

# Examen Práctico: Arrays y Cadenas en Python

## Instrucciones Generales:

- **Duración:** 60 minutos (1 hora).
  - **Recursos:** Prohibido el uso de internet, apuntes o colaboración. Solo puede utilizar el IDE Thonny.
  - **Estructura:** El examen se divide en dos secciones principales, que se calificarán por separado: arrays y cadenas.
  - **Entrega:** Guarda el archivo Python con tu nombre y súbelo a la tarea de Moodle. Puedes añadir el texto que necesites mediante comentarios en el código.
  - **Evaluación:** Todos los ejercicios valen lo mismo. Los dos primeros hacen media aritmética para el RA6a (Se han escrito programas que utilicen arrays) y los dos segundos igual para el RA6g (Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.).
- 

## Ejercicio 1: Gestión de Inventario (Búsqueda y Máximo)

Se le proporciona una matriz (lista de listas) que simula el inventario de una pequeña tienda. Cada fila representa un producto con la estructura:

ID_Producto	Precio_USD	Stock_Unidades
101	15.50	45
102	5.00	120
103	30.99	15
104	8.75	70
105	22.00	30

Debe crear un programa que realice las siguientes acciones:

1. **Búsqueda por ID:** Pida al usuario un **ID\_Producto**. Recorra la matriz y, si lo encuentra, imprima su precio y stock. Si no lo encuentra, muestre el mensaje: "**ID no encontrado.**".
2. **Producto más Caro:** Calcule y muestre el **ID** del producto que tiene el **precio más alto** en el inventario.

---

## Ejercicio 2: Procesamiento de Matriz (Transposición Parcial)

Dada la siguiente matriz cuadrada M:

$$M = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

1. Cree una **nueva matriz** T (transpuesta) de 3x3.
2. Copie el contenido de la matriz M a T, pero realice una **transposición parcial**: solo los elementos que NO están en la **diagonal principal** deben ser transpuestos.
3. Los elementos de la **diagonal principal** deben permanecer en su posición original en T.

$$T = \begin{bmatrix} 10 & 40 & 70 \\ 20 & 50 & 80 \\ 30 & 60 & 90 \end{bmatrix}$$

---

## Ejercicio 3: Análisis de Texto y Conteo (Cadenas)

Se le proporciona el siguiente texto:

```
texto = "El Python es un lenguaje de programación muy popular. Python se usa en ciencia de datos y desarrollo web."
```

Debe crear un programa que, utilizando métodos de cadena:

1. **Conteo de Palabras Clave:** Cuente cuántas veces aparece la palabra "**Python**" en el texto, considerando que debe ser la palabra completa (es decir, que no debe contar subcadenas si fuera el caso). *Pista: Considere convertir el texto a minúsculas y usar un método de cadena.*
2. **Inversión de Frase:** Convierta el texto a **mayúsculas** e invierta el **orden de las palabras** (no las letras), para que se lea de atrás hacia adelante.

**Ejemplo de Inversión de Palabras:** "Hola mundo" → "mundo Hola".

---

## Ejercicio 4: Validación de Formato (Expresiones Regulares)

El departamento de sistemas requiere una función de validación de un código de producto con el siguiente formato estricto:

### Formato Requerido:

- Debe tener una longitud exacta de **8 caracteres**.
- Debe empezar con dos **letras mayúsculas (A-Z)**.
- Debe ser seguido por un **guion (-)**.
- Debe terminar con **tres dígitos (0-9)**.
- Las dos posiciones restantes (4 y 5) deben contener **letras minúsculas (a-z)**.

Escriba una función que reciba una cadena (el código) y:

1. Utilice una **única expresión regular** para validar el formato.
  2. Devuelva **True** si el código es válido y **False** si no lo es.
  3. Pruebe su función con los siguientes ejemplos e imprima el resultado:
    - AB-xy123 (Debe ser **True**)
    - A B-xy123 (Debe ser **False**)
    - ZX-aBc555 (Debe ser **False**)
    - XY-ab99 (Debe ser **False**)
- 

## Referencias de ayuda

Método
<code>cadena.lower()</code>
<code>cadena.upper()</code>

`cadena.split()`

`separador.join(lista)`

`len(cadena)`

`cadena[inicio:fin:paso]`

Carácter	Descripción	Ejemplo	Coincide con
<code>^</code>	Coincide con el <b>inicio</b> de la cadena.	<code>^A</code>	La cadena empieza por 'A'.
<code>\$</code>	Coincide con el <b>final</b> de la cadena.	<code>5\$</code>	La cadena termina con '5'.

