

Facultad De Ingeniería En Electricidad Y Computación  
CCPG1043 / CCPG1801 - Fundamentos De Programación  
Tercera Evaluación / PAE 2026-2027/ Abril 24, 2026

**Compromiso de Honor**

Yo, ....., matrícula ..... del paralelo ..... de Fundamentos de Programación, declaro que he sido informado y conozco las normas disciplinarias que rigen a la ESPOL, en particular el Código de Ética y el Reglamento de Disciplina. Al aceptar este compromiso de honor, asumo la responsabilidad de realizar este examen de manera honesta, sin recurrir a prácticas de plagio, fraude o deshonestidad académica por medios físicos o electrónicos. Me comprometo a responder las preguntas con mis propias habilidades y conocimientos, sin recibir ayuda no autorizada. Además, garantizo que respetaré los derechos de propiedad intelectual y **no divulgaré ni copiaré el contenido del examen.**

**Acepto el presente compromiso**, como constancia de haber leído y aceptado la declaración anterior y me comprometo a seguir fielmente las directrices que se indican para la realización de la presente evaluación. Estoy consciente que el incumplimiento del presente compromiso anulará automáticamente mi evaluación y podría ser objeto del inicio de un proceso disciplinario.

**Firma compromiso:** .....

| Tema 1 | Tema 2 | Tema 3 | Tema 4 |                                 |
|--------|--------|--------|--------|---------------------------------|
|        |        |        |        | Firma<br>Revisé mi calificación |

**Funciones, métodos y accesos de referencia en Python.**

| Cadenas   | Listas  | Diccionarios:  | Pandas:   | pandas as pd   |
|---|---|--|---|--|
| cadena.islower()<br>cadena.isupper()<br>cadena.isdigit()<br>cadena.isalpha()<br>cadena.lower()<br>cadena.upper()<br>cadena.split(separador)<br>cadena.capitalize()<br>cadena.title()<br>cadena.count(subcadena)<br>cadena.replace(actual, nueva)<br>int(cadena) | lista.append(...)<br>lista.extend(...)<br>lista.count(...)<br>lista.index(...)<br>lista.pop()<br>lista.insert(pos, elem )<br>elemento in lista<br>range(inicio,fin, salto )<br>separador.join(lista )<br>len(lista)<br>sum(lista)<br>max(lista) | clave,valor in dic.items()<br>clave in dic<br>dic.keys()<br>dic.values()<br>dic[clave]<br>dic.get(clave, valor_defecto)<br>dic.setdefault(clave, valor_defecto)<br>dic.update(nuevo_dic) | df =<br>pd.DataFrame(...)<br>df.head()<br>df.tail()<br>df.info()<br>df.describe()<br>df["col"]<br>df[ lista_col ]<br>df.loc[ fila ]<br>df.iloc[ idx_fila ]<br>df["col"].sum()<br>df["col"].mean()<br>df["col"].unique() | df["col"].idxmax()<br>df["col"].reset_index()<br>df["col"].max()<br>df["col"].min()<br>df["col"].count()<br>df["col"].value_counts()<br>df.sort_values(by= "col", ascending=True )<br>df.groupby("col1")["col2"].sum()<br>df["col"].mask(condición, valor)<br>pd.read_csv("file.csv")<br>df.to_csv("file.csv") |

**IMPORTANTE:**

- Use únicamente estructuras vistas en clase.
- No use estructuras no permitidas.
- Cada función debe cumplir exactamente con lo solicitado.
- No se requiere validación adicional.

## TEMA 1 [20 Puntos]

En el sistema se ingresa una cadena de texto con números enteros positivos separados por comas, tal como en el **ejemplo**: "12,5,abc,30,7,0,15"

En los valores que se ingresa, se asume que siempre existirá al menos un número válido.

**1.1 [5 Puntos]** Debe implementar un programa que, usando únicamente un ciclo while recorra todos los elementos de la cadena de texto.

**1.2 [5 Puntos]** Al recorrer cada uno de los elementos, debe:

- Validar si es un número entero positivo mayor que cero.
- Solo si cumple con la condición, debe de agregarlo a una lista llamada: **Lst\_Num\_Validos**

**1.2 [10 Puntos]** Al finalizar el proceso, deberá:

- Mostrar la lista completa con los valores almacenados.
- Presentar la cantidad de números que tiene la lista.
- Presentar la suma total de todos los elementos.
- Presentar cuál es el número mayor de todos.
- Presentar cuál es el número menor de todos
- Presentar el promedio de los elementos de la lista

**Restricciones obligatorias:**

- Solo puede usar el ciclo while para el recorrido

**Ejemplo:**

La cadena de texto ingresada sería: "12,5,abc,30,7,0,15"

**El resultado que se espera es:**

- lst\_validos = [12, 5, 30, 7, 15]
- suma = 69
- máximo = 30
- mínimo = 5
- promedio = 13.8

## TEMA 2 [20 Puntos]

Se dispone de una lista de palabras: ["rojo", "azul", "rojo", "verde", "azul", "rojo"]

**2.1 [10 Puntos]** Debe implementar un programa que, a partir de la lista, construya un diccionario donde:

- La clave sea cada palabra (color)
- El valor sea la cantidad de veces se repite esa palabra en la lista

**2.2 [10 Puntos]** Cuando tenga el diccionario armado debe:

- Mostrar el diccionario completo (**2 Pts**)
- Mostrar la palabra que más se repite (**4 Pts**)
- Mostrar cuántas veces aparece esa palabra (**4 Pts**)

**Usando el ejemplo, la salida esperada es:**

```
{"rojo": 3, "azul": 2, "verde": 1}
Palabra más frecuente: rojo
Cantidad: 3
```

### TEMA 3 [30 Puntos]

El sistema recibe una cadena de texto con registros de usuarios en el siguiente formato:

- "ana,23,activo&LUIS,19,inactivo&maria,abc,activo&PEDRO,30,activo"

#### Considere:

- I. La estructura del registro tiene: **nombre,edad,estado**
- II. Cada grupo está separado del otro por medio de del signo: "&"
- III. Se asume que siempre el ingreso de la cadena de texto será ingresado como en el ejemplo.

**3.1 [5 Puntos]** Debe implementar la función **Fnc\_Procesar\_Registros(cadena)**, la cual:

- a) Debe separar los grupos de registros por el caracter "&".
- b) La función debe de devolver una lista, la cual debe contener todos los grupos de registros que recuperó de la cadena original.

**3.2 [19 Puntos]** Debe implementar la función **Fnc\_Separa\_Campos(Lst\_Grupo\_Registros)**, la cual:

1. Dentro de cada grupo de registro que recuperó, ahora debe de separar los campos por medio de la "," (3 pts).
2. Cuando recupere los campos debe, en el orden en que se indica a continuación, realizar los siguientes procesos ( de no hacerlo en el orden solicitado, tendrá una penalización del 40% sobre el puntaje que obtenga en esta pregunta):

- a) **Campo nombre**, debe validar que solo tiene letras, si cumple con esta condición lo debe de normalizar, sino cumple, debe de mantenerlo igual (3 pts).
- b) **Campo edad**: debe validar que el valor del campo sea un valor numérico, entero positivo entre 1 y 100, en caso de que no cumpla con estas condiciones debe de reemplazar el valor del campo por un -1 (3 pts).
- c) **Campo estado**: debe validar que en campo estado se encuentra la palabra "activo" o "inactivo" en cualquier combinación posible, **ejemplo**: ACTivo, activo, actiVO, InacTivo; si cumple con esta condición, debe de convertirlo a mayúscula, caso contrario, deberá reemplazar el valor por la palabra: **"NONE"** (3 pts).

3. Una vez validados los campos, solo se deben considerar los registros que cumplan simultáneamente (3 pts):

- El campo estado sea igual a "ACTIVO"
- El campo edad sea diferente de -1

4. Solo si cumple con esta condición deberá (3 pts):

- Ingresar el valor del campo **nombre**, a la lista: **Lst\_Nombres\_Usuarios**
- Ingresar el valor del campo **edad**, a la lista: **Lst\_Edad\_Usuarios**

5. Al finalizar, la función debe de devolver las listas (1 pts): Lst\_Nombres\_Usuarios, Lst\_Edad\_Usuarios

**3.3 [6 Puntos]** Escriba un programa en Python que:

- i. Presente la cadena que será usara en el ejercicio (Use el ejemplo de esta pregunta) (2 pts).
- ii. Llama a cada una de las funciones que implemento en el orden correcto (2 pts).
- iii. Presente por pantalla las listas: Lst\_Nombres\_Usuarios, Lst\_Edad\_Usuarios (2 pts).

## TEMA 4 [30 Puntos]

Se dispone del archivo `ventas_tienda.csv`, que contiene información sobre las ventas realizadas en distintas ciudades a través de una plataforma digital, el archivo tiene las siguientes columnas

|   |
|---|
| Producto,Categoria,Ciudad,Cantidad,Precio |
|---|

### La estructura del archivo es la siguiente:

|  |
|--|
| Producto -> Nombre del producto registrado en el sistema<br>Categoria ->(Ej: Electrónica, Hogar, Ropa, etc.)<br>Ciudad -> Lugar en donde se realizó la venta<br>Cantidad -> Cantidad de elementos que se vendió<br>Precio -> valor que se cobró al cliente |
|--|

### Ejemplo del archivo:

|  |
|--|
| Producto,Categoria,Ciudad,Cantidad,Precio<br>Laptop,Electronica,Guayaquil,2,900<br>Mouse,Electronica,Quito,5,20<br>Teclado,Electronica,Guayaquil,3,25<br>Silla,Hogar,Cuenca,1,150<br>Mesa,Hogar,Guayaquil,2,200<br>Laptop,Electronica,Quito,1,950<br>Mouse,Electronica,Cuenca,4,18 |
|--|

Debe implementar un programa en Python usando Pandas que realice lo siguiente:

#### 4.1 [2 puntos] Lectura:

- Cargue el archivo en un DataFrame llamado **df**

#### 4.2 [6 puntos] Presente:

- Muestre en pantalla las primeras 5 filas del DataFrame (2 pts).
- Muestre la información general del DataFrame (2 pts).
- Muestre un resumen estadístico de las columnas numéricas (2 pts).

#### 4.3 [8 puntos] Calcule:

- Calcule y muestre la suma total de la columna: **Cantidad** (2 pts).
- Calcule y muestre el promedio de la columna **Precio** (2 pts).
- Calcule y muestre el valor máximo de la columna **Precio** (2 pts).
- Calcule y muestre el valor mínimo de la columna **Precio** (2 pts).

#### 4.4 [2 puntos] Analice:

- Recupere y muestre los valores únicos de la columna **Categoria**

#### 4.5 [4 puntos] Agrupe:

- Calcule y muestre el total de **Cantidad** vendida por **Ciudad**

#### 4.6 [2 puntos] Ordene:

- Ordene el DataFrame por Precio de menor a mayor

#### 4.7 [4 puntos] Modifique:

- Reemplace los valores de **Cantidad** menores a 2 por 0 usando **mask** (2 pts).
- Muestre el DataFrame resultante (2 pts).

#### 4.8 [2 puntos] Exporte:

- Guarde el nuevo DataFrame (el resultado del 4.7) en un archivo llamado: `ventas_procesadas.csv`