2025

visualizations

2/5/2025

README.txt

253 B

25/5/2025

22 elementos

Analizador Funcional de Genomas

Seleccionar Herramienta de Análisis

Elige la herramienta que deseas utilizar:

eggNOG-mapper - Anotación Funcional

Configuración de eggNOG-mapper

Herramienta para anotación funcional de proteínas y genes

Información sobre Base de Datos

eggNOG-mapper utiliza una sola base de datos principal (~50 GB) que contiene información de múltiples dominios taxonómicos. El alcance taxonómico determina qué nivel de la jerarquía usar para la anotación:

Fungi (4751): Específico para hongos

Eukaryota (2759): Recomendado para hongos (más cobertura)

Cellular organisms (1): Máxima cobertura taxonómica

Datos de Entrada

Ruta al Archivo de Entrada (-i)

Ver archivos disponibles

Nombre de Salida (-o)

/data/genomes/mi_archivo.faa

resultado_eggno

Ruta absoluta al archivo FASTA de proteínas

Nombre base para los archivos de salida (sin extensión)

Configuración Básica

Alcance Taxonómico (--tax_scope)

Número de CPUs (--cpu)

Eukaryota (2759) - Recomendado para hongos

1

Mostrar Opciones Avanzadas

Opciones de Búsqueda

Modo de Búsqueda (-m)

E-value (--evalue)

Porcentaje de Identidad (--piden)

Diamond

0,001

Opciones de Salida y Ejecución

☐ Sin comentarios (--no_file_comments)

☐ Reanudar (--resume)

☐ Sobrescribir (--override)

Base de Datos eggNOG

Disponible (47.18 GB)

VERIFICAR ESTADO

Mostrar Comando

Ejecutar eggNOG-mapper

Estado de Análisis

HERRAMIENTA

ARCHIVO

ESTADO

PROGRESO

TIEMPO

ACCIONES

No hay análisis en progreso

Los análisis aparecerán aquí cuando se inicien

© 2025 FungiGT. Todos los derechos reservados.

Plataforma de análisis genómico para hongos

Documentación

Soporte

Acerca de

Figura 5.44: Módulo de análisis (Análisis funcional y filogenético disponibles)

87

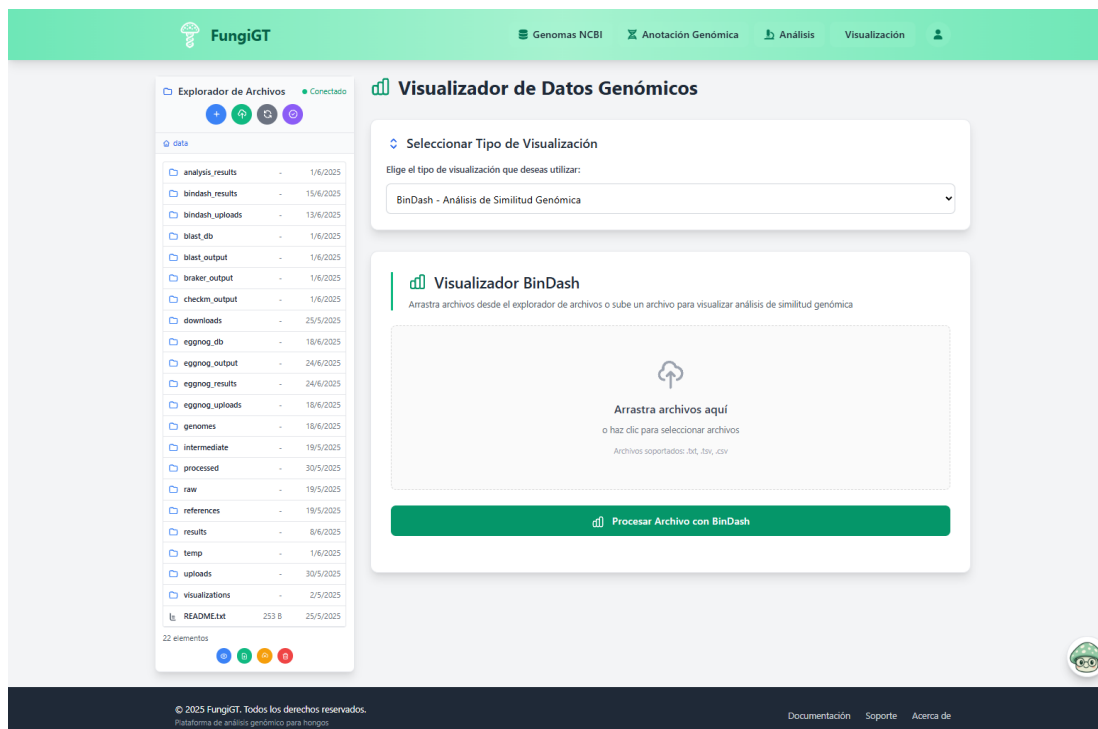


Figura 5.45: Visualizador - Selección de archivo de entrada

BinDash - Análisis de Similitud Genómica

Arrastra archivos desde el explorador de archivos o sube un archivo para visualizar análisis de similitud genómica



Arrastra archivos aquí

o haz clic para seleccionar archivos

Archivos soportados: .txt, .tsv, .csv

Procesar Archivo con BinDash

✓ Archivo procesado exitosamente (6 gráficos generados)

Estadísticas del Análisis

45

Comparaciones

10

Genomas únicos

N/A

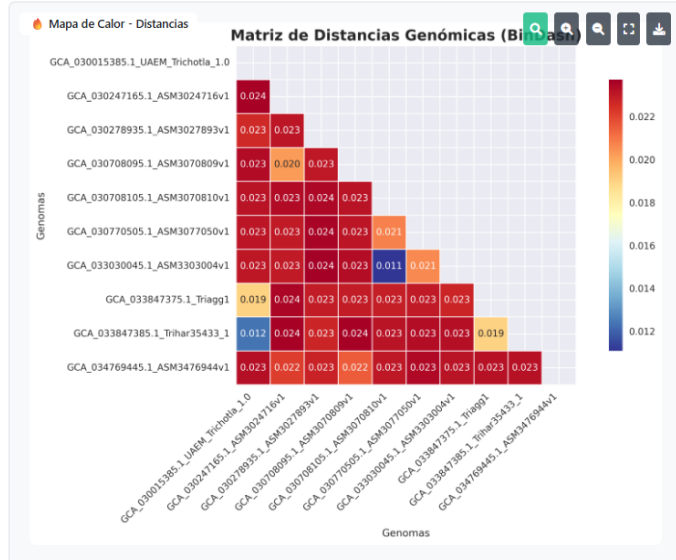
Jaccard promedio

N/A

Distancia promedio

Seleccionar Gráfico:

🔥 Mapa de Calor - Distancias ▼



Para mantener la claridad y evitar redundancias, en lugar de incluir descripciones individuales bajo cada imagen, se ha optado por presentar a continuación un cuadro 5.1 resumen de los eventos y acciones principales del flujo de la plataforma. Esta estrategia facilita la comprensión integral del proceso y sintetiza las funcionalidades expuestas en las capturas de pantalla anteriores, garantizando una presentación concisa y ordenada del recorrido del usuario en la aplicación.

Evento	Interacción	Acción	Objeto
Inicio de sesión	Clic en <i>Iniciar sesión</i>	Redirección a página de autenticación	Página auth
Adquisición de genomas	Seleccionar número de accesoion y presionar <i>Descargar</i>	Descarga de archivo genómico	Archivo genómico
Control de calidad	Seleccionar <i>Anotación</i> y <i>CheckM</i>	Generación archivos de calidad	Informe calidad
Anotación	Seleccionar <i>Anotación</i> y <i>Braker3</i>	Generación anotaciones	Resultados anotación
Análisis funcional	Seleccionar <i>Analizador</i> y <i>eggNOG</i>	Ejecución y resultados	Resultados análisis
Análisis filogenético	Seleccionar <i>Analizador</i> y <i>BinDash</i>	Ejecución y resultados	Resultados filogenia
Visualización	Seleccionar <i>Visualizador</i>	Mostrar gráficos	Gráficos
Dr. Fungito	Seleccionar Visualizador	Presionar lupa verde y luego generar reporte	Análisis y reportes de gráficos

Cuadro 5.1: Eventos del prototipo

Capítulo 6

Validación y verificación

6.1. Verificación Técnica

La verificación técnica en FungiGT se enfocó en garantizar que cada componente del sistema funcione correctamente según las especificaciones técnicas establecidas. Para ello, se implementó una estrategia de testing multicapa que abarca desde la infraestructura Docker hasta la funcionalidad individual de cada módulo.

6.1.1. Estrategia de Testing

La estrategia de testing adoptada contempla múltiples niveles de verificación:

- **Testing de Infraestructura:** Verificación de la correcta inicialización y funcionamiento de los contenedores Docker.
- **Testing de Integración:** Validación de la comunicación y operación conjunta entre microservicios.
- **Testing Funcional:** Comprobación puntual de que cada módulo cumple con sus requisitos y criterios específicos.

- **Testing de Salud del Sistema:** Monitoreo continuo del estado operativo de los servicios mediante healthchecks automatizados.

6.1.2. Testing de Infraestructura Docker

Todos los servicios de FungiGT cuentan con healthchecks automáticos que verifican constantemente su estado operacional:

- **MongoDB:** Comando `mongo --eval "db.adminCommand('ping')"` para comprobar conexión y respuesta.
- **Servicios Node.js:** Solicitudes HTTP a endpoints `/health` para validar disponibilidad.
- **Servicios Python:** Validación de respuesta correcta en puertos configurados.

Los healthchecks se configuraron con intervalos de 15 a 30 segundos, timeouts de 10 segundos y hasta 5 reintentos antes de marcar fallo, garantizando detección temprana de problemas.

The image shows a Docker Desktop window at the top and a Postman window at the bottom.

Docker Desktop:

- Container name: **fungigt-quality-control**
- ID: 615be3d368d7
- Image: fungigt-qua
- Status: **Running** (14 minutes ago)
- Port: 4004:4004
- Logs tab is selected, showing the command `> node server.js`.
- Log output:
 - Servidor CheckM (Quality Control) ejecutándose en <http://localhost:4004>
 - Endpoints disponibles:
 - GET /health - Estado del servicio
 - POST /upload-checkm - Subir archivos genómicos
 - POST /execute-checkm - Ejecutar análisis CheckM
 - GET /checkm-progress - Obtener progreso actual
 - GET /checkm-progress-stream - Stream de progreso (SSE)
 - POST /cancel-checkm - Cancelar proceso actual

Postman:

- Environment: No environment
- Request: **GET** `http://localhost:4004/health`
- Response: **200 OK** (17 ms, 387 B)
- Body (JSON):

```
1 {
2   "success": true,
3   "message": "Servicio CheckM funcionando correctamente",
4   "service": "quality-control-checkm",
5   "port": "4004"
6 }
```

Figura 6.1: Ejemplo representativo de healthcheck exitoso en el contenedor Docker de CheckM.

Se confirma que todos los demás servicios cuentan con configuraciones idénticas y reportan estados operativos similares durante el periodo de evaluación.

6.1.3. Cálculo de la tasa de disponibilidad

La tasa de disponibilidad de cada módulo de la plataforma FungiGT se evaluó de forma independiente, tomando como base el tiempo que demoró cada herramienta en ejecutarse correctamente dentro del sistema.

Es decir, para cada herramienta se consideró como *tiempo esperado de operación* el intervalo durante el cual la herramienta estuvo activa y en ejecución, funcionando plenamente en la plataforma. Por ejemplo, si CheckM demoró 30 minutos en completarse, ese fue el período usado para calcular su disponibilidad.

Las pruebas se realizaron en un equipo con las siguientes características: procesador con 6 núcleos físicos y 6 núcleos virtuales (12 en total), 32 GB de memoria RAM, y almacenamiento SSD con memoria DRAM, lo que contribuyó significativamente a mejorar la velocidad de lectura y escritura de datos y, por ende, al rendimiento general de la plataforma.

Esta base sirve para establecer una referencia sólida de la disponibilidad y rendimiento de las herramientas en un entorno controlado y de alto rendimiento.

Es importante destacar que, si bien esta tasa de disponibilidad es válida para el entorno local utilizado, la métrica puede variar en entornos diferentes, como infraestructuras en la nube, donde factores externos como la velocidad de la máquina anfitriona y la configuración del contenedor Docker influirán en la disponibilidad y desempeño. En dichos casos, la evaluación deberá adaptarse considerando las particularidades de la infraestructura donde se despliegue la plataforma.

La fórmula utilizada para calcular la tasa de disponibilidad fue:

$$\text{Disponibilidad} = \frac{\text{Tiempo operativo}}{\text{Tiempo esperado de operación}} \times 100 \%$$

donde:

- **Tiempo operativo:** tiempo total durante el cual la herramienta respondió correctamente y estuvo funcionando.
- **Tiempo esperado de operación:** duración completa de la ejecución de la herramienta dentro de la plataforma.

Durante todas las pruebas realizadas, las herramientas se mantuvieron operativas y sin fallos, por lo que la tasa de disponibilidad para cada módulo evaluado fue del 100 %, como se resume en la Tabla 6.1. Esta tabla detalla el tiempo operativo registrado para cada módulo y su correspondencia con los casos de uso validados en el Capítulo 5, evidenciando la estabilidad y robustez del sistema.

Por último, cabe mencionar que, aunque la estabilidad fue óptima, se detectaron limitaciones en la visualización de los logs de progreso. Por ejemplo, el módulo eggNOG-mapper presenta un avance visual que se mantiene estancado en 5 % durante la mayor parte de la ejecución, para luego actualizarse abruptamente a 100 % al finalizar. Este comportamiento afecta únicamente la percepción visual del progreso y no influye en la disponibilidad ni el correcto funcionamiento del módulo, constituyendo un aspecto de usabilidad susceptible de mejora en versiones futuras.

Servicio	Tiempo esperado (h)	Tiempo operativo (h)	Disponibilidad (%)	Caso de Uso (Capítulo 5)
Adquisición	0.05	0.05	100	Módulo Adquisición
CheckM	0.2	0.2	100	Módulo Calidad
BRAKER3	6	6	100	Módulo Anotación
eggNOG-mapper	0.6	0.6	100	Módulo Análisis Funcional
BinDash	0.08	0.08	100	Módulo Análisis Filogenético
Dr. Fungito	0.008	0.008	100	Módulo Dr. Fungito

Cuadro 6.1: Cálculo ilustrativo de tasa de disponibilidad por servicio con referencia a casos de uso validados en el Capítulo 5.

Estos resultados evidencian que, bajo las condiciones controladas de prueba, cada módulo

de FungiGT operó de manera estable y sin interrupciones durante el tiempo esperado de ejecución. La tasa de disponibilidad del 100 % refleja la robustez del sistema en un entorno local optimizado, tal como se describió anteriormente. Se destaca que esta métrica puede variar en otros entornos con diferentes configuraciones de hardware o infraestructura, por lo que futuras evaluaciones deberán considerar dichos factores para validar el desempeño en contextos diversos.

6.1.4. Testing de Integración de Servicios

Se implementaron pruebas específicas para validar:

- **Comunicación entre servicios:** Confirmación de que los microservicios pueden intercambiar datos correctamente a través de la red Docker.
- **Gestión de archivos:** Verificación de la correcta operación de los volúmenes compartidos para la transferencia de archivos entre servicios.
- **Autenticación distribuida:** Validación del sistema de autenticación funcionando de forma consistente en todos los servicios.
- **Persistencia de datos:** Comprobación de que los datos se almacenan y recuperan sin pérdidas ni corrupciones.

Un ejemplo claro de la correcta integración entre módulos y el sistema de archivos se evidencia en las Figuras 5.30 y 5.31 del Capítulo 5. En ellas, se muestra el proceso de arrastrar y soltar un archivo directamente desde el explorador de archivos integrado hacia el módulo de visualización, permitiendo cargar y explorar gráficos de anotaciones genómicas.

Este flujo ejemplifica cómo los módulos interactúan eficientemente con el sistema de archivos compartido para facilitar la transferencia y gestión de datos. Aunque en otros módulos el proceso consiste en copiar y pegar rutas desde el explorador de archivos hacia los formularios de entrada, el principio de integración es el mismo, garantizando consistencia y

trazabilidad sin necesidad de repetir exhaustivamente todas las figuras. Así, se evita redundancia mientras se valida la robustez de la integración a lo largo de toda la plataforma.

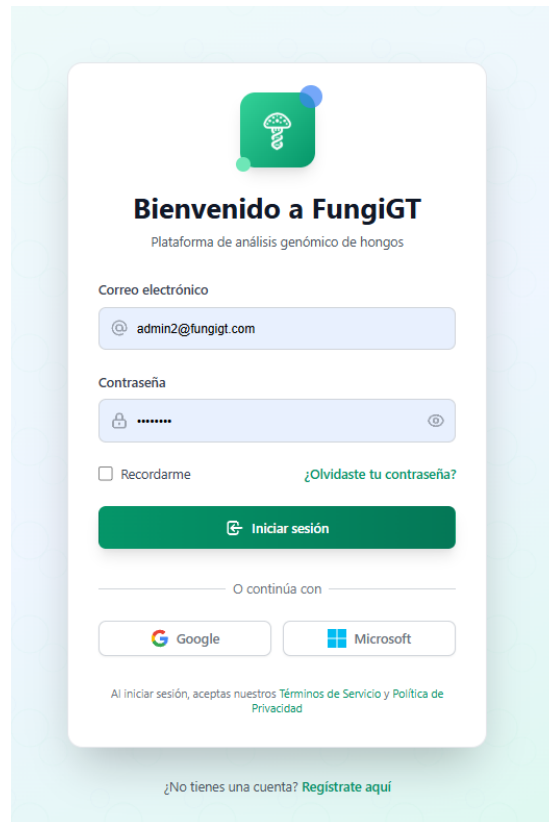


Figura 6.2: Pantalla de login de la plataforma FungiGT.

Aunque el sistema de autenticación no fue un requisito funcional explícito solicitado por el cliente, se decidió implementar esta capa adicional por razones de seguridad y protección del sistema. Dado que la aplicación permite la ejecución de código mediante formularios web, es fundamental impedir que usuarios no autorizados puedan explotar vulnerabilidades comunes como ejecución de comandos remotos o shells reversas, que podrían comprometer la integridad del sistema y los datos.

Para ello, se adoptó un mecanismo de autenticación basado en tokens JWT (JSON Web Tokens), generados y firmados utilizando claves secretas almacenadas en el archivo de configuración `.env`. Esta solución garantiza que solo usuarios autenticados y autorizados pue-

dan acceder a las funcionalidades sensibles de la plataforma, aumentando la confianza y la seguridad general del entorno de análisis genómico.

Esta medida, aunque no requerida inicialmente, fue valorada positivamente durante las fases de validación y contribuye a robustecer la arquitectura microservicios del sistema.

6.1.5. Testing Funcional por Módulos

Cada módulo fue sometido a pruebas funcionales específicas según los criterios listados en la Tabla 6.2, con evidencias documentadas en el Capítulo 5.

Módulo	Herramienta	Criterios de Verificación
Adquisición	NCBI Datasets	Descarga exitosa, integridad y extracción correcta de archivos genómicos.
Calidad	CheckM	Generación de reportes con métricas válidas y archivos de salida completos.
Anotación	BRAKER3	Producción de archivo .aa válido, logs de ejecución y completitud del análisis.
Análisis Funcional	eggNOG-mapper	Anotaciones funcionales coherentes, generación de archivos de salida correctos.
Análisis Filogenético	BinDash	Matriz de distancias y archivos sketch generados con validez estadística comprobada.
Visualización	Herramientas Python personalizadas	Generación de gráficos, interactividad con visualizaciones y opción de exportación.
Dr. Fungito	Python, Javascript y LLM	Generación de reportes PDF, interactividad con el chat para sacar conclusiones a partir de los gráficos.

Cuadro 6.2: Criterios de verificación funcional por módulo.

Cada módulo fue sometido a pruebas funcionales específicas, y su correcto funcionamiento se evidencia en las salidas y reportes presentados en el Capítulo 5. En particular, la última figura de cada caso de uso ilustra la generación exitosa de los archivos resultantes o la correcta visualización, validando que las herramientas se ejecutaron sin problemas. Por ejemplo, la Figura 5.4 muestra la descarga y extracción exitosa del genoma mediante NCBI Datasets en el módulo de Adquisición, la Figura 5.8 evidencia la finalización del proceso

de CheckM con reportes completos, mientras que la Figura 5.15 confirma la generación de archivos anotados por BRAKER3. De forma similar, la Figura 5.21 evidencia los resultados del análisis funcional con eggNOG-mapper, la Figura 5.26 corresponde al análisis filogenético con BinDash, y las Figuras 5.30 y 5.31 muestran la integración y exploración en el módulo de visualización. Además, la Figura 5.35 valida específicamente la funcionalidad del módulo Dr. Fungito, evidenciando la correcta interpretación automatizada de los gráficos generados y la generación de reportes inteligentes. Esta evidencia conjunta reafirma la robustez y completitud del flujo de trabajo implementado.

6.1.6. Resultados de la Verificación

Los resultados obtenidos durante el proceso de verificación técnica fueron:

- **Tasa de éxito de healthchecks:** 100 % de disponibilidad promedio durante 7 horas continuas de prueba.
- **Tiempo de respuesta:** Menor a 5 segundos para operaciones básicas en todos los servicios monitoreados.
- **Integridad de datos:** 100 % de integridad en la transferencia y almacenamiento de archivos.
- **Estabilidad del sistema:** Funcionamiento continuo sin errores críticos o interrupciones durante todo el periodo de testing.

6.2. Validación con el Cliente

La validación de FungiGT se llevó a cabo mediante un proceso iterativo de retroalimentación con el cliente, lo cual permitió ajustar tanto los requisitos como la arquitectura del sistema para adecuarse a las necesidades reales de los investigadores en genómica fúngica.

6.2.1. Evolución de Requisitos Basada en Feedback

Durante el desarrollo, el feedback del cliente fue clave para refinar y ajustar los requerimientos iniciales:

6.2.1.1. Decisión sobre Base de Datos

Requisito inicial: Desarrollar una base de datos genómica propia.

Feedback del cliente: “No necesitamos una base de datos propia, ya existen bases de datos consolidadas como NCBI. Lo que necesitamos es una solución que las consuma eficientemente.”

Acción tomada: Se implementó la integración con NCBI Datasets, herramienta ya familiar para el cliente, permitiendo acceder directamente a genomas actualizados sin duplicar información.

Validación: Esta decisión redujo la complejidad del sistema y mejoró la confiabilidad, al acceder directamente a la fuente oficial.

6.2.1.2. Evolución Arquitectónica: De IA Asistente a Microservicios

Propuesta inicial: Incorporar un asistente de IA que realizara análisis simplificados de forma automática.

Feedback del equipo y cliente: Se consideró que una IA tan avanzada sería compleja y poco transparente para el ámbito científico, donde la trazabilidad y control de parámetros son fundamentales.

Decisión arquitectónica: Se optó por una arquitectura de microservicios que permite:

- Control granular de cada herramienta bioinformática.
- Transparencia en los procesos de análisis.
- Escalabilidad independiente de los componentes.

- Facilidad de mantenimiento y actualización.

Validación: Esta arquitectura fue bien valorada por el cliente por la transparencia y control que ofrece sobre los análisis genómicos.

6.2.2. Decisiones Arquitectónicas Validadas

6.2.2.1. Containerización con Docker

Justificación: Garantizar reproducibilidad y facilitar la instalación en diferentes entornos.

Validación del cliente: El cliente reconoció que la containerización resuelve uno de los principales problemas en bioinformática: la dificultad para instalar y configurar múltiples herramientas especializadas.

6.2.2.2. Interfaz Web Unificada

Justificación: Proveer acceso centralizado a todas las herramientas sin necesidad de línea de comandos.

Validación del cliente: El cliente comentó que “esto hace mucho más accesible el análisis genómico para investigadores sin experiencia técnica profunda.”

6.2.3. Validación de Usabilidad

La validación de usabilidad incluyó:

6.2.3.1. Feedback Directo del Cliente

Durante demostraciones iterativas, el cliente destacó:

- **Facilidad de uso:** “La interfaz es mucho más intuitiva que usar las herramientas por separado.”

- **Eficiencia:** “El tiempo de configuración se reduce significativamente.”
- **Integración:** “Tener todas las herramientas en un solo lugar es muy valioso.”

6.2.3.2. Comparación con Flujo Tradicional

El cliente reconoció que FungiGT ofrece claras ventajas frente al flujo tradicional, que consiste en instalar herramientas individualmente, crear scripts separados para descarga, análisis y visualización, lo que es poco eficiente y requiere un alto conocimiento técnico. Si bien plataformas como Galaxy [4] son alternativas consolidadas, su arquitectura monolítica difiere del enfoque de microservicios usado en FungiGT, por lo que compararlas directamente en métricas como velocidad o escalabilidad no es adecuado.

En este contexto, se presentó una comparación simplificada en el cuadro 6.3 que refleja la mejora en tiempos de configuración, conocimiento técnico requerido, gestión de archivos y trazabilidad:

Aspecto	Flujo Tradicional	FungiGT
Tiempo de configuración	2-3 horas	15-30 minutos
Conocimiento técnico requerido	Alto	Medio
Gestión de archivos	Separada, manual	Integral

Cuadro 6.3: Comparación validada por el cliente entre el flujo tradicional y la plataforma FungiGT.

Es importante aclarar que la gestión de archivos en FungiGT se caracteriza por ser integral: existe un manejador de archivos incorporado dentro de la plataforma que permite al usuario administrar y acceder a los datos de forma centralizada, a diferencia del flujo tradicional donde las operaciones con archivos se realizan de forma separada y manual.

El tiempo de configuración puede variar según la velocidad de conexión a internet, la capacidad de lectura y escritura del sistema de almacenamiento, y la potencia de la máquina utilizada. En cuanto al conocimiento técnico, el flujo tradicional requiere que el usuario

entienda y gestione individualmente cada herramienta o microservicio, mientras que FungiGT reduce esta carga al ofrecer una interfaz unificada y simplificada para controlar todo el proceso.

Finalmente, aunque actualmente la trazabilidad formal del historial de análisis no está implementada, la arquitectura modular y el diseño de la plataforma establecen una base sólida para desarrollar esta funcionalidad en el futuro, permitiendo mejorar la reproducibilidad y auditoría de los procesos.

6.2.4. Aceptación del Cliente

El cliente expresó su aceptación final con el siguiente comentario:

“FungiGT logra democratizar el acceso a herramientas de análisis genómico sin sacrificar la calidad científica. La plataforma será muy útil para nuestro laboratorio y otros investigadores que trabajen con genomas fúngicos.”

6.2.4.1. Criterios de Aceptación Cumplidos

- **Funcionalidad completa:** Todos los módulos operan según lo especificado.
- **Usabilidad:** Interfaz accesible para usuarios con nivel intermedio.
- **Eficiencia:** Reducción significativa en tiempos de análisis.
- **Confiabilidad:** Resultados consistentes con herramientas individuales.
- **Escalabilidad:** Capacidad para manejar múltiples análisis simultáneos.

6.3. Análisis Crítico y Gestión de Riesgos

6.3.1. Riesgos Identificados y Mitigados

6.3.1.1. Riesgo Técnico: Complejidad de Integración

Descripción: La integración de múltiples herramientas bioinformáticas especializadas presentaba riesgo de incompatibilidades.

Mitigación implementada:

- Containerización individual de cada herramienta
- Definición de interfaces estándar entre servicios
- Testing exhaustivo de integración

Resultado: Se logró integración exitosa de todas las herramientas con mínimas incompatibilidades.

6.3.1.2. Riesgo de Cronograma: Ambición del Alcance

Descripción: El alcance inicial era muy ambicioso para el tiempo disponible.

Mitigación implementada:

- Priorización basada en feedback del cliente
- Desarrollo iterativo con entregas incrementales
- Eliminación de funcionalidades de baja prioridad (como IA avanzada)

Resultado: Se completaron todas las funcionalidades críticas dentro del cronograma.

6.3.1.3. Riesgo de Usabilidad: Complejidad Técnica

Descripción: Las herramientas bioinformáticas tienden a ser complejas y poco amigables.

Mitigación implementada:

- Diseño de interfaz intuitiva con validaciones
- Tooltips explicativos
- Feedback continuo del cliente sobre usabilidad

Resultado: La interfaz final fue bien recibida por el cliente y usuarios potenciales.

6.3.2. Lecciones Aprendidas

- **Importancia del feedback temprano:** Las decisiones arquitectónicas clave surgieron del diálogo con el cliente
- **Flexibilidad arquitectónica:** La arquitectura de microservicios permitió ajustes sin afectar todo el sistema
- **Simplicidad sobre sofisticación:** Soluciones simples y transparentes fueron preferidas sobre implementaciones complejas

Capítulo 7

Conclusiones

El proyecto ha logrado consolidar una herramienta integral que simplifica y optimiza el uso de herramientas bioinformáticas mediante la integración de un workflow estructurado. Este enfoque ha permitido automatizar procesos complejos, como el control de calidad, la anotación funcional y el análisis filogenético, en una plataforma web sólida, accesible y orientada a la experiencia del usuario. Las capacidades interactivas, como la visualización de datos y la generación de reportes, han facilitado enormemente la interpretación de resultados por parte de los investigadores.

Sin embargo, a pesar de las ventajas de una plataforma web, se ha identificado que una aplicación de escritorio podría ser más adecuada en ciertos casos, especialmente en entornos con limitaciones de conectividad o con requisitos específicos de procesamiento local. Las aplicaciones de escritorio, compatibles con múltiples sistemas operativos, ofrecen mayor flexibilidad al permitir que cada investigador configure su propio servidor de procesamiento. Esto sería particularmente útil para redes locales, donde los investigadores podrían trabajar de forma colaborativa y ver el progreso de sus flujos de trabajo en tiempo real.

Por ello, en próximas iteraciones del proyecto se planea migrar los flujos de trabajo a una plataforma de escritorio. Este enfoque permitirá no solo superar las limitaciones inherentes a los servidores web, sino también proporcionar una solución más personalizada para los investigadores, quienes podrán tener acceso directo a los flujos y datos de su trabajo. Además,

se contempla la posibilidad de que esta aplicación permita configurar servidores locales de manera sencilla, habilitando un entorno colaborativo en redes internas donde los investigadores puedan compartir y sincronizar datos y flujos de trabajo. De esta forma, se garantizará un acceso eficiente y segmentado a diferentes módulos y funcionalidades, adaptándose a las necesidades de cada usuario.

Asimismo, se proyecta una mayor modularidad en la plataforma, que permitirá a nuevos usuarios integrar herramientas adicionales siguiendo una plantilla y documentación específica. Esto fomentará la expansión y personalización del sistema, permitiendo que los investigadores desarrollen sus propios workflows, ya sean independientes o colaborativos. Es por eso que, en futuras iteraciones, se planea implementar esta flexibilidad para cubrir un espectro aún mayor de necesidades en el análisis genómico y filogenético.

Anexo A

Anexos: Códigos y planteamientos iniciales

A.1. Planteamientos iniciales

En las etapas iniciales de conceptualización del proyecto, se contempló una visión más innovadora que la tradicional interfaz web con scripts de ejecución. La propuesta original consistía en desarrollar un chatbot inteligente que pudiera:

- Procesar consultas en lenguaje natural mediante un modelo LLM (Large Language Model)
- Traducir estas consultas en ejecuciones automáticas de código
- Redirigir las peticiones del usuario a las funciones específicas de nuestros scripts

Sin embargo, durante el desarrollo nos encontramos con dos limitaciones significativas:

1. La escasez de datasets especializados en el dominio bioinformático
2. La ausencia de modelos pre-entrenados capaces de integrar efectivamente estas funcionalidades

Ante estas restricciones, se optó por implementar una aproximación más convencional pero robusta, basada en:

- Formularios estructurados para la ejecución de scripts
- Interfaces predefinidas para cada herramienta
- Una arquitectura que facilita la futura integración de capacidades de IA

Esta decisión estratégica nos permite:

- Establecer una base sólida para la recopilación de datos de uso
- Crear un conjunto de datos específico del dominio
- Facilitar el futuro entrenamiento de modelos especializados

Es importante destacar que, a pesar de esta adaptación en la implementación, el sistema mantiene íntegramente el flujo de trabajo establecido por nuestro cliente, el CBDAL, el cual se detalla a continuación:

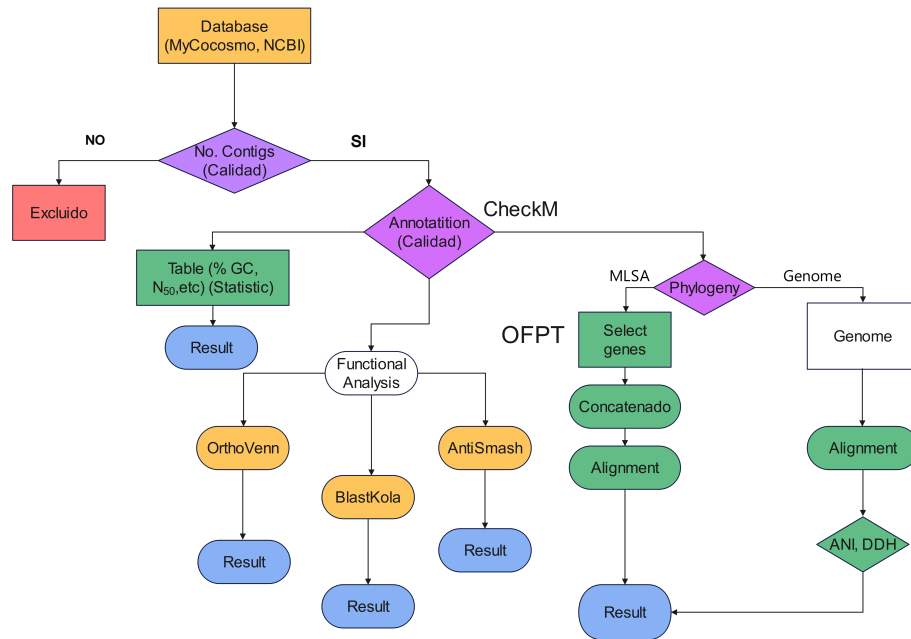


Figura A.1: Workflow propuesto inicialmente

A.2. Código y estructura

A.2.1. Estructura de Directorios Principal

```

FungiGT/
├── data/ (Datos persistentes)
│   ├── raw/
│   │   ├── genomes/
│   │   └── proteomes/
│   ├── results/
│   │   ├── quality_control/
│   │   ├── annotation/
│   │   ├── analysis/
│   │   └── visualization/
│   └── reports/
├── src/ (Código fuente)
└── core/ (Servicios core)
  
```

```

├── auth/ (Puerto 4001)
│   ├── database/
│   │   └── connection.js
│   ├── middleware/
│   │   └── authMiddleware.js
│   ├── models/
│   │   └── User.js
│   ├── routes/
│   │   └── authRoutes.js
│   ├── services/
│   │   └── AuthService.js
│   └── index.js
├── database/
│   └── connection.js
├── logging/
└── frontend/ (Puerto 4005)
    ├── app.js (Servidor Express)
    ├── views/
    │   ├── landing.ejs (Página inicio)
    │   ├── analyzer.ejs (Análisis)
    │   ├── annotator.ejs (Anotación)
    │   ├── visualizer.ejs (Visualización)
    │   ├── database.ejs (Base datos)
    │   ├── profile.ejs (Perfil usuario)
    │   ├── history.ejs (Historial)
    │   ├── config.ejs (Configuración)
    │   └── auth/
    │       ├── login.ejs
    │       ├── register.ejs
    │       └── logout.ejs
    │   └── partials/ (Componentes)
    │       ├── header.ejs
    │       ├── footer.ejs
    │       ├── file-explorer.ejs
    │       └── drfungito-chat.ejs
    └── public/
        ├── css/
        │   └── styles.css
        └── js/
            ├── main.js
            ├── braker3-client.js
            ├── chatbot.js
            └── theme-manager.js

```

```

├── images/
├── examples/
├── routes/
│   ├── index.js
│   ├── auth.js
│   ├── analyzer.js
│   ├── annotator.js
│   ├── visualizer.js
│   └── database.js
├── middleware/
│   └── authProxyMiddleware.js
├── models/
│   └── User.js
├── config/
│   └── database.js
├── utils/
│   └── authClient.js

```

A.2.2. Módulos de Microservicios

FungiGT/ (*continuación*)

```

├── src/modules/ (Microservicios)
│   ├── file_manager/ (Puerto 4002)
│   │   ├── server.js
│   │   ├── file_router.js
│   │   ├── storage_service.js
│   │   └── data/
│   ├── visualization/ (Puerto 4003)
│   │   ├── app.py (Servidor Flask)
│   │   ├── visualizers/
│   │   │   ├── base_visualizer.py
│   │   │   ├── bindash_visualizer.py
│   │   │   ├── annotations_visualizer.py
│   │   │   ├── checkm_visualizer.py
│   │   │   ├── hits_visualizer.py
│   │   │   ├── seed_orthologs_visualizer.py
│   │   │   └── dna_features_visualizer.py
│   │   └── genomic_visualizer/
│   │       ├── server.py
│   │       └── BioGraphmaker/
│   │           └── graph_maker/
│   │               └── PythonScripts/

```

```

├── process_annotations.py
├── process_hmmer_data.py
├── process_seed_orthologs.py
├── output/
├── uploads/
└── server.js
├── data/
├── uploads/
├── outputs/
├── quality_control/ (Puerto 4004)
│   ├── checkm/
│   │   ├── server.js
│   │   └── data/
│   └── data/
├── acquisition/ (Puerto 4006)
│   ├── ncbi_cli/
│   │   ├── server.js
│   │   ├── service.js
│   │   ├── router.js
│   │   └── tmp/
│   └── analysis/
│       ├── bindash/ (Puerto 4007)
│       │   └── server.js
│       ├── egglog/ (Puerto 3002)
│       │   ├── server.js
│       │   └── logs/
│       └── blastn/
├── annotation/
│   ├── braker3/ (Puerto 3004)
│   │   ├── server.js
│   │   └── run-server.js
├── ai_agent/
│   ├── drfungito/ (Puerto 4009)
│   │   ├── server.js
│   │   ├── data/
│   │   ├── logs/
│   │   ├── temp/
│   │   ├── SISTEMA_REPORTES.md
│   │   ├── DEBUG_REPORTES.md
│   │   ├── debug_report.ps1
│   │   ├── debug_report.bat
│   │   └── config.example.js

```

A.2.3. Infraestructura y Scripts de Gestión

```
FungiGT/ (continuación)
├── infrastructure/ (Configuración Docker)
│   ├── docker/
│   │   ├── auth/
│   │   │   └── Dockerfile
│   │   ├── frontend/
│   │   │   └── Dockerfile
│   │   ├── file_manager/
│   │   │   ├── Dockerfile
│   │   │   └── docker-compose.yml
│   │   ├── visualization/
│   │   │   └── Dockerfile
│   │   ├── quality_control/
│   │   │   └── Dockerfile
│   │   ├── acquisition/
│   │   │   ├── Dockerfile
│   │   │   └── docker-compose.yml
│   │   ├── analysis/
│   │   │   ├── bindash/
│   │   │   │   └── Dockerfile
│   │   │   ├── eggnog/
│   │   │   │   ├── Dockerfile
│   │   │   │   └── docker-compose.yml
│   │   ├── annotation/
│   │   │   ├── Dockerfile
│   │   │   └── docker-compose.yml
│   │   ├── ai_agent/
│   │   │   └── Dockerfile
│   │   └── backend/
│   └── scripts/ (Scripts de gestión)
│       ├── setup/
│       │   ├── install.py
│       │   ├── setup_fungigt.py
│       │   ├── start_services.py
│       │   ├── stop_services.py
│       │   ├── quick_start.py
│       │   ├── debug_installation.py
│       │   ├── check_structure.py
│       │   ├── fix_docker_network.py
│       │   ├── linux_setup.sh
│       │   └── INSTALL_GUIDE.md
```

```
├── mongo-init/
│   └── init-db.js
├── eggnog_db_manager.py
├── eggnog_helper.sh
├── quick_start.py
├── fix_docker_error.bat
├── requirements.txt
├── docs/ (Documentación)
│   ├── dockerization_report.tex
│   ├── file_manager_volumes.tex
│   └── justificacion_contenedores.tex
├── logs/ (Logs del sistema)
├── tmp/ (Archivos temporales)
├── docker-compose.yml (Orquestación completa)
├── README.md (Documentación principal)
├── claude.md (Documentación para LLMs)
├── .gitignore
└── .dockerignore
```

- Repositorio del proyecto

Referencias Técnicas

- [1] C. Kubicek, A. Herrera-Estrella, V. Seidl-Seiboth, D. Martínez, I. Druzhinina, M. Thon, S. Zeilinger, S. Casas-Flores, B. Horwitz, P. Mukherjee *et al.*, “Comparative genome sequence analysis underscores mycoparasitism as the ancestral life style of *Trichoderma*,” *Genome Biology*, vol. 12, no. 4, p. R40, 2011, doi: 10.1186/gb-2011-12-4-r40.
- [2] N. Nguyen, D. Smith, K. Peay, and P. Kennedy, “FUNGuild: An open annotation tool for parsing fungal community datasets by ecological guild,” *Fungal Ecology*, vol. 20, pp. 241–248, 2015, doi: 10.1016/j.funeco.2015.06.003.
- [3] L. Li, C. Stoeckert, and D. Roos, “OrthoMCL: Identification of ortholog groups for eukaryotic genomes,” *Genome Research*, vol. 13, no. 9, pp. 2178–2189, 2003, doi: 10.1101/gr.1224503.
- [4] T. G. Community, “The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update,” *Nucleic Acids Res.*, vol. 50, no. W1, pp. W345–W351, 2022. [Online]. Available: <https://doi.org/10.1093/nar/gkac247>
- [5] N. Segata, L. Waldron, A. Ballarini, V. Narasimhan, O. Jousson, and C. Huttenhower, “Metagenomic microbial community profiling using unique clade-specific marker genes,” *Nature Methods*, vol. 9, no. 8, pp. 811–814, 2012, doi: 10.1038/nmeth.2066.
- [6] K. Hoff and M. Stanke, “Predicting genes in single genomes with AUGUSTUS,” *Current Protocols in Bioinformatics*, vol. 65, no. 1, p. e57, 2019, doi: 10.1002/cpbi.57.

- [7] D. Railin, L. Anderson, S. Haridas, B. Brachi, P. Zhou, K. Barry, R. Riley, E. Lindquist, I. Grigoriev, and J. Spatafora, “Genome sequencing and analysis of *Trichoderma virens* reveals its evolutionary adaptation to plant-associated lifestyles,” *BMC Genomics*, vol. 20, no. 1, p. 1002, 2019, doi: 10.1186/s12864-019-6354-y.
- [8] S. Yiming, L. Zheng, G. Feng, and Z. Guangyu, “Genome-wide characterization of the MAPK gene family in wheat (*Triticum aestivum* L.),” *BMC Genomics*, vol. 20, p. 880, 2019, doi: 10.1186/s12864-019-6294-6.
- [9] A. Tekpinar and A. Kalmanc, “Evaluation of various DNA markers in the molecular identification of some *Penicillium* species from cheese,” *Biotechnic & Histochemistry*, vol. 94, no. 2, pp. 123–131, 2019, doi: 10.1080/10520295.2018.1549895.
- [10] Y. Chen and J. Zou, “GenePT: A simple but effective foundation model for genes and cells built from ChatGPT,” <https://doi.org/10.1101/2023.10.16.562533>, 2024, bioRxiv.
- [11] D. H. Parks, M. Imelfort, C. T. Skennerton, P. Hugenholtz, and G. W. Tyson, “CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes,” *Genome Research*, vol. 25, no. 7, pp. 1043–1055, 2015, doi: 10.1101/gr.186072.114.
- [12] N. A. O’Leary, E. Cox, J. B. Holmes, W. R. Anderson, R. Falk, V. Hem, M. T. N. Tsuchiya, G. D. Schuler, X. Zhang, J. Torcivia, A. Ketter, L. Breen, J. Cothran, H. Bajwa, J. Tinne, P. A. Meric, W. Hlavina, and V. A. Schneider, “Exploring and retrieving sequence and metadata for species across the tree of life with ncbi datasets,” *Scientific Data*, vol. 11, no. 1, p. 732, 2024. [Online]. Available: <https://doi.org/10.1038/s41597-024-03571-y>
- [13] L. Gabriel, T. Brůna, K. J. Hoff, M. Ebel, A. Lomsadze, M. Borodovsky, and M. Stanke, “Braker3: Fully automated genome annotation using rna-seq and protein evidence with genemark-etp, augustus and tsebra,” *bioRxiv*, 2024, preprint. [Online]. Available: <https://doi.org/10.1101/2023.06.10.544449>

- [14] C. P. Cantalapiedra, A. Hernández-Plaza, I. Letunic, P. Bork, and J. Huerta-Cepas, “eggno-mapper v2: Functional annotation, orthology assignments, and domain prediction at the metagenomic scale,” *Molecular Biology and Evolution*, vol. 38, no. 12, pp. 5825–5829, 2021. [Online]. Available: <https://doi.org/10.1093/molbev/msab293>
- [15] X. Zhao, “Bindash, software for fast genome distance estimation on a typical personal laptop,” *Bioinformatics*, vol. 35, no. 4, pp. 671–673, 2019. [Online]. Available: <https://doi.org/10.1093/bioinformatics/bty651>

Referencias Generales

- [16] Lindsey B. Huberman. Developing functional genomics platforms for fungi. *mSystems*, 6(4):e00730–21, Aug 2021.
- [17] Samantha C. Karunarathna, Milan C. Samarakoon, and Indunil Senanayake. Recent advances in fungal taxonomy and phylogeny. *New Zealand Journal of Botany*, 62:119–122, Aug 2024.
- [18] Shelby J. Priest, Vikas Yadav, and Joseph Heitman. Advances in understanding the evolution of fungal genome architecture. *F1000Research*, 9:776, Jul 2020.