|  |
| --- |
| **SGBD-IA Manual del sistema** |
| **Harold Stiven Ávila**  **Yesenia Castro**  **Yeison Alfredo Micolta**  **Héctor Andrés Mora**  **Jorge Albeiro Rivera** |
|  |
| **SISNOVA S.A.S** |
|  |
| **2025** |
|  |

Contenido

**Pág.**

[PRESENTACION 4](#_Toc189830687)

[INTRODUCCIÓN 5](#_Toc189830688)

[PROPÓSITO DEL MANUAL 5](#_Toc189830689)

[Objetivos del manual 5](#_Toc189830690)

[APLICATIVO DE ESCRITORIO SGBD-IA 6](#_Toc189830691)

[MARCO TEORICO DEL PROYECTO 7](#_Toc189830692)

[Tecnologias utilzadas 7](#_Toc189830693)

[Metodologia 8](#_Toc189830694)

[DIAGRAMAS DE MODELAMIENTO 9](#_Toc189830695)

[Diagrama Arquitectónico 9](#_Toc189830696)

[Diagrama de paquetes 10](#_Toc189830697)

[Diagrama de Casos de uso 11](#_Toc189830698)

[GESTION DE LA INFORMACION 11](#_Toc189830699)

[COMPONENTES TÉCNICOS 14](#_Toc189830700)

[CONFIGURACION DEL ENTORNO 14](#_Toc189830701)

[Instrucciones configuración del entorno 14](#_Toc189830702)

[Ejecución de la aplicación 17](#_Toc189830703)

Ilustraciones

[Ilustración 1 Diagrama Metodologia CRISP-DM 9](#_Toc189829482)

[Ilustración 2 Diagrama de Arquitectura 9](#_Toc189829483)

[Ilustración 3 Diagrama de paquetes 10](#_Toc189829484)

[Ilustración 4 Diagrama casos de usos 11](#_Toc189829485)

[Ilustración 5 Clase FileManager (Java) 12](#_Toc189829486)

[Ilustración 6 Función main (Python) 13](#_Toc189829487)

[Ilustración 7 Funcion predict\_img (Python) 13](#_Toc189829488)

[Ilustración 8 Descarga de JDK 15](#_Toc189829489)

[Ilustración 9 Descarga apache netbeans 15](#_Toc189829490)

[Ilustración 10 Descarga de python 16](#_Toc189829491)

[Ilustración 11 Verificar version de Python 16](#_Toc189829492)

[Ilustración 12 Comando instalación opencv 16](#_Toc189829493)

[Ilustración 13 Comando instalación tensorflow 17](#_Toc189829494)

[Ilustración 14 Comando instalación numpy 17](#_Toc189829495)

[Ilustración 15 Iniciando apache netbeans 17](#_Toc189829496)

[Ilustración 16 Visual studio code 18](#_Toc189829497)

[Ilustración 17 Carga de archivos Java en netbeans 18](#_Toc189829498)

[Ilustración 18 Carga de archivo python en VSC 19](#_Toc189829499)

[Ilustración 19 Verificación de archivos en netbeans 19](#_Toc189829500)

[Ilustración 20 Verificación de archivo Python 19](#_Toc189829501)

[Ilustración 21 Ejecución de UDENARBDMS 20](#_Toc189829502)

[Ilustración 22 Ejecución de UDENARSQL 20](#_Toc189829503)

[Ilustración 23 Ejecución de UDENAR-IA 21](#_Toc189829504)

[Ilustración 24 Ventana principal de UDENARSQL 22](#_Toc189829505)

# PRESENTACION

Bienvenido al manual del sistema SGBD-IA. Este documento ha sido creado para proporcionar una guía completa sobre el funcionamiento de nuestro sistema, diseñado para gestionar bases de datos optimizando el uso de datos no estructurados como lo son las imágenes haciendo uso de la inteligencia artificial.

El presente manual proporciona la información necesaria para realizar la instalación y exploración del software SGBD-IA. Ofrece una guía detallada de la arquitectura y el funcionamiento del software, permitiendo a los desarrolladores comprender y trabajar con el sistema de manera efectiva. Se proporciona una descripción completa de las distintas partes del desarrollo del aplicativo, asegurando que cualquier persona que desee acceder al software pueda hacerlo, manteniendo así la calidad y estabilidad del sistema.

# INTRODUCCIÓN

Este manual detalla al aplicativo de escritorio SGBD-IA, proporcionando una visión general de sus funcionalidades y estructura. Se incluye un Marco Teórico del Proyecto, explicando las tecnologías utilizadas y la metodología adoptada para su desarrollo. Además, se presentan diversos Diagramas de Modelamiento, como el diagrama arquitectónico, el diagrama de paquetes y el diagrama de casos de uso, que ayudan a comprender mejor la estructura y el flujo del sistema.

La gestión de la Información es otro aspecto crucial que se aborda, describiendo cómo se introducen y procesan los datos para la predicción dentro del sistema gestor de bases de datos. También se proporcionan detalles técnicos en la sección de Componentes Técnicos, donde se explican las tecnologías y herramientas empleadas para desarrollar el backend y el frontend del sistema.

Finalmente, el manual incluye instrucciones detalladas para la configuración del entorno, abarcando tanto la instalación de las bibliotecas necesarias como la carga de los archivos para su correcto funcionamiento. Se ofrecen pasos específicos para la instalación y despliegue del sistema, asegurando que los usuarios puedan configurar y utilizar el SGBD-IA de manera efectiva.

# **PROPÓSITO DEL MANUAL**

El propósito de este manual es proporcionar toda la información necesaria para la instalación y uso efectivo del sistema SGBD-IA. Este documento está diseñado para ayudar a los usuarios a comprender la arquitectura y el funcionamiento del software, facilitando tanto a desarrolladores como a usuarios finales la implementación y operación del sistema. Además, este manual ofrece una guía detallada que asegura que cualquier persona con conocimiento básicos en lenguajes de programación y SQL interesada pueda acceder y trabajar con el software de manera eficiente, manteniendo la calidad y estabilidad del sistema.

# **Objetivos del manual**

**Objetivo general**

Brindar la información necesaria para poder realizar la instalación y configuración del software.

**Objetivos específicos**

* Detallar cómo se configura el entorno de desarrollo.
* Ofrecer información exhaustiva sobre la instalación del software.
* Describir el funcionamiento de los componentes del sistema, tanto en el motor de bases de datos (backend) como en el cliente (frontend).

# APLICATIVO DE ESCRITORIO SGBD-IA

SGBD-IA es un gestor de bases de datos simplificado que ha sido desarrollado como una solución para la gestión y procesamiento eficiente de datos no estructurados como son las imagenes. Permite a los usuarios realizar consultas DML y predecir imágenes con la ayuda de un modelo de inteligencia artificial.

El objetivo principal del SGBD-IA es facilitar la consulta de datos no estructurados (imágenes).

Para el desarrollo de SGBD-IA se fundamentó en el fuerte acoplamiento de redes neuronales artificiales directamente en el kernel, permitiendo la predcción y ejecución de consultas DML de imágenes. Este proceso se logró mediante el uso de algoritmos especializados, como la Redes Neuronales Convolucionales (CNN), que facilitan el procesamiento de imágenes. Además, se implementó un modelo avanzado de predicción.

La arquitectura del sistema se compone de dos modulos, el primer modulo fue desarrollado en el lenguaje compilado Java, que gestiona la lógica del software, el segundo fue desarrollado en el lenguaje interpretado Pyhton, en este se procesan las imágenes y se hace uso del modelo para la predicción de las mismas; mientras que el Frontend está implementado únicamente en Java, proporcionando una interfaz gráfica en la cual los usuarios finales visualizan de forma correcta los datos.

SGBD-IA se centra en un matodologia sólida para el desarrollo del software, específicamente la metodología CRISP-DM, esencial para asegurar una implementación exitosa. Esta junto con los algoritmos especializados, garantizan que el sistema cumpla con sus objetivos propuestos de manera precisa y eficiente, asegurando resultados óptimos que son fundamentales tanto para su operación actual como para futuros desarrollos.

# MARCO TEORICO DEL PROYECTO

SGBD-IA se estructura en torno a varios componentes clave que guian su desarrollo y despliegue, para que este pueda gestionar los datos, ejecutar las consultas DML y realizar las predicciones de manera correcta. Este marco proporciona una visión de las tecnologías que permiten consolidar los modulos requeridos que se acoplen a los objetivos y necesidades de proyecto, a continuación, se presenta una descripción breve de las tecnologías más relevantes utilizadas en este desarrollo.

# Tecnologias utilzadas

* **Java:** Es el lenguaje de programacion principal empleado en SGBD-IA. En SGBD-IA, Java se utiliza para desarrollar la lógica del sistema y realizar las operaciones necesarias sobre las consultas.
* **Python:** Es el lenguaje de programacion secundario empleado en SGBD-IA. Es conocido por su simplicidad y potencia, especialmente en el ámbito de aprendizaje automatico y la minería de datos. Python se utiliza para desarrollar los algoritmos de inteligencia artificial que procesan las imágenes, incluyendo el modelo de predicción.
* **JSQLParser:** Es un parseador de sentencias SQL. JSQLParser se encarga de traducir las sentencias SQL en una jerarquia de clases Java en SGBD-IA, permitiendo que el sistema interprete de manera correcta las consultas DML lanzadas por los usuarios.
* **CNN (Convolutional Neural Network)**: Son un tipo de red neuronal artificial (ANN) que son usadas eficazmente para procesar datos no estructurados como imágenes y videos. En SGBD-IA, las CNN se utilizan para la creación, entrenamiento, prueba y evaluación del modelo de predicción creado.

clasificar y predecir imágenes que luego son retornadas con etiquetas.

# Metodologia

Para el desarrollo del SGBD-IA se ha utilizado la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), una metodología reconocida en el campo de la minería de datos y el aprendizaje automático. Esta metodología se estructura en seis fases que guían el proceso de desarrollo del aplicativo de manera sistemática y efectiva.

La **Comprensión del Negocio** es la primera fase, en la cual se identifican las necesidades y objetivos específicos del area educativa para la planificación curricular.

En la fase de **Comprensión y preparación de los Datos**, Se utilizó el conjunto de datos Fashion-MNIST para el entrenamiento y prueba del modelo, este conjunto de dato contas de 70.000 imágenes de las cuales 60.000 fuero usadas para el entrenamiento y 10.000 para pruebas del modelo generado.

El conjunto de datos Fashion-MNIST está compuesto por 10 clases, cada imagen está en escala de grises con dimensiones de 28x28

Las clases están etiquetadas de la siguiente manera:

* 0 T-shirt/top
* 1 Trouser
* 2 Pullover
* 3 Dress
* 4 Coat
* 5 Sandal
* 6 Shirt
* 7 Sneaker
* 8 Bag
* 9 Ankle boot

Durante la fase de **Modelado**, se desarrollaró y entrenaró un modelo de CNN para la clasificación y predicción de imágenes.

La fase de **Evaluación** implicó pruebas rigurosas al modelo para asegurar su precisión y eficiencia en la predicción de imágenes.

Finalmente, en la fase de **Despliegue**, se realizó la exportación del modelo en formato h5, posteriormente se integró a SGBD-IA a través de el modulo creado en Python el cual se conecta al resto del sistema usando sockets con lo cual el sistema queda completamente funcional para su uso.



Ilustración 1 Diagrama Metodologia CRISP-DM

# DIAGRAMAS DE MODELAMIENTO

# Diagrama Arquitectónico

**Arquitectura**

La arquitectura del sistema gestor de bases de datos simplificado (SGBD-IA) utiliza Java en conjunto con Python como backend y Java como Frontend se estructura como se puede observar en el diagrama aquitectonico que se muestra a continuación:

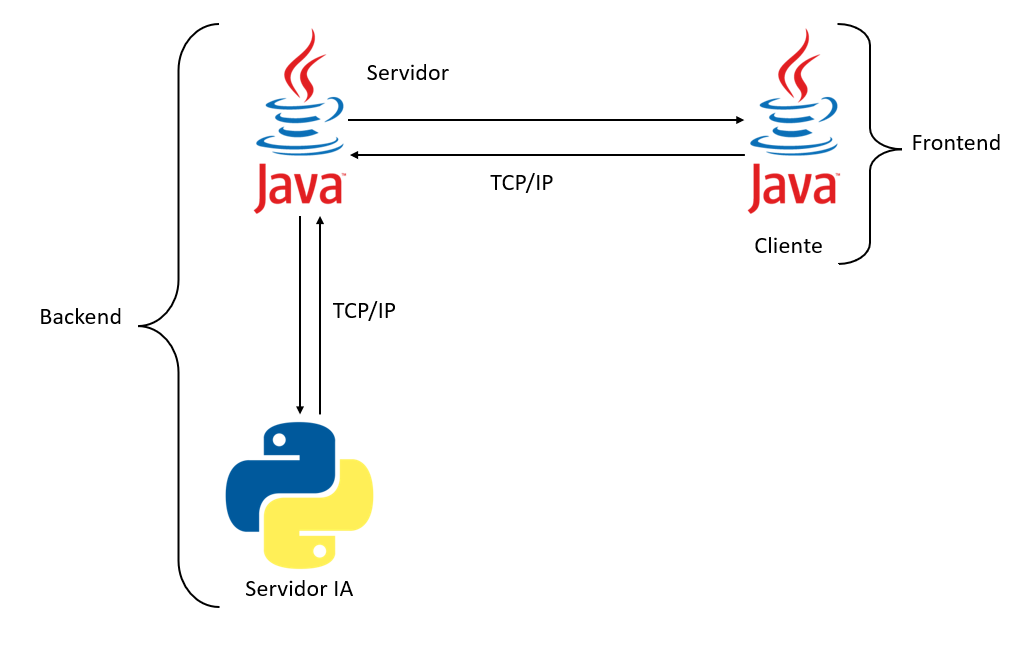


Ilustración 2 Diagrama de Arquitectura

* **Frontend:** El frontend desarrollado en Java se encarga de la interfaz de usuario y la interacción directa con los usuarios finales.
  + **Java:** Utilizado para la lógica del cliente (Frontend) y comunicación con el motor de bases de datos (Backend).
* **Backend:** El backend actúa como el servidor de aplicación y maneja la lógica de negocio.
  + **Java:** Utilizado para implementar la lógica del motor de bases de datos (Backend)<, incluyendo algoritmos y servicios.
  + **Python:** Utilizado para el procesamiento de las imágenes y la conexión del modelo de predicción con el motor de base de datos.

# Diagrama de paquetes

El diagrama de paquetes del SGBD-IA detalla los paquetes y las relaciones cruciales entre ellos para representar el sistema de manera integral. El proceso se inicia con los paquetes de información principales en el Backend, el cual emplea los lenguajes de programación Java y Python, y en el Frontend, únicamente se us el lenguaje Java. Estos componentes trabajan en conjunto mediante la conexión TCP/IP para la comunicación entre el cliente y el servidor, permitiendo la transferencia de datos entre ellos y posibilitando así la interacción y funcionalidad del sistema. Como se muestra en la imagen, el Backend aprovecha los modelos y métricas de inteligencia artificial haciendo uso del modulo desarrollado en Python, así como la sección de acceso a datos en el modulo codificado en Java. La parte del frontend, se evidencia el uso de la biblioteca JFrame la cual ya viene integrada en Java. Esta estructura organizada y cohesionada no solo facilita la comprensión del sistema SGBD-IA, sino también su mantenimiento continuo.

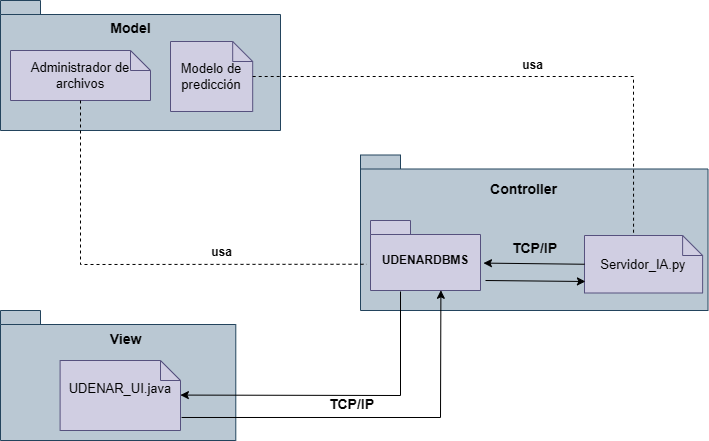


Ilustración 3 Diagrama de paquetes

# Diagrama de Casos de uso

El diagrama de casos de uso proporciona una representación visual de las interacciones entre los usuarios y el SGBD-IA. Este sistema esta diseñado para realizar consultas en las que se puedan usar datos no estructurados (imágenes) de manera mas eficiente. Además de esta función principal, el sistema también proporciona características comunes que contienen los demás sistemas gestores de bases de datos relacionales. A continuacion se muestra como se representa el diagrama destacando las funciones principales que ofrece SGBD-IA y cómo los usuarios pueden interactuar con el sistema.

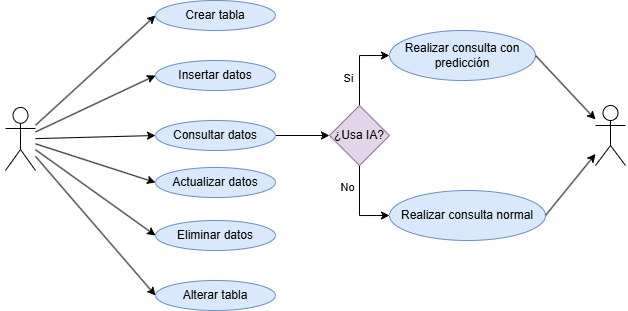


Ilustración 4 Diagrama casos de usos

# GESTION DE LA INFORMACION

En el SGBD-IA, la gestión de la información se basa en el almacenamiento de datos estructurados y no estructurados proporcionados por el usuario que interactúa con la interfaz frontend Java para Ingresar la información necesaria mediante la escritura manual de sentencias SQL. Las sentencias SQL enviadas por los usuarios a través de la interfaz frontend Java son cargadas y procesadas en el mortor de base de datos para permitir el intercambio de recursos y llevar a cabo el procesamiento de la información.

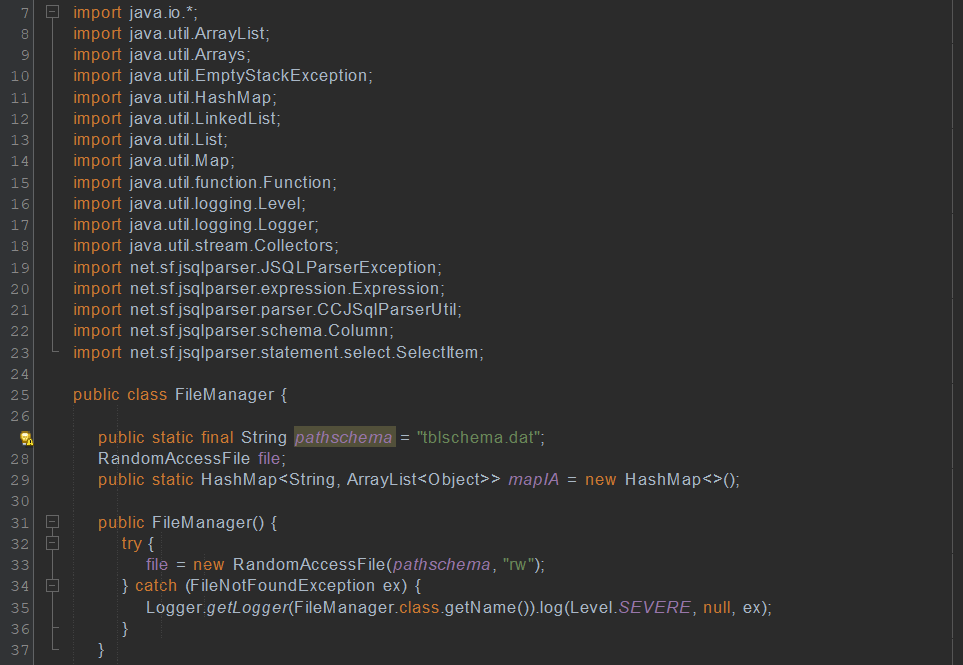


Ilustración 5 Clase FileManager (Java)

En el backend, el motor de bases de datos (UDENARBDMS) utiliza la calse “FileManager.java” para la creación de archivos binarios con dos tipos de formato, para el schema hace uso del formato “.dat” y para las tablas usa el formato “.tbl”.

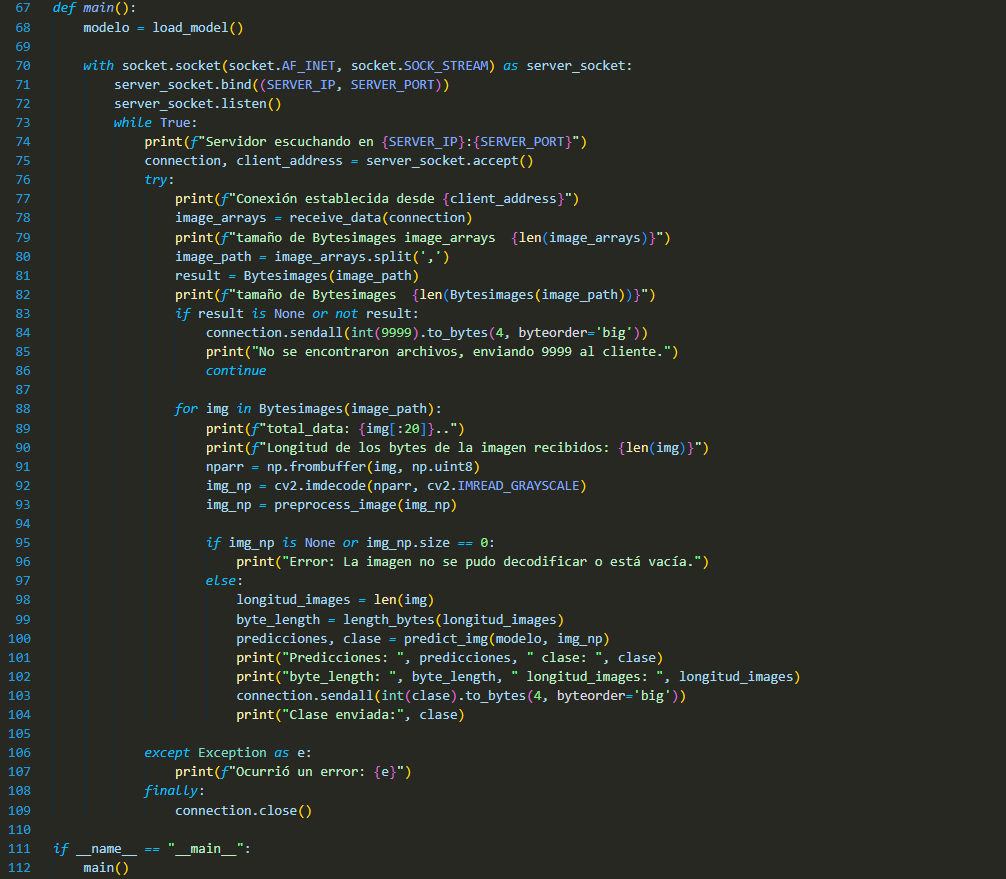


Ilustración 6 Función main (Python)

La función main que se encuentra en el modulo UDENAR-IA, es la principal y se encarga de ejecutar las funciones para la carga del modelo, la conexión con el modulo UDENARBDMS, procesar las imágenes, realizar las predicciones de las imágenes y enviar las predicciones al motor de bases de datos.

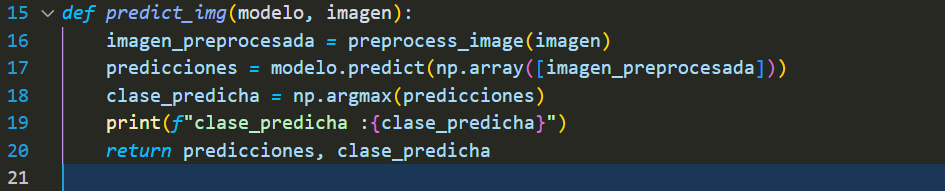


Ilustración 7 Funcion predict\_img (Python)

La función predict\_img es la que predice las imágenes ya procesadas haciendo uso del modelo de predicción.

# COMPONENTES TÉCNICOS

Para el desarrollo del software SGBD-IA se han utilizado tecnologías que aseguran el rendimiento y otras cualidades del aplicativo. Estas tecnologías están seleccionadas para resolver las necesidades presentes y futuras del sistema. En la siguiente tabla se describen los componentes según su nivel de relevancia:

|  |  |  |
| --- | --- | --- |
| **CATEGORIA** | **ELEMENTO** | **RELEVANCIA** |
| **Generales** | | |
| Sistema operativo | Windows | Alta |
| **Backend** | | |
| Lenguaje de programación | Java 19.0.2 | Alta |
| Lenguaje de programación | Python 3.12.3 | Alta |
| Framework | Pandas 2.2.2 | Media |
| Framework | Cv2 4.9.0 | Media |
| Framework | Tensorflow 2.16.0 | Media |
| Framework | Numpy 1.26.4 | Media |
| **Frontend** | | |
| Lenguaje de Programación | Java 19.0.2 | Alta |
| Framework | JFrame | Alta |

# CONFIGURACION DEL ENTORNO

# Instrucciones configuración del entorno

Para configurar y ejecutar la aplicación de manera correcta, es necesario seguir un proceso detallado de pasos e instalaciones. A continuación, se describen las instrucciones para realizar una instalación adecuada:

1. **Instalación de JDK 19.0.2.**

Inicialmente instalamos el Java Development kit que se encuentra de manera gratuita en la pagina

<https://www.oracle.com/java/technologies/javase/jdk19-archive-downloads.html>, se desplaza hacia abajo hasta encontrar las opciones de instalación para Windows

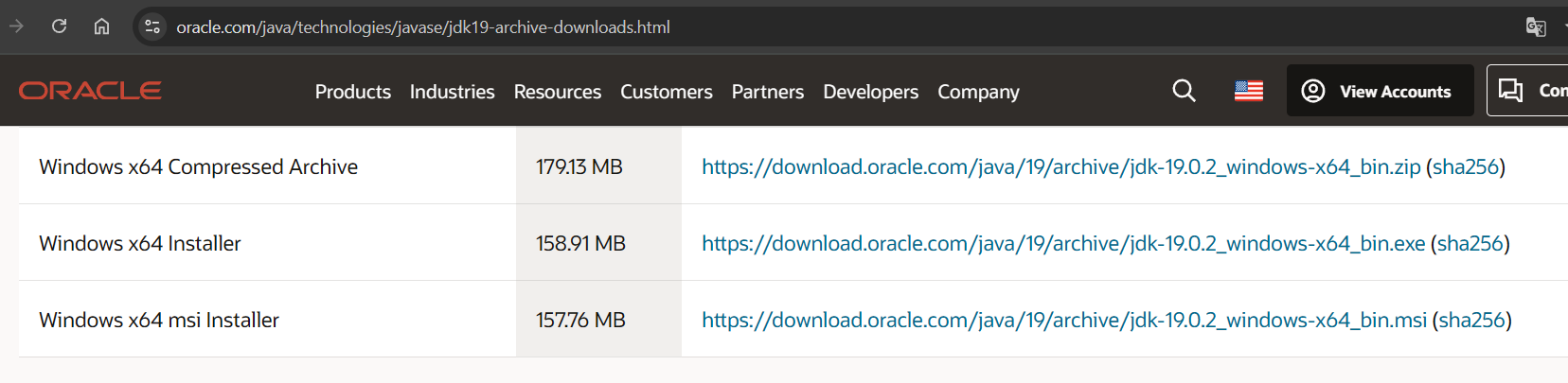


Ilustración 8 Descarga de JDK

1. **Instalación de Apache Netbeans.**

Para la instalación de Apache Netbeans se dirige a la pagina <https://netbeans.apache.org/front/main/index.html>, luego darle click en el botón “Download” que aparece en la parte principal de la página y proceda con la instalación.

****

Ilustración 9 Descarga apache netbeans

1. **Instalación de Python.**

Para empezar, se debe instalar Python 3.12.3, el cual se consigue de manera gratuita en la página <https://www.python.org/>, Dirigirse a la pestaña de descargas y buscar el lanzamineto especifico 3.12.3, dar Clik en el boton de descarga, y proceder con la intalacion.

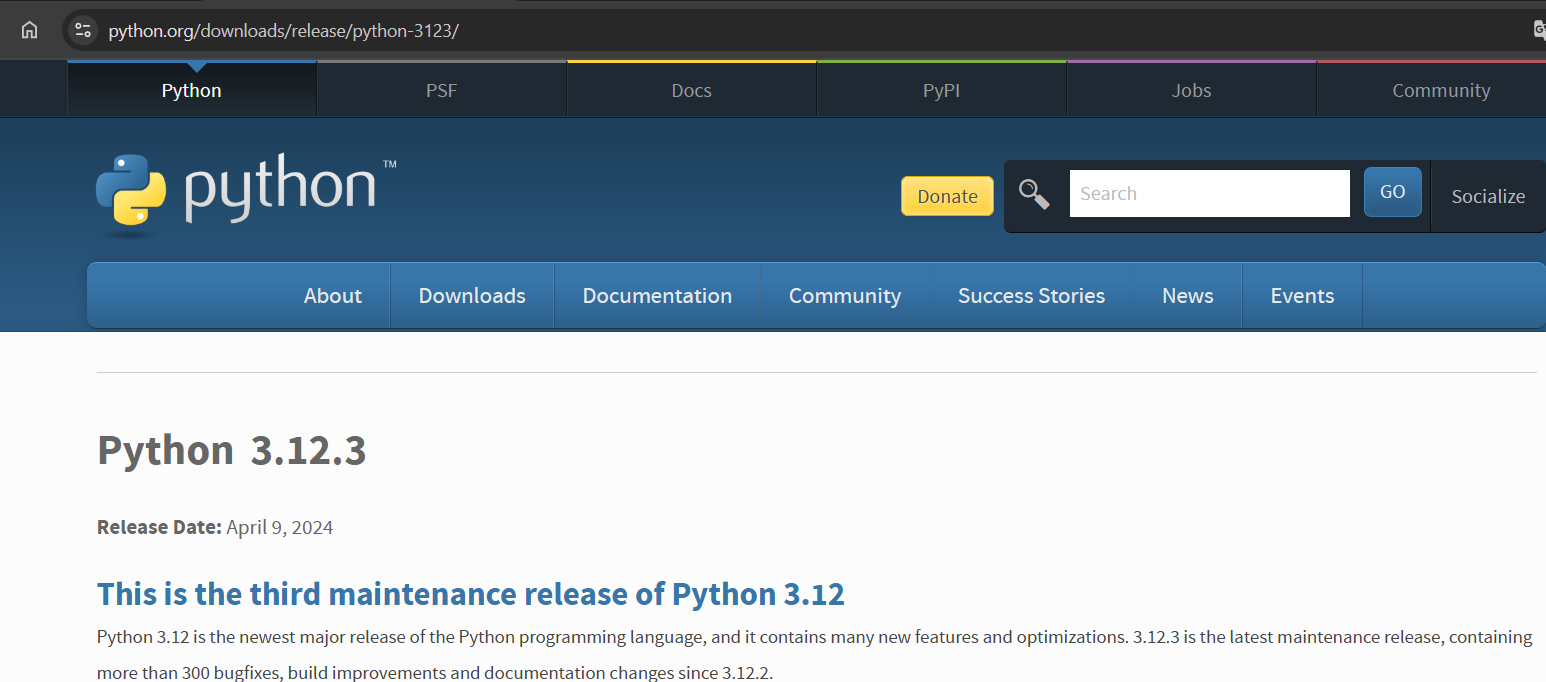


Ilustración 10 Descarga de python

Para verificar su instalación, diríjase a la consola de comandos en Windows (cmd) y escriba el commando



Ilustración 11 Verificar version de Python

1. **Instalacion de bibliotecas para Python**.

Para la creación del entorno virtual en Flask, accede a la consola de comandos y ejecuta los siguientes comandos dentro del directorio del aplicativo.

* **Instalación de opencv 4.9.0.80**

Se digita en la terminal de Windows el siguiente comando “**pip install opencv-python==4.9.0.80**” luego presiona la tecla “enter” y la instalación se realizará.



Ilustración 12 Comando instalación opencv

* **Instalacion de tensorflow 2.16.0**

Se digita en la terminal de Windows el siguiente comando “**pip install tensorflow==2.16.0**” luego presiona la tecla “enter” y la instalación se realizará.



Ilustración 13 Comando instalación tensorflow

* **Instalación de numpy 1.26.4**

Se digita en la terminal de Windows el siguiente comando “**pip install numpy==1.26.4**” luego presiona la tecla “enter” y la instalación se realizará.

****

Ilustración 14 Comando instalación numpy

# Ejecución de la aplicación

Para acceder al aplicativo, primero debes tener instalado Java SE Development Kit 19.0.2 y Apache Netbeans IDE para la ejecución de código Java, Python 3.11.5 y Visual Studio Code para la ejecución del código Python, herramientas con las cuales podrás usar el aplicativo de manera correcta.

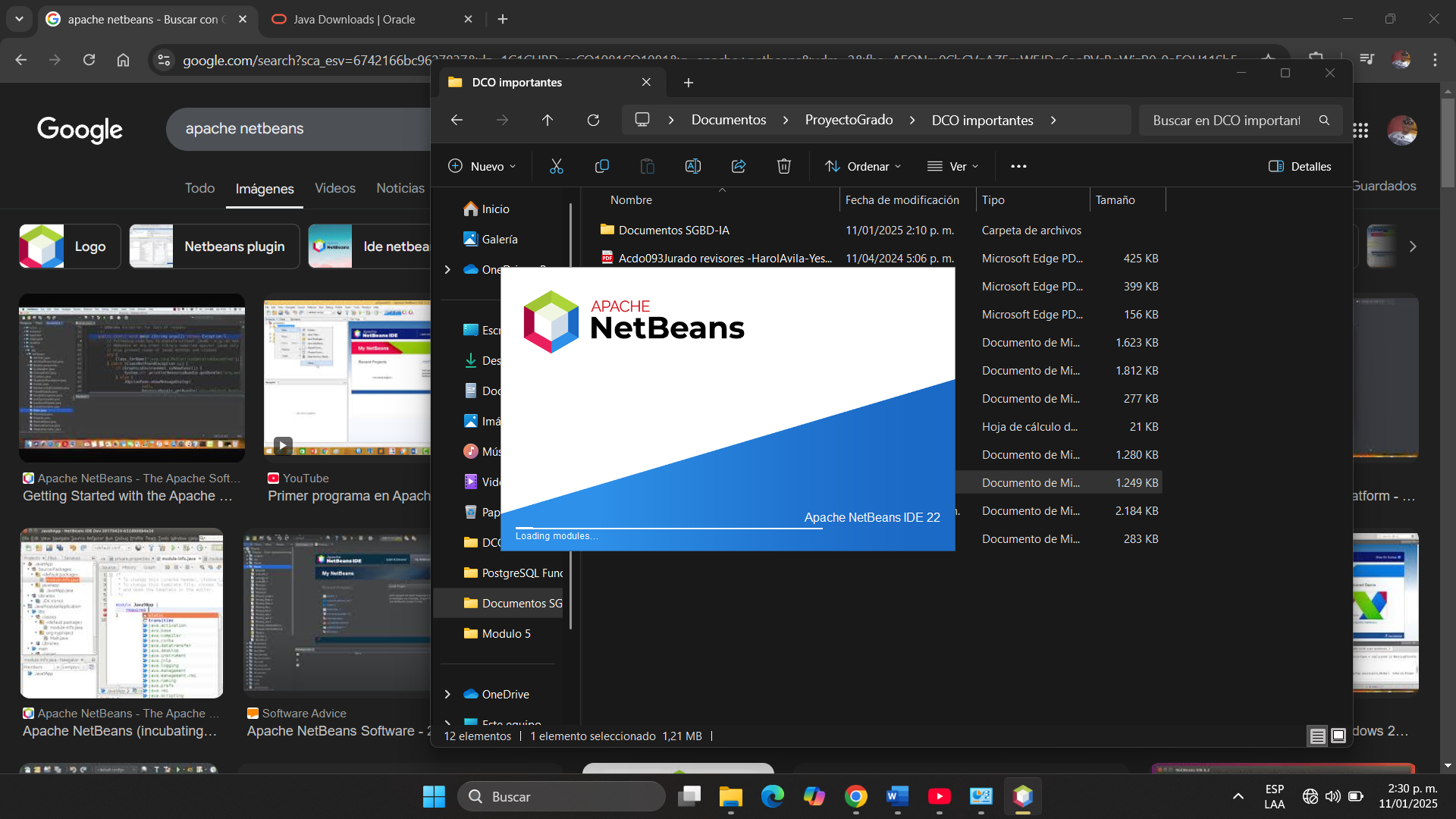


Ilustración 15 Iniciando apache netbeans

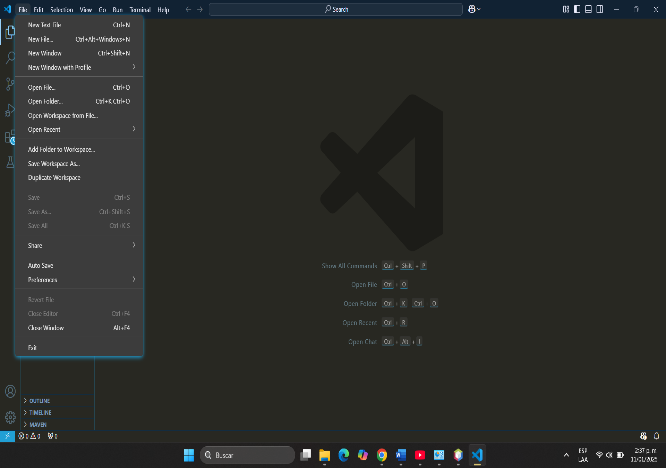


Ilustración 16 Visual studio code

* + 1. **Carga de archivos**

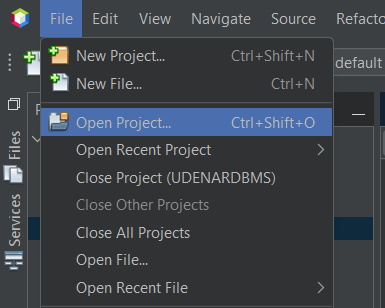


Ilustración 17 Carga de archivos Java en netbeans

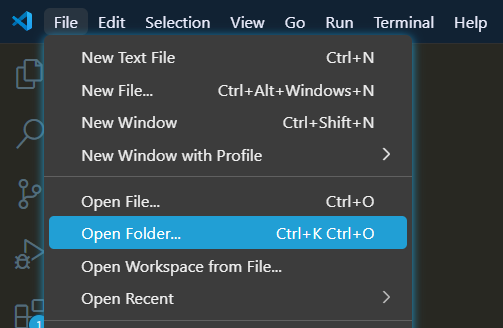


Ilustración 18 Carga de archivo python en VSC

Una vez teniendo esas herramientas instaladas procedemos a cargar las carpetas en las cuales se encuentran los scrips, en Apache Netbeans cargamos los de Java y en Visual Studio Code el script de Python.

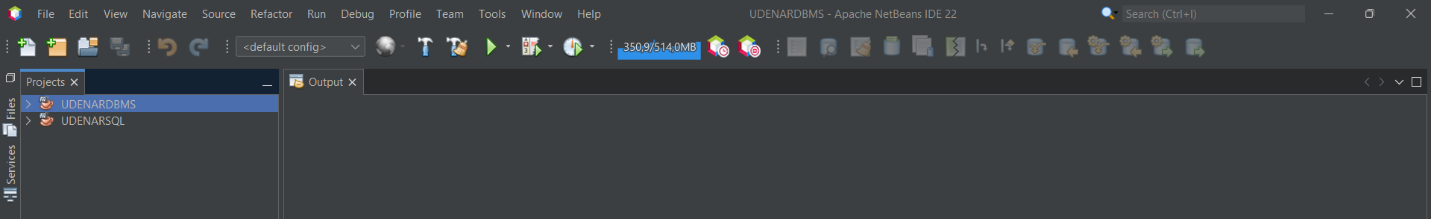


Ilustración 19 Verificación de archivos en netbeans

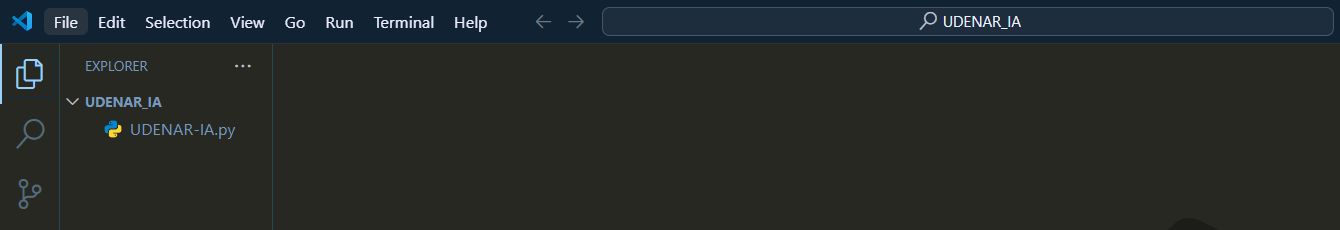


Ilustración 20 Verificación de archivo Python

En Apache Netbeans debes tener dos folders o carpetas llamadas UDENARDBMS y UDENARSQL, en Visual Studio Code debes tener la carpeta UDENAR\_IA.

* + 1. **Ejecución de la aplicacion**

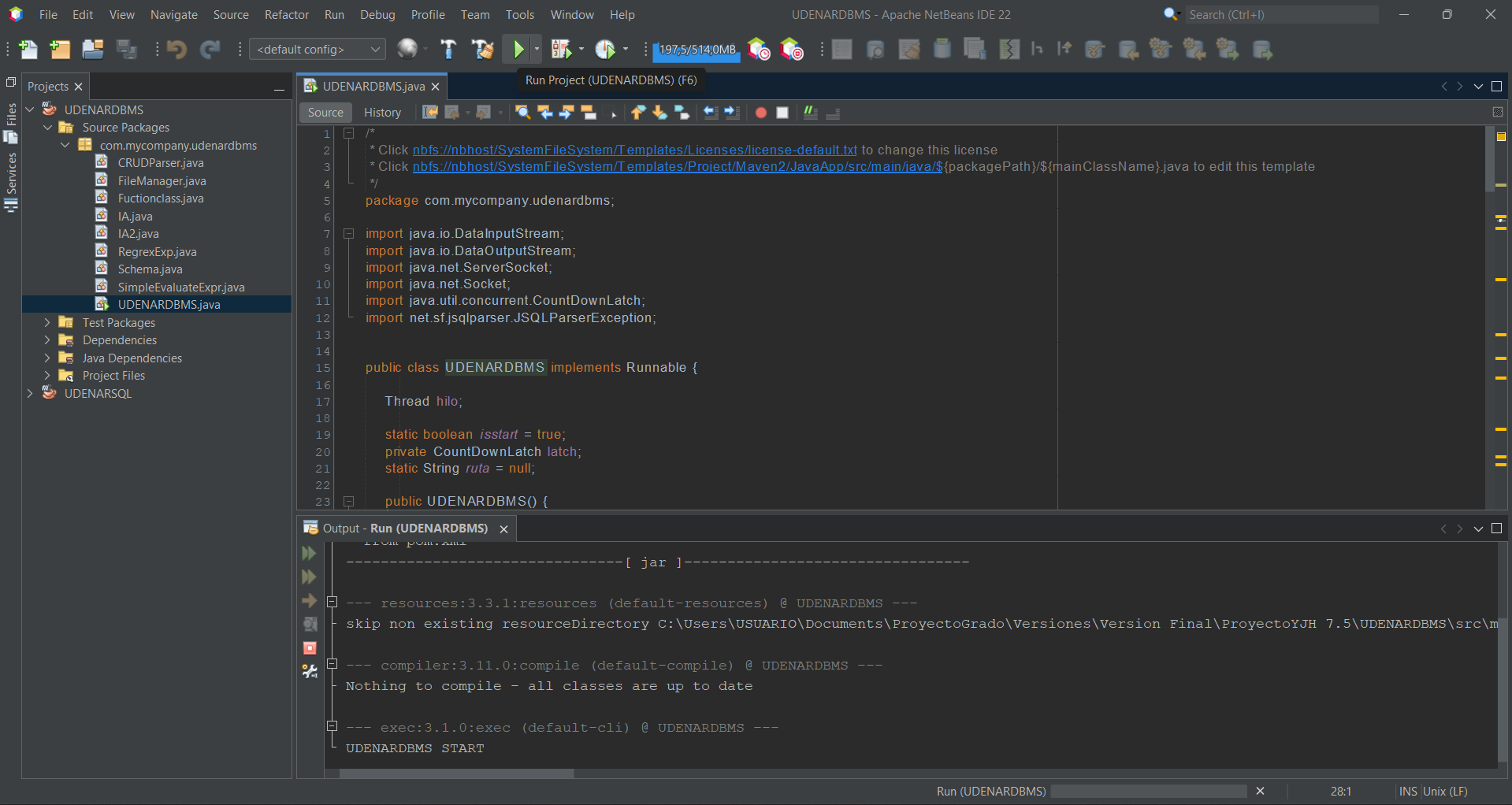


Ilustración 21 Ejecución de UDENARBDMS

Para iniciar el motor de base de datos debes ubicarte en la siguiente ruta “UDENARBDMS/Source Packages/com.mycompany.udenarbdms/UDENARBDMS.java”, dando doble click en UDENARBDMS.java queda seleccionado y se abrirá el script de Java para iniciar el motor de bases de datos, por ultimo, se da click en el botón play de color verde o presiona la tecla F6, en la parte inferior se abre la terminal y tendrá el siguiente mensaje “UDENARBDMS START” lo cual indica que el motor ha iniciado de manera correcta.

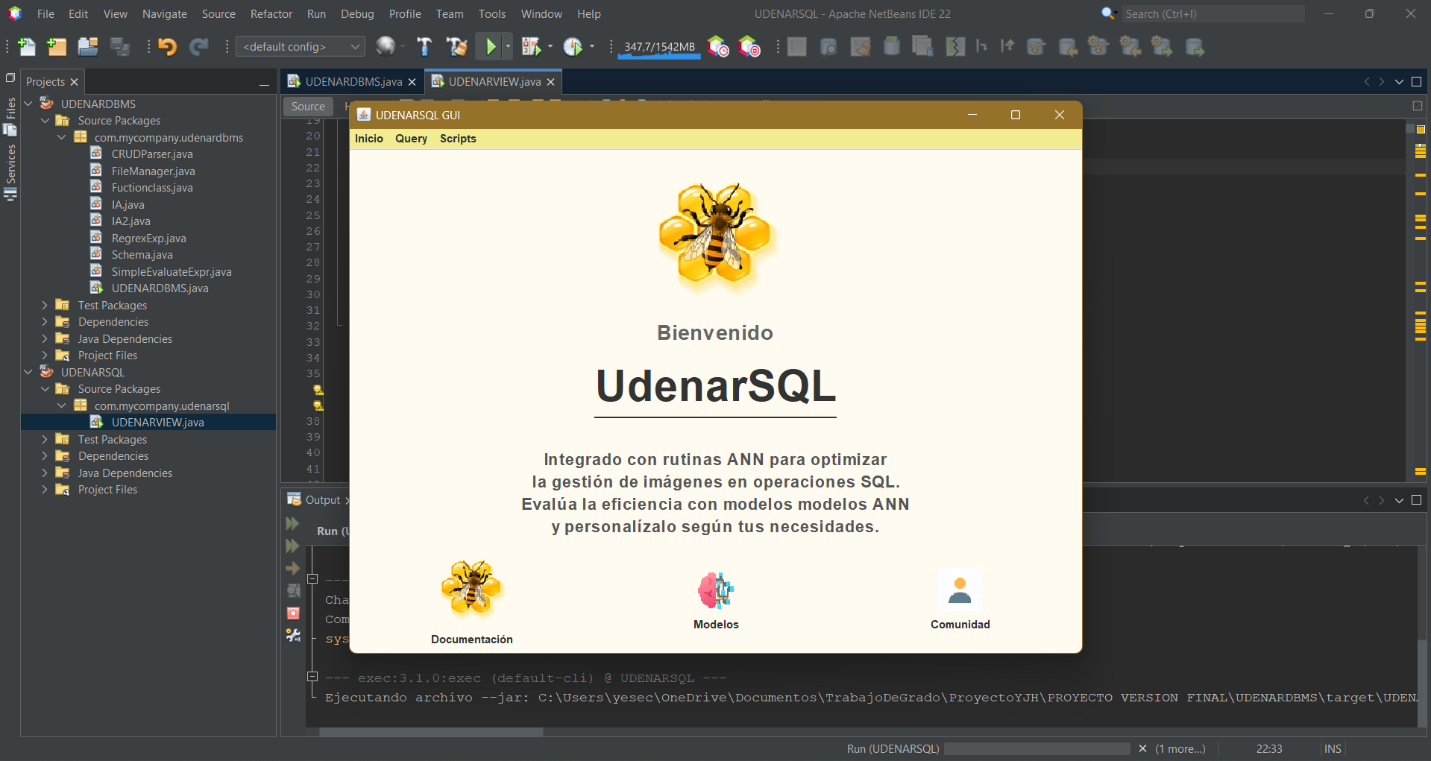


Ilustración 22 Ejecución de UDENARSQL

Luego debes ubicarte en la siguiente ruta para iniciar la interfaz gráfica del usuario “UDENARSQL/Source Packages/com.mycompany.udenarsql/UDENARSQL.java”, dando doble click en UDENARVIEW.java queda seleccionado y se abrirá el script de Java para iniciar la interfaz gráfica del cliente, nuevamente click en el botón play de color verde o presiona la tecla F6, en la parte inferior se abre la terminal y se desplegará la venta, lo cual indica que el cliente ha iniciado de manera correcta.

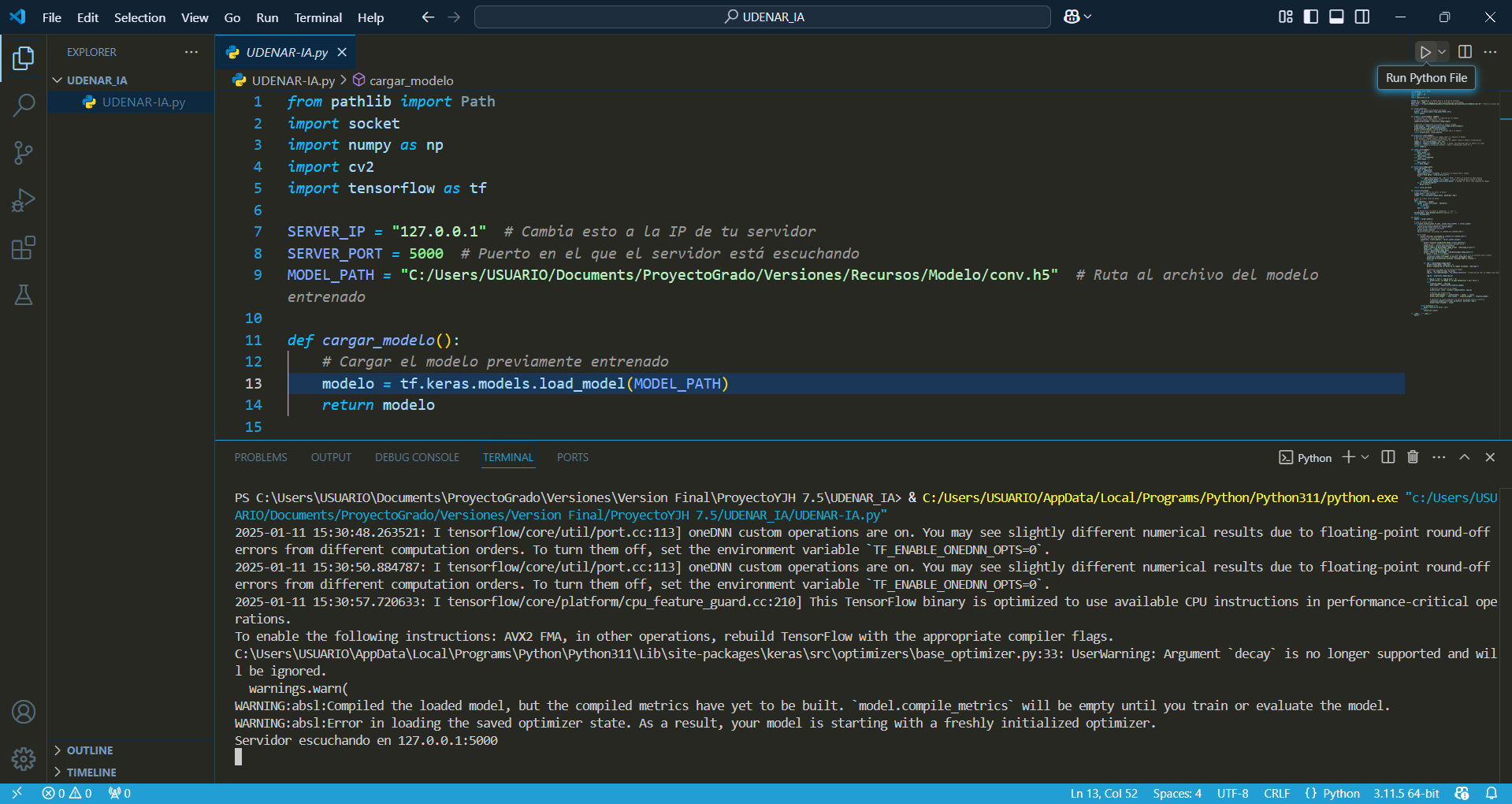


Ilustración 23 Ejecución de UDENAR-IA

Por últmo para ejecutar el modulo de predicciones en Visual Studio Code debes dar click en el archivo UDENAR-IA.py, luego click en el botón play ubicado en la parte superior derecha, en la parte inferior se abrirá la terminal y tendrá el siguiente mensaje “Servidor escuchando en 127.0.0.1:5000” lo que indica que está listo para su funcionamiento.



Ilustración 24 Ventana principal de UDENARSQL