# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD INGENIERÍA,
ARQUITECTURA Y DISEÑO

**BIOINGENIERÍA**

# Predicción de promotores dependientes del factor $\sigma$70 en *E. coli* aplicando modelado de Redes Neuronales con Validación Cruzada


Tesis que presenta:
**Hernández Ponce Braulio Andrés**


Para obtener el título de:
**BIOINGENIERO**


Director de Tesis: **Dr. Dante Alberto Magdaleno Moncayo**


Ensenada, B. C., a 22 de mayo del 2024

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO

**"Predicción de promotores dependientes del factor σ70 en *E. coli* aplicando modelado de Redes Neuronales con Validación Cruzada"**

TESIS

PARA CUBRIR LOS REQUISITOS NECESARIOS PARA OBTENER EL TÍTULO DE

**Bioingeniero**

PRESENTA

**Hernández Ponce Braulio Andrés**
**1277856**

A quien el Comité de Tesis autoriza el trabajo terminal, después de haber efectuado una revisión minuciosa del mismo y de acuerdo con el Art. 19 del R.G.E.P.E.P, las y los señores profesores emiten los siguientes votos aprobatorios mediante rubrica:

Dr. Dante Alberto Magdaleno Moncayo
del programa Bioingeniero
**DIRECTOR**

Dra. Haydeé López Rodríguez
del programa Ingeniero en Nanotecnología
SINODAL

Dr. Rubén César Villarreal Sánchez del programa Bioingeniero
SINODAL

**"Por la Realización Plena del Ser"**

C.c.p.- Archivo
C.c.p.- Minutario

- **Promotores**
- **Bioinformática**
- *E. coli*
- **Aprendizaje automático**

## Agradecimientos

*A mi asesor y director de tesis Dr. Dante Alberto Magdaleno Moncayo, a quien le debo mi gusto por el conocimiento y la investigación.*

*A mis familiares por su apoyo no solo emocional si no también en lo material, agradezco inmensamente todo su esfuerzo.*

*A mis compañeros y compañeras quienes se convirtieron en mis hermanos de título.*

# Contenido

# Tabla de imágenes

# Resumen

## Predicción de promotores dependientes del factor $\sigma 70$ en E. coli aplicando modelado de Redes Neuronales con Validación Cruzada

El proyecto se enfocó en la aplicación de herramientas bioinformáticas y métodos de análisis de secuencias para la predicción de promotores dependientes de la subunidad sigma 70 en la bacteria *E. coli*. El objetivo principal fue recopilar y curar un conjunto de datos de secuencias genómicas de *E. coli*, con énfasis en las regiones promotoras y posteriormente utilizar estas secuencias para modelar un programa apto para la predicción de promotores de *E. coli*. Este enfoque involucró la extracción de características relevantes de las secuencias de ADN, así como el empleo de herramientas bioinformáticas para el análisis y la predicción de promotores. Además, el proyecto exploró el uso de técnicas de machine learning, como las redes neuronales, para el análisis de las secuencias genómicas y la predicción de promotores. Se espera que los resultados de esta investigación contribuyan al desarrollo de métodos para la identificación de promotores dependientes de sigma 70 en *E. coli*, lo cual podría tener aplicaciones significativas en la ingeniería genética, la biotecnología y la investigación en biología molecular. Este proyecto busca integrar la bioinformática, el análisis de secuencias genómicas y el machine learning para avanzar en la comprensión y predicción de promotores en la bacteria *E. coli.*

# Capítulo 1. Introducción

## La Bioinformática

La bioinformática ha surgido como una disciplina científica en respuesta a la creciente necesidad de interpretar la abundante información presente en las secuencias de DNA, RNA y proteínas. Con la difusión de las técnicas de secuenciación de DNA y proteínas, así como el aumento significativo en el volumen de secuencias almacenadas en bancos de datos, ha surgido la demanda de desarrollar algoritmos especializados. Estos algoritmos tienen como objetivo catalogar secuencias, analizar similitudes entre ellas y descubrir sus propiedades estructurales y funcionales. La bioinformática se caracteriza por ser una disciplina interdisciplinaria, fundamentada en la conjunción de la biología y las ciencias de la computación. No obstante, su alcance se extiende considerablemente al integrar elementos de la fisicoquímica, las matemáticas, la estadística y la probabilidad. Este enfoque integrado permite abordar de manera integral la complejidad de la información biológica contenida en las secuencias, proporcionando herramientas y perspectivas valiosas para la investigación en diversos campos científicos (Barnetche, 2007).

Actualmente, han surgido herramientas que derivan de la profundización del conocimiento en esta área, la Secuenciación de Nueva Generación (NGS) es una de estas herramientas, esta tecnología tiene un alto rendimiento permitiendo la secuenciación masiva de pares de bases de muestras de AND o ARN. Es así que se pueden generar una inmensa cantidad de datos en horas, y es en este punto que la bioinformática tiene la capacidad de analizar todos estos datos asumiendo un papel muy importante en el procesamiento (Branco, 2021).

## *Escherichia coli*

*E. coli* es una bacteria Gram-negativa con forma de bastón que pertenece a Enterobacteriales, Enterobacteriaceae, Escherichia, incluyendo cepas comensales y patógenas. Puede causar una variedad de enfermedades como infecciones del tracto urinario o enfermedades extraintestinales y debido a la propagación que ha

tenido y la aparición de resistencia, el tratamiento se vuelve cada vez más limitado (Shen, 2023).

### *Escherichia coli* K-12

*E. coli* K-12 es uno de los organismos modelos más utilizados en los trabajos de los investigadores. Gracias a esto el conocimiento sobre los mecanismos reguladores en este organismo se encuentra altamente bien caracterizados de manera experimental. Y debido a todo este trabajo es que podemos encontrar varias bases de datos como PRODORIC, que se centra en los motifs de unión del factor de transcripción (TF) para diferentes bacterias, RegPrecise contiene regulones microbianos reconstruidos por genómica comparada o BioCyc, que incluye una gran cantidad de genomas microbianos. Para el caso de *E. coli* K-12, las bases datos mejor actualizadas con datos seleccionados manualmente de literatura pertenecen a RegulonDB y EcoCyc (Salgado, 2024).

### Promotores

Las secuencias promotoras son segmentos de ADN ubicados río arriba del sitio de inicio de la transcripción (TSS) o + 1, donde se une la enzima ARN polimerasa (RNAP) para llevar a cabo la transcripción genética, se encuentran ubicados en regiones proximales a un gen y en la región 5’ de la secuencia donde el gen comienza a ser transcrito (NIH, 2024). En las bacterias, esta interacción sólo es posible cuando una proteína adicional, un factor sigma (σ), interactúa con la RNAP. La función principal de los factores σ es redirigir la RNAP a promotores específicos, otorgando especificidad al reconocimiento del promotor (Martinez, 2021).

La transcripción de las bacterias como *E. coli* es iniciada por la ARN polimerasa que es una enzima multi-unidad, la subunidad sigma 70 de esta enzima reconoce las secuencias especificas hacia arriba del gen y permite al resto de la enzima unirse, entonces estas secuencias río arriba donde se une la proteína sigma, representa una secuencia promotora, los cuales tienen segmentos -35 y -10 bp, estas últimas siendo ubicaciones río arriba desde el sitio de inicio de la transcripción conocidos también como 35 box y 10 box. En *E. coli* la 35 box tiene una secuencia

consenso “TTGACA” y la 10 box tiene una secuencia consenso “TATAAT” (Jiménez, 2013).

## Factores de transcripción de *E.coli*

Los factores de transcripción son proteínas que se unen al DNA para controlar los genes. Los factores de transcripción tienen funciones fundamentales en casi todos los procesos biológicos como desarrollo y crecimiento, así también en las respuestas a factores ambientales. Estas proteínas estimulan o reprimen la tasa transcripcional de sus genes blanco al unirse a regiones promotoras específicas elementos cis, lo que desencadena en la activación o desactivación de cascadas de señalización de genes. Los sitios de unión a factores de transcripción (elementos cis-reguladores o motivos) son secuencias de DNA que influyen de manera temporal y espacial en la actividad transcripcional. Múltiples elementos en cis forman módulos de regulación cis (CRMs), los cuales integran las señales de múltiplos factores de transcripción, resultando en un control combinacional con patrones específicos de regulación (Hernández, 2017).

Hay varios factores sigma en la ARN polimerasa de *E. coli*, que dependen del medio ambiente y de los genes. Estos factores se utilizan como elementos distintivos de las secuencias promotoras que se encuentran en el ADN. Siendo seis, los diferentes tipos de factores sigma: $\sigma 24, \sigma 28, \sigma 32, \sigma 38, \sigma 54, \sigma 70$. Cada uno de estos factores tienen su propia función, por ejemplo, el factor $\sigma 70$ es responsable de la transcripción de la mayoría de los genes bajo condiciones de mantenimiento. El factor $\sigma 24$ y $\sigma 32$ son responsables de la respuesta al choque térmico, el $\sigma 28$, es responsable de los genes flagelares, el factor $\sigma 38$ es de respuesta al estrés durante la transición de fase de crecimiento exponencial a la fase estacionaria y el factor $\sigma 54$ es de respuesta al metabolismo de nitrógeno (Sifat,, 2020).

**TABLA 1. Factores sigma para la transcripción y sus funciones.**

| Factores sigma y su función | |
|---|---|
| **FACTOR** | **FUNCIÓN** |
| $\boldsymbol{\sigma 24}$ | Factor de estrés en respuesta al choque térmico. |
| $\boldsymbol{\sigma 28}$ | Responsable de genes flagelares. |
| $\boldsymbol{\sigma 32}$ | Factor de estrés en respuesta al choque térmico. |
| $\boldsymbol{\sigma 38}$ | Respuesta al estrés durante la transición de fase de crecimiento exponencial a la fase estacionaria. |
| $\boldsymbol{\sigma 54}$ | Respuesta al metabolismo del nitrógeno. |
| $\boldsymbol{\sigma 70}$ | Encargado de la expresión basal en condiciones de mantenimiento. |

Por ejemplo, el factor $\sigma 70$ que es importante para la expresión de genes constitutivos, reconoce una expresión consenso de 35 pb de longitud aproximadamente, con dos elementos claves, la caja -10 y la caja -35, y en este sentido la producción de ARN en el sitio de inicio de la transcripción es el resultado de la interacción de estos elementos cis, con la región central que constituye al promotor (Wu, 2023).

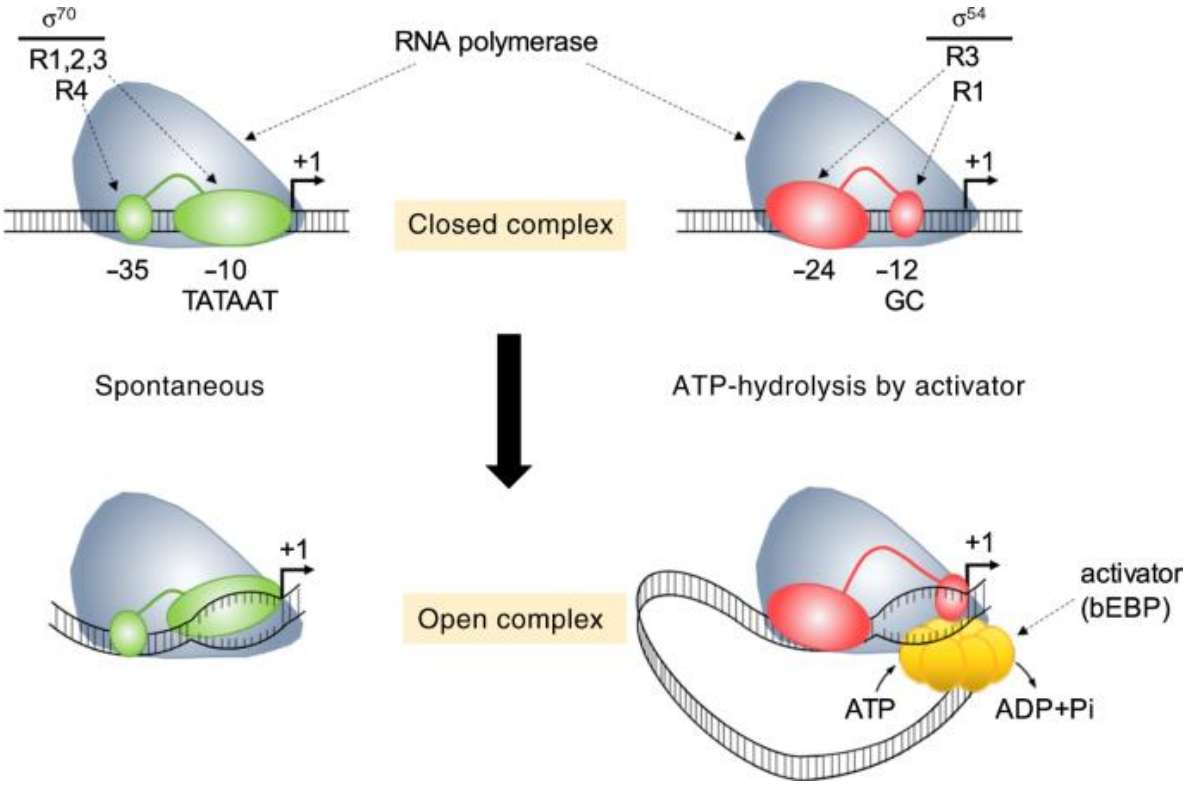



*Imagen 1. Iniciación de la transcripción en promotores controlados por sigma 70 y sigma 54 en bacterias. Los factores dirigen la ARN polimerasa a promotores bacterianos para formar el complejo cerrado. Las regiones (R) 1,2,3 y 4 de $\sigma^{70}$ se unen a elementos de ADN en la posición −10 (secuencia de consenso: TATAAT) y −35, mientras que R1 y R3 de $\sigma^{54}$ se unen a secuencias en la posición −12 (secuencia de consenso: GC) y −24 aguas arriba del sitio de inicio de la transcripción (+1) (Engl, 2020).*

## Aplicaciones del conocimiento de los promotores de *E.coli*

### Proteínas recombinantes:

La producción de PR en bacterias es una tecnología que surgió hace cerca de 30 años y respondió a una necesidad de proveer proteína de uso terapéutico con un abasto asegurado (que no dependiera de fuentes animales) y calidad constante. La primera PR aprobada para su uso en humanos fue la insulina humana producida en *E. coli* por la empresa Genentech. Muchas de las PR terapéuticas requieren modificaciones post–traduccionales que no pueden ser llevadas a cabo por cepas bacterianas silvestres, por lo que son preferentemente producidas por células eucariontes superiores (Lara, 2011). *E. coli* es el huésped bacteriano más popular para la producción de proteínas recombinantes y esto es debido a su rápida tasa de crecimiento que tiene un tiempo de 20 minutos de generación en condiciones óptimas, también porque hay herramientas bien desarrolladas de manipulación

molecular junto con su bien conocida biología, y además puede lograr una alta densidad celular con la utilización de reactivos de cultivo económicos (Arshpreet Bhatwa, 2021).

## RegulonDB

Esta plataforma es una base de datos que proporciona información sobre la regulación genética de *E. coli* K-12 en el que se utiliza MongoDB, que es un administrador de bases de datos, lo cual permite tener un esquema de datos flexible que permite la adaptación del usuario a la heterogeneidad de la información biológica. Un datamart es un sistema de almacenamiento de datos que se centra en áreas temáticas específicas y, por lo tanto, se llega a un análisis eficiente de la información. En RegulonDB hay cinco datamarts diferentes que a su vez se divide en otros datos subyacentes de cada categoría; genes, operones, regulon, sigmulon y unidades genéticas (Salgado, 2024).

## Predicción o búsqueda de promotores en el ADN

En el área de la bioinformática, el tema de predicción o búsqueda de promotores es una tarea que implica la identificación de estas secuencias y para esto hay métodos que deben utilizarse según la complejidad del análisis que se realice y las herramientas con las que se cuente para ello.

### Métodos experimentales

- **Mapeo del sitio de inicio de la transcripción:** de esta manera se accede al ADN para la identificación de elementos reguladores, mapeando los nucleosomas que bordean las NDR, generalmente mediante marcas de histonas, incluyendo modificaciones “activas” de histonas, como la metilación H3K4, y la acetilación de K3K27, de modo que las enzimas que modifican las colas de las histonas y las chaperonas que depositan las subunidades de nucleosomas son más activas cerca de los sitios de inicio de la transcripción, lo que generalmente ocurre de manera bidireccional tanto en los promotores como en los potenciadores de genes para producir ARNm estables (Henikoff, 2020).

- **Reporter Merger Test:** Estos son ensayos de fusión reporteros/informadores, que funciona mediante marcadores visibles, estos son agregados a los candidatos promotores y si al probar estos marcadores se producen enzimas funcionales para la síntesis de lo que resulta de la transcripción de ese gen y su producción muestra un marco de lectura abierto y el marcador resaltará de manera evidente en la producción (He, 2020).
- **Sort-Seq**: Se utiliza clasificación de células activadas por fluorescencia (FACS) basada en cambios en la fluorescencia debido a los promotores mutados combinados con secuenciación para identificar las ubicaciones específicas del sitio de unión de la transcripción en el genoma (Ireland, 2020).

**Métodos computacionales**

- **Análisis de Secuencias de ADN**: Hay secuencias o elementos altamente conservados en especies que nos dan a conocer la presencia de un promotor, en bacterias están los elementos -10 y -35, además cajas TATA o cajas CG que son iniciadores de la transcripción que si bien no en todos los organismos es igual, hay varias secuencias consenso de estos elementos, sin embargo se ha visto una mayor incidencia en promotores que contienen elementos TATA, y para la búsqueda de estos elementos hay herramientas que periten la alineación de genomas para buscar cambión entre estos o similitudes (Brázda, 2021).
- **Uso de softwares:** Se puede utilizar programas para mapeo de promotores que son basados en el uso de matrices de peso de posición (PWM) de los motifs -10 y -35, teniendo en cuenta la distribución de la longitud de espaciador entre los motifs y su distancia con el sitio de inicio de la transcripción (TSS) (Wu, 2023)..
- **Aprendizaje automático:** Recientemente se ha comenzado a implementar estas herramientas de aprendizaje automático para el reconocimiento de promotores, TSS, y secuencias reguladoras, como ejemplo, maquina de soporte de vectores o redes neuronales (Wu, 2023).

# Capítulo 2. Antecedentes

## Aprendizaje Automático (Machine Learning):

Las técnicas de aprendizaje automático (ML) y aprendizaje profundo (DL) desempeñan un papel importante en el análisis de datos a comparación de métodos estadísticos. Se están usando estas técnicas ML se han utilizado en la genómica, ya sea para, identificación de sitio de empalme, regiones promotoras, clasificación de genes relacionados a enfermedades, identificación de inicio de la transcripción (TSS), identificación de proteínas y más (Bhandari, 2021).

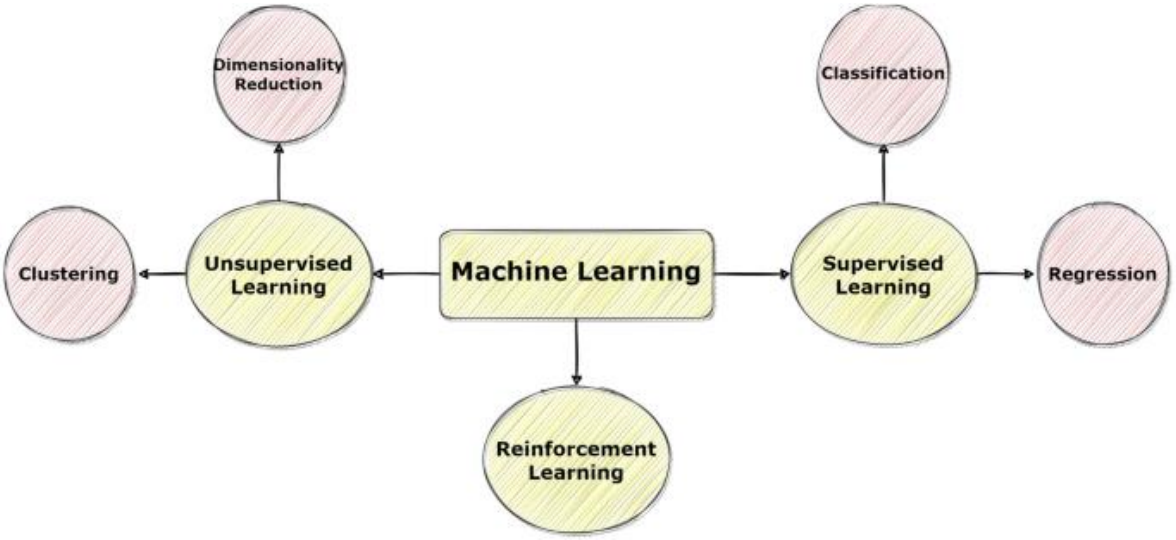



*Imagen 2. Metodologías y técnicas del ML*

### Tipos de aprendizaje automático

- **Aprendizaje supervisado:** el programa de computadora deriva una función entre las entradas y las salidas a partir de un conjunto de datos de entrenamiento etiquetados. La intervención humana juega un papel importante en el aprendizaje supervisado. Las personas no solo etiquetan el resultado del conjunto de entrenamiento, sino que también seleccionan las características, los algoritmos e incluso los parámetros de control de los algoritmos basándose en varias suposiciones de los algoritmos (Kang, 2020).
- **Aprendizaje no supervisado**: a diferencia del aprendizaje supervisado, este tipo de aprendizaje automático no requiere datos etiquetados. El aprendizaje no supervisado suele utilizarse cuando no se conocen las relaciones entre

las variables de entrada. El aprendizaje no supervisado proporciona el patrón de variables de entrada y, en su mayoría, presenta diferentes grupos construidos en función de los datos de entrada (Kang, 2020).

- **Aprendizaje semisupervisado:** Los algoritmos de aprendizaje semisupervisados pueden entrenar un modelo con datos etiquetados y sin etiquetar, lo que puede proporcionar una mayor precisión en comparación con el modelo supervisado que utiliza datos etiquetados muy limitados (Kang, 2020).
- **Aprendizaje por refuerzo**: en el aprendizaje por refuerzo, los agentes observan el entorno, realizan algunas acciones y obtienen algunas recompensas (negativas/positivas) en función de la acción seleccionada, y luego el modelo se actualiza en consecuencia (Kang, 2020).

**Técnicas de Machine Learning**

- **Redes Neuronales:** Son modelos matemáticos/computacionales en los que se aprende con redes neuronales artificiales. Las neuronas artificiales en una red están interconectadas a través de funciones matemáticas que interactúan entre sí para producir un valor de salida o respuesta. En esta red, la salida de una neurona influye en las neuronas subsiguientes, y la manera en que lo hace, ya sea activación o inhibición, puede variar de una neurona a otra según un factor que se evalúa en cada ciclo de aprendizaje. Además de este factor de propagación, los valores que determinan el umbral que define si una neurona se activará son evaluados mediante funciones matemáticas cuyos parámetros se ajustan ciclo tras ciclo. Las redes neuronales artificiales adquieren conocimiento mediante la modificación de los parámetros de conectividad y los umbrales de activación de cada neurona, de modo que minimizan una función de pérdida que evalúa la red al presentar conjuntos de datos verdaderos durante su entrenamiento (Reyna, 2020).

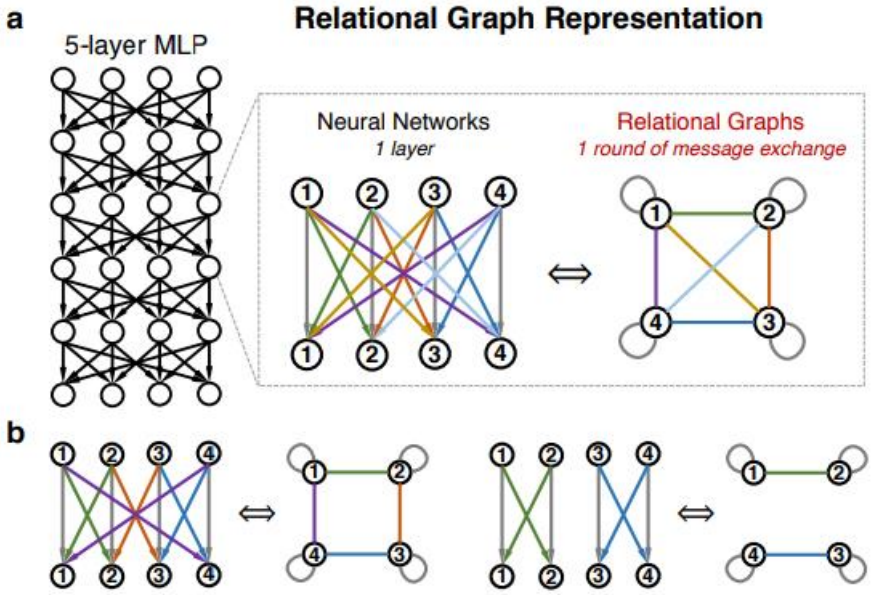



*Imagen 3. a) Una capa de red neuronal se puede ver como un gráfico relacional donde se conectan nodos los cuales intercambian mensajes, b) y puede haber diferentes capas y gráficos (You, 2020).*

- **Máquina de Soporte de Vectores:** Son modelos de aprendizaje supervisado diseñados para el análisis de clasificación y regresión de datos. Utilizando un conjunto de ejemplos de entrenamiento, cada uno etiquetado como perteneciente a una de dos categorías posibles, los algoritmos de máquinas de vectores de soporte construyen un modelo que asigna nuevos ejemplos a una de estas categorías, transformándolo en un clasificador lineal binario no probabilístico. En términos prácticos, las máquinas de vectores de soporte pueden entenderse como modelos que mapean los puntos de entrada a un espacio de características de mayor dimensión, con el objetivo de encontrar el hiperplano que los separe y maximice el margen entre las clases (Reyna, 2020).

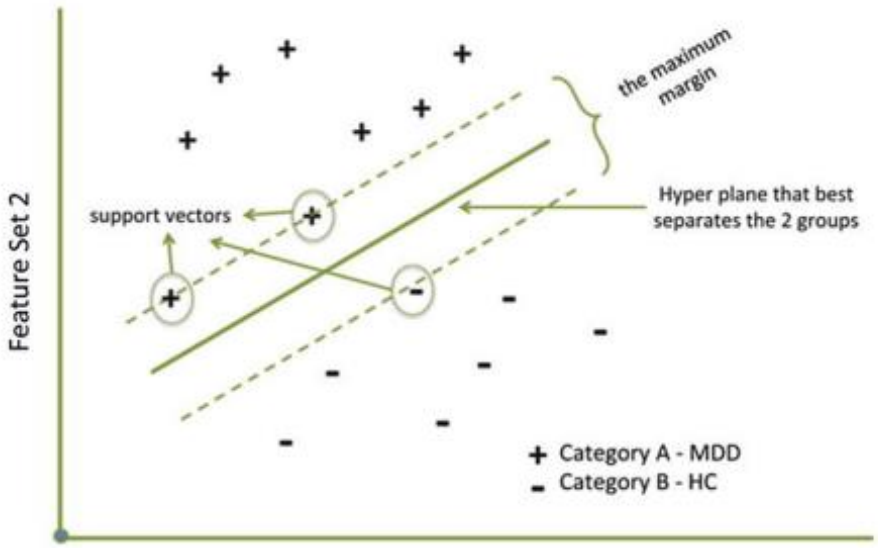



*Imagen 4. Ilustración del hiperplano que separa al máximo los vectores de apoyo correspondientes a cada una de las dos clases que se van a predecir, el trastorno depresivo mayor (TDM) y los controles sanos (HC). Ejemplo de aplicación (Pisner, 2020).*

## Aplicaciones del Aprendizaje Automático

### Bioinformática:

- Clasificación: Es una de las tareas más estudiadas del aprendizaje automático, su principio se basa en el atributo predicho para predecir la clase del atributo objetivo especificado por el usuario. Suelen implementarse en estos casos las redes bayesianas, árboles de decisión, redes neuronales (Yang, 2020).

- Agrupación: Se pueden agrupar secuencias con algunas características iguales y explorar la información efectiva de secuencias desconocidas a partir de funciones y estructuras conocidas. A diferencia con algoritmos de clasificación es que en la agrupación no implementa una categoría establecida (Yang, 2020).

## Trabajo en Aprendizaje Automático

SAPPHIRE, un clasificador basado en redes neuronales para la predicción del promotor sigma 70 en Pseudomonas fue entrenado en el trabajo de (Coppens, 2020), este modelo se entrenó utilizando un conjunto de datos de 170 promotores

únicos de Pseudomonas sigma 70, 94 de estas secuencias se tomaron por ser validados experimentalmente de *P. aeruginosa* y *P. putida* sigma 70, y los 76 restantes se recuperaron de la base de datos de NCBI. Se extrajeron aleatoriamente regiones intergénicas que no se anotaron como promotoras en el genoma de *P. aeruginosaë* proporcionando un conjunto de ejemplo negativos para el entrenamiento. En este trabajo se utilizó una codificación one-hot para transmitir las características del entrenamiento a la red neuronal. La arquitectura central de esta red neuronal consta de dos capas consecutivas completamente conectadas, que alimentan a una tercera capa de un solo nodo. Los resultados arrojados después de la optimización de este modelo dieron valores de 76.6 y 88.1% para sensibilidad media y la especificidad media respectivamente en una validación cruzada cinco veces en el conjunto de entrenamiento. Finalmente se concluye que esta herramienta en el futuro podrá compilar a otros factores y especies sigma, dependiendo de la disponibilidad de conjuntos de datos experimentales.

Cr-Prom (M. Shujaat, 2021), es otro modelo basado en redes neuronales convolucionales para la predicción de promotores, en este caso de arroz. El conjunto de datos utilizado fue uno desarrollado por ProRice, extrayéndose 4220 datos de PlantProm que contienen los TSS de varias especies de plantas y los datos negativos se obtuvieron de datos del proyecto de anotación de arroz (RAP-DB) siendo el mismo número de datos que los positivos. La longitud de cada secuencia positiva y negativa fue de 251 pb y el modelo dividió el conjunto de datos en un 70% para el entrenamiento y un 30% para la prueba de validación. En este trabajo se codificaron los datos por one-hot para convertir los datos a binarios de cuatro dimensiones para cada nucleótido. En caso de las redes neuronales convolucionales, tienen un gran beneficio, el cual es que no requiere una extracción preliminar de funciones. Para evaluar el desempeño se hizo una validación cruzada 5 veces, logrando una sensibilidad de 98.57% y una especificidad del 99.9%, una precisión del 99.1% y un MCC de 0.9839. estos datos representan un resultado de discriminación exitoso entre las secuencias de ADN promotoras y no promotoras teniendo resultados superiores en comparación a otras técnicas existentes.

En la investigación de (Pi-Jing Wei a, 2022), Se recopilaron secuencias promotoras de seis cepas de *Nannochloropsis*. Se utilizó la herramienta CD-HIT para eliminar el 80% de similitud en las secuencias de promotores de cada cepa. Esto sugiere que se buscaba diversidad en las secuencias de promotores. En lugar de seleccionar aleatoriamente regiones no promotoras del mismo genoma como muestras negativas, se utilizó un método de codificación dentro del grupo para generar un conjunto de datos negativos. Esto indica que se prestó atención a la generación de un conjunto de datos de entrenamiento equilibrado y representativo. Se desarrolló un modelo de red neuronal convolucional densamente conectada (DNPPro) con secuencias de codificación one-hot como entrada. Esto sugiere que se empleó un enfoque de aprendizaje profundo para la predicción de promotores. Y se menciona que el método DNPPro tiene la capacidad de extraer características de secuencias largas en mayor medida en comparación con los métodos de procesamiento de secuencias comúnmente utilizados.

En (Wang, 2023), el objetivo de su trabajo fue el desarrollo de herramientas para el diseño de novo de promotores activos y para la predicción de su fuerza. En el estudio se utilizan dos enfoques, el primero llamado DRSAdesign que se basa en una red generativa y herramientas de predicción de fuerza para el diseño de promotores. DRSAdesign se basa en un proceso de arriba hacia abajo, en el que los promotores generados se clasifican como reales o falsos, seguido de una predicción de su fuerza. Y simultáneamente, se llevó a cabo una generación basada en restricciones de promotores sigma70 para entrenar nuestro modelo supervisado desarrollado, y el modelo se utilizó además para predecir la fuerza de los nuevos promotores generados; este método se le denominó Ndesign. El enfoque del aprendizaje automático fue de clasificación, promotores fuertes y débiles, se entrenó usando RegulonDB ajustando la longitud de promotor a 50pb y se codificaron las características utilizando one-hot lo que mejoró la precisión del modelo además de que en conjunto de validación cruzada en la arquitectura se logró obtener un buen entrenamiento.

Otro trabajo llamado SELECTOR, se llevó a cabo en cinco pasos principales en su desarrollo, primeramente, la recopilación de datos, los cuales se obtuvieron de RegulonDB, que incluyen promotores verificados experimentalmente con tres niveles de anotación, confirmadas, fuertes y débiles. El segundo paso se basa en la codificación en cuatro esquemas, cada uno utilizando estrategias estadísticas exactas como características monocatenarias o frecuencia de aparición. Después se utiliza un esquema eficaz para minimizar la tasa de error de generalización de varios modelos predictivos y ha demostrado eficacia y estabilidad. Posteriormente se utilizaron modelos de clasificación de árboles en 2 niveles pues el predictor SELECTOR, es capaz de realizar tareas de clasificación binaria y de determinación del tipo de promotor específico al que pertenece el promotor predicho. Finalmente se evalúa SELECTOR, un enfoque interpretable, apilado y basado en aprendizaje automático para la predicción precisa de promotores bacterianos y los tipos de promotores específicos a los que pertenecen. SELECTOR utilizó cinco esquemas de codificación de secuencias de ADN diferentes para codificar las secuencias promotoras e integra cinco populares algoritmos de conjunto basados en árboles para construir modelos SELECTOR estables y apilados. La validación cruzada y pruebas independientes demostraron la efectividad de los modelos SELECTOR apilados al superar a los métodos existentes, incluidos MULTiPly, iPromoter-2L, PCSF, vwZ-curve, Stability e iPro54 (Li, 2021).

Otro trabajo de identificación de promotores donde se utiliza una base de datos de UCI Machine Learning Repository que tiene 106 secuencias de ADN, con 57pb cada una. Para el desarrollo de la herramienta se usó Jupyter Notebook Integrated Develoment Enviroment que es un entorno de programación en lenguaje Python. Las secuencias de ADN se enlistaron, se dividieron las secuencias de ADN en nucleótidos individuales, después se convierte el conjunto de datos en pandas Data Frame, además se implementa la codificación one-hot y la división de datos se le da un tamaño de 0.23 de 1. Los resultados se etiquetan de manera binaria donde 1 es “Promotor” y 0 es “No promotor”. Las observaciones que obtuvieron proponen una mayor precisión que otras metodologías existentes, con un 88% de precisión de un clasificador y un 96% de precisión con el uso de red neuronal (Khan, 2020).

Otro trabajo en el que se aplica un modelo computacional se usa una base de datos para el entrenamiento de *E. coli* que se descargaron de RegulonDB, una fuente confiable debido a que las secuencias están validadas experimentalmente. En el estudio se usan 3382 señales promotoras positivas y 3382 negativas, que a su vez la base de daros positiva contenía 1591 señales positivas fuertes y 1792 débiles. En el estudio se implementa la validación cruzada en el proceso de entrenamiento. Debido a que el modelo cuenta con 2 capas se observó en resultados que la primera capa contó con los mejores resultados a comparación de la segunda capa en todas las métricas. Se concluye que el modelo propuesto se evaluó con un conjunto de datos de referencia y superó otros modelos con los que se comparó, tanto como en la identificación del promotor, como en la fuerza (Tayara, 2020).

En un estudio de (Zhu, 2021), se propone un marco computacional llamado Depicter para predecir promotores TATA y no TATA específicos de especies. Se desarrolla en cuatro pasos, primeramente, con la recopilación y el procesamiento de datos de referencia riguroso y objetivo es un paso fundamental para establecer un modelo de predicción robusto, El estudio recopila secuencias promotoras de cuatro especies incluido *Homo sapiens (H. sapiens), Mus musculus (M. musculus), Drosophila melanogaster (D. melanogaster)* y *Arabidopsis thaliana (A. thaliana)* de la base de datos EPDNew que recopila promotores validados experimentalmente. Para el entrenado del modelo se implementó la codificación one-hot para construir matrices de características dispersas y se modela una red neuronal convolucional y la red de cápsulas. La concatenación de CNN y la red cápsula se utilizó para construir el marco de Depicter, que mostró su superioridad en comparación con el marco de conexión de CNN y el mecanismo de atención. orprendentemente, en términos de siete mediciones de desempeño y tres tipos específicos de promotores en las cuatro especies diferentes, hubo un total de 84 resultados de desempeño. Entre ellos, Depicter logró 78 mejores resultados de rendimiento predictivo en comparación con otros métodos existentes.

# Capítulo 3. Objetivos y justificación

## Objetivo General

Predicción de promotores dependientes de factores de transcripción sigma 70 en *E. coli* implementando aprendizaje automático basado en Redes Neuronales con Validación Cruzada.

## Objetivos Específicos

1. Recopilación y curado de un conjunto de datos de secuencias genómicas de *E. coli*, que incluye secuencias de promotores dependientes del factor sigma 70.
    - Recopilación de secuencias de control promotoras y no promotoras, para utilizar como conjunto de entrenamiento y prueba.
2. Desarrollo de una herramienta de aprendizaje automático para la predicción de promotores sigma 70 de *E. coli*.
3. Realizar una predicción de promotores de *E. coli* utilizando el modelo entrenado.

## Justificación

A pesar de la importancia de los promotores, hay una notoria falta de investigación en este campo específico. Hay trabajos realizados anteriormente donde se destaca la necesidad de seguir investigando más a fondo las técnicas de aprendizaje automático para darle una aplicación bioinformática, sin embargo, aún es un área muy desatendida en la búsqueda de secuencias promotoras debido a la poca diversidad de información validada experimentalmente. Es por eso que en este estudio se busca sentar un precedente en la aplicación del aprendizaje automático para expandir este tipo de modelos a otras especies con el objetivo de garantizar resultados positivos que permitan la identificación de promotores para la aplicación en las áreas de la biotecnología.

# Capítulo 4. Métodos y materiales

## Metodología 1. Obtención de los datos

### 1.1 Recopilación de los datos

- Se obtuvieron archivos FASTA que contenían los datos de *E. coli*. Por la gran cantidad de datos ya comprobados de manera experimental se utilizó la cepa K12.

    1. Los datos de *E.coli* K-12, son aquellos más abarcados en investigaciones de esta naturaleza por sus bases de datos de alta confianza.

    2. Se reunió un conjunto de datos de promotores *de E. coli*. Estos sirvieron como datos de entrenamiento.

    3. Los datos para el entrenamiento del algoritmo fueron recabados de la base de datos RegulonDB usada en el trabajo de (Towsey, 2008), encontrados en https://regulondb.ccg.unam.mx/datasets/browser/PromoterSet.

    4. Se descargaron en tabla de datos los promotores sigma 70 de *E. coli* K-12.

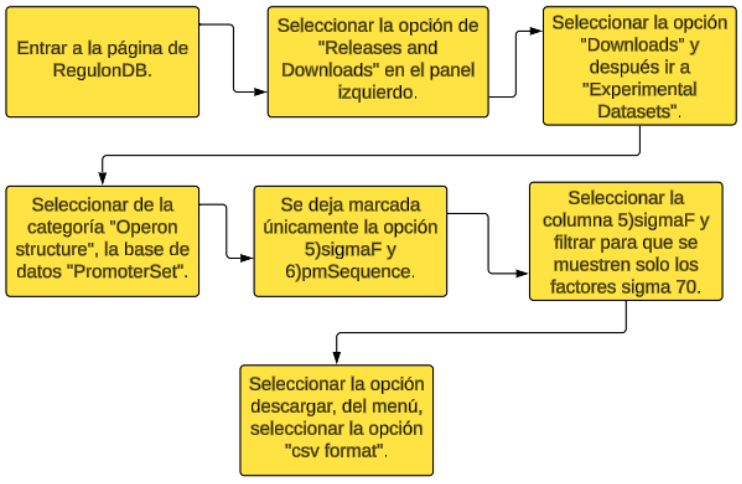



*Imagen 5. Metodología 1.1. Breve descripción del portal RegulonDB*

### 1.2 Preprocesamiento de los datos promotores

1. Se revisó el archivo csv descargado de RegulonDB con el fin de verificar el estado del archivo, distribución de filas y columna con la información correcta.

2. Se revisó toda la información de los datos, cabe mencionar que esta base de datos contiene 55 secuencias promotoras confirmadas, 784 secuencias promotoras fuertes y 1159 secuencias promotoras débiles. Dando un total de 1998 secuencias promotoras con longitud de 81pb. No se encontró alguna inconsistencia, sin embargo, fue necesario darle formato a la tabla con el fin de utilizar el archivo csv y no un archivo fasta, esto debido a la cantidad de datos y a la herramienta que se utilizó para modelar, por lo que fue más factible usar el archivo csv, a este último se agregó una fila para titular la columna con el nombre de “Secuencia” y poder dar instrucciones de que columna debió leer el programa.

3. Los datos negativos fueron extraídos de las secuencias codificantes del genoma de *E. coli* K-12 obtenido de NCBI https://www.ncbi.nlm.nih.gov/nuccore/U00096.2.

4. Se realizó una secuencia de comandos con las instrucciones de generar un archivo fasta, con secuencias generadas aleatoriamente usando como base la parte codificante para alimentar la generación de secuencias, se generaron 1000 secuencias con la misma longitud de pb que los datos positivos.

5. Una vez que se generó el archivo se hizo la revisión del mismo para cerciorarse de que no se haya equivocado el programa.

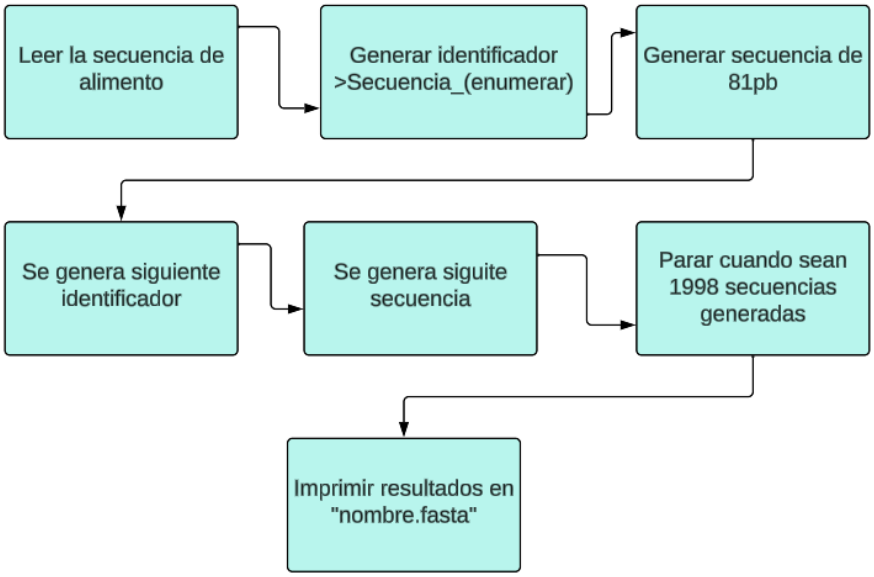



*Imagen 6. Generación de las secuencias promotoras negativas.*

## Metodología 2. Modelo de aprendizaje automático basado en Redes Neuronales con Validación Cruzada

Para la realización del modelo de Red Neuronal hay muchas plataformas de codificación Python en la web. En este trabajo se optó por el uso de GOOGLE COLLAB que es una plataforma de codificación que cuenta con múltiples librerías para el entrenamiento de modelos de aprendizaje automático como sklearn y tensorflow. Además, la plataforma te ofrece la opción de vincular los archivos con el almacenamiento en la nube de Drive por lo que llamar los archivos necesarios desde una carpeta personal fue muy conveniente.

El modelado conllevó diferentes partes que se describen a continuación:

## 2.1 Bibliotecas

1. En la primera parte del modelo se llamaron las bibliotecas que se usaron:

**TABLA 2. Bibliotecas python.**

| Herramienta | Descripción |
|---|---|
| **Tensorflow** | Biblioteca de código abierto para machine learning y deep learning. Permite construir y entrenar modelos de manera eficiente. |
| **Sklearn** | Scikit-learn es una biblioteca para aprendizaje automático y minería de datos que proporciona herramientas simples y eficientes. |
| **Bio** | Biopython es una colección de herramientas para biología computacional, que incluye módulos para bioinformática y genómica. |
| **Pandas** | Biblioteca que ofrece estructuras de datos flexibles y herramientas de análisis de datos para manipulación y limpieza de datos. |
| **Numpy** | Biblioteca fundamental para la computación científica en Python, proporcionando matrices y funciones matemáticas de alto rendimiento. |
| **Pickle** | Módulo para serializar y deserializar objetos de Python, permitiendo almacenar y recuperar estructuras de datos de manera eficiente. |

## 2.2 Procesamiento de los datos

1. Se leyeron los archivos desde una carpeta en Google drive donde anteriormente se habían cargado tantos los datos positivos como los negativos.
2. La lectura se realizó declarando ambos archivos.
    - La base de datos en formato csv se declaró como promotores reales.

- El programa se aseguró de que la primera columna fuera llamada “Secuencia”.
- La base de datos en formato fasta se declaró como promotores falsos.

3. Con estos datos, se crearon dos data frames diferentes, y con esto se evitó el etiquetado incorrecto de datos según fuera promotor o no.
    - En el caso de los promotores positivos se asignó en la segunda columna llamada “Promotor” el número 1, haciendo referencia a dato positivo.
    - En el caso de los promotores falsos se asignó en la columna llamada “Promotor” el número 0, haciendo referencia a dato negativo.
4. Se convirtieron los datos de texto a datos numéricos, convirtiendo los datos de secuencias a enteros.
5. Los datos enteros después se convirtieron en una secuencia de datos enteros de codificación one-hot en 4 dimensiones.

### 2.3 Modelado

1. En esta parte se declararon en **x** los datos para el entrenamiento y para la prueba. Y se declararon en **y** los datos binarios con los que se etiquetaron a promotores positivos como 1 y falsos como 0.

2. Se inició un modelo secuencial vacío.

3. Después se fueron añadiendo capas al modelo de la siguiente manera:
    - La primera capa densa consta de 256 nodos y se usó una función Lineal.
    - Después de esta capa se añadió un Dropout de 0.5 para que apague el 50% de los nodos de la capa anterior con el fin de evitar el sobre ajuste.
    - Después de esto se concatenaron las capas con los dropout con valor de 0.5, las capas siguientes fueron: 128 nodos, y 64 nodos.
    - Al final se añadió una última capa de un solo nodo y con activación sigmoide. Como resultado produce una salida binaria y por esto la

función sigmoide, donde la salida representa la probabilidad de pertenecer a una clase binaria.

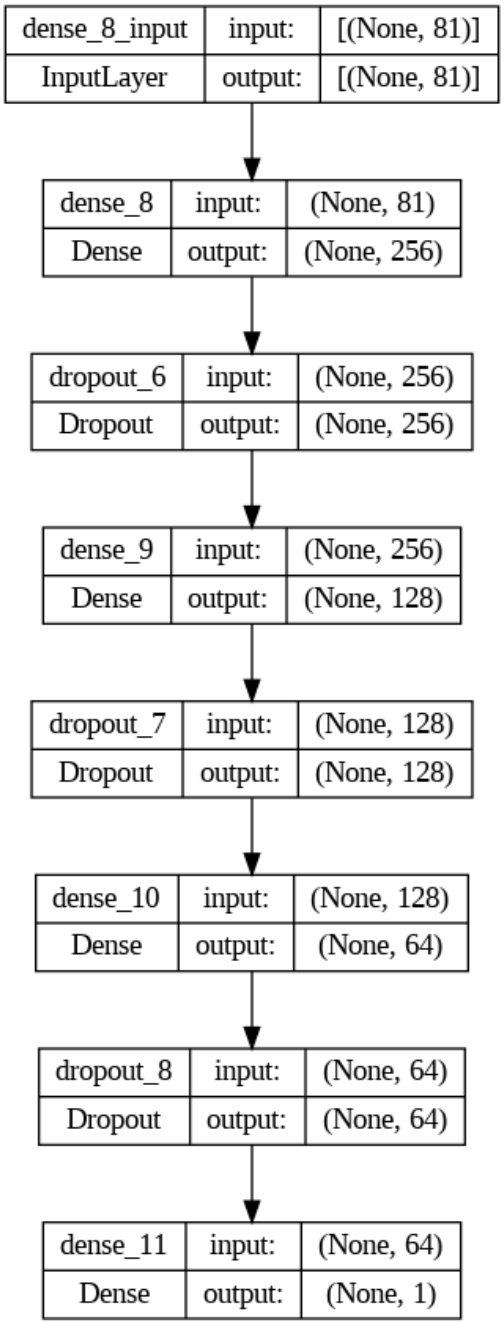



*Imagen 7. Arquitectura del modelo de Redes neuronales.*

4. Se compiló el modelo y se entrenó con 131 épocas, y con un conjunto de datos que representa el 67% para entrenamiento y un 33% para la prueba.

5. Se configuró la validación cruzada con 5 pliegues (k-folds) y se realizó entrenando el modelo en cada pliegue y evaluando en el conjunto de prueba de ese pliegue.
    - 108 épocas.
    - 35% para prueba y 65% para entrenamiento.

6. Al final se imprimieron los resultados de precisión.

7. Se guardó el modelo en formato h5 para su posterior aplicación en la siguiente metodología.

## Metodología 3. Predicción y aplicación del modelo

### 3.1 Algoritmo para la predicción

Para el trabajo de predicción se necesitaron de los siguientes requisitos:

**TABLA 3. Requisitos para la predicción.**

| REQUISITO | OBTENCIÓN | ESTADO |
|---|---|---|
| **MODELO .h5** | Este es el modelo de redes neuronales que se entrenó con secuencias promotoras y no promotoras con validación cruzada. | Obtenido |
| **TOKENIZER .h5** | Este es un archivo que ayudó a configurar los datos que usaremos para la predicción para que se adapten a la estructura del modelo y los datos con los que se entrenaron. | Obtenido |
| **SECUENCIAS PARA PREDICCIÓN ARCHIVO FASTA** | Se usaron secuencias del entrenamiento para la validación de la predicción y algunas secuencias de otras especies. | Obtenido |
| **EQUIPO PARA PROCESAMIENTO DE LOS DATOS (COMPUTADORA, SISTEMA LINUX)** | Para la parte de las predicciones se requirió un equipo con las especificaciones y capacidades que se requieren para la programación de los códigos que se necesitan para emplear el modelo. | Obtenido |
| **CÓDIGO PYTHON PARA LA PREDICCIÓN** | Finalmente se realizó un código con comandos para el procesamiento de datos de entrada, modelo y salida de resultados. | Obtenido |

**Para el código de predicción:**

1. Se utilizaron las librerías necesarias para el procesamiento de archivos y datos (mismas librerías usadas en el entrenamiento del modelo).

2. Se declararon tanto el modelo h5 como el tokenizer h5 con los nombres “model” y “tokenizer” respectivamente para ser llamados al programa desde la ruta de los archivos.

3. Se declaró el archivo fasta de prueba con las secuencias a predecir como “exseq.fasta” para ser llamado al programa desde la ruta de los archivos.

4. Se realizó una iteración de las secuencias o secuencia en el archivo a predecir, extrayendo la secuencia y leyendo las primeras 81pb (este es el número de pares de bases que tienen los datos de entrenamiento).

5. Se ajustó un umbral de 0.8 para discriminar las predicciones que tengan menos del 0.8 de probabilidades de que fueran un promotor.

6. Se enumeraron los resultados en un archivo txt con la información:

    - Probabilidad de ser promotor: “valor”.
    - Probabilidad de no ser promotor: “valor”
    - Además, añadió 1 para secuencias probablemente promotoras y 0 a las no promotoras.

**Para los datos a predecir:**

Este fue un archivo fasta que contenía varias secuencias, las cuales se ordenan a continuación:

**TABLA 4. Secuencias para la predicción.**

| No. de secuencia | Secuencia | Promoto |
|---|---|---|
| 1 | AGGTACTTACGTACCTTACCTCGGCCTACTCTCGTAATCGACTTAACCACGTCCGACTAGTACGTGCTCACAGTCTCTTT | SI |
| 2 | tgccaactggcaggtcaaccgaatgcagacatcgcaggcgggatgtgtcagcatcagcgtTacgctagtttcacccggggg | SI |
| 3 | tttgttgcgatttagcgcgcaaatctttacttatttacagaacttcggcattatcttgccGgttcaaattacggtagtgat | SI |
| 4 | ttttcaccacaagaatgaatgttttcggcacatttctccccagagtgttataattgcggtCgcagagttggttacgctcat | SI |
| 5 | TGTGCACACACATGTGTACACACACACATGTGTGAGAGAGAGGAGAGAGGAGAGAGGAGAGAGAGAGTATATGGAGGAGAG | NO |
| 6 | GCAATTGAAAACTTTCGTCGATCAGGAATTTGCCCAAATAAAACATGTCCTGCATGGCATTAGTTTGTTGGGGCAGTGCCC | NO |
| 7 | CAGTTTCTGCGTTCCACAAAGCGACTGTGTGCGAGCTGAACGGGCAATGCAGGAAGAGTTCTACCTGGAACTGAAAGAAGG | NO |
| 8 | ATTCGCCTCGTGAAAGAATATCATCTGCTGAACCCGGTCATTGTTGACTGCACTTCCAGCCAGGCAGTGGCGGATCAATAT | NO |
| 9 | ACAACCATGCGAGTGTTGAAGTTCGGCGGTACATCAGTGGCAAATGCAGAACGTTTTCTGCGTGTTGCCGATATTCTGGAA | NO |
| 10 | CTACTTCGGCGCTAAAGTTCTTCACCCCCGCACCATTACCCCCATCGCCCAGTTCCAGATCCCTTGCCTGATTAAAAATAC | NO |
| 11 | GTCACGCGCCCGTATTTCCGTGGTGCTGATTACGCAATCATCTTCCGAATACAGCATCAGTTTCTGCGTTCCACAAAGCGA | NO |
| 12 | tgcgttttcgctaatagttgacagatttatccgctccatcgcgggcggataatctcccctTccccaactttcttcgtaacg | SI |
| 13 | ccaattgccagcttaagtcgaaacaaggagactcgatatttaaatcggattacattttaaCtttagtaatattcttcagag | SI |
| 14 | atttaactcactaaagttaagaagattgaaaagtcttaaacatattttcagaataatcggAtttatatgtttgaaaattat | SI |

| | | |
|---|---|---|
| **15** | atttattgaccgtctaaatgagagtttttgatatataactacagacagctactataacttTcatctatttattcacagcgc | SI |
| **16** | aactaaaccgtggcacaaatgggcaatttatccatcggtaaaatactataaaatagctttAgaaaattcccccctggaaaga | SI |
| **17** | cgcgttaaatgctgaatctttacgcatttctcaaaccctgaaatcactgtatactttaccAgtgttgagaggtgagcaatg | SI |
| **18** | TTGGGCGAACGACGGGAATTGAACCCGCGCGTGGTGGATTCACAATCCACTGCCTTGATCCACTTGGCTACATCCGCCCCG | PRUEBA MICROALGA |
| **19** | AGATGAGTAAAAAAAAAAAATAAAGTGAAACGCTCACAACTATCAATTGTGATAATTGATATTGGTGTTAAATCAATGAAT | PRUEBA MICROALGA |
| **20** | TACCTTTGGTATTTATACCGCTACTCGAAATAAAATTTTTTTTATAAAAAATTTTATCTTTAAAAATTACATTTTTTTGGATACGAGCAATGTAGGAAGGTAACATTCTTATCCCATAAGAATTAAAGTAATTCATTGTTTATTTTATAT | PRUEBA MICROALGA |

# PIPELINE: Resumen de las Metodologías

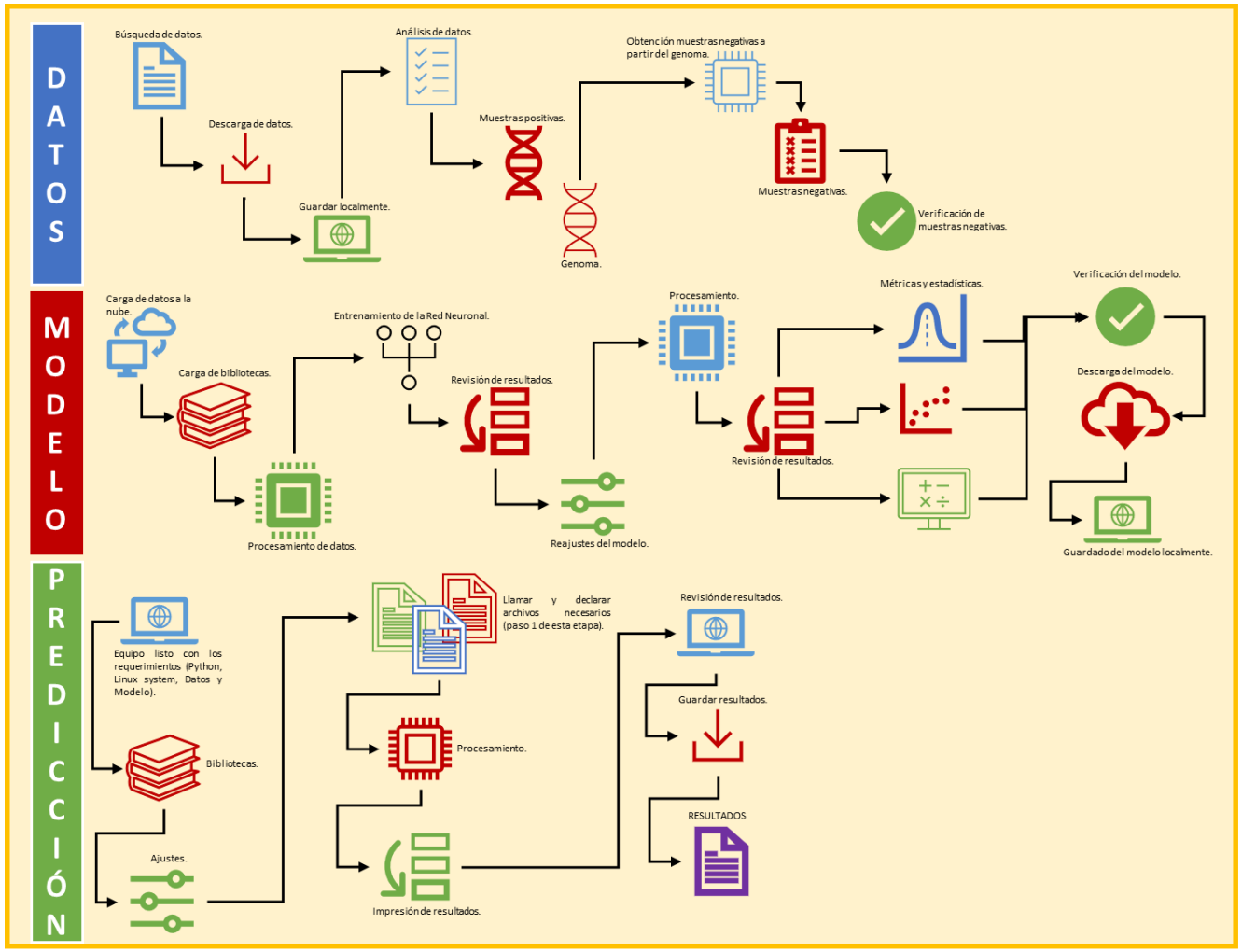



*Imagen 8. Pipeline del desarrollo del proyecto donde se muestran las tres etapas implementadas.*

# Capítulo 5. Resultados

## Resultados del Modelo de Redes Neuronales Sencillas

El modelo en un primer entrenamiento se obtuvo una precisión de entrenamiento de 0.838 y una precisión de prueba de 0.759.

En la siguiente figura se muestra una pendiente de precisión con el comportamiento según las épocas:

Pendiente de precisión del modelo sin la validación cruzada:

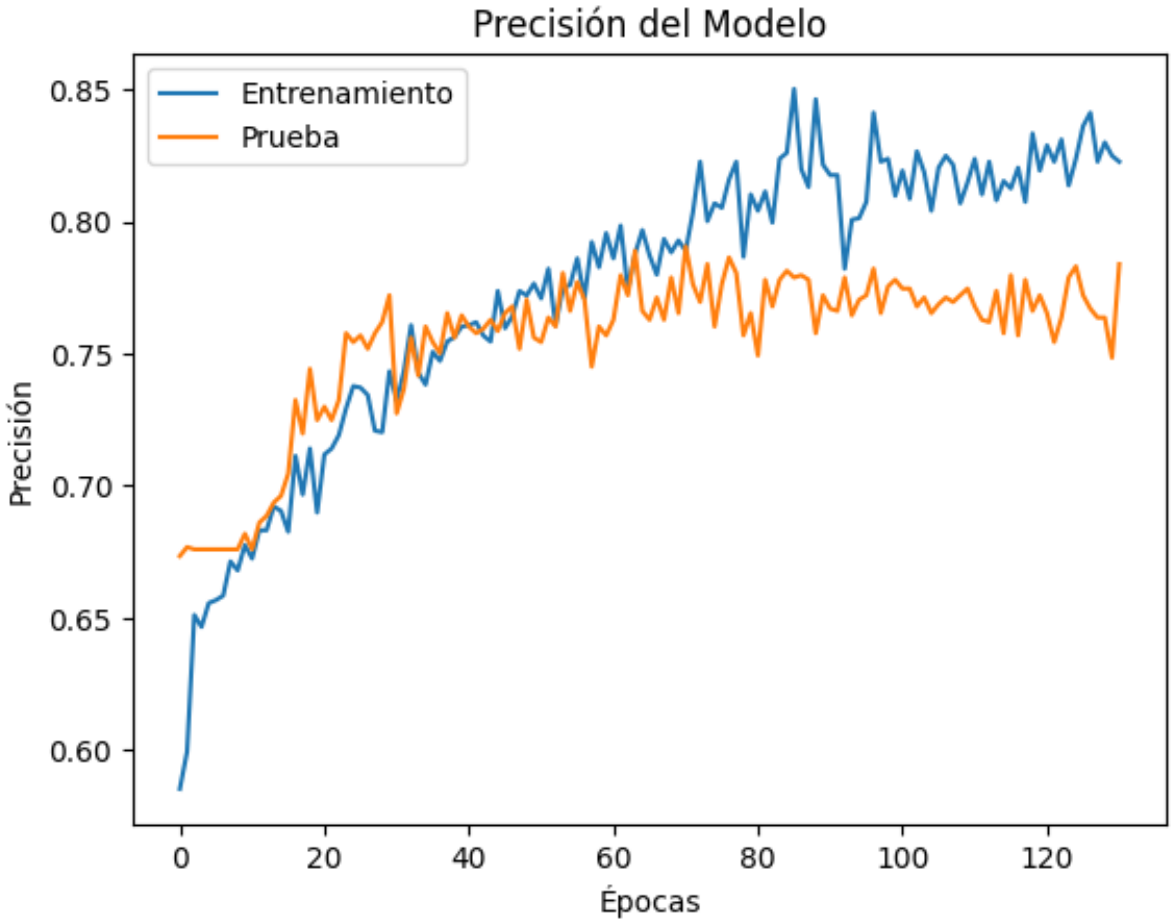



*Imagen 9. Pendiente del comportamiento del modelo en entrenamiento-prueba.*

Gráfico ROC y AUC del modelo:

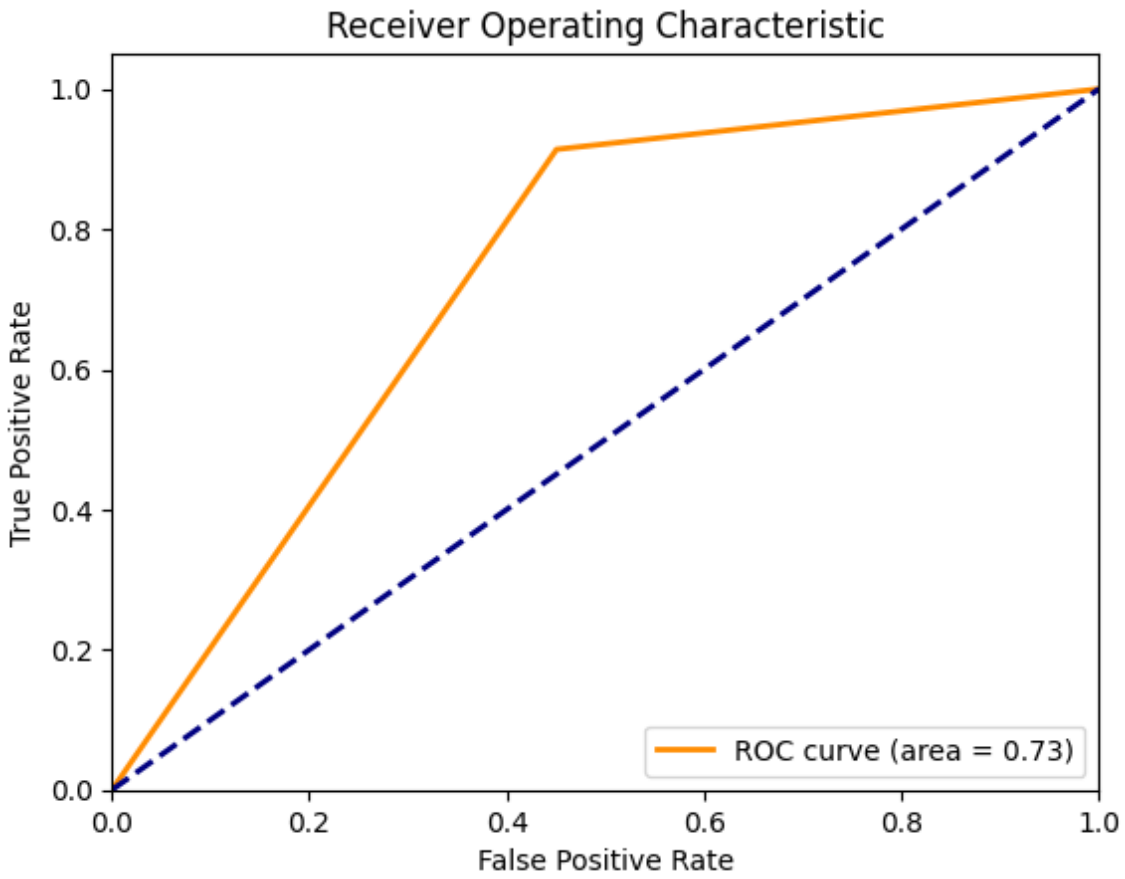



*Imagen 10. Gráfico de ROC y su AUC.*

**Matriz de confusión y métricas**

Se realizaron las siguientes métricas de rendimiento:

**TABLA 5. Distribución de la matriz de confusión.**

| | |
|---|---|
| TN: Número que representa las muestras negativas que fueron correctamente clasificadas como negativas. | FP: Número de muestras negativas que fueron incorrectamente clasificadas. |
| FN: Número de muestras positivas que fueron incorrectamente clasificadas. | TP: Número de muestras positivas que fueron correctamente clasificadas como positivas. |

Matriz de confusión del modelo:

$$\begin{bmatrix} 211 & 173 \\ 69 & 732 \end{bmatrix}$$

Precisión:

$$P = \frac{TP}{TP + FP} = \frac{732}{732 + 173} = \frac{211}{280} = 0.808$$

Exactitud:

$$A = \frac{TP + TN}{TP + FN + TN + FP} = \frac{732 + 211}{732 + 211 + 173 + 69}$$

$$A = \frac{943}{1185} = 0.795$$

Sensibilidad:

$$R = \frac{TP}{TP + FN} = \frac{732}{732 + 69} = \frac{732}{801} = 0.913$$

F1-Score:

$$F = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad}$$

$$F = 2 * \frac{0.808 * 0.913}{0.808 + 0.913} = 2 * \frac{0.732}{1.72}$$

$$F = 2 * 0.425 = 0.851$$

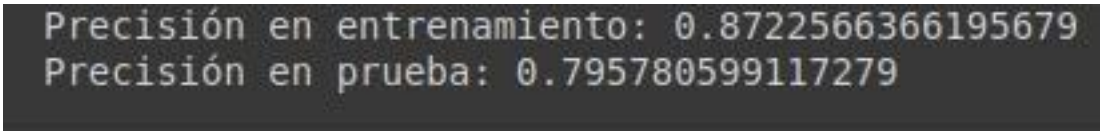



*Imagen 11. Resultados del modelo de Redes Neuronales Sencillas.*

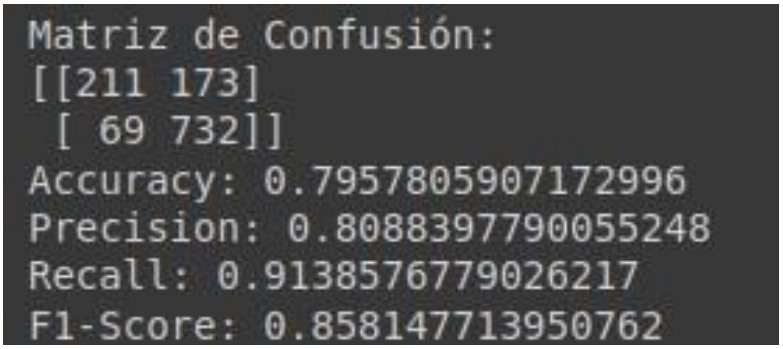



*Imagen 12. Resultados de las métricas del modelo de Redes Neuronales Sencillas.*

## Resultados del Modelo de Redes Neuronales con Validación Cruzada

Para favorecer el rendimiento del modelo, se trabajó una validación cruzada en 5 pliegues de la que se obtuvo una precisión media de 0.9319, a continuación, se muestra una gráfica de la pendiente de comportamiento global del entrenamiento y prueba:

Pendiente de precisión del modelo con la validación cruzada:

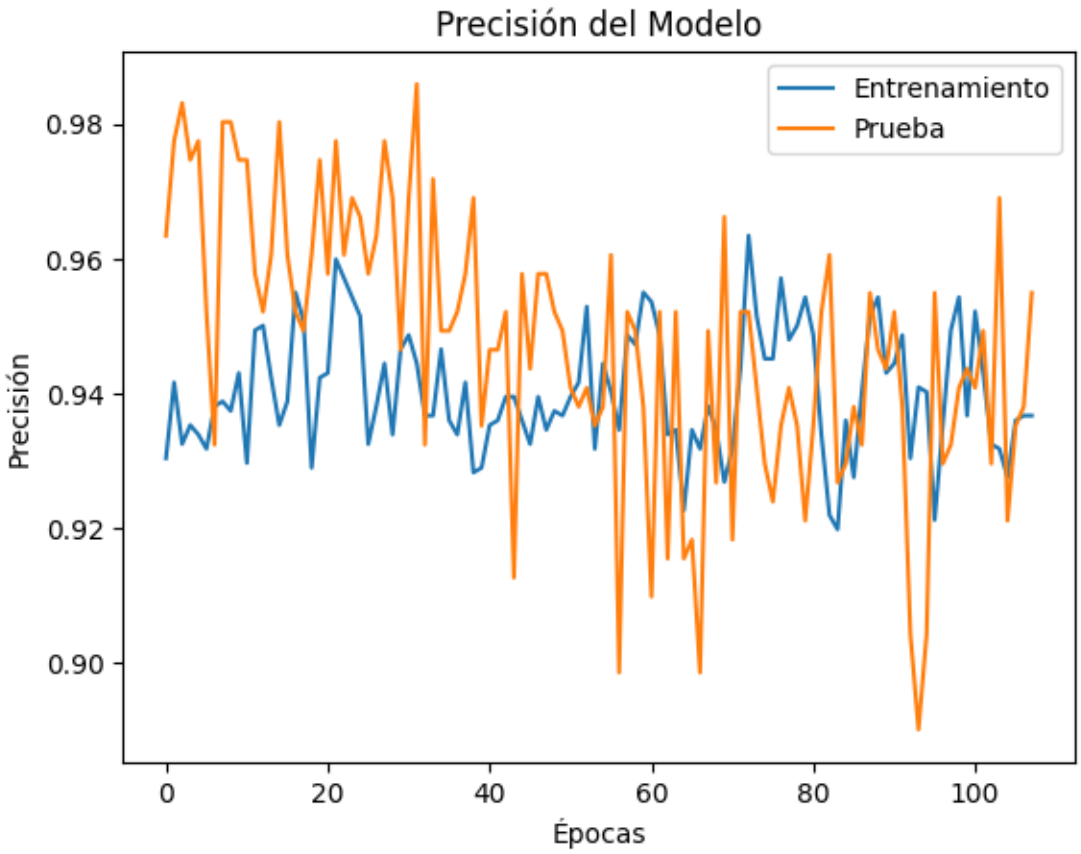



*Imagen 13. Comportamiento del modelo durante el entrenamiento y prueba.*

**Matriz de confusión y métricas**

También se desarrollaron las métricas, se muestran a continuación:

Matriz de confusión del pliegue 1:

$$\begin{bmatrix} 82 & 34 \\ 4 & 236 \end{bmatrix}$$

Matriz de confusión del pliegue 2:

$$\begin{bmatrix} 93 & 24 \\ 1 & 238 \end{bmatrix}$$

Matriz de confusión del pliegue 3:

$$\begin{bmatrix} 96 & 22 \\ 0 & 239 \end{bmatrix}$$

Matriz de confusión del pliegue 4:

$$\begin{bmatrix} 94 & 22 \\ 0 & 239 \end{bmatrix}$$

Matriz de confusión del pliegue 5:

$$\begin{bmatrix} 101 & 15 \\ 1 & 238 \end{bmatrix}$$

Promedio de las matrices:

$$Prom = \begin{bmatrix} 82 & 34 \\ 4 & 236 \end{bmatrix} + \begin{bmatrix} 93 & 24 \\ 1 & 238 \end{bmatrix} + \begin{bmatrix} 96 & 22 \\ 0 & 239 \end{bmatrix} + \begin{bmatrix} 94 & 22 \\ 0 & 239 \end{bmatrix} + \begin{bmatrix} 101 & 15 \\ 1 & 238 \end{bmatrix}$$

$$Prom = \begin{bmatrix} 82 + 93 + 96 + 94 + 101 & 34 + 24 + 22 + 22 + 15 \\ 4 + 1 + 0 + 0 + 1 & 236 + 238 + 239 + 239 + 238 \end{bmatrix}$$

$$Prom = \frac{\begin{bmatrix} 466 & 117 \\ 6 & 1190 \end{bmatrix}}{5} = \begin{bmatrix} \frac{466}{5} & \frac{117}{5} \\ \frac{6}{5} & \frac{1190}{5} \end{bmatrix}$$

Precisión media:

$$P = \frac{TP}{TP + FP} = \frac{\frac{1190}{5}}{\frac{1190}{5} + \frac{117}{5}} = \frac{238}{\frac{1307}{5}} = 0.910$$

Exactitud:

$$A = \frac{TP + TN}{TP + FN + TN + FP} = \frac{\frac{1190}{5} + \frac{466}{5}}{\frac{1190}{5} + \frac{466}{5} + \frac{117}{5} + \frac{6}{5}}$$

$$A = \frac{\frac{1656}{5}}{\frac{1779}{5}} = 0.930$$

Sensibilidad media:

$$R = \frac{TP}{TP + FN} = \frac{\frac{1190}{5}}{\frac{1190}{5} + \frac{6}{5}} = \frac{\frac{1190}{5}}{\frac{1196}{5}} = 0.995$$

F1-Score media:

$$F = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad}$$

$$F = 2 * \frac{0.912 * 0.931}{0.912 + 0.931} = 2 * \frac{0.849}{1.843}$$

$$F = 2 * 0.460 = 0.921$$

Resultados de las matrices:

**TABLA 6. Resultados de las matrices de confusión de cada pliegue.**

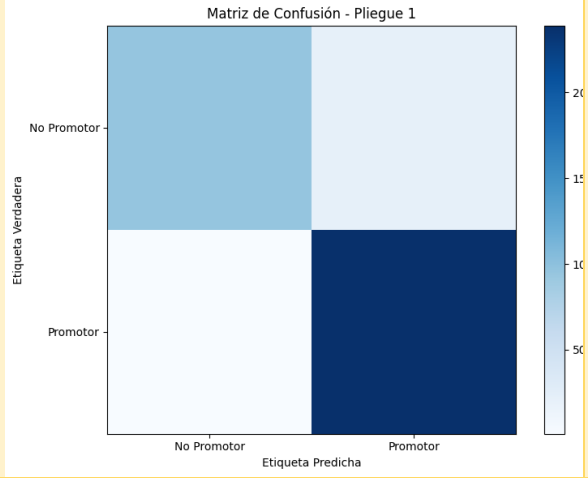
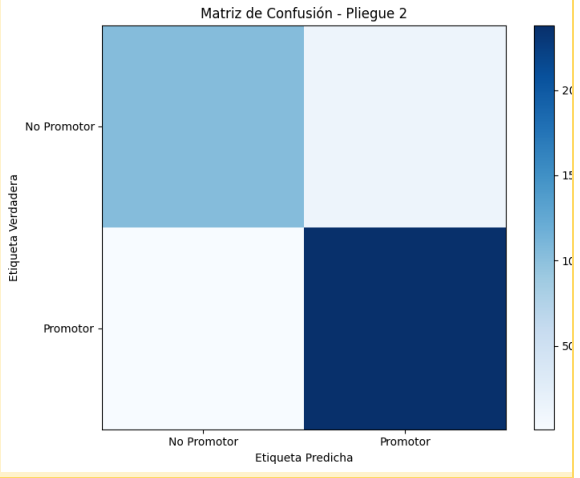
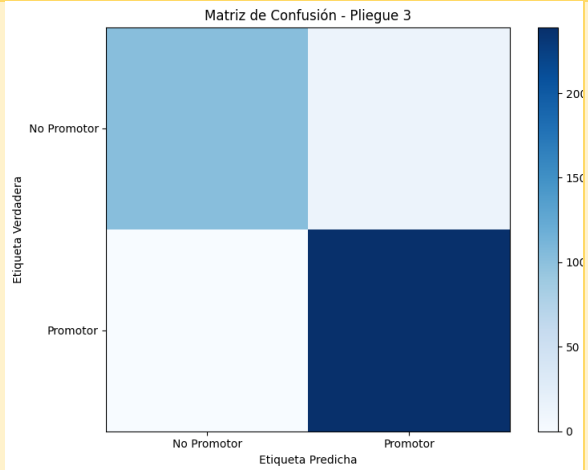
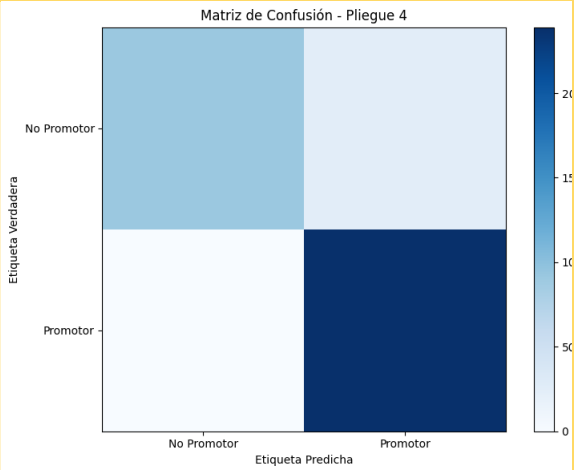
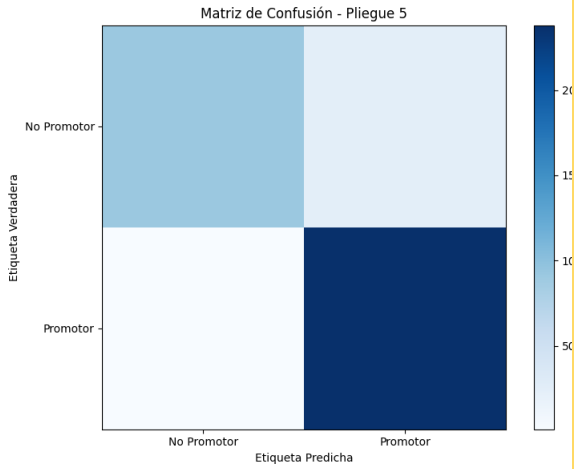
| | |
|---|---|
| 1 | 2 |
|  |  |
| 3 | 4 |
|  |  |
| 5 |  |

Gráfico ROC y AUC del modelo:

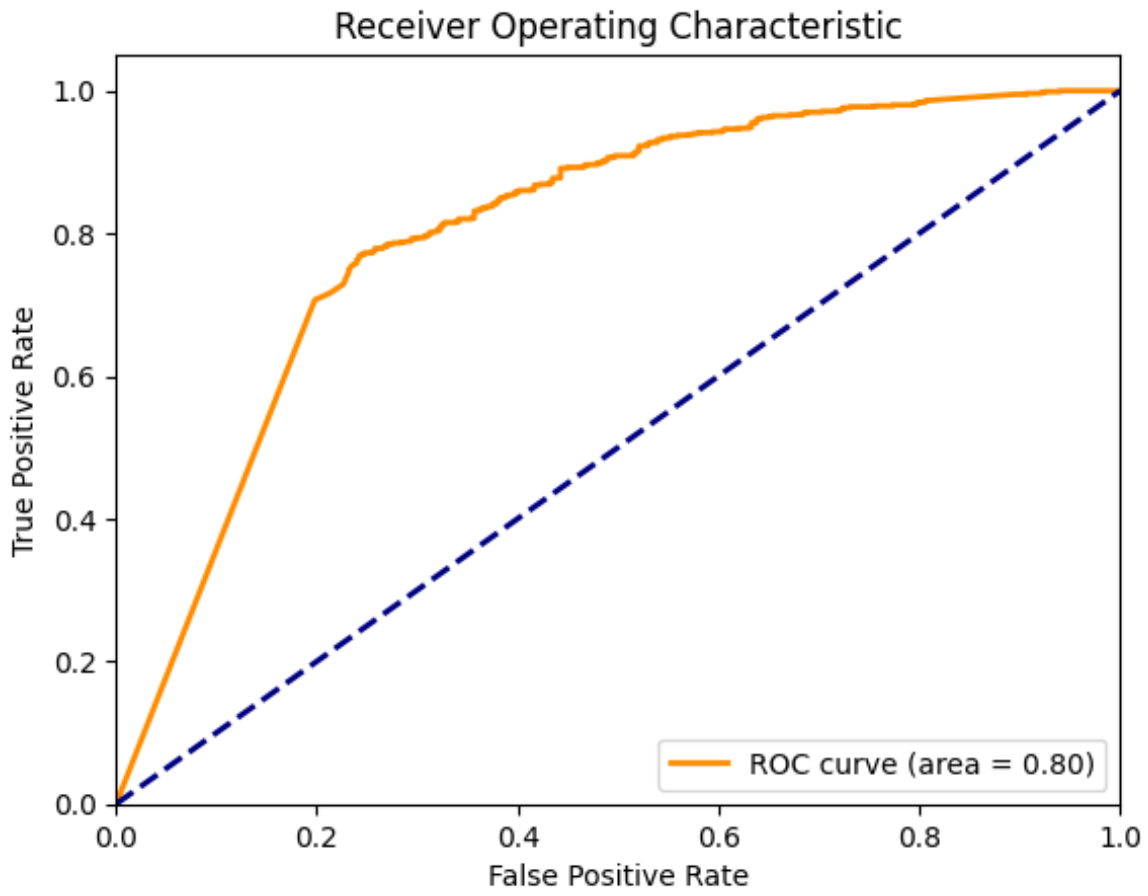



*Imagen 14. Gráfico de ROC y su AUC.*

| | Secuencia | Promotor |
|---|---|---|
| 0 | acactttcattgttttaccgttgctctgattaattgacgctaaagt... | 1 |
| 1 | tgccaactggcaggtcaaccgaatgcagacatcgcaggcgggatgt... | 1 |
| 2 | tttgttgcgatttagcgcgcaaatctttacttatttacagaacttc... | 1 |
| 3 | ttttcaccacaagaatgaatgttttcggcacatttctccccagagt... | 1 |
| 4 | agggaaaaaaataaaatttagtgctgtacagagcgcgttacaacac... | 1 |

Precisión media de la validación cruzada: 0.8818

*Imagen 15.Primer resultado del modelo de Redes Neuronales con validación cruzada.*

```
12/12 [==============================] - 0s 3ms/step
Matriz de Confusión - Pliegue 1:
[[ 82  34]
 [  4 236]]

12/12 [==============================] - 0s 2ms/step
Matriz de Confusión - Pliegue 2:
[[ 93  24]
 [  1 238]]

12/12 [==============================] - 0s 2ms/step
Matriz de Confusión - Pliegue 3:
[[ 96  20]
 [  0 239]]

12/12 [==============================] - 0s 2ms/step
Matriz de Confusión - Pliegue 4:
[[ 94  22]
 [  0 239]]

12/12 [==============================] - 0s 2ms/step
Matriz de Confusión - Pliegue 5:
[[101  15]
 [  1 238]]
```

*Imagen 16. Matrices de confusión de los 5 pliegues de la validación cruzada.*

```
Precisión media de la validación cruzada: 0.9319
Precision media de la validación cruzada: 0.9123
Recall media de la validación cruzada: 0.9950
F1-Score media de la validación cruzada: 0.9518
```

*Imagen 17. Resultados medios de las métricas del modelo de Redes Neuronales con Validación cruzada.*

## RESULTADO DE LAS PREDICCIONES

### Modelo sencillo de Redes Neuronales

**TABLA 7. Resultados de la predicción del modelo de Redes Neuronales Sencillas.**

| | | |
|---|---|---|
| **SECUENCIA 1** | AGGTACTTACGTACCTTACCTCGGCCTACTCTCGTAATCGACTTAACCACGTCCGACTAGTACGTGCTCACAGTCTCTTT | **ES PROM** |
| **Prob. Es promotor** | 0.4756 | SI |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 2** | TgccaactggcaggtcaaccgaatgcagacatcgcaggcgggatgtgtcagcatcagcgtTacgctagtttcacccggggg | |
| **Prob. Es promotor** | 0.6567 | SI |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 3** | TttgttgcgatttagcgcgcaaatctttacttatttacagaacttcggcattatcttgccGgttcaaattacggtagtgat | |
| **Prob. Es promotor** | 0.6477 | SI |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 4** | TtttcaccacaagaatgaatgttttcggcacatttctccccagagtgttataattgcggtCgcagagttggttacgctcat | |
| **Prob. Es promotor** | 1.0000 | SI |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 5** | TGTGCACACACATGTGTACACACACACATGTGTGAGAGAGAGGAGAGAGGAGAGAGGAGAGAGAGAGTATATGGAGGAGAG | |
| **Prob. Es promotor** | 0.6558 | NO |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 6** | GCAATIGAAAACTTTCGTCGATCAGGAATTTGCCCAAATAAAACATGTCCTGCATGGCATTAGTTTGTTGGGGCAGTGCCC | |

| | | |
|---|---|---|
| **Prob. Es promotor** | 0.1745 | NO |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 7** | CAGTTTCTGCGTTCCACAAAGCGACTGTGTGCG AGCTGAACGGGCAAT GCAGGAAGAGTICTACCTGGA ACTGAAAGAAGG | NO |
| **Prob. Es promotor** | 0.3495 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 8** | ATTCGCCTCGTGAAAGAATATCATCTGCTGAACCC GGTCATTGTTGACTGCACTTCCAGCCAGGCAGTG GCGGATCAATAT | NO |
| **Prob. Es promotor** | 0.6545 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 9** | ACAACCATGCGAGTGTTGAAGTTCGGCGGT ACATCAGTGGCAAATGCAGAACGTTTTCTGC GTGTTGCCGATATTCTGGAA | NO |
| **Prob. Es promotor** | 0.6725 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 10** | CTACTTCGGCGCTAAAGTTCTTCACCCCCGC ACCATTACCCCCATCGCCCAGTTCCAGATCC CTTGCCTGATTAAAAATAC | NO |
| **Prob. Es promotor** | 0.3039 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 11** | GTCACGCGCCCGTATTTCCGTGGTGCTGA TTACGCAATCATCTTCCGAATACAGCATCAG TTTCTGCGTTCCACAAAGCGA | NO |
| **Prob. Es promotor** | 0.6634 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 12** | Tgcgttttcgctaatagttgacagatttatccgctccatcgcgggcgg ataatctcccctTccccaactttcttcgtaacg | |

| | | |
|---|---|---|
| **Prob. Es promotor** | 0.9159 | SI |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 13** | Ccaattgccagcttaagtcgaaacaaggagactogatatttaaa tcggattacattttaaCtttagtaatattcttcagag | SI |
| **Prob. Es promotor** | 0.9971 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 14** | Atttaactcactaaagttaagaagattgaaaagtcttaaacatattttca gaataatcggAtttatatgtttgaaaattat | SI |
| **Prob. Es promotor** | 0.6646 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 15** | Atttattgaccgtctaaatgagagtttttgatatataactacaga cagctactataacttTcatctatttattcacagcgc | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 16** | Aactaaaccgtggcacaaatgggcaatttatccatcggtaaaata ctataaaatagctttAgaaaattccccctggaaaga | SI |
| **Prob. Es promotor** | 0.9871 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 17** | Cgcgttaaatgctgaatctttacgcatttctcaaaccctgaaatcactg tatactttaccAgtgttgagaggtgagcaatg | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 18** | TIGGGCGAACGACGGGAATIGAACCCGCGC GTGGTGGATICACAATCCACTGCCTTGATCC ACTTGGCTACATCCGCCCCG | Microalga |
| **Prob. Es promotor** | 0.5306 | |
| **Pred. Binaria** | No promotor | |

| | | |
|---|---|---|
| **SECUENCIA 19** | AGATGAGTAAAAAAAAAAAATAAAGTGAAAC GCTCACAACTATCAATTGTGATAATTGATATT GGTGTTAAATCAATGAAT | Microalga |
| **Prob. Es promotor** | 0.9792 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 20** | TACCTTTGGTATTTATACCGCTACTCGAAATAAA ATTTTTTTATAAAAAATTTTATCTTTAAAAATTACA TTTTTTTGGATACGAGCAATGTAGGAAGGTAAC ATTCTTATCCCATAAGAATTAAAGTAATTCATTGT TTATTTTATAT | Microalga |
| **Prob. Es promotor** | 0.3333 | |
| **Pred. Binaria** | No promotor | |

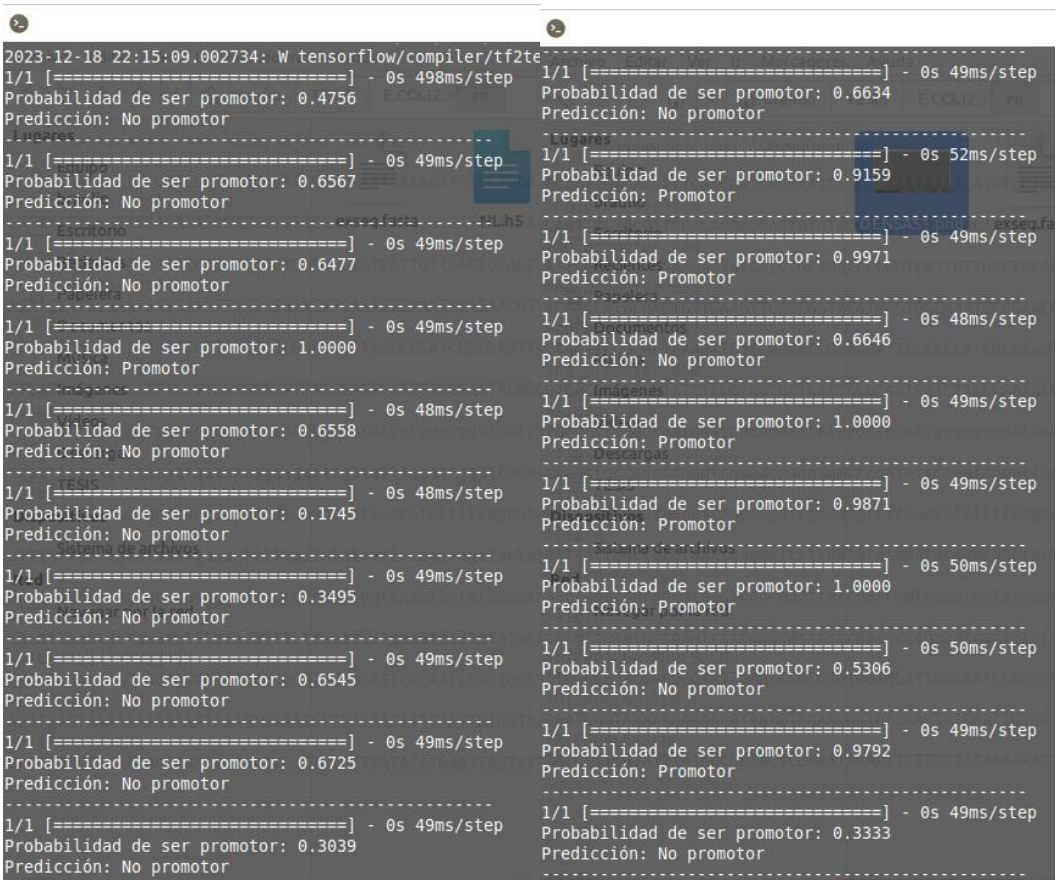



*Imagen 18. Predicciones de las secuencias usadas.*

**Modelo de Redes Neuronales con Validación cruzada**

**TABLA 8. Resultados de la predicción del modelo de Redes Neuronales con Validación Cruzada.**

| | | |
|---|---|---|
| **SECUENCIA 1** | AGGTACTTACGTACCTTACCTCGGCCTACTCTCGTA ATCGACTTAACCACGTCCGACTAGTACGTGCTCACA GTCTCTTT | **ES PROM** |
| **Prob. Es promotor** | 0.9979 | SI |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 2** | Tgccaactggcaggtcaaccgaatgcagacatcgcaggcggga tgtgtcagcatcagcgtTacgctagtttcacccggggg | SI |
| **Prob. Es promotor** | 0.9387 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 3** | Tttgttgcgatttagcgcgcaaatctttacttatttacagaacttcggca ttatcttgccGgttcaaattacggtagtgat | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 4** | Ttttcaccacaagaatgaatgttttcggcacatttctccccagagtgttataa ttgcggtCgcagagttggttacgctcat | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 5** | TGTGCACACACATGTGTACACACACACATGT GTGAGAGAGAGGAGAGAGGAGAGAGGAGAGA GAGAGTATATGGAGGAGAG | NO |
| **Prob. Es promotor** | 0.3204 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 6** | GCAATIGAAAACTTTCGTCGATCAGGAATTTGCC CAAATAAAACATGTCCTGCATGGCATTAGTTTGTT GGGGCAGTGCCC | NO |
| **Prob. Es promotor** | 0.3199 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 7** | CAGTTTCTGCGTTCCACAAAGCGACTGTGTGCG AGCTGAACGGGCAATGCAGGAAGAGTICTACCTG | |

| | | |
|---|---|---|
| | GAACTGAAAGAAGG | NO |
| **Prob. Es promotor** | 0.2653 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 8** | ATTCGCCTCGTGAAAGAATATCATCTGCTGAACCC GGTCATTGTTGACTGCACTTCCAGCCAGGCAGTG GCGGATCAATAT | NO |
| **Prob. Es promotor** | 0.9818 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 9** | ACAACCATGCGAGTGTTGAAGTTCGGCGGT ACATCAGTGGCAAATGCAGAACGTTTTCTGC GTGTTGCCGATATTCTGGAA | NO |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 10** | CTACTTCGGCGCTAAAGTTCTTCACCCCCGC ACCATTACCCCCATCGCCCAGTTCCAGATCC CTTGCCTGATTAAAAATAC | NO |
| **Prob. Es promotor** | 0.3199 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 11** | GTCACGCGCCCGTATTTCCGTGGTGCTGA TTACGCAATCATCTTCCGAATACAGCATCAG TTTCTGCGTTCCACAAAGCGA | NO |
| **Prob. Es promotor** | 0.3199 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 12** | Tgcgttttcgctaatagttgacagatttatccgctccatcgcgggcgg ataatctcccctTccccaactttcttcgtaacg | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 13** | Ccaattgccagcttaagtcgaaacaaggagactogatatttaaa tcggattacattttaaCtttagtaatattcttcagag | SI |
| **Prob. Es promotor** | 0.3199 | |
| **Pred. Binaria** | No promotor | |
| **SECUENCIA 14** | Atttaactcactaaagttaagaagattgaaaagtcttaaacatattttca gaataatcggAtttatatgtttgaaaattat | |

| | | |
|---|---|---|
| **Prob. Es promotor** | 1.0000 | SI |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 15** | Atttattgaccgtctaaatgagagtttttgatatataactacaga cagctactataacttTcatctatttattcacagcgc | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 16** | Aactaaaccgtggcacaaatgggcaatttatccatcggtaaaata ctataaaatagctttAgaaaattccccctggaaaga | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 17** | Cgcgttaaatgctgaatctttacgcatttctcaaaccctgaaatcactg tatactttaccAgtgttgagaggtgagcaatg | SI |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 18** | TIGGGCGAACGACGGGAATIGAACCCGCGC GTGGTGGATICACAATCCACTGCCTTGATCC ACTTGGCTACATCCGCCCCG | Microalga |
| **Prob. Es promotor** | 0.9734 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 19** | AGATGAGTAAAAAAAAAAAATAAAGTGAAAC GCTCACAACTATCAATTGTGATAATTGATATT GGTGTTAAATCAATGAAT | Microalga |
| **Prob. Es promotor** | 1.0000 | |
| **Pred. Binaria** | Promotor | |
| **SECUENCIA 20** | TACCTTTGGTATTTATACCGCTACTCGAAATAAA ATTTTTTTATAAAAAATTTTATCTTTAAAAATTACA TTTTTTTGGATACGAGCAATGTAGGAAGGTAAC ATTCTTATCCCATAAGAATTAAAGTAATTCATTGT TTATTTTATAT | Microalga |
| **Prob. Es promotor** | 0.8405 | |
| **Pred. Binaria** | Promotor | |

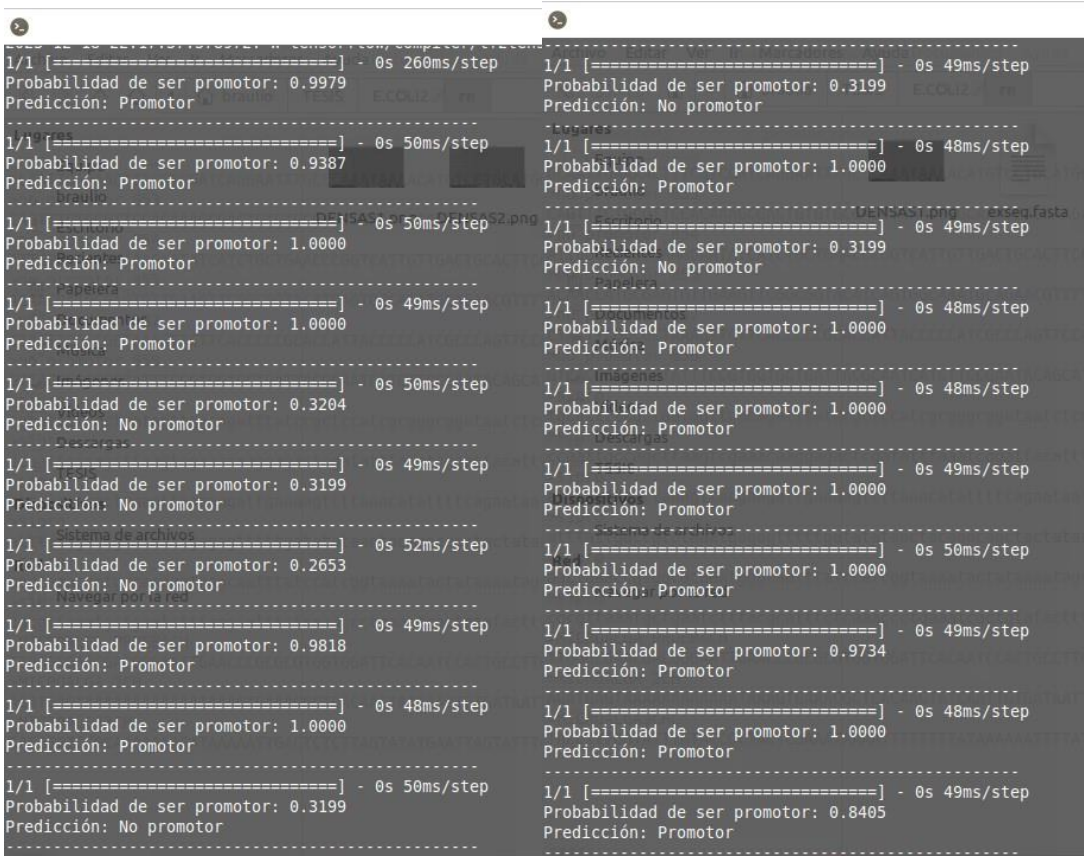



*Imagen 19. Predicciones realizadas por el Modelo de Redes Neuronales con Validación Cruzada.*

## COMPARATIVA DEL MODELO

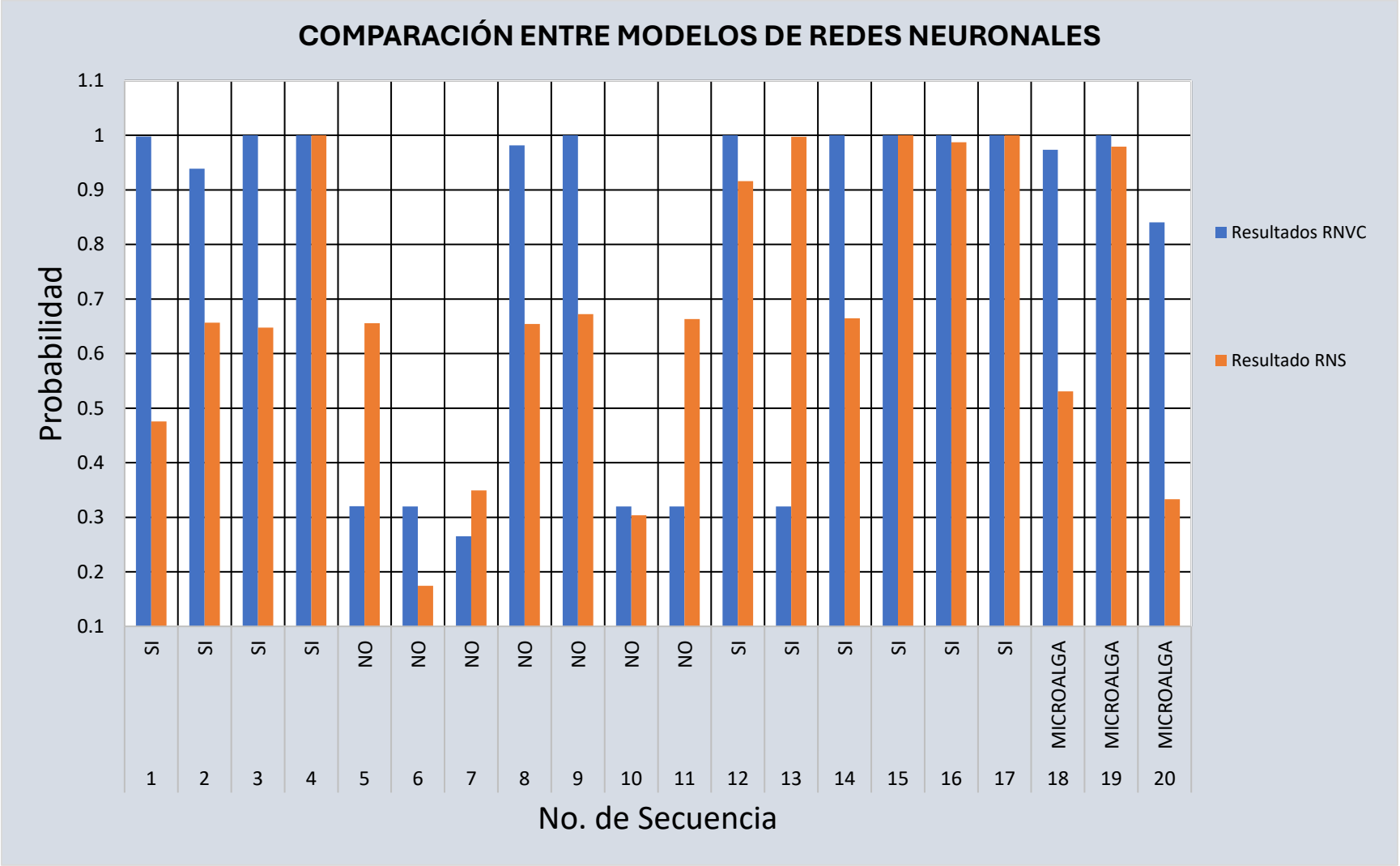



*Imagen 20. Comparación de predicciones de los modelos de Redes Neuronales.*

# Capítulo 6. Discusión

## Modelo de Redes Neuronales Sencillas

Los resultados del modelo fueron los esperados para la estrategia propuesta, en la que se optó por el diseño de un modelo con redes neuronales. En el desarrollo de una primera parte de la red neuronal, donde la estructura densa de la red neuronal permite un óptimo comportamiento en el entrenamiento las métricas generales del modelo fueron de 0.838 para la fase del entrenamiento y un 0.759 para la fase de prueba.

### Gráfico de comportamiento del modelo

Este comportamiento se puede observar también en la imagen 9, que representa el grafico de la pendiente de precisión del modelo contra las épocas con las que se entrenó del cual podemos destacar que, la parte de datos que se usaron en el entrenamiento fue incrementando en precisión con el paso de las épocas donde se observa que hubo una precisión aceptable mayor a 0.8 a partir de la época 70 aproximadamente y manteniéndose por encima de ese umbral, en cambio, para los datos que se usaron para la prueba del modelo, si bien, no inició con valores de precisión bajos, no llegó a tener el mismo rendimiento que en el entrenamiento, pudiendo deberse a la cantidad de datos para la prueba que en este caso se usó un 35% del total de datos para la fase de prueba.

### Curva ROC

Conforme a estos resultados se le pidió al programa un gráfico ROC y AUC (imagen 10), el cual se utiliza mucho para evaluar el rendimiento de los modelos de aprendizaje o de inteligencia artificial (Pethani, 2021), el gráfico consta de dos ejes, la sensibilidad y la especificidad, en nuestro caso contraponemos la tasa de verdaderos positivos contra la tasa de falsos positivos, consta también de una curva y una línea transversal en este caso que nos deja saber que representa el valor de 0.5 según lo que representa AUC que son las siglas en inglés para área bajo la curva, que, con relación a la curva nos dice una mayor área debajo de esa curva representa un mejor rendimiento del modelo, y cómo podemos observar la curva

ROC nos da un área por debajo de ella de 0.73, un valor muy parecido a lo observado en la precisión que se imprimé al compilar el modelo.

### Métricas

Para hacer un análisis más desarrollado, al programa se le pidió la realización de varias métricas, primeramente, la matriz de confusión en la que se obtuvieron los valores de 211 muestras falsas bien clasificadas y 732 muestras positivas bien clasificadas, y, por otro lado 69 muestras verdaderas mal clasificadas y 173 muestras negativas mal clasificadas. Con el uso de la matriz se realizaron los cálculos de las métricas Exactitud (Ac) con valor de 0.795, Precisión con valor de 0.808, Sensibilidad (RC) con valor de 0.913 y prueba de F1-score con valor de 0.858. Por lo que podemos observar que fue un buen rendimiento, que no está muy alejado de los trabajos anteriormente abordados.

## Modelo de Redes Neuronales con Validación Cruzada

La validación cruzada que utiliza subconjuntos aleatorios de datos, conocida como validación cruzada de k-folds, es un medio poderoso para probar la tasa de éxito de los modelos utilizados para la clasificación (Marcot, 2021). Se trabajó con 5 folds o pliegues para un número de datos de 4 mil muestras, en cada pliegue se hace un entrenamiento diferente y de los cuales se obtuvo una precisión media de 0.9319, un resultado mucho mayor al obtenido por redes neuronales convencionales.

### Gráfico de comportamiento del modelo

En la imagen 13 se nos presenta el gráfico de comportamiento del modelo, podemos observar que durante el entrenamiento mantuvo valores por encima del 0.92 que ya son bastante altos y que donde tuvo un poco más de problemas fue en la fase de prueba en la que si bien se observa más cambios, la precisión se mantuvo por arriba del 0.88 y que los valores de precisión oscilan entre 0.02-0.04 al paso de cada época.

### Curva ROC

Lo que nos muestra el grafico de la imagen 14 es que el área debajo de la curva aumento a un 0.80 y que tardo menos épocas en aumentar a un 0.80 en comparación a la curva ROC de las redes neuronales sencillas.

**Métricas**

La validación cruzada con pliegues conlleva procesamientos independientes de las muestras, por lo cual se obtuvieron 5 matrices, una por cada pliegue, y, para poder reportar los cálculos de las diferentes métricas, se realizó una operación básica de para promediar las 5 matrices. Sin embargo, cabe destacar que las matrices individuales reportaron entre 236-239 muestras verdaderas bien clasificadas, entre 82-101 muestras negativas bien clasificadas, entre 15-22 muestras negativas mal clasificadas y entre 0-4 muestras verdaderas mal clasificadas, siendo que las equivocaciones y los aciertos fueron aumentando en cada pliegue. Con el uso de la matriz promedio se realizaron los cálculos de las métricas medias de Exactitud (Ac) con valor de 0.93, Precisión con valor de 0.91, Sensibilidad (RC) con valor de 0.995 y prueba de F1-score con valor de 0.921. Por lo cual podemos verificar que efectivamente, el tratamiento del modelo con validación cruzada obtiene un aumento considerable en el rendimiento y podemos obtener como resultado una Red Neuronal capaz de conseguir buenos resultados.

**Predicciones**

En la imagen 20, observamos una gráfica que compara el valor de probabilidad que le asigna el programa según el modelo implementado en la predicción de las secuencias de la tabla 4, las barras en color naranja representan las predicciones con las redes neuronales sencillas y las barras azules representan las predicciones con las redes neuronales con validación cruzada, a simple vista se observa que el mejor rendimiento lo tuvo el modelo con validación cruzada, obtuvo más aciertos, en esta parte se añadieron 3 secuencias pertenecientes a una prueba con un modelo de red neuronal donde dieron como predicciones fuertes estas 3 secuencias con el fin de verificar la utilización del modelo de validación cruzada que si bien, las secuencias utilizadas para el entrenamiento son de *E. coli* K-12, las tres secuencias agregadas son de posibles promotores de cloroplasto de microalgas y se tiene documentada la relación de los factores sigma 70 con secuencias promotoras en cloroplastos de microalgas.

# Capítulo 7. Conclusión

Entre un Modelo de Redes Neuronales Sencillas y un Modelo de Redes Neuronales con Validación Cruzada se observó diferencias significativas en cuanto a su desempeño y capacidad predictiva.

Para el Modelo de Redes Neuronales Sencillas, se observó un comportamiento esperado, alcanzando una precisión de 0.838 en la fase de entrenamiento y 0.759 en la fase de prueba. Aunque mostró una mejora progresiva en la precisión durante las épocas de entrenamiento, la fase de prueba no logró mantener el mismo rendimiento, posiblemente debido a la cantidad limitada de datos utilizados para la evaluación.

En contraste, el Modelo de Redes Neuronales con Validación Cruzada demostró un desempeño notablemente superior, con una precisión media de 0.9319, obtenida mediante la validación cruzada de 5 folds. Este modelo exhibió una consistencia en la precisión tanto en el entrenamiento como en la fase de prueba, manteniendo valores altos por encima de 0.88 en ambas fases. Además, la curva ROC reveló un área bajo la curva (AUC) de 0.80, indicando un mejor rendimiento en comparación con el modelo de redes neuronales sencillas.

Las métricas obtenidas, como la precisión, la sensibilidad y la prueba de F1-score, respaldaron el alto rendimiento del Modelo de Redes Neuronales con Validación Cruzada, con valores medios de precisión de 0.91 y sensibilidad de 0.995. Estos resultados sugieren que el enfoque de validación cruzada permitió al modelo generalizar de forma más optima y mejorar su capacidad predictiva en comparación con el modelo sencillo de redes neuronales.

Además, las predicciones realizadas con ambos modelos confirmaron la superioridad del Modelo de Redes Neuronales con Validación Cruzada, mostrando un mayor número de aciertos y una mayor confianza en las predicciones realizadas, incluso en secuencias no incluidas en el conjunto de entrenamiento por lo que es un área de oportunidad para futuros trabajos de especies no muy bien anotadas de los que se deseé encontrar regiones promotoras para su aplicación biotecnológica.

# Referencias


Arshpreet Bhatwa, W. J.-Z. (2021). Challenges Associated With the Formation of Recombinant Protein Inclusion Bodies in Escherichia coli and Strategies to Address Them for Industrial Applications. *Bioeng. Biotechnol*. doi:https://doi.org/10.3389/fbioe.2021.630551

Barnetche, J. M. (2007). *La bioinformática como herramienta para la investigación en salud humana.* Obtenido de Salud Pública de México, REDALYC: https://www.redalyc.org/pdf/106/10649028.pdf

Bhandari, N. K. (2021). Comparison of machine learning and deep learning techniques in promoter prediction across diverse species. *PeerJ Computer Science 7*, e365.

Branco, I. &. (2021). Bioinformatics: new tools and applications in life science and personalized medicine. *Applied microbiology and biotechnology*, 937-951.

Brázda, V. B. (2021). Brázda, V., Bartas, M., & Bowater, R. P. . *Trends in Genetics 37*, 730-744.

Coppens, L. L. (22 de Septiembre de 2020). SAPPHIRE: a neural network based classifier for σ70 promoter prediction in Pseudomonas. *BMC Bioinformatics 21*, 415. doi:https://doi.org/10.1186/s12859-020-03730-z

Engl, C. J.-L. (2020). The route to transcription initiation determines the mode of transcriptional bursting in E. coli. *Nature communications, 11(1)*, 2422.

He, Y. Z. (2020). A reporter for noninvasively monitoring gene expression and plant transformation. *Horticulture research 7*.

Heidari, A. J. (2022). Machine learning applications for COVID-19 outbreak management. *Neural Computing and Applications 34*, 15313-15348.

Henikoff, S. H.-O. (2020). Efficient chromatin accessibility mapping in situ by nucleosome-tethered tagmentation. *Elife*.

Hernández, S. J. (enero de 2017). *CARACTERIZACIÓN MOLECULAR DE PROMOTORES DE LOS GENES DREB2 Y RAP2.4ADE Carica papaya L. VAR. MARADOL ROJA EN RESPUESTA A ESTRÉS ABIÓTICO.* Obtenido de Centro de Investigación Científica de Yucatán, A. C.: https://cicy.repositorioinstitucional.mx/jspui/bitstream/1003/439/1/PCB_BT_M_Tesis_2017_Reyes_Sandi.pdf

Ireland, W. T.-B. (2020). Deciphering the regulatory genome of Escherichia coli, one hundred promoters at a time. *Elife 9*.

Jiménez, P. E. (9 de JULIO de 2013). *Predicción de Promotores y Elementos Reguladores.* Obtenido de CINVESTAV: https://www.tamps.cinvestav.mx/~ertello/bioinfo/sesion11.pdf

Kang, Z. C. (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering 149*, 106773.

Khan, Z. A. (2020). Promoter identification in DNA sequences using machine learning. *In 2020 IEEE 17th India Council International Conference (INDICON)*, 1-4.

Lara, Á. R. (2011). *Producción de proteínas recombinantes en Escherichia coli.* Obtenido de Revista mexicana de ingeniería química, SCIELO: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-27382011000200006

Li, F. C. (2021). Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework. *Briefings in bioinformatics, 22(2)*, 2126-2140.

M. Shujaat, S. L. (2021). Cr-Prom: un modelo basado en redes neuronales convolucionales para la predicción de promotores de arroz. *IEEE Access*, 81485-81491. doi:doi: 10.1109/ACCESS.2021.3086102

Marcot, B. G. (2021). What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? *Computational Statistics, 36(3)*, 2009-2031.

Martinez, G. S.-R. (2021). DNA structural and physical properties reveal peculiarities in promoter sequences of the bacterium Escherichia coli K-12. *SN Applied Sciences, 3*, 1-10.

Mejía-Almonte, C. B.-V. (2020). Redefining fundamental concepts of transcription initiation in bacteria. *Nature Reviews Genetics 21(11)*, 699-714.

NIH. (3 de ENERO de 2024). *PROMOTOR.* Obtenido de NATIONAL HUMAN GENOME RESEARCH INSTITUTE: https://www.genome.gov/es/genetics-glossary/Promotor

Pethani, F. (2021). Promises and perils of artificial intelligence in dentistry. *Australian Dental Journal, 66(2)*, 124-135.

Pi-Jing Wei a, Z.-Z. P.-J.-Y.-S.-H. (2022). Predicción del promotor en nanochloropsis basada en redes neuronales convolucionales densamente conectadas. *METHODS*, 38-46.

Pisner, D. A. (2020). Support vector machine In Machine learning. *Academic Press.*, 101-121.

Reyna, M. C. (Octubre de 2020). *IDENTIFICACIÓN DE PROMOTORES σ54 BACTERIANOS CON BASE EN LA CONSERVACIÓN DE SECUENCIA NUCLEOTÍDICA.* Obtenido de Biocomputo.ibt.unam: https://biocomputo.ibt.unam.mx/tesis/maricela-maestria.pdf

S. M. (02 de 07 de 2020). iPromoter-BnCNN: a novel branched CNN-based predictor for identifying and classifying sigma promoters. *OXFORD Bioinformatics*, 4869-4875. doi:10.1093/bioinformatics/btaa609

Salgado, H. G.-C.-A.-C.-A.-V. (2024). RegulonDB v12. 0: a comprehensive resource of transcriptional regulation in E. coli K-12. *Nucleic Acids Research*, D255D264.

Sharma, A. J. (2020). Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access, 9*, 4843-4873.

Shen, S. S. (2023). Characteristics of antimicrobial peptide OaBac5mini and its bactericidal mechanism against Escherichia coli. *Frontiers in Veterinary Science*.

Tayara, H. T. (2020). Identification of prokaryotic promoters and their strength by integrating heterogeneous features. *Genomics, 112(2)*, 1396-1403.

Towsey, M. T. (2008). La predicción entre especies de promotores bacterianos utilizando una máquina de vectores de soporte. *Biología y química computacional*, 359–366. doi:10.1016/j.compbiolchem.2008.07.009

Wang, X. X. (2023). Deep Learning-Assisted Design of Novel Promoters in Escherichia coli. *Advanced Genetics, 4(4)*, 2300184.

Wu, X. L. (2023). Optimization of Constitutive Promoters Using a Promoter-Trapping Vector in Burkholderia pyrrocinia JK-SH007. *International Journal of Molecular Sciences 24(11)*, 9419.

Yang, A. Z. (2020). Review on the application of machine learning algorithms in the sequence data mining of DNA. *Frontiers in Bioengineering and Biotechnology, 8*, 1032.

You, J. L. (2020). Graph structure of neural networks. *In International Conference on Machine Learning*, 10881-10891.

Zhu, Y. L. (2021). Computational identification of eukaryotic promoters based on cascaded deep capsule neural networks. *Briefings in Bioinformatics, 22(4)*, bbaa299.

***Dedico este trabajo:***

*A mi familia y a Dios.*

*A mis padres Guillermo y Mayra,*

*mis hermanos Heriberto y Flor,*

*Les dedico este trabajo y agradezco de corazón todo el apoyo y amor incondicional.*