



DIRECCIÓN ACADÉMICA
VICERRECTORADO ACADÉMICO

Facultad de Ingeniería

Carrera de Ciencia de Datos e Inteligencia Artificial

Informe de Actividad de Investigación Formativa

**Periodo Académico
Octubre 2025 – febrero 2026**



Contenido

1.	Autores.....	3
2.	Personal Académico	3
3.	Resultados de Aprendizaje de la asignatura:	3
4.	Tema de la Actividad de la Investigación Formativa:	4
5.	Objetivos de la(s) actividad(es):	5
6.	Fecha de la ejecución:.....	5
7.	Desarrollo del Informe.....	6
7.1	Introducción.....	6
7.2	Descripción de la metodología	7
7.3	Descripción de la(s) acción(es) realizadas.....	8
7.4	Resultados.....	10
7.5	Bibliografía.....	11
8.	ANEXOS (Evidencias)	12



1. Autores

- Janeta Saigua Guadalupe Margarita

2. PERSONAL ACADÉMICO

- Director de Carrera:
Ing. Milton López Ramos Mgs.
- Profesor de Asignatura:
Mgs. José Andrés Zuñiga Cazorla

3. RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA:

- Comprender y aplicar los principios de la Programación Orientada a Objetos (POO), incluyendo encapsulamiento, herencia y polimorfismo en Python
- Implementar métodos especiales y sobrecarga de operadores en Python para mejorar la reutilización y legibilidad del código.
- Aplicar el patrón Singleton para gestionar instancias de clases de manera eficiente dentro de una aplicación en Python.
- Diseñar e implementar soluciones modulares siguiendo buenas prácticas de POO y principios SOLID en Python.
- Sustenta la gestión del conocimiento ajustada al entorno cultural y legal, tanto nacional como internacional
- Comprender la estructura y funcionamiento de los arreglos en NumPy para optimizar la manipulación de datos en Python.
- Utilizar operaciones matemáticas y estadísticas en NumPy para realizar cálculos eficientes en análisis de datos.
- Implementar técnicas avanzadas de indexación, segmentación y manipulación de arreglos multidimensionales en entornos de procesamiento numérico.
- Desarrollar aplicaciones prácticas que empleen NumPy para la optimización del procesamiento y análisis de datos en Python.
- Comprender los principios de transformación y limpieza de datos para mejorar la calidad de la información en análisis exploratorios.



- Utilizar herramientas de Python para detectar y manejar datos faltantes, valores atípicos y errores en los conjuntos de datos.
- Aplicar técnicas de visualización de datos con Matplotlib y Seaborn para interpretar patrones y tendencias en conjuntos de datos.
- Analizar la distribución y relaciones entre variables a través de métodos estadísticos y gráficos exploratorios
- Desarrollar aplicaciones modulares en Python utilizando estructuras reutilizables y organizadas.
- Implementar una API RESTful en Flask para gestionar peticiones HTTP y procesar datos dinámicos.
- Aplicar técnicas de concurrencia y programación asíncrona en aplicaciones web para mejorar su rendimiento.
- Integrar Flask con bases de datos SQL utilizando SQLAlchemy para la gestión eficiente de la información.
- Integra herramientas de programación modular en Python con bases de datos globales para modelar relaciones multivariadas y generar predicciones que orienten la toma de decisiones, articulando la innovación tecnológica (ODS 9) con la acción climática (ODS 13).

4. TEMA DE LA ACTIVIDAD DE LA INVESTIGACIÓN FORMATIVA:

Análisis predictivo del estrés académico en estudiantes mediante modelos de Machine Learning y técnicas de preprocesamiento de datos



5. OBJETIVOS DE LA(S) ACTIVIDAD(ES):

Objetivo general:

- Desarrollar un sistema web para el análisis predictivo del estrés académico en estudiantes, utilizando modelos de Machine Learning, estructuras de datos y visualización interactiva de resultados.

Objetivos Específicos

- Aplicar técnicas de limpieza, codificación y normalización de datos a partir de archivos CSV para garantizar la calidad de la información analizada.
- Entrenar y comparar modelos de Machine Learning, específicamente Regresión Logística y Arboles de Decisión, para la clasificación del nivel de estrés académico.
- Implementar estructuras de datos lineales y no lineales mediante el uso de arreglos y arboles de decisión para el procesamiento y análisis de la información
- Desarrollar un sistema web interactivo que permita la carga de datos, la ejecución de los modelos y la visualización de resultados.
- Visualizar los resultados del análisis y las predicciones mediante gráficos interactivos utilizando la librería Plotly para facilitar la interpretación de los datos

6. FECHA DE LA EJECUCIÓN:

03/02/2026



7. DESARROLLO DEL INFORME

7.1 Introducción.

El estrés académico es un problema frecuente en la población estudiantil universitaria y puede afectar de manera significativa el rendimiento académico, la salud física y el bienestar emocional de los estudiantes. Factores como la sobrecarga de actividades, la presión por obtener buenos resultados, la falta de una adecuada gestión del tiempo y el uso inadecuado de hábitos de estudio que contribuyen al aumento de los niveles de estrés, convirtiéndose en un tema de interés para el análisis desde distintas áreas del conocimiento.

La presente investigación formativa aborda esta problemática desde un enfoque interdisciplinario, integrando conceptos de ciencia de datos, Programación, estructura de datos e Inteligencia Artificial. Para ello, se utilizó un conjunto de datos relacionado con el estrés académico de estudiantes, el cual fue sometido a un proceso de limpieza, trasformación y análisis exploratorio, con el fin de garantizar la calidad de la información y facilitar su interpretación.

A partir de los datos preprocesados, se implementaron modelos de Machine Learning, específicamente Regresión Logística y Arboles de decisión, con el objetivo de predecir el nivel de estrés académico. El Árbol de Decisión permitió, además, reforzar el uso de estructuras de datos no lineales, mientras que el manejo de arreglos y cálculos estadísticos manuales evidencio la aplicación de estructuras de datos lineales dentro del desarrollo del proyecto.

Como complemento al análisis, se desarrolló un sistema web interactivo que permite la carga de archivos, el procesamiento de la información y la visualización de resultados de manera clara y dinámica. La interacción de gráficos interactivos mediante la librería Plotly facilitó la comprensión de los patrones identificados y mejoró la comunicación de los resultados obtenidos. De esta manera, el proyecto demuestra como la combinación de técnicas de análisis de datos, estructuras de datos y visualización interactiva puede contribuir a la compresión de problemáticas reales en el ámbito académico y apoyar la toma de decisiones orientadas a la prevención del estrés académico.



7.2 Descripción de la metodología

Para el desarrollo de la presente investigación formativa se aplicó una metodología de trabajo modular y progresiva, orientada al análisis de datos y a la implementación de técnicas básicas de inteligencia Artificial, integrando conceptos de Estructuras de Datos y desarrollo web.

En primer lugar, se realizó la recolección y preparación de los datos, utilizando un conjunto de datos en formato CSV de nominado Academic Stress Level, obtenido de la plataforma Kaggle. En esta etapa se llevó a cabo un proceso de preprocesamiento, en el cual se identificaron y trajeron valores nulos, además de codificación. Este procedimiento permitió que la información sea compatible con los algoritmos de Machine Learning utilizados posteriormente.

Como segunda fase, se diseñó la arquitectura del sistema, adoptando un enfoque modular para organizar el proyecto. Se separaron las funcionalidades en distintos archivos, diferenciando el preprocesamiento de datos, el entrenamiento de modelos, la visualización de resultados y la ejecución del servidor. Para la gestión del sistema y la interacción con el usuario, se implementó un servidor web utilizando el microframework Flask, el cual permitió recibir archivos, procesarlos y mostrar resultados de manera dinámica.

En la tercera fase se implementaron modelos de Inteligencia Artificial, entrenando y evaluando dos algoritmos de clasificación: Regresión logística y Árbol de Decisión. La Regresión Logística se utilizó como modelo base para comparación, mientras que el árbol de decisión fue seleccionado como modelo principal debido a su relación directa con las estructuras de datos no lineales, representadas mediante nodos y ramas. Para mejorar su comportamiento y evitar el sobreajuste, el modelo fue configurado con parámetros específicos con la profundidad máxima del árbol.



De manera complementaria, se integraron conceptos de estructura de datos, desarrollando una clase personalizada denominada Analizador Estructuras. A través de esta clase se implementaron algoritmos manuales para el cálculo de medidas de tendencia central, como la media y la mediana, recorriendo arreglos de datos mediante bucles y aplicada lógica de ordenamiento, sin el uso de funciones automáticas de librerías externas. Esto permitió reforzar el aprendizaje de estructuras de datos lineales y algoritmos básicos.

Finalmente, se desarrolló la interfaz de usuario, diseñando un dashboard interactivo mediante el uso del framework Bootstrap, con el objetivo de presentar los resultados de forma clara y ordenada. Para la visualización de los datos se utilizó la librería Plotly, la cual permitió generar gráficos de dispersión interactivos que facilitan la interpretación de los resultados y sirven como evidencia del análisis realizado.

7.3 Descripción de la(s) acción(es) realizadas

A. Fase de Ejecución y seguimiento

Durante esta fase se llevó a cabo la implantación práctica del sistema propuesto, transformando el diseño teórico en una aplicación funcional. En primer lugar, se realizó la codificación modular del proyecto, separando las funcionalidades en distintos archivos, lo que permitió organizar de mejor manera el proceso de desarrollo y facilitar la corrección de errores.

Posteriormente, se implementó el módulo de preprocesamiento de datos, encargado de limpiar el data set, manejar valores nulos y codificar las variables categóricas en valores numéricos, asegurando que los datos fueran compatibles con los modelos de Machine Learning. Durante esta etapa se identificaron errores relacionados con el formato de los archivos cargados por el usuario, por lo que se implementó una excepción personalizada y bloques de control try-except, con el objetivo de mejorar la robustez y estabilidad del sistema.



En cuanto al modelado, se entrenaron y evaluaron los algoritmos de regresión logística y Árbol de Decisión, realizando diferentes pruebas para ajustar los parámetros del árbol, con la profundidad máxima, con el fin de evitar el sobreajuste y mejorar su comportamiento general. Este proceso permitió comprender como la estructura del modelo influye directamente en los resultados obtenidos.

Finalmente, se integraron los distintos módulos dentro de un servidor web desarrollando con Flask, permitiendo que el usuario cargue el data set y obtenga los resultados de manera interactiva, incluyendo métricas de desempeño y visualizaciones gráficas.

B. Fase de socialización y Reflexión

Una vez finalizada la implementación del sistema, se procedió a la evaluación y análisis de los resultados obtenidos, no solo desde el punto de vista técnico, sino también académico. Se verificó que la aplicación cumple con su objetivo principal de analizar y predecir el nivel de estrés académico, además de ofrecer una explicación clara del proceso mediante una nota técnica incluida en la interfaz.

Durante esta fase se reflexionó sobre la importancia de integrar Inteligencia Artificial y Estructura de Datos en proyectos académicos, ya que esta combinación permite desarrollar sistemas más transparentes y comprensibles. El uso del Árbol de decisión facilitó la interpretación de los resultados al representar el proceso de clasificación mediante nodos y ramas, mientras que los cálculos manuales de la media y la mediana permitieron contrastar los resultados estadísticos con los obtenidos por los modelos.

Finalmente, se concluyó que, aunque la precisión del modelo no fue elevada, el proyecto cumplió con su propósito formativo, permitiendo fortalecer conocimientos en programación, análisis de datos y Machine Learning, así como comprender la complejidad del estrés académico como fenómeno multifactorial.



7.4 Resultados

Tras la ejecución del sistema de análisis de estrés académico, obtuvieron los siguientes resultados a partir del procesamiento del conjunto de datos seleccionado:

El modelo de Árbol de Decisión alcanzo una precisión del 17.86% en la clasificación de los niveles de datos de entrenamiento y prueba. La estructura generada por el modelo permitió representar el proceso de clasificación mediante una jerarquía de decisiones, organizada en nodos y ramas.

Mediante la clase personalizada AnalizadorEstructuras, se procesaron los resultados generados por el modelo utilizando arreglos de datos. A partir de este procesamiento se calcularon las medidas de tendencia central, obteniéndose una media de 3.75 y una mediana de 4.00, lo que indica una mayor concentración de los valores en niveles medios-altos de estrés académico dentro del conjunto de datos analizado.

Adicionalmente, el sistema presentó los resultados a través de una dashboard interactivo, donde se visualizaron de forma clara la precisión del modelo, las métricas estadísticas y el gráfico de dispersión. El gráfico interactivo generado con Plotly permitió observar la relación entre los valores reales y los valores predichos, facilitando el análisis visual de los datos procesados.



7.5 Bibliografía

Corbo, A. (03 de junio de 2025). *¿Qué es un árbol de decisiones?*

<https://builtin.com/machine-learning/decision-tree>

hiberus. (31 de Octubre de 2025). *Machine Learning: cómo el aprendizaje automático impulsa la transformación digital de las empresas.*

<https://www.hiberus.com/crecemos-contigo/machine-learning-la-inteligencia-artificial-transforma-negocio/>

kaggle . (2026). *Student Academic Stress Level.*

<https://www.kaggle.com/datasets/ayeshaimran1619/student-academic-stress-level?resource=download>

Pavlovych, A. (31 de Diciembre de 2025). *¿Qué es la visualización de datos? Herramientas y ejemplos.* <https://www.planeks.net/data-visualization-the-complete-guide/>

Scielo . (junio de 2015). *Estrés académico.*

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2077-28742015000200013



8. ANEXOS (EVIDENCIAS)

BLOQUE 3: Exploración inicial

Estas funciones permiten conocer la estructura del dataset, el tipo de variables, la cantidad de datos disponibles y detectar posibles valores nulos o inconsistencias.

```
df.head()
df.info()
df.describe()

✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140 entries, 0 to 139
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Timestamp        140 non-null    object  
 1   Your Academic Stage 140 non-null    object  
 2   Peer pressure     140 non-null    int64  
 3   Academic pressure from your home 140 non-null    int64  
 4   Study Environment 139 non-null    object  
 5   What coping strategy you use as a student? 140 non-null    object  
 6   Do you have any bad habits like smoking, drinking on a daily basis? 140 non-null    object  
 7   What would you rate the academic competition in your student life 140 non-null    int64  
 8   Rate your academic stress index      140 non-null    int64  
dtypes: int64(4), object(5)
memory usage: 10.0+ KB
```



```
from sklearn.metrics import accuracy_score

print("Accuracy SIN selección:", accuracy_score(y_test, y_pred_log))
print("Accuracy CON selección:", accuracy_score(y_test, y_pred_fs))

✓ 0.0s
```

Python

Accuracy SIN selección: 0.35714285714285715
Accuracy CON selección: 0.35714285714285715

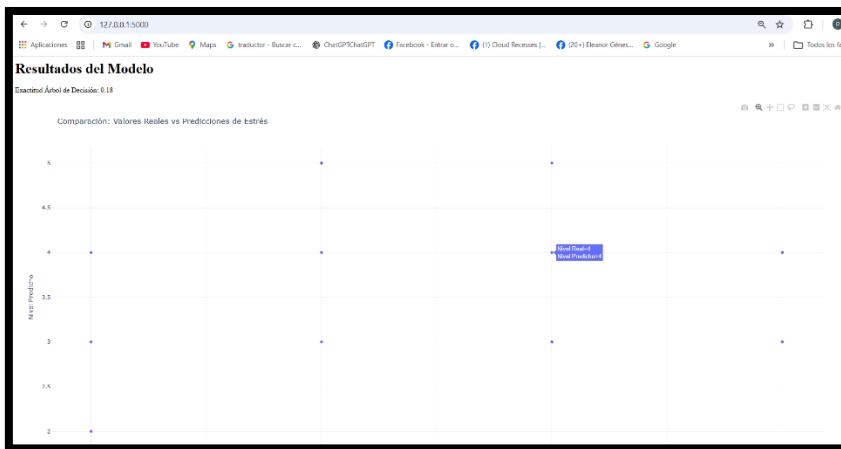
```
Símbolo del sistema
Downloaded plotly-6.5.2-py3-none-any.whl (9.9 MB) 9.9/9.9 MB 8.0 MB/s eta 0:00:00
Downloaded blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloaded click-8.3.1-py3-none-any.whl (108 kB) 108.3/108.3 kB 3.2 MB/s eta 0:00:00
Downloaded itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloaded jinja2-3.1.6-py3-none-any.whl (134 kB) 134.9/134.9 kB 7.8 MB/s eta 0:00:00
Downloaded markupsafe-3.0.3-cp311-cp311-win_amd64.whl (15 kB)
Downloaded narwhals-2.16.0-py3-none-any.whl (443 kB) 444.0/444.0 kB 14.0 MB/s eta 0:00:00
Downloaded werkzeug-3.1.5-py3-none-any.whl (225 kB) 225.0/225.0 kB 6.7 MB/s eta 0:00:00
Installing collected packages: narwhals, markupsafe, itsdangerous, click, blinker, werkzeug, plotly, jinja2, flask
  WARNING: The script plotly.get_chrome.exe is installed in 'C:\Users\WinUser\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script flask.exe is installed in 'C:\Users\WinUser\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed blinker-1.9.0 click-8.3.1 flask-3.1.2 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.3 narwhals-2.16.0 plotly-6.5.2 werkzeug-3.1.5

[notice] A new release of pip is available: 24.0 -> 26.0.1
[notice] To update, run: C:\Users\WinUser\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
C:\Users\WinUser>
C:\Users\WinUser>
C:\Users\WinUser>
```



```
Symbolo del sistema x + v
parso 0.8.5
pillow 12.1.0
platformdirs 4.5.0
plotly 6.5.2
prompt_toolkit 3.0.52
psutil 7.1.1
pycparser 0.2.3
Pygments 2.19.2
pynarsing 3.3.1
python-dateutil 2.9.0.post0
pytz 2025.2
pyzmq 27.1.0
scikit-learn 1.8.0
scipy 1.16.3
seaborn 0.13.2
six 1.17.0
stack-data 0.6.3
threadpoolctl 3.6.0
tornado 6.5.2
traitlets 5.1.10.3
typing_extensions 4.15.0
tzdata 2025.3
width 0.2.14
Werkzeug 3.1.5

[notice] A new release of pip is available: 24.0 -> 26.0.1
[notice] To update, run: C:\Users\WinUser\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
C:\Users\WinUser>
```



EXPLORADOR

- INVESTIGACIÓN FORMATIVA
- Archivo CSV
- Introducción para la app
- README.md
- visualización generada
- newplot.png
- app.py
- modelo.py
- preprocesamiento.py
- visualización.py

app.py x modelo.py newplot.png

```
from flask import Flask, render_template, request, send_file
import pandas as pd
import numpy as np
from preprocesamiento import limpiar_datos
from modelo import entrenar_modelos
from visualización import crear_grafico_resultados
app = Flask(__name__)
# 5. EXERCICIO PERSONALIZADA (Punto 5 de la rúbrica)
class ArchivoInvalidoError(Exception):
    pass
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # Procesamiento de datos
        # Llamada a la función de preprocesamiento
        # Llamada a la función de entrenamiento
        # Generación del gráfico de resultados
    return render_template('index.html')
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

```
PS C:\Users\WinUser\OneDrive\Desktop\Investigación Formativa Programación 2 Janeta Guadalupe Paralelo A> & C:\Users\WinUser\AppData\Local\Microsoft\WindowsApps\Python3.11.exe "C:\Users\WinUser\OneDrive\Desktop\Investigación Formativa Programación 2 Janeta Guadalupe Paralelo A\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: Do not use a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 102-091-956
* Debugger PIN: 102-091-956
127.0.0.1 - [06/feb/2026 12:55:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - [06/feb/2026 12:55:38] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - [06/feb/2026 12:55:38] "GET /favicon.ico HTTP/1.1" 404 -
Intento de procesamiento finalizado.
127.0.0.1 - [06/feb/2026 12:58:37] "POST / HTTP/1.1" 200 -
Intento de procesamiento finalizado.
127.0.0.1 - [06/feb/2026 12:58:37] "POST / HTTP/1.1" 200 -
127.0.0.1 - [06/feb/2026 12:58:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - [06/feb/2026 12:58:37] "POST / HTTP/1.1" 200 -
```

ESQUEMA

LÍNEA DE TIEMPO



```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
127.0.0.1 - - [06/Feb/2026 12:58:54] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [06/Feb/2026 12:58:54] "POST / HTTP/1.1" 200 -
* Detected change in 'c:\Users\WinUser\OneDrive\Desktop\Investigacion formativa Programacion 2 Janeta Guadalupe Paralelo A\modelo.py', r
* Detected change in 'c:\Users\WinUser\OneDrive\Desktop\Investigacion formativa Programacion 2 Janeta Guadalupe Paralelo A\modelo.py', r
* reloading
* Rebooting with stat
* Debugger is active!
* Debugger PIN: 102-061-956
127.0.0.1 - - [06/Feb/2026 13:03:28] "GET / HTTP/1.1" 200 -
Intento de procesamiento finalizado.
127.0.0.1 - - [06/Feb/2026 13:03:50] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [06/Feb/2026 13:07:20] "GET / HTTP/1.1" 200 -
Intento de procesamiento finalizado.
127.0.0.1 - - [06/Feb/2026 13:07:35] "POST / HTTP/1.1" 200 -
* Detected change in 'c:\Users\WinUser\OneDrive\Desktop\Investigacion formativa Programacion 2 Janeta Guadalupe Paralelo A\modelo.py', r
* reloading
* Rebooting with stat
* Debugger is active!
* Debugger PIN: 102-061-956
Lín. 21, Col. 1 (159 seleccionada) Espacios: 4 UTF-8
```

Investigación Formativa: Estrés Académico

Sube el CSV para procesar mediante IA y Estructuras de Datos

Seleccionar archivo Sin archivos seleccionados

Analizar Datos

