**Santiago de Cali, 31 de octubre del 2025**

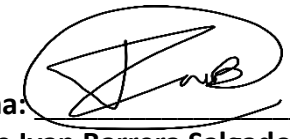
Doctor
**Diego Luis Linares O.**
Director Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana de Cali

**Asunto:** Presentación para evaluación del proyecto aplicado

Cordial Saludo,

Con el fin de cumplir con los requisitos exigidos por la Universidad para optar por el título de Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto denominado "Modelo predictivo de resistencia antibiótica en bacterias bucales mediante análisis de marcadores genómicos de patogenicidad.", el cual fue realizado por los estudiantes Jorge Ivan Barrera Salgado y Ana Luisa Sotelo Ariza con códigos 8993445 y 8993444 respectivamente, pertenecientes a la Maestría en Ciencia de Datos, bajo la dirección de Fabian Tobar Tosse. El suscrito director del Proyecto Aplicado autoriza para que se proceda a hacer la evaluación de este proyecto, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado y sustentado oficialmente.
Atentamente,

**Firma:** 
**Ana Luisa Sotelo Ariza.**
**C.C. 1000773821 de Bogotá.**

**Firma:** 
**Jorge Ivan Barrera Salgado.**
**C.C. 1014305420 de Bogotá.**

**Documentación anexa:**
Resumen del Proyecto Aplicado en formato digital.
Una copia digital (PDF) del documento del proyecto aplicado.

**FICHA RESUMEN**
**PROYECTO APLICADO – MAESTRÍA EN CIENCIA DE DATOS**

**TÍTULO: Modelo predictivo de resistencia antibiótica en bacterias bucales mediante análisis de marcadores genómicos de patogenicidad.**

1. **ÁREA DE TRABAJO:** Investigación biomédica.
2. **TIPO DE PROYECTO:** Aplicado.
3. **ESTUDIANTE(S):** Barrera Salgado Jorge Iván y Sotelo Ariza Ana Luisa.
4. **CORREO ELECTRÓNICO:** jbarreras@javerianacali.edu.co y anasoteloariza@javerianacali.edu.co
5. **DIRECCIÓN Y TELEFONO:** 3213842203 y 3227880972.
6. **DIRECTOR:** Tobar Tosse Fabian.
7. **VINCULACIÓN DEL DIRECTOR:** Docente del departamento ciencias básicas de la salud universidad javeriana de Cali.
8. **CORREO ELECTRÓNICO DEL DIRECTOR**: ftobar@javerianacali.edu.co
9. **PALABRAS CLAVE:**

   **MACHINE LEARNING**: Se refiere al proceso mediante el cual las computadoras aprenden de los datos y pueden mejorar su rendimiento en una tarea específica sin ser programadas explícitamente para esa tarea. Es una subdisciplina de la inteligencia artificial que se basa en algoritmos y modelos matemáticos para permitir que los sistemas informáticos aprendan patrones y tomen decisiones con mínima intervención humana.

   **MARCADORES**: En el contexto del análisis de datos y estadísticas, los marcadores son variables o indicadores utilizados para medir o representar ciertas características o comportamientos en un conjunto de datos. Los marcadores pueden ser biológicos, como los biomarcadores en medicina, o analíticos, como los indicadores de rendimiento en sistemas computacionales.

   **METAGENOMA**: El conjunto de material genético obtenido directamente de muestras ambientales con una mezcla de microorganismos, como bacterias, virus y hongos, que permite estudiar la diversidad y función de los microorganismos en su entorno natural, sin necesidad de aislar y cultivar los organismos individualmente.

   **PATOGENICIDAD**: La patogenicidad es un concepto crucial en la microbiología y la medicina, ya que, determina la habilidad de un patógeno para invadir un huésped y causar una enfermedad.

   **VIRULENCIA**: En el contexto de la microbiología y la medicina, la virulencia se refiere a la capacidad que tiene un agente patógeno, como un virus, una bacteria o un parásito, para

producir enfermedad en un organismo infectado. Es decir, la virulencia es la capacidad de un microorganismo de causar enfermedad en el huésped infectado, que facilitan la infección y el daño a los tejidos del huésped.

10. **FECHA DE INICIO:** 29 de abril de 2024
11. **FECHA DE FINALIZACIÓN:** 18 meses
12. **RESUMEN:**

El presente proyecto desarrolló un modelo de aprendizaje automático para la validación y recomendación personalizada de tratamientos antibióticos, basado en la identificación y clasificación de marcadores de patogenicidad en muestras de metagenomas. Se abordó el problema de la baja precisión en la detección de patógenos altamente resistentes, lo que limita la toma de decisiones terapéuticas en contextos clínicos.

Se construyó un conjunto de datos integrado por bacterias humanas, mayoritariamente orales, incluyendo información sobre genomas, fenotipos de resistencia y tratamientos antibióticos. Mediante técnicas de minería de datos y aprendizaje supervisado (XGBoost, Random Forest y Regresión Logística), así como modelos no supervisados (KMeans con PCA), se logró clasificar comunidades bacterianas, identificar perfiles de resistencia y proponer esquemas terapéuticos más efectivos.

Los resultados demostraron que el modelo XGBoost alcanzó el mejor rendimiento en la predicción de resistencia (F1 ponderado = 0.74), mientras que los clústeres generados permitieron caracterizar agrupamientos bacterianos con perfiles epidemiológicos y de resistencia distintos. Se concluye que este enfoque es viable para fortalecer la vigilancia epidemiológica y la medicina personalizada en el contexto de infecciones bacterianas resistentes.

# Modelo predictivo de resistencia antibiótica en bacterias bucales mediante análisis de marcadores genómicos de patogenicidad.

**Ana Luisa Sotelo Ariza 8993444.**

**Jorge Ivan Barrera Salgado 8993445.**

**Proyecto Aplicado para optar al título de**

**Magister en Ciencia de Datos.**

**Tobar Tosse Fabian.**

**FACULTAD DE INGENIERÍA Y CIENCIAS.**

**MAESTRÍA EN CIENCIA DE DATOS.**

**SANTIAGO DE CALI, OCTUBRE 31 DE 2025.**

TABLA DE CONTENIDO

# LISTA DE FIGURAS

## LISTA DE TABLAS



## LISTA DE ANEXOS

No se encuentran elementos de tabla de ilustraciones.

# 1 INTRODUCCIÓN

La identificación de infecciones bacterianas resistentes a tratamientos antibióticos ha representado un desafío crítico en la medicina moderna, debido al aumento de patógenos con alta virulencia y multirresistencia. La Organización Mundial de la Salud (OMS) ha advertido que esta problemática es una de las principales amenazas para la salud pública global. Estas infecciones prolongan las hospitalizaciones, elevan la mortalidad y aumentan significativamente los costos sanitarios. Por ello, la detección oportuna de marcadores de patogenicidad en muestras metagenómicas se vuelve fundamental para diseñar terapias antibióticas más efectivas y personalizadas.

Sin embargo, los métodos tradicionales de identificación bacteriana, como los cultivos, resultan lentos y limitados para capturar la diversidad microbiana en su totalidad. En este contexto, emergen técnicas computacionales como el aprendizaje automático (*machine learning*), capaces de analizar grandes volúmenes de datos genéticos, identificar patrones de resistencia y predecir comportamientos clínicos con mayor rapidez y precisión.

Durante este proyecto se integró un conjunto de datos derivados de múltiples fuentes de BV-BRC, incluyendo genes, fenotipos de resistencia y metadatos genómicos de bacterias asociadas al ser humano. Se emplearon algoritmos de clasificación (*XGBoost*, *Random Forest* y Regresión Logística) y *clustering* (*KMeans + PCA*) para modelar la resistencia antibiótica y caracterizar perfiles bacterianos. Todo el proceso fue automatizado desde la línea de comandos, permitiendo la recolección y preprocesamiento eficiente de miles de muestras.

Como resultado, se construyó un modelo robusto basado en *XGBoost* con un F1 ponderado de 0.74, y se identificaron cuatro agrupaciones bacterianas con patrones de resistencia diferenciados. Estas evidencias permiten validar el uso de aprendizaje automático como una herramienta eficaz para apoyar decisiones clínicas, establecer vigilancia epidemiológica y sugerir tratamientos antibióticos personalizados en casos de infecciones resistentes.

# 2 DEFINICIÓN DEL PROBLEMA

## 2.1 PLANTEAMIENTO DEL PROBLEMA

La identificación y tratamiento de infecciones bacterianas representaron un desafío crítico en la medicina moderna, especialmente debido al aumento de bacterias con alta virulencia y resistencia a los antibióticos convencionales. La OMS declaró que la resistencia a los antibióticos es una de las mayores amenazas para la salud mundial, la seguridad alimentaria y el desarrollo económico [6]. Las infecciones bacterianas resistentes no solo prolongaron las estancias hospitalarias y aumentaron los costos de atención médica, sino que también elevaron la mortalidad. En este contexto, la precisión en la identificación de marcadores de patogenicidad en muestras de metagenoma resultó crucial para desarrollar tratamientos personalizados que pudieran abordar estas infecciones de manera eficaz.

En el momento de ejecución del proyecto, la falta de métodos precisos y rápidos para identificar patógenos y sus factores de virulencia en muestras de metagenoma limitaba significativamente la capacidad de los profesionales de la salud para ofrecer tratamientos adecuados. Los métodos tradicionales, como los cultivos bacterianos, resultaban lentos y, a menudo, incapaces de detectar todas las bacterias presentes en una muestra. Además, las técnicas de secuenciación de próxima generación (*NGS*) generaban grandes cantidades de datos cuya interpretación resultaba compleja y requería herramientas avanzadas [7]. Sin una identificación precisa y rápida de los patógenos y sus características de resistencia, los tratamientos a menudo se administraban de manera empírica, lo que podía ser ineficaz y contribuir al aumento de la resistencia a los antibióticos [8].

Uno de los principales desafíos en este campo fue la capacidad de procesar y analizar grandes volúmenes de datos genómicos junto con la implementación efectiva de algoritmos de aprendizaje automático. La ciencia de datos ofreció soluciones prometedoras mediante técnicas avanzadas de análisis de datos y aprendizaje automático. Estas técnicas permitieron el procesamiento eficiente y análisis detallado de los datos genómicos, facilitando la identificación precisa y rápida de patógenos, sus factores de virulencia y los tratamientos antibióticos asociados a bacterias. A través de algoritmos de aprendizaje automático como *Random Forest*, Regresión Lineal, Regresión Logística o *Gradient Boosting*, fue posible entrenar modelos que reconocieran patrones específicos en los datos genómicos, permitiendo detectar bacterias patógenas y sus resistencias a antibióticos de manera más efectiva.

Para la fase de segmentación del estudio, se utilizó un conjunto de datos clínicos compuesto por muestras biológicas de pacientes humanos, que incluyen placa dental, mucosa bucal, saliva y tejido tumoral oral. Estas muestras se toman de microbiota oral y resistencia antimicrobiana, y proporcionan un enfoque aplicado para evaluar el modelo propuesto en contextos clínicos reales. Este enfoque permite estudiar la interacción entre comunidades

bacterianas presentes en la cavidad bucal y su relación con procesos infecciosos o patológicos, fortaleciendo la relevancia biomédica del modelo planteado.

El proyecto "Modelo predictivo de resistencia antibiótica en bacterias bucales mediante análisis de marcadores genómicos de patogenicidad." abordó esta problemática mediante el uso de técnicas avanzadas de análisis de datos, enfocándose en mejorar la precisión y rapidez en la identificación de bacterias patógenas y sus factores de virulencia a partir de metagenomas. La aplicación de algoritmos de aprendizaje automático a los datos masivos no solo facilitó la identificación efectiva de patógenos, sino que también permitió predecir su comportamiento y resistencia a antibióticos, proporcionando así una herramienta valiosa para la toma de decisiones clínicas y epidemiológicas.

## 2.2 FORMULACIÓN DEL PROBLEMA

¿Cómo pueden las técnicas de aprendizaje automático y el análisis de grandes volúmenes de datos genómicos mejorar la precisión en la identificación de factores de virulencia bacteriana en muestras de metagenoma y predecir la resistencia a los antibióticos?

### 2.2.1 PREGUNTAS DE SISTEMATIZACIÓN

- ¿Qué métodos de minería de datos fueron más adecuados para clasificar las diferentes comunidades bacterianas en muestras de metagenomas y cómo fueron implementados para identificar y caracterizar la diversidad microbiana?
- ¿Cómo se identificaron y analizaron marcadores específicos de patogenicidad y resistencia a antibióticos en las bacterias obtenidas de los metagenomas utilizando algoritmos de *machine learning*?
- ¿Qué pasos se requirieron para desarrollar un conjunto de datos de entrenamiento que incluyera bacterias y sus respectivos marcadores de patogenicidad y tratamientos antibióticos?
- ¿Qué algoritmos de *machine learning* fueron más efectivos para la clasificación de bacterias y la recomendación de tratamientos antibióticos basados en el perfil metagenómico?
- ¿Cómo se implementó y evaluó un modelo de *machine learning* para predecir la resistencia antibiótica y el perfil metagenómico, optimizando la precisión y rapidez en la identificación de antibióticos?

# 3 OBJETIVOS DEL PROYECTO

## 3.1 OBJETIVO GENERAL

Implementar un modelo de aprendizaje automático para la predicción de resistencia antibiótica en bacterias del microbioma bucal, basado en la identificación y clasificación de marcadores genómicos de patogenicidad presentes en muestras metagenómicas.

## 3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar un conjunto de datos de entrenamiento compuesto por bacterias bucales y sus respectivos marcadores genómicos de patogenicidad y resistencia, integrando información proveniente de bases bioinformáticas y de muestras clínicas humanas.

- Aplicar técnicas de minería de datos y análisis exploratorio para identificar patrones de diversidad microbiana en las comunidades bacterianas bucales y su relación con los perfiles genómicos.

- Analizar los marcadores genéticos asociados a la patogenicidad y la resistencia antibiótica, determinando su relevancia e interacciones mediante métodos de minería de datos.

- Implementar un modelo predictivo supervisado capaz de estimar la probabilidad de resistencia antibiótica en bacterias bucales, contribuyendo a la comprensión de los mecanismos de resistencia y a la optimización de futuras estrategias terapéuticas.

# 4 MARCO TEÓRICO Y ANTECEDENTES

## 4.1 MARCO TEÓRICO

En los últimos años, el estudio de los microorganismos y su impacto en la salud humana avanzó significativamente gracias a técnicas innovadoras como la metagenómica y la inteligencia artificial.

La metagenómica permitió analizar el material genético de comunidades microbianas directamente desde muestras ambientales, eludiendo la necesidad de cultivar los microorganismos en el laboratorio. Esto revolucionó la comprensión de la diversidad y funciones microbianas en distintos entornos. Paralelamente, los algoritmos de *machine learning* se consolidaron como herramientas poderosas para manejar y analizar grandes volúmenes de datos genéticos, mejorando la identificación de patógenos y la predicción de sus funciones.

Este marco teórico presenta la integración de estas tecnologías, resaltando su relevancia en la identificación de factores de virulencia bacteriana y la personalización de tratamientos clínicos. Se desarrollaron y aplicaron conceptos clave de la metagenómica, los factores de virulencia bacteriana y el *machine learning*, como eje central del análisis y validación de los resultados en este trabajo.

### Metagenómica

La metagenómica fue la técnica empleada para estudiar el material genético recuperado directamente de muestras ambientales, lo que permitió el análisis de comunidades microbianas sin necesidad de cultivar los microorganismos [10]. El proyecto utilizó este enfoque basado en la extracción de ADN de muestras orales humanas, seguido de la secuenciación y análisis del material genético para caracterizar las bacterias presentes. La posibilidad de analizar grandes volúmenes de datos genómicos facilitó la comprensión de la diversidad y función de los microbiomas humanos [12].

Este método resultó esencial para identificar patógenos no cultivables, lo cual contribuyó a una visión más completa de las enfermedades infecciosas. Se respaldó en estudios que evidenciaron su eficacia en la detección de nuevas especies microbianas y en la caracterización genética en distintos entornos [11][12][13].

### Factores de Virulencia Bacteriana

Los factores de virulencia bacteriana fueron claves en este proyecto, ya que permitieron detectar cómo las bacterias lograban evadir el sistema inmunológico, colonizar tejidos y causar enfermedad. Estos factores incluyeron toxinas, adhesinas y enzimas que degradan tejidos [13]. Su identificación facilitó la clasificación de bacterias patógenas y respaldó el desarrollo del modelo predictivo de resistencia y tratamiento personalizado.

Las investigaciones consultadas aportaron una base sólida sobre los mecanismos de infección

y estrategias terapéuticas. Los autores [13] resaltaron temas comunes en patogenicidad, mientras que [14] los autores explicaron procesos de adhesión bacteriana, y [15] los autores propusieron modelos de vacunación basados en estos factores.

### Algoritmos de Machine Learning en Metagenómica

En el desarrollo del proyecto se implementaron algoritmos de *machine learning* para analizar datos metagenómicos. Estos modelos como *Random Forest*, Regresión Logística y *XGBoost* permitieron clasificar datos, predecir resistencia antibiótica y descubrir patrones complejos invisibles para métodos tradicionales [2][3].

La inclusión de estos algoritmos favoreció la predicción de genes de resistencia antimicrobiana y la identificación de nuevos factores de virulencia [2]. El proyecto se sustentó en investigaciones como las de los autores [16], [17], y [18], quienes validaron el uso del *machine learning* para extraer conocimiento desde datos genéticos complejos.

### Integración de Machine Learning y Metagenómica

La integración entre metagenómica y *machine learning* fue el eje del modelo predictivo desarrollado. Esta combinación permitió aumentar la precisión diagnóstica, anticipar resistencias y proponer tratamientos personalizados, apoyándose en el perfil genético de las bacterias detectadas [22].

Los análisis desarrollados en el proyecto confirmaron que el uso conjunto de estas técnicas mejoró la efectividad terapéutica. Referencias como [22], [20] y [21], respaldaron metodológicamente esta integración.

### Impacto de la Metagenómica y Machine Learning en la Medicina Personalizada

Durante el proyecto, se demostró que la medicina personalizada se vio fortalecida por la sinergia entre la metagenómica y el *machine learning*. Estas herramientas permitieron diseñar terapias basadas en la composición del microbioma y en la resistencia específica del patógeno, reduciendo efectos adversos y tiempos de recuperación [19].

Los autores [23] abordaron los desafíos clínicos de aplicar estos métodos, mientras que Los autores [24] y [16] reforzaron la importancia de la vigilancia genómica en tiempo real para enfrentar las infecciones resistentes.

### Posibles Modelos en Base al Marco Teórico y Referencias Aplicadas en el Proyecto

1. **Modelo de clasificación**: Se implementó un modelo supervisado de clasificación para identificar patógenos y categorizar muestras de metagenomas según su resistencia antibiótica*. Random Forest* y *XGBoost* demostraron ser los más eficaces.

2. **Modelo de regresión**: Se utilizó un enfoque de regresión logística para evaluar relaciones entre características genómicas y respuesta a antibióticos, facilitando predicciones sobre efectividad terapéutica.
3. **Modelo de agrupamiento**: A través del algoritmo K-Means, se realizó segmentación no supervisada de los datos, identificando grupos bacterianos con perfiles de resistencia y patogenicidad similares.
4. **Modelo de detección de anomalías**: Aunque no fue el enfoque central, se valoró la utilidad de modelos como **Isolation Forest** para detectar patrones genómicos inusuales que podrían estar vinculados a nuevas formas de resistencia.

### 4.2 ANTECEDENTES

En el estudio [7] se realizó una revisión exhaustiva de las etapas que componen la metagenómica, desde el muestreo y la secuenciación hasta el análisis de datos genómicos. Dicha investigación sirvió como guía metodológica para estructurar el flujo de trabajo de obtención, limpieza y análisis de los datos en el presente proyecto. Sin embargo, el estudio [7] no abordó estrategias de minería de datos ni el uso de modelos de aprendizaje automático, lo que marcó una diferencia clave con nuestra propuesta.

El trabajo [8] exploró el potencial diagnóstico de la metagenómica en contextos clínicos, demostrando su relevancia para identificar infecciones complejas directamente desde muestras humanas. Aunque reforzó el fundamento clínico de nuestro enfoque, no incluyó modelos predictivos ni algoritmos de clasificación, por lo que el presente proyecto complementa dicho vacío mediante el uso de aprendizaje automático.

Por su parte, la revisión sistemática [24] recopiló diversas investigaciones que combinaban metagenómica con técnicas de aprendizaje automático, contextualizando el uso de modelos supervisados como Random Forest y XGBoost. Sin embargo, ningún estudio analizado en [24] abordó específicamente la clasificación de factores de virulencia ni su relación con tratamientos antibióticos personalizados, lo que justificó la construcción de un modelo propio en esta investigación.

Finalmente, el estudio [25] destacó cómo el aprendizaje automático ha mejorado significativamente el análisis de datos metagenómicos y la predicción de resistencia antibiótica. Aunque [25] se centró en aplicaciones generales, sin un enfoque clínico, sus aportes metodológicos sirvieron de base para seleccionar los algoritmos más robustos implementados en el presente trabajo.

# 5 DISEÑO Y ARQUITECTURA DEL PROYECTO

## 5.1 DISEÑO DE LA ARQUITECTURA

El presente proyecto se estructuró bajo un enfoque de Ciencia de Datos aplicada a la bioinformática, con el objetivo de desarrollar un modelo de aprendizaje automático capaz de predecir la resistencia antibiótica en bacterias del microbioma bucal, a partir del análisis de marcadores genómicos de patogenicidad obtenidos de muestras bacterianas humanas.

Para alcanzar este propósito, se diseñó una arquitectura modular e integrada, que garantiza la trazabilidad completa desde la adquisición y procesamiento de los datos genómicos hasta la generación de modelos predictivos e interpretables, orientados a apoyar la comprensión de los mecanismos de resistencia presentes en las comunidades microbianas orales.
El sistema se concibió como un flujo *end-to-end*, compuesto por las siguientes fases principales:

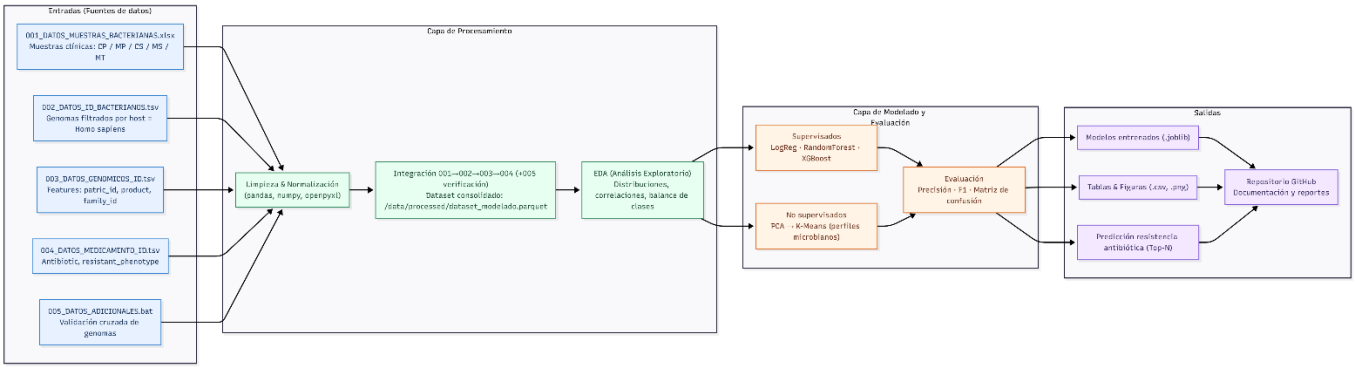



*Figura 1: Diagrama de Arquitectura.*

### 5.1.1 ENTRADAS (FUENTES DE DATOS)

Incluye la recopilación de muestras clínicas (placa dental, saliva y tejido tumoral) y la descarga de datos genómicos desde la base *BV-BRC*, empleando su interfaz de línea de comandos `(p3-scripts)`.
En esta etapa se generaron los archivos 002 a 005, que contienen información biológica, genómica y fenotípica de las bacterias analizadas.

### 5.1.2 CAPA DE PROCESAMIENTO

A través de *scripts* en Python, se normalizan y unifican los archivos obtenidos, cruzando identificadores únicos de genomas (*genome_id*) y eliminando duplicados.
Esta integración produce un conjunto consolidado de datos que servirá de base para la generación del *dataset* de entrenamiento.
Además, se realiza el análisis exploratorio de datos en donde se aplican técnicas de minería

de datos y análisis estadístico para identificar patrones de resistencia, distribuciones por país y año, y relaciones entre marcadores de virulencia y antibióticos.

#### 5.1.3 CAPA DE MODELADO Y EVALUACIÓN

Se implementan algoritmos como *XGBoost*, *Random Forest* y *K-Means* para predecir resistencias, clasificar bacterias y agrupar comunidades microbianas con comportamientos similares.

#### 5.1.4 SALIDAS

Finalmente, los resultados se almacenan en carpetas estructuradas dentro del repositorio *GitHub*, incluyendo modelos entrenados, métricas de validación, figuras y reportes técnicos.

### 5.2 CONFIGURACIÓN DEL ENTORNO

#### 5.2.1 HERRAMIENTAS UTILIZADAS

Para la implementación del proyecto se emplearon herramientas de código abierto que permiten garantizar la **reproducibilidad**, **eficiencia** y **trazabilidad** de todo el proyecto. A continuación, se describen las principales:

- **Anaconda:** Utilizada para la gestión del entorno de desarrollo, instalación de dependencias y control de versiones de librerías. Su uso permitió mantener un entorno aislado y reproducible durante todo el ciclo del proyecto.
- **Python 3.10:** Lenguaje base para la ejecución de los procesos de limpieza, análisis exploratorio y modelado de datos, gracias a su versatilidad y amplia disponibilidad de librerías científicas.
- **Git y GitHub:** utilizados para el control de versiones del código, documentación y datos. Se trabajó mediante ramas (main, dev, experiment) y commits descriptivos que aseguran la
- **Mermaid:** Herramienta de diagramación empleada para la representación visual de la arquitectura del sistema, flujos de datos y relaciones entre procesos. Su integración en el repositorio facilita la documentación estructurada y la comprensión de la arquitectura del proyecto.
- **BV-BRC Command Line Interface (p3-scripts):** Herramienta empleada para la extracción de datos genómicos y fenotípicos desde la base ***Bacterial and Viral Bioinformatics Resource Center (BV-BRC),*** permitiendo filtrar resultados por especie (*Homo sapiens*) y generar archivos en formato. tsv.

  Entre sus principales ventajas se destacan:

  - **Extensa cobertura de datos:** alberga millones de genomas bacterianos y virales completamente curados, con anotaciones verificadas y actualizadas de forma continua por organismos especializados.

- **Infraestructura bioinformática integrada:** ofrece tanto una interfaz web como una línea de comandos (CLI) que facilitan la automatización de procesos de consulta, descarga y análisis de datos.
- **Diversidad de información biológica: i**ncluye datos sobre genes, familias de proteínas, fármacos, mecanismos de resistencia antimicrobiana, funciones biológicas y anotaciones moleculares, lo que permite abordar estudios desde distintas perspectivas.
- **Automatización del análisis:** permite ejecutar consultas y generar salidas reproducibles mediante comandos especializados como p3-get-genome-features, p3-get-genome-drugs o p3-get-family-data, optimizando la extracción y estructuración de grandes volúmenes de datos genómicos.

Para realizar la extracción de los datos se debe usar el siguiente modelo [26]:

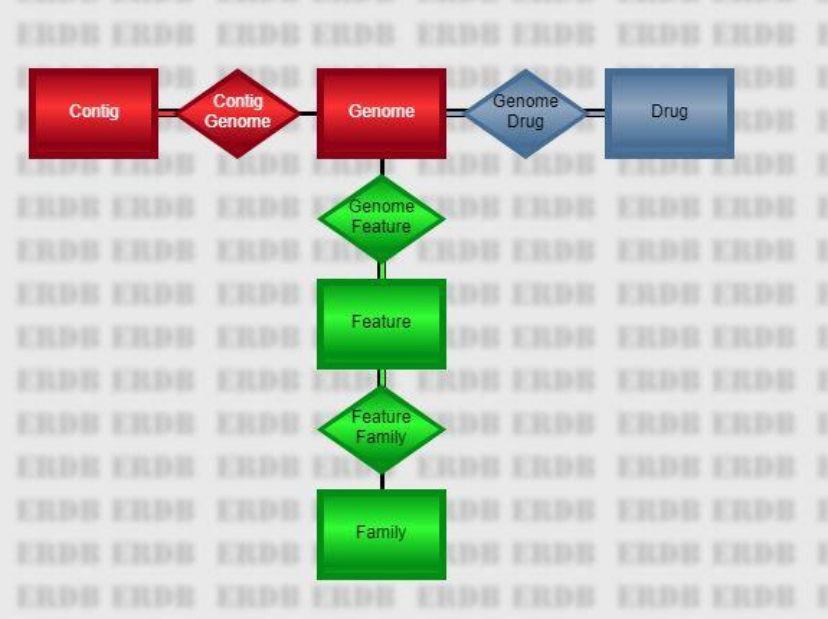



*Figura 2: Modelo base BV-BRC.*

- **Genome (Genoma)**

Un **genoma** corresponde al conjunto completo de secuencias de ADN y sus respectivas anotaciones, que representan la mejor estimación del material genético de un organismo. El BV-BRC provee múltiples herramientas para acceder a la información genómica mediante su *CLI*, entre las que se destacan:

- *p3-all-genomes*: lista todos los genomas disponibles o un subconjunto filtrado.
- *p3-get-genome-data*: obtiene metadatos de genomas específicos.
- *p3-get-genome-features*: accede a las características (genes, regiones o elementos funcionales) asociadas a cada genoma.

- *p3-get-genome-contigs*: recupera las secuencias de ADN correspondientes a cada genoma.
- *p3-get-genome-drugs*: extrae información relacionada con la resistencia antimicrobiana vinculada a los genomas.

En los archivos exportados, las columnas asociadas a esta entidad utilizan el prefijo **genome.**

**Ejemplo:** *genome.genome_name* → nombre del genoma bacteriano.

- **Contig**

Un **contig** es una secuencia continua de ADN dentro de un genoma. Puede representar un **cromosoma completo**, un **plásmido** o un **fragmento parcial de ADN** ensamblado. Los contigs se obtienen a partir de los identificadores de genoma mediante el comando:

- *p3-get-genome-contigs*

Las columnas asociadas a esta entidad utilizan el prefijo **contig.**

**Ejemplo**: *contig.length* → longitud del contig en pares de bases.

- **Drug (Fármaco o agente antimicrobiano)**

La entidad **Drug** almacena la información relacionada con los **fármacos antimicrobianos** utilizados en tratamientos y pruebas de resistencia. Constituye la base de los datos de resistencia antimicrobiana en BV-BRC y permite vincular cada fármaco con los genomas en los que se ha identificado resistencia o sensibilidad, los comandos utilizados para obtener la información:

- *p3-all-drugs*: lista todos los fármacos registrados.
- *p3-get-drug-genomes*: obtiene datos de resistencia a dichos fármacos en distintos genomas.

Las columnas correspondientes emplean el prefijo **drug.**

**Ejemplo:** *drug.molecular_formula* → fórmula molecular del compuesto.

- **Feature (Característica o elemento genómico)**

Una **feature** representa una región funcional de interés dentro del genoma, que puede corresponder a:

- Un **gen** (zona que codifica una proteína),
- Un **sitio de ARN**,
- Una **secuencia CRISPR**, o
- Una **región reguladora**.

Cada feature pertenece exclusivamente a un único genoma, aunque puede estar

distribuida en varios contigs, algunos de los comandos que se pueden usar son:

- *p3-get-genome-features*: obtiene las características de uno o varios genomas.
- *p3-get-family-features*: lista las features asociadas a familias de proteínas.
- *p3-get-feature-data*: recupera información detallada de una feature específica.

El identificador único de cada feature es *patric_id*, y las columnas de salida emplean el prefijo **feature**.

**Ejemplo:** *feature.location* → ubicación de la feature dentro del genoma.

- **Family (Familia de proteínas)**

Una **familia** agrupa proteínas o features que se consideran **homólogas isofuncionales**, es decir, que comparten similitud estructural y funcional.
Este nivel de información permite estudiar relaciones evolutivas y patrones de funcionalidad conservados entre diferentes especies bacterianas, algunos comandos para obtener la información son:

- *p3-get-family-features*: obtiene las features pertenecientes a una o varias familias.
- *p3-get-family-data*: recupera la información general de dichas familias.

Las columnas correspondientes a esta entidad utilizan el prefijo **family.**

**Ejemplo:** *family.product* → producto o función asociada a la familia proteica.

### 5.2.2 LIBRERÍAS DE PYTHON

El entorno incluye librerías de uso general y especializado para el análisis, modelado y visualización de datos, en la Tabla1 se describen las principales librerías utilizadas y su función dentro del flujo de trabajo.

| LIBRERÍA | FUNCIÓN PRINCIPAL |
|---|---|
| **PANDAS** | Manipulación y limpieza de datos tabulares (.csv, .tsv, .xlsx). |
| **NUMPY** | Cálculo numérico, manejo de matrices y arrays multidimensionales. |
| **SCIPY** | Aplicación de pruebas estadísticas y métricas matemáticas. |
| **SCIKIT-LEARN** | Implementación de modelos supervisados (Regresión Logística, *Random Forest, XGBoost*) y no supervisados (*K-Means, PCA*). |
| **XGBOOST** | Algoritmo optimizado de *boosting* para clasificación de resistencia antibiótica. |
| **IMBALANCED-LEARN** | Rebalanceo de clases mediante técnicas como *SMOTE* y *RandomUnderSampler.* |
| **MATPLOTLIB / SEABORN** | Visualización de datos y resultados (gráficos, matrices de confusión, distribuciones). |
| **OPENPYXL** | Lectura de archivos Excel (.xlsx) y conversión a estructuras pandas. |
| **JOBLIB** | Serialización y almacenamiento de modelos entrenados. |
| **TQDM** | Seguimiento del progreso de tareas en bucles (entrenamiento). |
| **PYTHON-DOTENV** | Manejo seguro de variables de entorno (rutas, credenciales). |
| **PYARROW / FASTPARQUET** | Almacenamiento eficiente en formato *Parquet* para datasets procesados. |

*Tabla 1: Descripción librerías Python.*

## 5.3 GESTIÓN DEL CONTROL DE VERSIONES

Con el fin de garantizar una adecuada gestión del código, la trazabilidad de los cambios y la colaboración sincrónica del equipo de trabajo se procedió a la creación de un repositorio en la plataforma GitHub bajo el nombre:
PROYECTO_GRADO_MAESTRIA_DATOS_2025
**https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025**
**Este repositorio constituye el espacio centralizado donde se almacenan y versionan los datos, scripts, notebooks, experimentos y documentación relacionados con el proyecto titulado:**
"Modelo predictivo de resistencia antibiótica en bacterias bucales mediante análisis de marcadores genómicos de patogenicidad.".

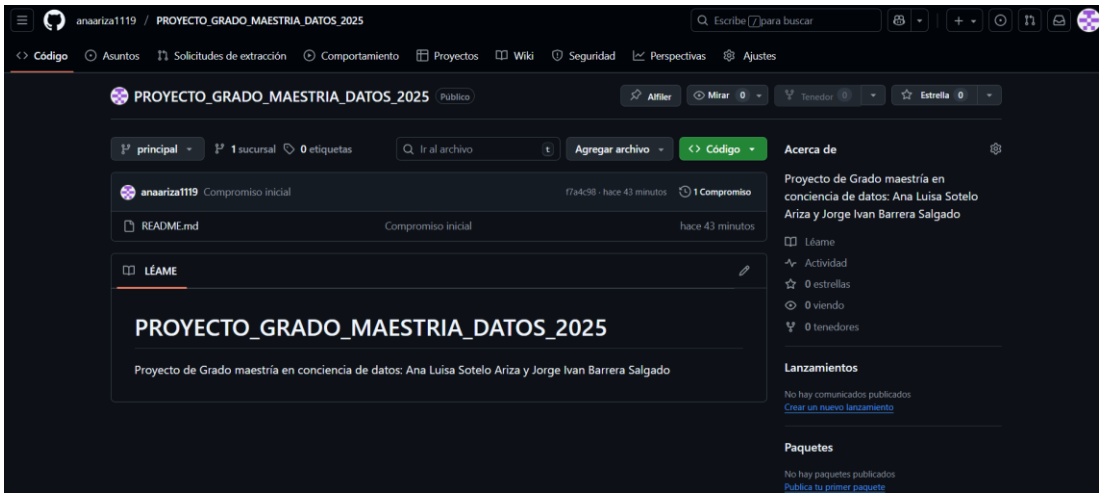


*Figura 3: Creación repositorio.*

### 5.3.1 ORGANIZACIÓN

La organización del repositorio responde a las buenas prácticas de ingeniería de software y ciencia de datos, lo cual permite una adecuada separación entre datos, código, reportes y resultados experimentales.

- **data/:** Contiene los conjuntos de datos en sus diferentes estados: crudos **(raw)**, procesados y fuentes externas suministradas por el director de proyecto (external).
- **notebooks/:** Almacena los cuadernos de análisis exploratorio-utilizados durante las fases iniciales de experimentación.
- **src/:** Concentra el código fuente del proyecto, organizado en subcarpetas para el preprocesamiento de datos, la implementación de modelos, la evaluación de resultados, la generación de visualizaciones y utilidades de apoyo.

- **experiments/:** Incluye los resultados de los experimentos realizados, así como los registros de ejecución, métricas de validación y modelos entrenados guardados.
- **reports/:** Concentra la documentación formal, incluyendo figuras, gráficos y el informe final de la investigación.
- **tests/:** Reservado para la ejecución de pruebas de código que respalden la calidad y consistencia del desarrollo.
- **Archivos base:** En la raíz del repositorio se encuentran documentos fundamentales como README.md, .gitignore, requirements.txt y environment.yml, que describen el proyecto, definen exclusiones de control de versiones y especifican las dependencias necesarias para la reproducción de los experimentos.

# 6 OBTENCIÓN Y PREPARACIÓN DE LOS DATOS

## 6.1 OBJETIVO DEL CAPITULO

Desarrollar un conjunto de datos de entrenamiento compuesto por bacterias bucales y sus respectivos marcadores genómicos de patogenicidad y resistencia, integrando información proveniente de bases bioinformáticas y de muestras clínicas humanas.

### 6.1.1 DESCRIPCIÓN

La fase de obtención y preparación de los datos constituye el punto de partida del proceso analítico, pues permite construir un conjunto de datos estructurado, confiable y trazable que servirá como base para los modelos de aprendizaje automático.

En este proyecto, todas las fuentes de información fueron obtenidas directamente del repositorio *Bacterial and Viral Bioinformatics Resource Center (BV-BRC)* mediante su interfaz de línea de comandos (*Command Line Interface – CLI)*, conocida como `p3-scripts`.

Cada consulta permitió recuperar información funcional y fenotípica de bacterias aisladas de muestras humanas del microbioma bucal (*host_name = Homo sapiens*). A continuación, se describen las cuatro fuentes de datos utilizadas, el proceso de limpieza, normalización e integración, y la estructura final del *dataset* consolidado.

## 6.2 OBTENCIÓN DE DATOS

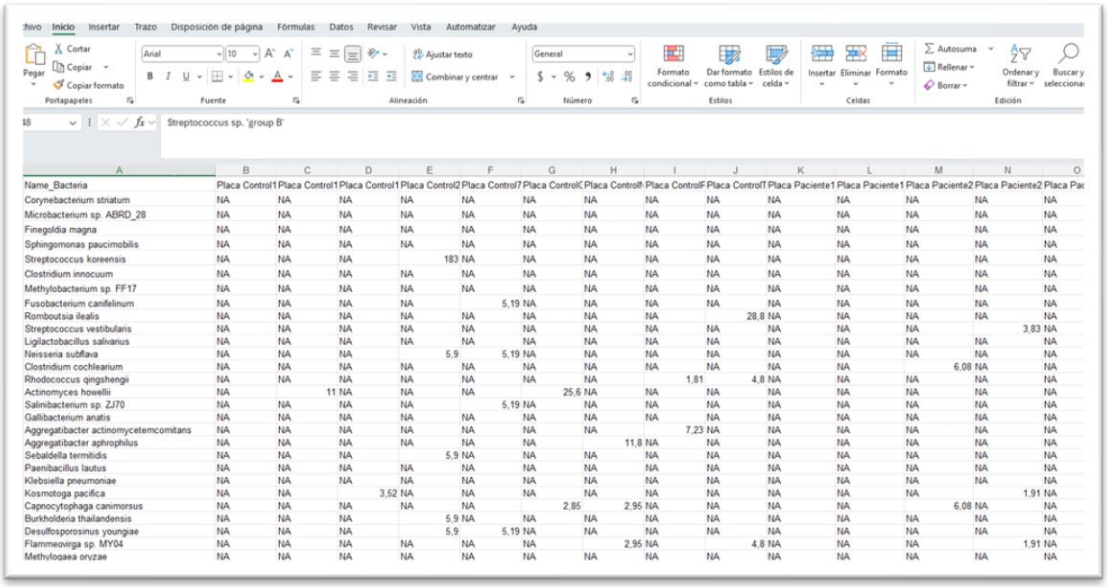
### 6.2.1 ARCHIVO: 001_DATOS_MUESTRAS_BACTERIANAS.xlsx

#### 6.2.1.1 DESCRIPCIÓN

Este archivo constituye la base inicial del estudio y contiene los nombres de las bacterias identificadas en las muestras biológicas humanas, clasificadas por tipo de muestra.
Las muestras incluyen **placa dental**, **saliva** y **tejido tumoral oral**, diferenciadas entre grupos **control** y **paciente**

Este archivo se encuentra alojado en el repositorio:

https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025/tree/main/data/external



| Name_Bacteria | Placa Control1 | Placa Control1 | Placa Control1 | Placa Control2 | Placa Control7 | Placa Control( | Placa Control( | Placa Control( | Placa Control( | Placa Control7 | Placa Paciente1 | Placa Paciente1 | Placa Paciente2 | Placa Paciente2 | Placa Pac |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Corynebacterium striatum | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Microbacterium sp. ABRD_28 | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Finegoldia magna | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Sphingomonas paucimobilis | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Streptococcus koreensis | NA | NA | NA | | 183 | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Clostridium innocuum | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Methylobacterium sp. FF17 | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Fusobacterium canifelinum | NA | NA | NA | NA | | 5,19 | NA | NA | NA | NA | NA | NA | NA | NA | |
| Romboutsia ilealis | NA | NA | NA | NA | NA | NA | NA | NA | NA | 28,8 | NA | NA | NA | NA | |
| Streptococcus vestibularis | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | | 3,83 | NA |
| Ligilactobacillus salivarius | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Neisseria subflava | NA | NA | NA | | 5,9 | 5,19 | NA | NA | NA | NA | NA | NA | NA | NA | |
| Clostridium cochlearium | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | | 6,08 | NA | |
| Rhodococcus qingshengii | NA | NA | NA | NA | NA | NA | NA | | 1,81 | 4,8 | NA | NA | NA | NA | |
| Actinomyces howellii | NA | | 11 | NA | NA | | 25,6 | NA | NA | NA | NA | NA | NA | NA | |
| Salinibacterium sp. ZJ70 | NA | NA | NA | NA | | 5,19 | NA | NA | NA | NA | NA | NA | NA | NA | |
| Gallibacterium anatis | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Aggregatibacter actinomycetemcomitans | NA | NA | NA | NA | NA | NA | NA | | 7,23 | NA | NA | NA | NA | NA | |
| Aggregatibacter aphrophilus | NA | NA | NA | NA | NA | NA | | 11,8 | NA | NA | NA | NA | NA | NA | |
| Sebaldella termitidis | NA | NA | NA | | 5,9 | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Paenibacillus lautus | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Klebsiella pneumoniae | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Kosmotoga pacifica | NA | NA | | 3,52 | NA | NA | NA | NA | NA | NA | NA | NA | | 1,91 | NA |
| Capnocytophaga canimorsus | NA | NA | NA | NA | NA | | 2,85 | 2,95 | NA | NA | NA | NA | 6,08 | NA | |
| Burkholderia thailandensis | NA | NA | NA | | 5,9 | NA | NA | NA | NA | NA | NA | NA | NA | NA | |
| Desulfosporosinus youngiae | NA | NA | NA | | 5,9 | 5,19 | NA | NA | NA | NA | NA | NA | NA | NA | |
| Flammeovirga sp. MY04 | NA | NA | NA | NA | NA | NA | | 2,95 | NA | 4,8 | NA | NA | NA | 1,91 | NA |
| Methylogaea oryzae | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | |

*Figura 4: Archivo 001_DATOS_MUESTRAS_BACTERIANAS.*

#### 6.2.1.2 ORIGEN

Datos clínicos obtenidos de pacientes humanos. Este archivo fue proporcionado como insumo inicial para realizar las búsquedas correspondientes en BV-BRC.

#### 6.2.1.3 COLUMNAS

| COLUMNA | DESCRIPCIÓN |
|---|---|
| **NAME_BACTERIAS** | Nombre científico de la bacteria aislada. |
| **CP** | *Control Placa*: muestra de placa dental del grupo control. |
| **MP** | *Muestra Paciente Placa*: muestra de placa dental de pacientes. |
| **CS** | *Control Saliva*: muestra de saliva del grupo control. |
| **MS** | *Muestra Paciente Saliva*: muestra de saliva de pacientes. |
| **MT** | *Muestra Tumoral*: tejido tumoral oral de pacientes. |

*Tabla 2: Variables archivo 001_DATOS_MUESTRAS_BACTERIANAS.*

#### 6.2.1.4 PROPÓSITO

Establecer la lista base de especies bacterianas presentes en diferentes tipos de muestras para su posterior búsqueda e identificación genómica en BV-BRC.

### 6.2.2 ARCHIVO: 002_DATOS_ID_BACTERIANOS.tsv

#### 6.2.2.1 DESCRIPCIÓN

Este archivo está conformado por los ID de cada bacteria, con sus nombres completos, se filtra por medio del parámetro ***'hostname': 'Humanos/Homo sapiens'.***

Este archivo se encuentra alojado en el repositorio:

https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025/tree/main/data/raw

#### 6.2.2.2 ORIGEN

El origen del archivo corresponde a la tabla **`p3-all-genomes`** de la base de datos BV-BRC, en donde se encuentran los ID de las bacterias, los comandos usados para su extracción son:

```
p3-all-genomes --eq genome_name,Corynebacterium striatum |
p3-get-genome-data  --eq host_name,"Homo sapiens"
 --attr genome_name --attr host_name >> 002_DATOS_ID_BACTERIANOS.tsv
```

*Figura 5: Comando 002_DATOS_ID_BACTERIANOS.*

002_DATOS_ID_BACTERIANOS.tsv

| | genome.genome_id | genome.genome_name | genome.host_name |
|---|---|---|---|
| 1 | | | |
| 2 | 82380.59 | Microbacterium oxydans strain VHA | Homo sapiens |
| 3 | 82380.60 | Microbacterium oxydans strain VHA | Homo sapiens |
| 4 | 82380.61 | Microbacterium oxydans strain VHA | Homo sapiens |
| 5 | 82380.63 | Microbacterium oxydans VHA | Homo sapiens |
| 6 | 2029109.3 | Microbacterium sp. UMB0228 | Human, Homo sapiens |
| 7 | 2916212.3 | Microbacterium sp. p3-SID336 | Homo sapiens |
| 8 | 2916213.3 | Microbacterium sp. p3-SID337 | Homo sapiens |
| 9 | 2916214.4 | Microbacterium sp. p3-SID338 | Homo sapiens |

*Figura 6: Datos 002_DATOS_ID_BACTERIANOS.*

#### 6.2.2.3 COLUMNAS

| COLUMNA | DESCRIPCIÓN |
|---|---|
| **GENOME_ID** | ID del genoma. |
| **GENOME_NAME** | Nombre de la bacteria |
| **HOST_NAME** | Especie huésped de donde se aisló la bacteria. |

*Tabla 3: Variables archivo 002_DATOS_ID_BACTERIANOS.*

#### 6.2.2.4 PROPÓSITO

El propósito de este *dataset* es Identificar los ID de las bacterias que se encuentran en las muestras bucales.

### 6.2.3 ARCHIVO: 003_DATOS_FAMILIAS.tsv

#### 6.2.3.1 DESCRIPCIÓN

Este archivo contiene las propiedades taxonómicas de cada bacteria como la familia, especie y género.

Este archivo se encuentra alojado en el repositorio:

https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025/tree/main/data/raw

#### 6.2.3.2 ORIGEN

El origen del archivo corresponde a la tabla **`p3-get-genome-data`** de la base de datos BV-BRC, contiene las familias de bacterianas filtradas por el genoma id, los comandos usados para la extracción son los siguientes:

```
p3-echo -t genome_id    43770.574   |p3-get-genome-data --attr genome_id --attr genome_name --attr
genus --attr species --attr taxon_id --attr taxon_lineage_names --attr taxon_lineage_ranks --attr
family --eq host_name,"Homo sapiens" >>  003_DATOS_FAMILIAS.tsv
```

*Figura 7: Comando 003_DATOS_GENOMICOS_ID.*

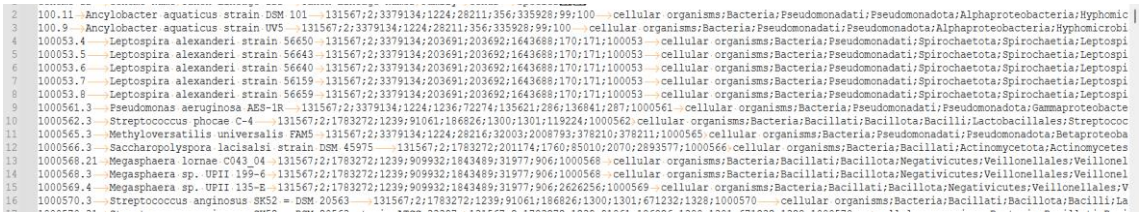


*Figura 8: Datos 003_DATOS_GENOMICOS_ID.*

#### 6.2.3.3 COLUMNAS

| COLUMNA | DESCRIPCIÓN |
|---|---|
| **GENOME ID** | Identificador único del genoma en BV-BRC. |
| **GENOME NAME** | Nombre del genoma o cepa. |
| **TAXON LINEAGE IDS** | Identificadores numéricos de cada nivel taxonómico. |
| **TAXON LINEAGE NAMES** | Nombres jerárquicos de la clasificación taxonómica. |
| **FAMILY** | Familia taxonómica del organismo. |
| **GENUS** | Género al que pertenece el organismo. |
| **SPECIES** | Especie del organismo. |

*Tabla 4: Variables archivo 003_DATOS_GENOMICOS_ID.*

#### 6.2.3.4 PROPÓSITO

El propósito de este *dataset* es relacionar las familias bacterianas y agruparlas.

### 6.2.4 ARCHIVO: 004_DATOS_MEDICAMENTO_ID.tsv

#### 6.2.4.1 DESCRIPCION

Este archivo contiene los antibióticos y las resistencias de las bacterias.

Este archivo se encuentra alojado en el repositorio:

https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025/tree/main/data/raw

#### 6.2.4.2 ORIGEN

El origen de la base corresponde a la tabla **`p3-get-genome-drugs`** de la base BV_BRC**,** el comando usado para su extracción es el siguiente:

```
p3-echo -t genome_id 1318.969 | p3-get-genome-drugs --attr genome_id --attr antibiotic --attr resistant_phenotype >> resistencia.tsv
```

*Figura 9: Comando 004_DATOS_MEDICAMENTO_ID*

| Taxon ID | Genome ID | Genome Name | Antibiotic | Resistant Phenotype | Measurement | Measurement Sign | Measurement Value | Measurement Unit | Laboratory Typing Method |
|---|---|---|---|---|---|---|---|---|---|
| 28901 | 28901.3228 | Salmonella enterica strain N55377 | ampicillin | 32.0 | 32.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |
| 590 | 590.13329 | Salmonella enterica SRR3057226 | amoxicillin/clavulanic acid | 1.0 | 1.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |
| 1773 | 1733.5277 | Mycobacterium tuberculosis 14.0609388 | amikacin | 0.25 | 0.25 | | mg/L | MIC | XGBoost Model 202503101.0 W1 sc |
| 562 | 562.99686 | Escherichia coli 395c9a32-7bb9-11e9-a8d3-68b59976a384 | amikacin | 2.0 | 2.0 | | mg/L | MIC | XGBoost Model 202503101.0 W |
| 287 | 287.34586 | Pseudomonas aeruginosa F3520 | amikacin | 8.0 | 8.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: 0.70, CI[0.62, |
| 562 | 562.68593 | Escherichia coli strain AB4-2 | ampicillin/sulbactam | 16.0 | 16.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 sc |
| 573 | 573.33434 | Klebsiella pneumoniae strain NR5091 | ampicillin | 32.0 | 32.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: 0.9 |
| 562 | 562.13854 | Escherichia coli strain 505077 | ampicillin/sulbactam | 16.0 | 16.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 sc |
| 562 | 562.111149 | Escherichia coli 54823 | amoxicillin/clavulanic acid | 4.0 | 4.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: 0.97, C |
| 28901 | 28901.5905 | Salmonella enterica strain FSIS11814817 | azithromycin | 4.0 | 4.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |
| 1313 | 1313.27064 | Streptococcus pneumoniae strain 14913_3#2 | azithromycin | 0.25 | 0.25 | | mg/L | MIC | XGBoost Model 202503101 |
| 59201 | 59201.417 | Salmonella enterica subsp. enterica 17-7739 | ampicillin | 4.0 | 4.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |
| 573 | 573.67141 | Klebsiella pneumoniae KP_NORM_URN_2013_95505 | aztreonam | 8.0 | 8.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |
| 562 | 562.76321 | Escherichia coli strain BfR-EC-17652 | aztreonam | 16.0 | 16.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |
| 470 | 470.20062 | Acinetobacter baumannii Acb_18 | ampicillin/sulbactam | 16.0 | 16.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 sc |
| 470 | 470.7748 | Acinetobacter baumannii strain MRSN7310 | ampicillin | 32.0 | 32.0 | | mg/L | MIC | XGBoost Model 202503101.0 W1 score: |

*Figura 10: Datos 004_DATOS_MEDICAMENTO_ID.*

#### 6.2.4.3 COLUMNAS

| COLUMNA | DESCRIPCIÓN |
|---|---|
| **TAXON ID** | Identificador taxonómico del organismo. |
| **GENOME ID** | Identificador único del genoma en BV-BRC. |
| **GENOME NAME** | Nombre del genoma o cepa. |
| **ANTIBIOTIC** | Antibiótico evaluado. |
| **RESISTANT PHENOTYPE** | Resultado de resistencia o susceptibilidad. |
| **MEASUREMENT** | Tipo de medición realizada. |
| **MEASUREMENT SIGN** | Signo que acompaña la medición (>, <, =). |
| **MEASUREMENT VALUE** | Valor numérico de la medición. |
| **MEASUREMENT UNIT** | Unidad del valor medido (µg/mL, mm). |
| **LABORATORY TYPING METHOD** | Método de laboratorio usado. |
| **LABORATORY TYPING METHOD VERSION** | Versión del método de laboratorio. |
| **LABORATORY TYPING PLATFORM** | Plataforma o equipo empleado. |
| **VENDOR** | Proveedor del kit o método. |
| **TESTING STANDARD** | Norma usada (CLSI, EUCAST, etc.). |
| **TESTING STANDARD YEAR** | Año de la norma aplicada. |
| **COMPUTATIONAL METHOD** | Método computacional de análisis. |
| **COMPUTATIONAL METHOD VERSION** | Versión del método computacional. |
| **COMPUTATIONAL METHOD PERFORMANCE** | Desempeño del método computacional. |
| **EVIDENCE** | Tipo de evidencia (experimental o computacional). |
| **SOURCE** | Fuente del dato o base de origen. |

*Tabla 5: Variables archivo 004_DATOS_MEDICAMENTO_ID.*

#### 6.2.4.4 PROPÓSITO

El propósito de este *dataset* es poder identificar los antibióticos que son efectivos contra cada bacteria para encontrar tendencias en sus características.

## 6.3 PREPARACIÓN Y LIMPIEZA DE DATOS

La limpieza y preparación de los datos se realizaron utilizando Python (versión 3.10) y las librerías `pandas, numpy y openpyxl`, dentro de un entorno Anaconda.
Las principales tareas ejecutadas fueron:

### 6.3.1 NORMALIZACIÓN DE COLUMNAS

Se estandarizan los nombres de columnas y el contenido textual para garantizar la compatibilidad entre las distintas fuentes de datos.
Se aplicó la función `normalizar_columnas(df)` para:

- Eliminar espacios en blanco en los encabezados.
- Convertir todos los nombres a minúsculas.
- Reemplazar espacios por guiones bajos (_).
- Eliminar caracteres especiales con expresiones regulares ([^a-z0-9_]).

```python
def normalizar_columnas(df: pd.DataFrame) → pd.DataFrame:
    df = df.copy()
    df.columns = (
        df.columns
          .str.strip()
          .str.lower()
          .str.replace(" ", "_")
          .str.replace("[^a-z0-9_]", "", regex=True)
    )
    return df
```

*Figura 11: Código normalización columnas.*

Posteriormente se ejecutó la función limpiar_textos_y_nulos(df) para normalizar el contenido de cada celda:

- Remover espacios iniciales/finales.
- Convertir a minúsculas.
- Reemplazar valores vacíos, “NA”, “N/A” o guiones por NaN.
- Sustituir todos los valores nulos (NaN) por 0, tanto en variables numéricas como de texto.
- Eliminar valores duplicados.
- En el caso particular de la fuente 001_DATOS_MUESTRAS_BACTERIANAS.xlsx, los valores negativos fueron convertidos a cero.
  Esta decisión se fundamenta en que las variables presentes en este conjunto de datos representan magnitudes biológicas o experimentales (por ejemplo, conteos

de bacterias, concentraciones o identificadores numéricos), las cuales no pueden asumir valores negativos en un contexto real.

- Por tanto, cualquier valor negativo identificado se considera un error de registro, digitación o transformación durante la captura de los datos. Sustituirlos por cero garantiza la coherencia biológica de la información y evita distorsiones en los análisis estadísticos posteriores, ya que el valor cero refleja de forma más adecuada la ausencia o no detección de un valor medible, en lugar de representar una magnitud inválida.

```python
def limpiar_textos_y_nulos(df: pd.DataFrame) -> pd.DataFrame:
    """
    Limpieza general:
      - Strings: strip/lower; vacíos comunes → NaN.
      - Elimina filas totalmente vacías y duplicadas.S
      - Numéricas: NaN → 0; negativos → 0.
      - Texto: NaN → "0" y, si el texto representa un número negativo, → "0".
    """
    df = df.copy()
    vacios = {"", " ", "-", "na", "n/a", "NA", "N/A"}

    # ── 1) Limpieza de texto base ──
    for col in df.columns:
        if df[col].dtype == "object":
            df[col] = df[col].apply(lambda x: x.strip().lower() if isinstance(x, str) else x)
            df[col] = df[col].apply(lambda x: np.nan if isinstance(x, str) and x in vacios else x)

    # ── 2) Quitar filas vacías y duplicadas ──
    df = df.dropna(how="all").drop_duplicates()

    # ── 3) Numéricas reales: NaN → 0; negativos → 0 ──
    num_cols = df.select_dtypes(include=[np.number]).columns
    if len(num_cols):
        df[num_cols] = df[num_cols].fillna(0)
        df[num_cols] = df[num_cols].applymap(lambda x: 0 if x < 0 else x)

    # ── 4) Objetos que contienen números negativos en texto ──
    # patrón: acepta guion normal o unicode (-) y decimales con punto o coma
    neg_pattern = re.compile(r"^\s*[--]\s*\d+(?:[.,]\d+)?\s*$")

    obj_cols = df.select_dtypes(include=["object"]).columns
    for col in obj_cols:
        # a) reemplazar NaN por "0"
        df[col] = df[col].fillna("0")

        # b) detectar strings que representan números negativos
        mask_neg_text = df[col].astype(str).apply(lambda s: bool(neg_pattern.match(s)))

        # c) para esas filas, poner "0"
        if mask_neg_text.any():
            df.loc[mask_neg_text, col] = "0"

        # d) adicional: intentar coercer a número para detectar negativos ocultos tipo " -3,5 "
        tmp = (
            df[col]
            .astype(str)
            .str.replace("-", "-", regex=False)   # guion unicode → normal
            .str.replace(",", ".", regex=False)   # coma decimal → punto
        )
        tmp_num = pd.to_numeric(tmp, errors="coerce")
        mask_neg_num = tmp_num < 0
        if mask_neg_num.any():
            df.loc[mask_neg_num, col] = "0"

    return df
```

*Figura 12: Código normalización celdas.*

En la siguiente figura se observa el resultado de la normalización de columnas y celdas aplicada a las cinco fuentes de datos (001, 002, 003, 004).

En el encabezado de cada *DataFrame* se evidencia la eliminación de espacios y caracteres especiales, la conversión a minúsculas y la sustitución de espacios por guiones bajos, esta estandarización garantiza que todas las tablas mantengan una estructura coherente y compatible para los procesos de integración posteriores.
Además, durante la limpieza del contenido se reemplazaron los valores nulos y los campos vacíos por **0**, lo que evita errores en los cruces o análisis numéricos.

```
Head 001:
            name_bacteria  10cp  15cp  ... 3mt  5mt     nombre_normalizado
0  [clostridium] hylemonae  1.78  1.83  ...   0    0  Clostridium hylemonae
1   [clostridium] innocuum     0     0  ...   0  111   Clostridium innocuum
2    abiotrophia defectiva     0  25.7  ...   0    0  Abiotrophia defectiva

[3 rows x 48 columns]

Head 002 (muestra en memoria):
  genomegenome_id                genomegenome_name genomehost_name
0        82380.59  microbacterium oxydans strain vha    homo sapiens
1        82380.60  microbacterium oxydans strain vha    homo sapiens
2        82380.61  microbacterium oxydans strain vha    homo sapiens

Head 003 (muestra):
    genome_id  ... featurefamily_id
0  43770.1307  ...              0.0
1  43770.1307  ...              0.0
2  43770.1307  ...              0.0

[3 rows x 4 columns]

Head 004 (muestra):
    genome_id  ... genome_drugresistant_phenotype
0  43770.1401  ...                              0
1  43770.1401  ...                              0
2  43770.1401  ...                              0

[3 rows x 4 columns]

Head 005 (muestra):
  genomegenome_id  ...      genomehost_name
0        1318.969  ...  human, homo sapiens
1        1328.227  ...  human, homo sapiens
2        1334.234  ...  human, homo sapiens

[3 rows x 3 columns]
```

*Figura 13: Resultado estabilización columnas y celdas.*

También se aplica la función `limpiar_textos_y_nulos()`, para realizar la depuración de duplicados la cual permitió reducir significativamente el tamaño de las fuentes de datos, eliminando registros repetidos que podían generar inconsistencias en los cruces.

Por ejemplo, el archivo **002_DATOS_ID_BACTERIANOS.tsv** pasó de **1.302.275 registros iniciales** a **543.716 registros únicos**, lo que representa una reducción del **58%**, según los logs de ejecución.

```
Backup → C:\Users\anaso\OneDrive\Documentos\PROYECTO_GRADO_MAESTRIA_DATOS_2025\data\interim\001_normalizado.csv (275, 48)
Backup → C:\Users\anaso\OneDrive\Documentos\PROYECTO_GRADO_MAESTRIA_DATOS_2025\data\interim\002_normalizado.csv | filas in=1302275 → out=543716 | dups removidos=758559
Backup → C:\Users\anaso\OneDrive\Documentos\PROYECTO_GRADO_MAESTRIA_DATOS_2025\data\interim\003_normalizado.csv | filas in=5111836 → out=5111836 | dups removidos=0
Backup → C:\Users\anaso\OneDrive\Documentos\PROYECTO_GRADO_MAESTRIA_DATOS_2025\data\interim\004_normalizado.csv | filas in=23750 → out=22599 | dups removidos=1151
Backup → C:\Users\anaso\OneDrive\Documentos\PROYECTO_GRADO_MAESTRIA_DATOS_2025\data\interim\005_normalizado.csv | filas in=818056 → out=394911 | dups removidos=423145
```

*Figura 14: Log eliminar duplicados.*

### 6.3.2 ESTANDARIZACIÓN DE NOMBRES CIENTÍFICOS

La estandarización de los nombres científicos constituyó una de las etapas más importantes dentro del proceso de preparación y limpieza de los datos, ya que de ella dependía la correcta vinculación entre los registros bacterianos del archivo 001 y los identificadores genómicos del archivo 002.

En las bases de datos biológicas, es común encontrar variaciones en la escritura de los nombres taxonómicos debido a diferencias en las fuentes, errores de digitación o convenciones propias de cada laboratorio. Ejemplos frecuentes incluyen el uso de abreviaturas *(sp., spp., strain),* códigos de colección *(ATCC, DSM, NCTC),* tildes, mayúsculas o corchetes para indicar reclasificaciones. Estas inconsistencias pueden provocar que un mismo organismo sea tratado como diferentes entidades al realizar cruces o análisis comparativos, generando falsos negativos y pérdida de información biológicamente relevante [27],[28],[29].

Para evitarlo, se desarrolló la función `limpiar_nombre_cientifico()`, la cual normaliza todos los nombres bacterianos al formato "Género especie", siguiendo las reglas del sistema binomial de nomenclatura establecido por *Linnaeus* [30]. Este formato asegura que:

- El género se escriba con inicial mayúscula.
- La especie se escriba en minúscula.
- Se eliminen tildes, corchetes, paréntesis y cualquier caracter especial.
- Se supriman abreviaturas taxonómicas no válidas como *sp., spp., strain, ATCC, DSM,* entre otras.

```python
def limpiar_nombre_cientifico(nombre: str) → str | None:
    """
    Estandariza a 'Género especie':
      - Quita tildes, corchetes, paréntesis, comillas y puntos.
      - Elimina 'sp', 'spp' y metadatos de cepa (strain, ATCC, DSM, etc.).
      - NO fragmenta tokens con '_' o '-' (se descartan completos si no son alfabéticos).
      - Evita tomar como 'especie' términos de sitio/aislado comunes (eye, skin, soil, etc.).
    """
    import re, unicodedata

    if not isinstance(nombre, str) or not nombre.strip():
        return None

    # 1) Normalización básica
    n = ''.join(c for c in unicodedata.normalize('NFD', nombre) if unicodedata.category(c) ≠ 'Mn')
    n = re.sub(r'[\[\]\(\)\'\"{}\.]', '', n)

    # 2) Cortar metadatos de cepa/colecciones y lo que siga
    n =
re.sub(r'\b(strain|type|isolate|atcc|dsm|nctc|ccug|jcm|kctc|nbrc|cbs|subsp|subspecies|umb|mmrc|cw|we
)\b.*',
                '', n, flags=re.IGNORECASE)

    # 3) Quitar sp/spp como palabras completas
    n = re.sub(r'\bsp\.?\b|\bspp\.?\b', '', n, flags=re.IGNORECASE)

    # 4) Espacios prolijos (¡sin reemplazar '_' ni '-'!)
    n = re.sub(r'\s+', ' ', n).strip()

    # 5) Tokenizar por ESPACIO y quedarse solo con tokens 100% alfabéticos
    tokens_orig = n.split()
    tokens_alpha = [t for t in tokens_orig if re.fullmatch(r'[A-Za-z]+', t)]

    if not tokens_alpha:
        return None

    genero = tokens_alpha[0].capitalize()

    # Lista corta de términos no-especie comunes (puedes ampliarla si ves otros)
    STOP_ESPECIE = {
        "eye","skin","soil","water","aquifer","sediment","lake","river","seawater","marine",
        "stool","feces","fecal","urine","blood","saliva","oral","nasal","vaginal","gut",

    return genero
```

*Figura 15: Código limpiar nombre científico.*

El proceso permitió que todas las observaciones siguieran un estándar taxonómico uniforme, indispensable para la correcta identificación y trazabilidad entre las distintas fuentes de información [31].

#### 6.3.2.1 EJEMPLOS DEL PROCESO DE ESTANDARIZACIÓN

| NOMBRE ORIGINAL | NOMBRE ESTANDARIZADO | DESCRIPCIÓN DEL CAMBIO APLICADO |
|---|---|---|
| **[Clostridium] innocuum** | **Clostridium innocuum** | Eliminación de corchetes y mayúscula inicial en género. |
| **Clostridium sp.** | **Clostridium** | Eliminación de abreviatura *sp.* (especie no determinada). |
| **Bacillus subtilis strain 168** | **Bacillus subtilis** | Remoción de “strain 168”. |
| **Actinomyces sp .** | **Actinomyces** | Supresión de punto y espacio adicional. |
| **Lactobacillus spp.** | **Lactobacillus** | Eliminación de *spp.* (múltiples especies). |
| **Escherichia coli ATCC 25922** | **Escherichia coli** | Limpieza de código de colección. |
| **Microcella sp.** | **Microcella** | Abreviatura *sp.* eliminada. |

*Tabla 6: Ejemplos del proceso de estandarización.*

Tras la ejecución del proceso, se verificó la ausencia de sufijos como *sp.* o *spp.* en los campos normalizados, confirmando que todos los registros bacterianos quedaron correctamente estandarizados. Este control se realizó mediante expresiones regulares, obteniéndose 0 ocurrencias de dichos términos, lo que evidencia la eficacia del proceso de limpieza.

```
Head 001:
             name_bacteria  10cp  15cp  ... 3mt  5mt     nombre_normalizado
0  [clostridium] hylemonae  1.78  1.83  ...   0    0  Clostridium hylemonae
1   [clostridium] innocuum     0     0  ...   0  111   Clostridium innocuum
2    abiotrophia defectiva     0  25.7  ...   0    0  Abiotrophia defectiva
```

*Figura 16: Resultado estandarizar nombre científico.*

## 6.4 INTEGRACIÓN DE DATOS

Desarrollar un conjunto de datos de entrenamiento compuesto por bacterias bucales y sus respectivos marcadores genómicos de patogenicidad y resistencia.

Las principales tareas ejecutadas fueron:

### 6.4.1 CONVERSIÓN DE TIPO DE DATOS

La función `_ensure_str()` convierte columnas específicas a tipo *string* para evitar errores de comparación durante las uniones (*merge*).

```
#──────────────────────────────────────────────────
# Paso 1 - Asegurar tipo str
#──────────────────────────────────────────────────
def _ensure_str(df, cols):
    for c in cols:
        if c in df.columns:
            df[c] = df[c].astype(str)
    return df
```

*Figura 17: Comando conversión de tipo de datos*

### 6.4.2 GRÁFICOS DE COINCIDENCIAS

La función `pie_match_vs_nomatch()` calcula el número y porcentaje de coincidencias y no coincidencias entre dos conjuntos de datos y genera un gráfico tipo torta que representa esa proporción.

```python
def pie_match_vs_nomatch(matched_count, base_count, title, out_path=None):
    matched = int(matched_count)
    base = int(base_count) if base_count else 0
    no_match = max(base - matched, 0)
    pct_m = round((matched / base * 100), 2) if base else 0
    pct_nm = round((no_match / base * 100), 2) if base else 0

    sizes = [matched, no_match]
    labels = ['Cruzan', 'No cruzan']
    colors = [PALETTE[0], PALETTE[1]]
    explode = (0.05, 0)

    plt.figure(figsize=(6, 6))
    wedges, texts, autotexts = plt.pie(
        sizes,
        explode=explode,
        labels=None,
        colors=colors,
        startangle=90,
        shadow=True,
        autopct='%1.1f%%',
        pctdistance=0.75,
        wedgeprops={'linewidth': 1, 'edgecolor': 'white'}
    )
    plt.title(title, fontsize=14, fontweight='bold')
    plt.axis('equal')
    plt.legend(
        wedges,
        [f"{lab} ({val/base*100:.1f}%)" for lab, val in zip(labels, sizes) if base > 0],
        loc='upper left',
        bbox_to_anchor=(0.02, 0.98),
        frameon=True,
        fancybox=True,
        shadow=False,
        fontsize=10
    )
    plt.tight_layout()
    if out_path:
        os.makedirs(os.path.dirname(out_path), exist_ok=True)
        plt.savefig(out_path, dpi=300, bbox_inches='tight')
        plt.close()
    else:
        plt.show()

    return {
        "matched": matched,
        "nomatch": no_match,
        "base": base,
        "pct_matched": pct_m,
        "pct_nomatch": pct_nm
    }
```

*Figura 18:Comando gráficos de coincidencias*

#### 6.4.1.1 EXPORTACIÓN DE REPORTES DE UNIÓN

La función `export_join_reports()` realiza las uniones entre tablas, identifica los registros coincidentes y no coincidentes, guarda los resultados en archivos .csv y genera el gráfico correspondiente.

```python
#————————————————————————————————————
# Paso 3 - Unión y reportes
#————————————————————————————————————
def export_join_reports(left_df, right_df, key, left_name, right_name, out_dir, base_is='left'):
    os.makedirs(out_dir, exist_ok=True)

    inner = left_df.merge(right_df, how='inner', on=key, suffixes=(f"_{left_name}",
f"_{right_name}"))
    left_only = left_df[~left_df[key].isin(inner[key])]
    right_only = right_df[~right_df[key].isin(inner[key])]

    if base_is == 'left':
        base_total = left_df[key].nunique(dropna=True)
    elif base_is == 'inner':
        base_total = inner[key].nunique(dropna=True)
    else:
        base_total = left_df[key].nunique(dropna=True)

    matched_total = inner[key].nunique(dropna=True)

    inner_path = os.path.join(out_dir, f"inner_{left_name}_vs_{right_name}.csv")
    left_only_path = os.path.join(out_dir, f"no_{right_name}_en_{left_name}.csv")
    right_only_path = os.path.join(out_dir, f"no_{left_name}_en_{right_name}.csv")

    inner.to_csv(inner_path, index=False, encoding='utf-8')
    left_only.to_csv(left_only_path, index=False, encoding='utf-8')
    right_only.to_csv(right_only_path, index=False, encoding='utf-8')

    pie_path = os.path.join(out_dir, f"pie_{left_name}_vs_{right_name}.png")
    title = f"Relación entre {left_name} y {right_name}"
    pie_stats = pie_match_vs_nomatch(matched_total, base_total, title, pie_path)

    resumen_row = {
        "union": f"{left_name}_vs_{right_name}",
        "clave": key,
        "base": base_is,
        "base_total": base_total,
        "matched": pie_stats["matched"],
        "no_matched": pie_stats["nomatch"],
        "pct_matched": pie_stats["pct_matched"],
        "pct_no_matched": pie_stats["pct_nomatch"],
        "inner_csv": inner_path,
        "left_only_csv": left_only_path,
        "right_only_csv": right_only_path,
        "pie_png": pie_path
    }
    return inner, left_only, right_only, resumen_row
```

*Figura 19: Comando exportación de reportes de unión*

#### 6.4.1.2 INTEGRACIONES PROGRESIVAS

Se realizan tres integraciones consecutivas: (a) entre 001 y 002 por el campo *nombre_normalizado*, (b) entre la unión previa y 003 por *genome_id*, y (c) entre la unión resultante y 004 por el mismo campo.

```python
#-------------------------------------------
# Paso 5 - Integraciones
#-------------------------------------------
df001_unicos = _ensure_str(df001_unicos, ['nombre_normalizado'])
df002 = _ensure_str(df002, ['nombre_normalizado'])

inner_001_002, no_002_en_001, no_001_en_002, row_001_002 = export_join_reports(
    left_df=df001_unicos,
    right_df=df002,
    key='nombre_normalizado',
    left_name='001',
    right_name='002',
    out_dir=os.path.join(ruta_out, "001_vs_002"),
    base_is='left'
)

df001_002 = inner_001_002.rename(columns={
    "genomegenome_id": "genome_id",
    "genomegenome_name": "genome_name"
})
df001_002 = _ensure_str(df001_002, ['genome_id'])
df003 = _ensure_str(df003, ['genome_id'])

inner_001002_003, no_003_en_001002, no_001002_en_003, row_001002_003 = export_join_reports(
    left_df=df001_002[['nombre_normalizado','genome_id','genome_name']].drop_duplicates(),
    right_df=df003[['genome_id']].drop_duplicates(),
    key='genome_id',
    left_name='001_002',
    right_name='003',
    out_dir=os.path.join(ruta_out, "001_002_vs_003"),
    base_is='left'
)

df004 = _ensure_str(df004, ['genome_id'])
inner_0010023_004, no_004_en_0010023, no_0010023_en_004, row_0010023_004 = export_join_reports(
    left_df=inner_001002_003[['genome_id']].drop_duplicates(),
    right_df=df004[['genome_id']].drop_duplicates(),
    key='genome_id',
    left_name='001_002_003',
    right_name='004',
    out_dir=os.path.join(ruta_out, "001_002_003_vs_004"),
    base_is='left'
)
```

*Figura 20:Comando integraciones progresivas.*

### 6.4.1.3 RESUMEN GENERAL DE INTEGRACIONES

Los resultados de cada integración se resumen en un archivo *(resumen_integracion.csv)* que contiene los totales de registros base, coincidencias, porcentajes y rutas de salida generadas.

```
#-----------------------------------------------
# Paso 6 - Resumen general
#-----------------------------------------------
resumen = pd.DataFrame([row_001_002, row_001002_003, row_0010023_004])
resumen_path = os.path.join(ruta_out, "resumen_integracion.csv")
resumen.to_csv(resumen_path, index=False, encoding='utf-8')
print("Resumen guardado en:", resumen_path)
```

*Figura 21:Comando resumen general de integraciones.*

#### 6.4.1.4 RESUMEN IMPRESO DE RESULTADOS

Se imprime en consola el resumen final de las tres integraciones con sus respectivos totales, porcentajes y nombres de archivos exportados.

```
#-----------------------------------------------
# Paso 8 - Mostrar resumen
#-----------------------------------------------
print("\n========== RESUMEN DE INTEGRACIÓN ==========\n")
for fila in [row_001_002, row_001002_003, row_0010023_004]:
    print(f" Unión: {fila['union']}")
    print(f"  - Base de referencia: {fila['base']}")
    print(f"  - Total base: {fila['base_total']:,}")
    print(f"  - Coinciden: {fila['matched']:,} ({fila['pct_matched']}%)")
    print(f"  - No coinciden: {fila['no_matched']:,} ({fila['pct_no_matched']}%)")
    print(f"  - CSV coincidencias: {os.path.basename(fila['inner_csv'])}")
    print(f"  - CSV no coinciden (der): {os.path.basename(fila['right_only_csv'])}")
    print(f"  - Gráfico: {os.path.basename(fila['pie_png'])}")
    print("-" * 60)
```

*Figura 22:Comando resumen impreso de resultados.*

### 6.4.1.5 RESULTADOS DE LA INTEGRACIÓN

El análisis de integración de las bases de datos **001, 002, 003 y 004** permitió evaluar el grado de coincidencia entre las distintas fuentes de información. En la primera comparación (**001_vs_002**) se alcanzó una concordancia del **78,85%**, lo que evidencia una relación inicial consistente entre ambas. Con la incorporación de la base **003**, la coincidencia aumentó a **90,62%**, indicando una mejora en la homogeneidad y compatibilidad de los registros. Sin embargo, al añadir la base **004**, el nivel de concordancia descendió a **82,99%**, posiblemente debido a la falta de estudios clínicos, en la base genómica inicial. En conjunto, estos resultados confirman una integración globalmente estable y confiable, para el entrenamiento del modelo de datos, destacando la preservación de **23** bacterias, **109.831** *Genome_id* y más de **4** millones de registros asociados a las resistencias antibióticas.

Estas figuras se encuentran alojadas en el repositorio:

https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025/tree/main/reports/figures

**Gráficos por cada cruce:**

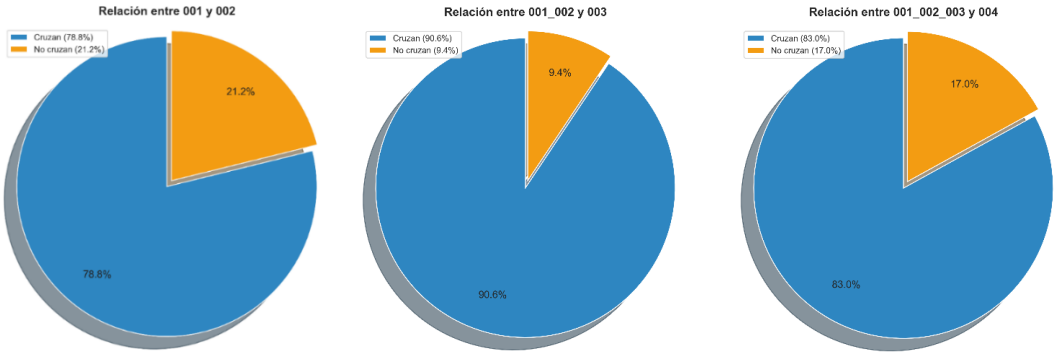



*Figura 23:Graficos pie_001_vs_002_vs_003_vs_004.*

| RELACION | TOTAL BASE | COINCIDEN | % COINCIDEN | NO COINCIDEN | % NO COINCIDEN |
|---|---|---|---|---|---|
| **001_vs_002** | 260 | 205 | 78.85% | 55 | 21.15% |
| **001_002_vs_003** | 146,034 | 132,339 | 90.62% | 13,695 | 9.38% |
| **001_002_003_vs_004** | 132,339 | 109,831 | 82.99% | 22,508 | 17.01% |

*Tabla 7: Resultado integración Objetivo 1.*

Durante la integración se realizaron tres uniones progresivas:

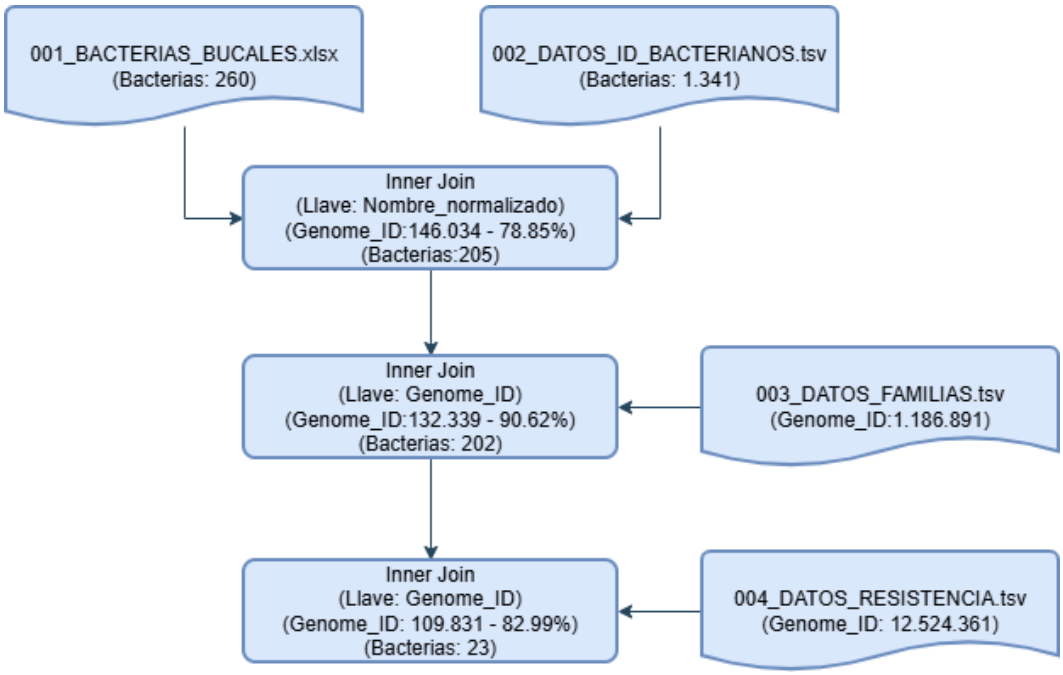



*Figura 24:Diagrama relación de resultados*

1. Entre los archivos 001 y 002, el 78.85 % de las especies (205 de 260) presentaron coincidencias por el campo nombre_normalizado.

2. En la segunda integración, al incorporar el archivo 003, se mantuvieron 132,339 registros coincidentes sobre un total de 146,034, equivalentes al 90.62 %.

3. En la última unión, con el archivo 004, se obtuvieron 109,831 coincidencias de 132,339 registros, lo que corresponde al 82.99 %.

El número de especies únicas se redujo de 260 iniciales a 23 después de las integraciones, como resultado de la eliminación de registros duplicados o sin correspondencia entre tablas.

Como resultado del proceso de integración y comparación entre las distintas bases de datos, se generaron los siguientes archivos, los cuales recopilan la información obtenida en cada cruce.
Estos archivos se encuentran disponibles en el repositorio del proyecto, en la ruta:

**https://github.com/anaariza1119/PROYECTO_GRADO_MAESTRIA_DATOS_2025/tree/main/data/processed**

Los archivos generados son los siguientes:

- **Inner_001_vs_002.csv:** Contiene los registros coincidentes entre las bases 001 y 002.
- **Inner_001_002_vs_003.csv:** Consolida las coincidencias resultantes de la unión entre las bases 001, 002 y 003.
- **Inner_001_002_003_vs_004.csv:** Integra la información de las cuatro bases de datos, mostrando las coincidencias globales.
- **No_001_en_002.csv:** Agrupa los registros presentes en la base 001 que no se encontraron en la base 002.
- **No_001_002_en_003.csv:** Identifica los registros de las bases 001 y 002 que no aparecen en la base 003.
- **No_001_002_003_en_004.csv:** Muestra los registros existentes en las tres primeras bases que no están presentes en la base 004.

# 7 ANÁLISIS EXPLORATORIO DE MUESTRAS CLÍNICAS.

## 7.1 OBJETIVO DEL CAPITULO

El objetivo de este capítulo es realizar un análisis exploratorio integral de las muestras clínicas obtenidas en el estudio, con el fin de identificar patrones de abundancia, distribución y variabilidad microbiana entre distintos tipos de muestras (placa control, placa de pacientes, saliva de controles, saliva de pacientes y tejido tumoral). A partir de los datos previamente normalizados, se implementaron procedimientos estadísticos y visuales que permiten caracterizar la composición bacteriana global, determinar las especies dominantes en cada grupo y explorar posibles similitudes o diferencias entre las condiciones clínicas analizadas.

### 7.1.1 DESCRIPCIÓN

En esta fase mediante el uso de herramientas de análisis en Python (principalmente pandas, seaborn y matplotlib), se generaron resúmenes cuantitativos y gráficos comparativos que describen la distribución de abundancias relativas, la variabilidad intra e intergrupal y la identificación de bacterias potencialmente asociadas con muestras tumorales.
Este proceso incluye la normalización de proporciones, la visualización de las 30 bacterias más representativas mediante mapas de calor, la determinación de los 10 taxones más abundantes por tipo de muestra y la evaluación de la distribución global mediante diagramas de caja (boxplots).

Asimismo, se efectuó una comparación cruzada entre los perfiles tumorales y los de otros grupos, con el fin de detectar bacterias comunes y aquellas exclusivas de los tejidos tumorales, aportando así una primera aproximación exploratoria al comportamiento diferencial del microbioma clínico.

En conjunto, este análisis constituye una fase descriptiva y comparativa preliminar, fundamental para orientar los siguientes capítulos enfocados en la validación estructural y el estudio taxonómico y de resistencia antimicrobiana.

### 7.1.2 AGRUPACIÓN DE MUESTRAS.

El código desarrollado realiza un análisis de las muestras clínicas mediante los grupos de muestras (placa y saliva de controles y pacientes, además de tejido tumoral) a partir de los nombres de las columnas. Posteriormente, se calcula la abundancia total y relativa de cada bacteria en los distintos grupos, generando un resumen comparativo.

```
# ──────────────────────────────────────────────────────
# Paso 2: Identificar grupos de muestras
# ──────────────────────────────────────────────────────
grupos = {
    'CP': [c for c in df.columns if 'control' in c and 'placa' in c],
    'MP': [c for c in df.columns if 'placa_paciente' in c],
    'CS': [c for c in df.columns if 'saliva_control' in c],
    'MS': [c for c in df.columns if 'saliva_paciente' in c],
    'MT': [c for c in df.columns if 'turmor_paciente' in c]
}

for g, cols in grupos.items():
    print(f"{g}: {len(cols)} columnas detectadas")
```

*Figura 25:Comando Identificar Grupos.*

```
CP: 9 columnas detectadas
MP: 10 columnas detectadas
CS: 7 columnas detectadas
MS: 10 columnas detectadas
MT: 10 columnas detectadas
```

*Figura 26:Resultado Identificar Grupos*

### 7.1.3 ABUNDANCIA POR GRUPO.

En este paso se elabora un resumen de abundancia por grupo, sumando los valores de cada tipo de muestra para obtener la distribución total de bacterias. Luego, se aplica una normalización proporcional que convierte los valores absolutos en abundancias relativas, permitiendo comparar la composición microbiana entre los distintos grupos clínicos.

```
# ──────────────────────────────────────────────────────
# Paso 3: Resumen de abundancia por grupo
# ──────────────────────────────────────────────────────
resumen = {}
for grupo, cols in grupos.items():
    resumen[grupo] = df[cols].sum(axis=1)

df_resumen = pd.DataFrame(resumen)
df_resumen["nombre_normalizado"] = df["nombre_normalizado"]

# Normalizar proporciones
df_resumen_prop = df_resumen.set_index("nombre_normalizado")
df_resumen_prop = df_resumen_prop.div(df_resumen_prop.sum(axis=0), axis=1)

plt.figure(figsize=(10, 12))
sns.heatmap(df_resumen_prop.head(30), cmap="YlGnBu", cbar_kws={'label':
'Proporción relativa'})
plt.title("Abundancia relativa de las 30 bacterias principales",
fontsize=14, fontweight='bold')
plt.xlabel("Tipo de muestra", fontsize=12)
plt.ylabel("Bacterias", fontsize=12)
plt.tight_layout()
plt.show()
```

*Figura 27:Comando Abundancia por Grupos.*

El mapa de calor muestra la distribución de abundancia relativa de las bacterias detectadas en los diferentes tipos de muestras (CP, MP, CS, MS y MT). Se observan perfiles microbianos diferenciados entre grupos, con una mayor representación de géneros como *Streptococcus*, Prevotella y *Capnocytophaga* en las muestras de pacientes, especialmente en tejido tumoral, lo que sugiere una posible asociación entre estas bacterias y las condiciones patológicas analizadas.

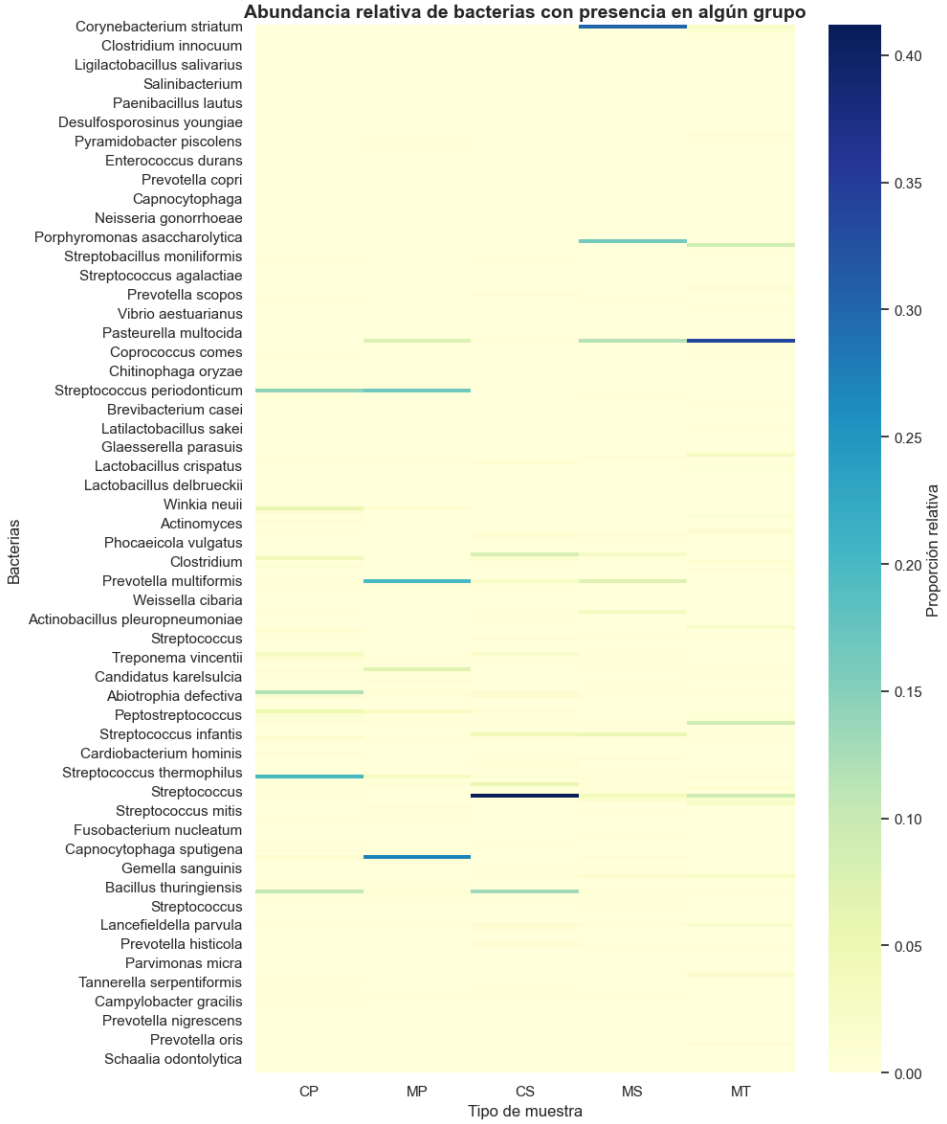



*Figura 28:Grafico Abundancia 30 bacterias.*

### 7.1.4 GRÁFICOS DE BARRAS DE ABUNDANCIA

El código identifica las bacterias más abundantes en los diferentes tejidos y las compara con otros grupos, determinando cuáles son comunes y cuáles son exclusivas o dominantes en cada muestra, con el fin de resaltar posibles asociaciones microbianas específicas del tejido.

```
# ──────────────────────────────────────────────────────────
# Paso 5: Top bacterias por tipo de muestra (barras multicolor sin error
bars)
# ──────────────────────────────────────────────────────────
top_n = 10
for grupo in grupos.keys():
    top = df_resumen.nlargest(top_n, grupo)[["nombre_normalizado", grupo]]

    plt.figure(figsize=(8, 5))
    sns.barplot(
        data=top,
        x=grupo,
        y="nombre_normalizado",
        palette=PALETTE[:len(top)],  # cada barra con color distinto
        errorbar=None  # ← evita mostrar la línea negra
    )

    plt.title(f"Top {top_n} bacterias más abundantes en {grupo}",
fontsize=13, fontweight='bold')
    plt.xlabel("Nivel de abundancia", fontsize=11)
    plt.ylabel("Bacteria", fontsize=11)
    plt.grid(axis='x', linestyle='--', alpha=0.5)
    plt.tight_layout()
    plt.show()
```

*Figura 29:Código de gráficos de barras.*

**Gráfico de barras top 10 de bacterias con mayor abundancia en las muestras de tejido tumoral (MT):**

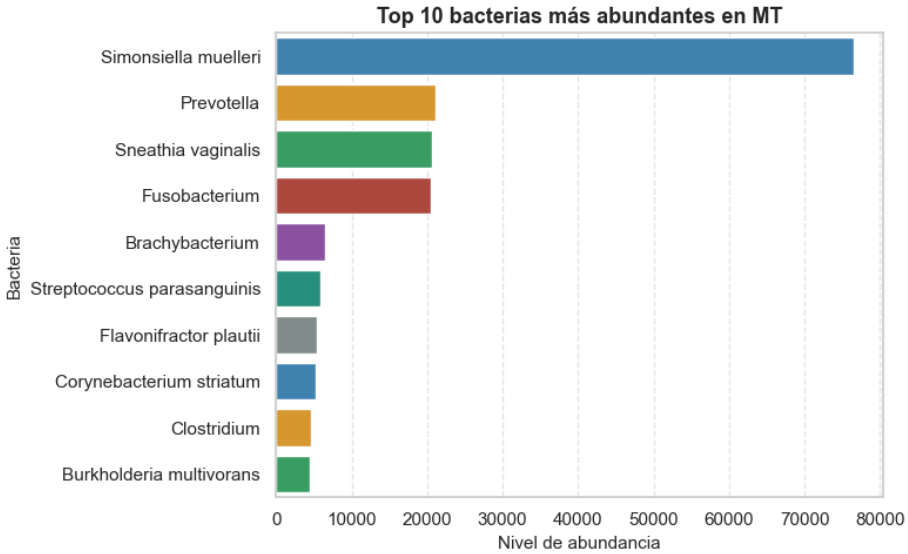



*Figura 30:Grafica bacterias más abundantes (MT).*

Se observa que *Simonsiella muelleri* muestra una dominancia marcada, superando ampliamente a las demás especies. En niveles intermedios destacan *Prevotella*, *Sneathia vaginalis* y *Fusobacterium*, todas ellas previamente asociadas con procesos inflamatorios y disbiosis en tejidos orales. Las demás bacterias, aunque menos abundantes, contribuyen al perfil microbiano característico del tejido tumoral.

**Gráfico de barras diez bacterias más abundantes en las muestras de saliva de pacientes (MS)**.

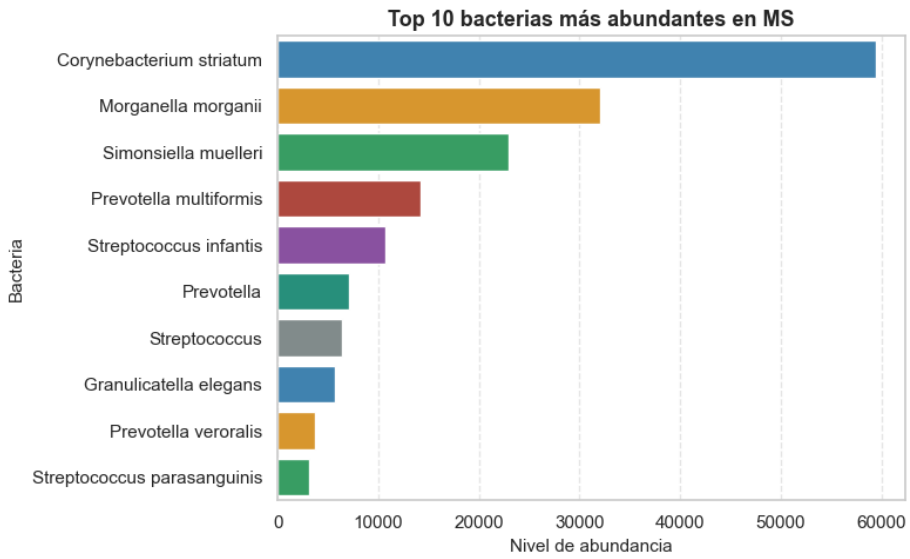
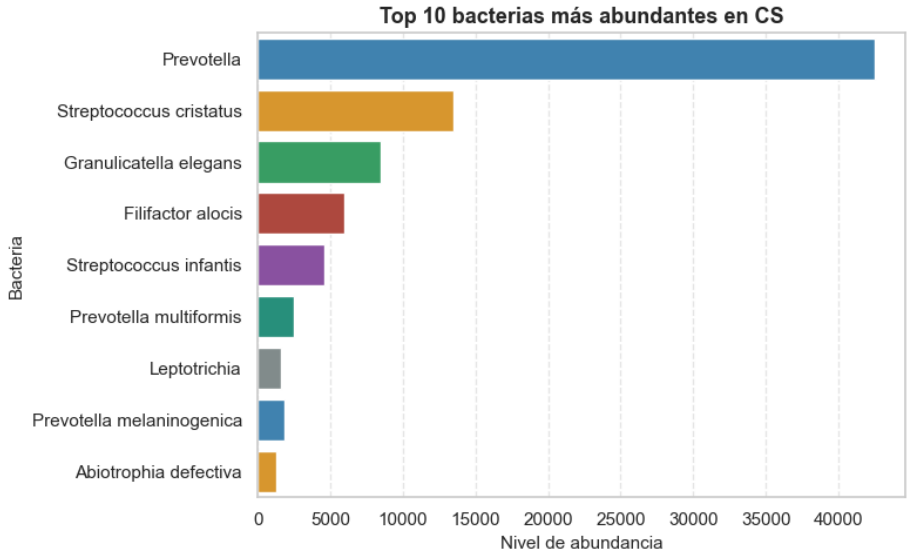



*Figura 31:Grafica bacterias más abundantes (MS).*

Se observa un claro predominio de *Corynebacterium striatum* y *Morganella morganii*, seguidas por *Simonsiella muelleri* y *Prevotella multiformis*. Estas especies destacan por su alta prevalencia en ambientes orales alterados y su posible implicación en procesos infecciosos o inflamatorios. La presencia de géneros como *Streptococcus* y *Prevotella* evidencia una comunidad microbiana mixta, con coexistencia de bacterias comensales y oportunistas características de las condiciones patológicas analizadas.

**Gráfico de barras diez bacterias más abundantes en las muestras de saliva de pacientes (MS)**.




El gráfico muestra las bacterias más abundantes en saliva control (CS), destacando el predominio de *Prevotella*, seguida de *Streptococcus cristatus* y *Granulicatella elegans*, lo que refleja un perfil microbiano típico de microbiota oral saludable.

### 7.1.5 COMPARACIÓN DE MUESTRAS

El código compara las bacterias más abundantes en tejido tumoral (MT) con las de otros grupos. Selecciona las 15 especies más representativas de cada conjunto, identifica las bacterias compartidas y las exclusivas del tumor, y luego imprime ambos listados para resaltar posibles diferencias microbianas asociadas al tejido tumoral.

```python
# ──────────────────────────────────────────────────────
# Paso 6: Comparación con tumores (MT)
# ──────────────────────────────────────────────────────
top_mt = set(df_resumen.nlargest(15, 'MT')["nombre_normalizado"])
otros_totales = df_resumen.drop(columns=['nombre_normalizado',
'MT']).sum(axis=1)
top_otros = set(df.loc[otros_totales.nlargest(15).index,
"nombre_normalizado"])

comunes = top_mt.intersection(top_otros)
solo_mt = top_mt.difference(top_otros)

print("\n═══ Bacterias presentes tanto en tumores como en otros grupos ═══")
for b in comunes:
    print(f" - {b}")
print("\n═══ Bacterias exclusivas o dominantes en tumores (MT) ═══")
for b in solo_mt:
    print(f" - {b}")
```

*Figura 32:Código comparación con tumores.*

**Bacterias presentes tanto en tumores como en otros grupos**

- Prevotella
- Corynebacterium striatum
- Simonsiella muelleri

**Bacterias exclusivas o dominantes en tumores (MT)**

- Streptococcus parasanguinis
- Prevotella veroralis
- Sneathia vaginalis
- Leptotrichia trevisanii
- Brachybacterium
- Clostridium
- Burkholderia multivorans
- Lancefieldella párvula
- Fusobacterium
- Brachybacterium huguangmaarense
- Flavonifractor plautii

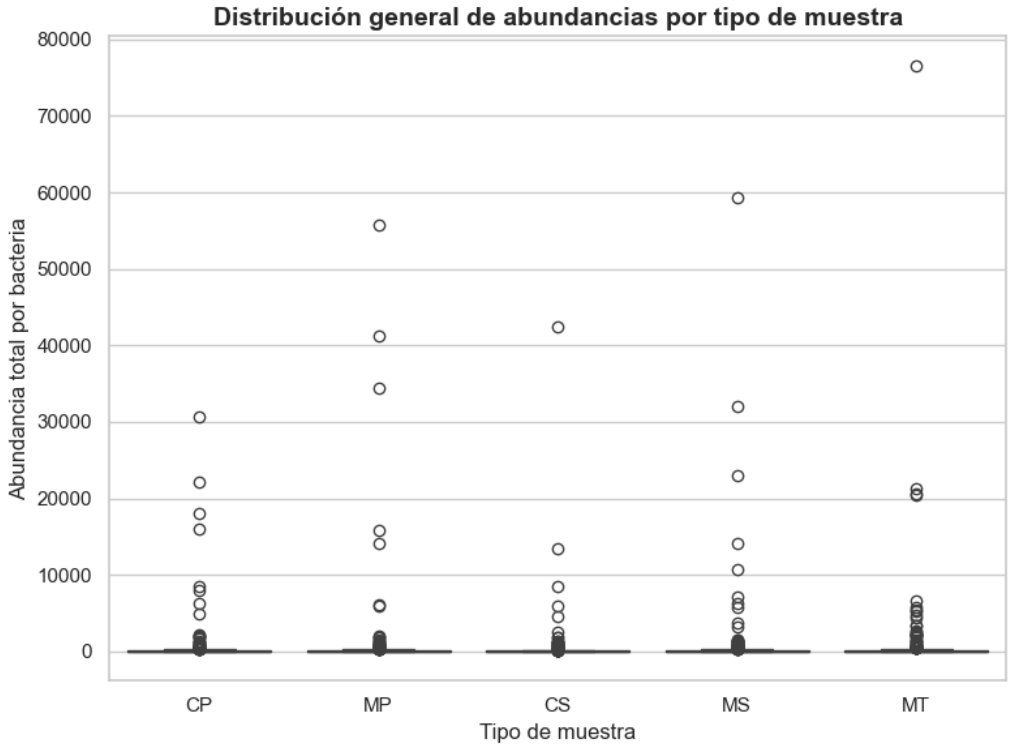



*Figura 33:Grafico distribución general por tipo.*

El gráfico presenta la distribución de la abundancia total por bacteria en diferentes tipos de muestras (CP, MP, CS, MS, MT).
En el eje vertical se muestra la abundancia total, mientras que en el eje horizontal se encuentran los diferentes tipos de muestras. Los puntos representan las mediciones individuales de abundancia bacteriana para cada tipo de muestra, y la dispersión de estos puntos indica la variabilidad dentro de cada grupo.

Se observa una concentración de los puntos cercanos a la base del gráfico, lo que sugiere que la mayoría de las muestras tienen abundancias bacterianas bajas.

Sin embargo, también se identifican algunos puntos dispersos a alturas mucho mayores, lo que indica que existen muestras con abundancias significativamente altas. Este patrón refleja la heterogeneidad de la abundancia bacteriana entre los tipos de muestra analizados.

# 8 ANÁLISIS EXPLORATORIO TAXONÓMICO Y DE RESISTENCIA

## 8.1 OBJETIVO DEL CAPÍTULO

Analizar la distribución taxonómica y los patrones de resistencia antimicrobiana presentes en las bacterias identificadas en las muestras clínicas, con el fin de caracterizar los fenotipos de resistencia, los antibióticos más frecuentes y su relación con los distintos niveles taxonómicos, a partir de los datos procesados del estudio.

### 8.1.1 DESCRIPCIÓN

El código realiza un análisis exploratorio de los datos de resistencia bacteriana contenidos en el archivo *df_004_inner_full.csv.* Primero, identifica los valores taxonómicos más representativos (familia, género y especie) y genera gráficos de barras para visualizar su frecuencia. Luego, determina los antibióticos y fenotipos de resistencia más comunes, mostrando su distribución mediante diagramas de barras y gráficos circulares.

Posteriormente, evalúa la asociación entre familias bacterianas y fenotipos de resistencia mediante un mapa de calor (*heatmap*), que permite observar patrones de susceptibilidad o resistencia predominantes por grupo taxonómico. Finalmente, compara los tipos de evidencia disponibles (experimental vs. predicción computacional), identificando la proporción de información derivada de laboratorio, lo que proporciona una visión integral del perfil de resistencia en las muestras analizadas.

# 9 CONCLUSIONES Y TRABAJOS FUTUROS

## 9.1 CONCLUSIONES

El desarrollo de este proyecto permitió avanzar significativamente en la caracterización de comunidades bacterianas a partir de datos metagenómicos, aplicando técnicas de minería de datos y modelos de aprendizaje automático. A lo largo de la investigación, se integraron múltiples fuentes de información (resistencia, genómica y metadatos clínicos) para construir un *dataset* robusto y limpio, lo cual fue clave para llevar a cabo análisis exploratorios, clasificación supervisada y *clustering* no supervisado sobre las bacterias orales presentes en humanos.

Entre los logros más relevantes del estudio, se destacan los siguientes:

- **Clasificación efectiva de bacterias:** mediante el uso de algoritmos como Random Forest y XGBoost, se logró identificar patrones de resistencia antibiótica con una precisión global del 74%, validando la utilidad de los modelos supervisados en contextos de alta variabilidad biológica.
- **Agrupamiento natural de comunidades bacterianas:** el modelo de clustering K-

Means reveló cuatro grupos diferenciados de bacterias, los cuales presentaron perfiles particulares de resistencia y distribución geográfica, fortaleciendo la hipótesis de que ciertos clústeres comparten características patogénicas y epidemiológicas.

- **Asociación estadística entre clústeres y resistencia:** a través de pruebas estadísticas como chi-cuadrado, se demostró que existe una relación significativa entre los grupos bacterianos identificados y los fenotipos de resistencia, lo cual respalda el uso de estos enfoques como herramientas para la vigilancia y predicción clínica.

Sin embargo, uno de los objetivos más ambiciosos del proyecto —**implementar un modelo que recomiende tratamientos antibióticos personalizados a partir del perfil metagenómico y la clasificación bacteriana**— no pudo completarse en su totalidad debido a la falta de una base de datos clínicamente etiquetada que relacione genomas bacterianos con tratamientos exitosos o fallidos en pacientes reales.

Pese a esta limitación, los resultados obtenidos constituyen una **base sólida para que futuras investigaciones** avancen hacia la integración de modelos de recomendación antibiótica.

### 9.2 TRABAJOS FUTUROS

- **Construcción de una base de datos clínico-terapéutica:** que incluya el historial de tratamientos antibióticos, resultados clínicos, bacterias detectadas, y su perfil genético. Esta base permitiría entrenar modelos capaces de predecir la eficacia de un antibiótico en función del metagenoma del paciente.
- **Integración con modelos de recomendación basados en aprendizaje supervisado o reforzado:** una vez disponible esta información, sería viable implementar sistemas que no solo predigan la resistencia, sino que también sugieran el tratamiento más adecuado con base en evidencia histórica y patrones genéticos.
- **Validación clínica de los modelos:** los algoritmos propuestos deben ser evaluados en entornos hospitalarios o mediante colaboración con instituciones médicas para verificar su aplicabilidad en casos reales.
- **Expansión del análisis a otras comunidades bacterianas:** si bien esta tesis se centró en bacterias orales humanas, los métodos desarrollados pueden ser escalables a microbiomas intestinales, respiratorios u hospitalarios, donde también es crítica la resistencia a antibióticos.

## 10 REFERENCIAS BIBLIOGRÁFICAS


[1] F. Author, "Title of paper," *Journal Name*, vol. 10, no. 5, pp. 99-102, Jan. 2020.

[2] J. Smith, "Metagenomics: A powerful tool to investigate microbial communities and their functions," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 3, pp. 899-2103, March 2019.

[3] A. Patel, "Machine Learning: A comprehensive survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 1234-1252, June 2018.

[4] A. Kumar, "Virulence factors and their role in pathogenicity," *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 870-6724, Dec. 2017.

[5] B. Brown, "Statistical markers and their applications in data analysis," *IEEE Access*, vol. 7, pp. 877-3092, May 2019.

[6] Organización Mundial de la Salud, "Informe mundial sobre la resistencia a los antimicrobianos," 2019.

[7] C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata, "Shotgun metagenomics, from sampling to analysis," *Nature Biotechnology*, vol. 35, no. 9, pp. 833-844, 2017.

[8] C. Y. Chiu and S. A. Miller, "Clinical metagenomics," *Nature Reviews Genetics*, vol. 20, no. 6, pp. 341-355, 2019.

[9] E. A. Dinsdale, R. A. Edwards, D. Hall, et al., "Functional metagenomic profiling of nine biomes," *Nature*, vol. 452, no. 7187, pp. 629-632, 2008.

[10] J. Handelsman, "Metagenomics: Application of Genomics to Uncultured Microorganisms," *Microbiology and Molecular Biology Reviews*, vol. 68, no. 4, pp. 669-685, 2004.

[11] J. Qin, R. Li, J. Raes, et al., "A human gut microbial gene catalogue established by metagenomic sequencing," *Nature*, vol. 464, no. 7285, pp. 59-65, 2010.

[12] C. S. Riesenfeld, P. D. Schloss, and J. Handelsman, "Metagenomics: genomic analysis of

microbial communities," *Annual Review of Genetics*, vol. 38, pp. 525-552, 2004.

[13] B. B. Finlay and S. Falkow, "Common themes in microbial pathogenicity revisited," *Microbiology and Molecular Biology Reviews*, vol. 61, no. 2, pp. 136-169, 1997.

[14] J. Pizarro-Cerdá and P. Cossart, "Bacterial adhesion and entry into host cells," *Cell*, vol. 124, no. 4, pp. 715-727, 2006.

[15] R. Rappuoli and A. Aderem, "A 2020 vision for vaccines against HIV, tuberculosis and malaria," *Nature*, vol. 473, no. 7348, pp. 463-469, 2011.

[16] K. Qu, F. Guo, X. Liu, et al., "Application of Machine Learning in Microbiology," *Frontiers in Microbiology*, vol. 10, p. 827, 2019.

[17] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nature Reviews Genetics*, vol. 16, no. 6, pp. 321-332, 2015.

[18] G. Arango-Argoty, E. Garner, A. Pruden, L. S. Heath, P. Vikesland, and L. Zhang, "DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data," *Microbiome*, vol. 6, no. 1, p. 23, 2018.

[19] R. Knight, J. Jansson, D. Field, et al., "Unlocking the potential of metagenomics through replicated experimental design," *Nature Biotechnology*, vol. 30, no. 6, pp. 513-520, 2012.

[20] M. Pop and S. L. Salzberg, "Bioinformatics challenges of new sequencing technology," *Trends in Genetics*, vol. 24, no. 3, pp. 142-149, 2008.

[21] B. Liu, Y. Liu, J. Li, et al., "Learning-based methods for predicting functional roles of metagenome-derived antibiotic resistance genes," *Scientific Reports*, vol. 9, p. 2773, 2019.

[22] C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata, "Shotgun metagenomics, from sampling to analysis," *Nature Biotechnology*, vol. 35, no. 9, pp. 833-844, 2017.

[23] C. Y. Chiu and S. A. Miller, "Clinical metagenomics," *Nature Reviews Genetics*, vol. 20, no. 6, pp. 341-355, 2019.

[24] A. Dutta and G. Dutta, "Machine learning algorithms in metagenomics and human

microbiome studies: A systematic review," *Genomics, Proteomics & Bioinformatics*, vol. 18, no. 4, pp. 420-431, 2020.

[25] Y. Liu and X. Ding, "Applications of machine learning in metagenomics," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 1898-1910, 2021.

[26] *Bacterial and Viral Bioinformatics Resource Center (BV-BRC)*, U.S. Department of Energy Systems Biology Knowledgebase (KBase) and National Institutes of Health (NIH), 2024. [Online]. Available: https://www.bv-brc.org. Accessed: Oct. 25, 2025.

[27] S. Federhen, "The NCBI Taxonomy database," *Nucleic Acids Research*, vol. 40, no. D1, pp. D136–D143, 2012.

[28] C. Wittouck *et al.*, "A genome-based species taxonomy of the Lactobacillus genus complex," *Nature Microbiology*, vol. 5, pp. 251–259, 2020.

[29] P. Parks *et al.*, "A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life," *Nature Biotechnology*, vol. 36, pp. 996–1004, 2018.

[30] C. Linnaeus, *Systema Naturae*, 10th ed., Stockholm: Laurentius Salvius, 1758.

[31] M. K. Tindall, "The use of taxonomic names in microbiology and the role of nomenclature," *International Journal of Systematic and Evolutionary Microbiology*, vol. 61, pp. 2775–2780, 2011.