

TECNOLÓGICO DE MONTERREY

Escuela de Ingeniería y Ciencias

Campus Puebla

TI3001C

ANALÍTICA DE DATOS Y HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL I



Tecnológico
de Monterrey

Facilitador: Rigoberto Cerino Jiménez
cerino.rigoberto@tec.mx

28/Agosto/2025

ACTIVIDADES DE REFUERZO



- ▶ Solicitar al usuario que ingrese una frase y luego imprimir la cantidad de vocales que se encuentran en dicha frase.
- ▶ Escribir un código que calcule el interés que genera un monto ingresado a través de los años. Se solicita al usuario un capital C , a un interés I (que oscila entre 0 y 100 y hace referencia al porcentaje), durante M años y se desea saber en cuanto se habrá convertido ese capital en “ M ” años, sabiendo que es acumulativo.
- ▶ Solicitar al usuario que ingrese un número N de salarios en una empresa. Mostrar en pantalla el número de usuario que percibe el salario mayor.
- ▶ El día juliano correspondiente a una fecha es un número entero que indica los días que han transcurrido desde el 1 de enero del año indicado. Queremos crear un programa principal que al introducir una fecha nos diga el día juliano que corresponde. Para ello podemos hacer las siguientes subrutinas:
 - **LeerFecha:** Nos permite leer por teclado una fecha (día, mes y año).
 - **DiasDelMes:** Recibe un mes y un año y nos dice los días de ese mes en ese año.
 - **EsBisiesto:** Recibe un año y nos dice si es bisiesto.
 - **Calcular_Dia_Juliano:** recibe una fecha y nos devuelve el día juliano.

Nota: para que un año sea bisiesto debe ser divisible por 4 y no debe ser divisible por 100, excepto que también sea divisible por 400.

- ▶ Escribe un programa que:

Pida al usuario dos números y un operador matemático (+, -, *, /, **).

Convierta los números a float.

Realice la operación ingresada y muestre el resultado.

Controle posibles errores como: División por cero, entrada no numérica y operador inválido.

- ▶ Escribe un programa que reciba el gasto mensual de un cliente y lo clasifique en las siguientes categorías:

- **"Cliente Básico"** si gasta menos de \$500.

- **"Cliente Frecuente"** si gasta entre \$500 y \$2000.

- **"Cliente VIP"** si gasta más de \$2000.

- ▶ Además, si el cliente es VIP, **debe recibir un 5% de descuento en su próxima compra.**

- ▶ Escribe un programa que pida un número N y genere la secuencia de Fibonacci hasta el N-ésimo término.

Tip: La secuencia de Fibonacci comienza con 0 y 1, y cada número siguiente es la suma de los dos anteriores.

Ejercicio 1: Análisis de Ventas de Productos Electrónicos

Descripción: Esta actividad consiste en realizar un programa en Python que lea datos de un archivo de texto llamado Ventas_semanales.txt, el cual contiene información sobre las ventas diarias de productos electrónicos en una empresa durante una semana. El programa debe ofrecer al usuario varias opciones para analizar y mostrar estos datos.

Instrucciones:

Leer y Procesar el Archivo de Ventas: El archivo Ventas_semanales.txt contiene datos de ventas en el siguiente formato:

<fecha producto cantidad precio_unitario>.

Cada línea del archivo representa una venta individual y está separada por espacios. Asegúrate de que el archivo se encuentra en la misma carpeta que el script de Python.

Implementar un Menú de Opciones:

Debes crear un menú que muestre las siguientes opciones al usuario (**opcional**):

1. Mostrar ventas totales por producto: Calcula el total de ventas de cada producto y muestra el resultado en la consola.
2. Mostrar el día con más ventas por producto: Determina el día en el que se registraron más ventas para cada producto y muestra esta información (**opcional**).
3. Imprimir el contenido del archivo: Muestra el contenido completo del archivo ventas_semanales.txt en la consola.
4. Salir: Finaliza la ejecución del programa.

▸ Ejercicio 1: Consolidación de Presupuestos de Departamentos

Contexto: La empresa tiene **tres departamentos** con presupuestos separados en diferentes archivos.

▸ Datos iniciales

presupuesto_marketing.csv

presupuesto_ventas.csv

presupuesto_finanzas.csv

▸ Tareas:

Cargar los tres archivos CSV en DataFrames diferentes.

Usar `pd.concat()` para unir los DataFrames en una única tabla.

Agregar una columna con el porcentaje de gasto respecto al presupuesto.

Ordenar el DataFrame por año.

Preguntas de análisis:

¿Qué departamento tuvo el mayor gasto en relación con su presupuesto?

¿Cuál tuvo la mayor diferencia entre presupuesto y gasto?

▸ Ejercicio 2: Registro de Pedidos en Diferentes Ciudades

Contexto: Se tienen registros de pedidos en diferentes ciudades, cada una en un archivo distinto.

▸ Datos iniciales

CDMX.csv

Guadalajara.csv

Monterrey.csv

▸ Tareas:

Unir los tres DataFrames en uno solo usando concat().

Agregar una columna con el nombre de la ciudad según el archivo de origen.

Calcular el total de ingresos por ciudad.

Preguntas de análisis:

¿Qué ciudad generó más ingresos?

¿Cuántos pedidos se registraron en total?

▸ Relación entre Empleados y Salarios

Contexto: Se tiene una lista de empleados y otra con sus salarios, pero no todos los empleados tienen salario registrado.

▸ Datos iniciales

empleados.csv

salarios.csv

▸ Tareas:

- Fusionar ambos DataFrames con merge() para ver qué empleados tienen salario registrado.
- Realizar el tipo de unión para identificar empleados sin salario registrado.
- Realizar el tipo de unión para identificar salarios sin empleados registrados.

Preguntas de análisis:

¿Cuántos empleados no tienen salario registrado?

¿Existe algún salario sin un empleado correspondiente?

▸ Relación entre Proveedores y Productos

Contexto: Se tiene una lista de proveedores y otra con productos, pero no todos los productos tienen un proveedor asignado.

▸ Datos iniciales

proveedores.csv
productos.csv

▸ Tareas:

- Unir los DataFrames usando merge().
- Usar how="left" para identificar productos sin proveedor asignado.
- Usar how="right" para encontrar proveedores sin productos asociados.

Preguntas de análisis:

- ¿Cuántos productos no tienen un proveedor asignado?
- ¿Cuántos proveedores no tienen productos registrados?