

Facultad De Ingeniería En Electricidad Y Computación
 CCPG1043 / CCPG1801 - Fundamentos De Programación
 Segunda Evaluación / PAE 2026-2027/ Abril 17, 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	Total	
				_____ Firma Revisé mi calificación

Funciones, métodos y accesos de referencia en Python.

Diccionarios:	Pandas:	pandas as pd
clave,valor in dic.items() clave in dic dic.keys() dic.values() dic[clave] dic.get(clave, valor_defecto) dic.setdefault(clave, valor_defecto) dic.update(nuevo_dic)	df = pd.DataFrame(...) df.head() df.tail() df.info() df.describe() df["col"] df[lista_col] df.loc[fila] df.iloc[idx_fila] df["col"].sum() df["col"].mean() df["col"].unique()	df["col"].idxmax() df["col"].reset_index() df["col"].max() df["col"].min() df["col"].count() df["col"].value_counts() df.sort_values(by= "col", ascending=True) df.groupby("col1")["col2"].sum() df["col"].mask(condición, valor) pd.read_csv("file.csv") 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 [35 Puntos]

Se requiere desarrollar un sistema de registro de calificaciones de estudiantes con el siguiente formato: "nombre;nota"

El ingreso termina cuando el usuario escribe exactamente: FIN

Ejemplo:

```
Ingrese registro: Juan Perez;8
Ingrese registro: Maria Aguirre;9
Ingrese registro: Juan Perez;7
Ingrese registro: FIN
```

Requerimientos:

1.1 [10 Puntos] validar_registro(s) Esta función retorna **True** si la cadena **s** cumple con todo lo siguiente:

- Tiene 2 partes separadas por ;
- Nombre no vacío
- Nota es de tipo entero
- Nota válida entre 0 y 10

Caso contrario debe retornar **False**.

1.2 [10 Puntos] registrar_nota(dic, nombre, nota) Esta función almacenará los datos en el diccionario, considere lo siguiente:

- Usa el nombre como clave
- Guarda las notas como lista de enteros
- Si existe el nombre, debe de agregar la nueva nota sin borrar las anteriores.

1.3 [5 Puntos] calcular_promedios(dic) Esta función recibe el diccionario de las notas de los estudiantes y retorna un nuevo diccionario con el promedio de cada estudiante con el siguiente formato:

```
{nombre: promedio}
```

1.4 [10 Puntos] Programa principal:

Se debe:

- Declarar un diccionario vacío
- Leer registros hasta FIN
- Validar cada registro ingresado:
 - Si es válido, registrarlo al diccionario
 - Sino, ignorarlo
- Al final:
 - Mostrar diccionario de notas
 - Mostrar promedios

TEMA 2 [20 Puntos]

Una clínica veterinaria lleva el control de las mascotas atendidas durante el día. La información que ingresan en el sistema tiene la siguiente estructura:

Ejemplo: (Recuerde que el diccionario puede tener más datos, pero mantiene la estructura)

```
d_atenciones = {  
    "Dr. Pérez": ["Luna", "Max", "Luna"],  
    "Dra. López": ["Max", "Sol", "Sol"],  
    "Dr. Ramos": ["Luna", "Toby", "Max", "Toby", "Toby"]  
}
```

Cada clave representa un veterinario y cada lista contiene las mascotas que atendió, una mascota puede aparecer repetida con varios doctores si tuvo varias atenciones.

Desarrolle:

2.1 [20 puntos] `atenciones_por_mascota(dic, mascota)` Esta función recibe el diccionario de atenciones y el nombre de una mascota.

Retorna un nuevo diccionario con la cantidad de veces que dicha mascota fue atendida por cada veterinario. **Formato:** {veterinario: cantidad}

Ejemplo:

```
d_mascota = {  
    "Dr. Pérez": 2,  
    "Dra. López": 0,  
    "Dr. Ramos": 1  
}
```

TEMA 3 [45 Puntos]

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

```
Ciudad,Categoria,Ventas,Clientes,Ingresos
...
```

- **Ciudad**
- **Categoria** (Ej: Electrónica, Hogar, Ropa, etc.)
- **Ventas** -> Número de ventas realizadas
- **Clientes** -> Número de clientes únicos
- **Ingresos** -> Ingreso total generado

Ejemplo del archivo:

Ciudad	Categoria	Ventas	Clientes	Ingresos
Quito	Electrónica	1200	800	240000
Quito	Ropa	900	600	90000
...
Guayaquil	Electrónica	1500	1000	300000
Guayaquil	Hogar	1100	700	165000
Cuenca	Ropa	700	500	70000

3.1 [5 puntos] Cargue el archivo en un **DataFrame** llamado **df** y muestre las primeras 5 filas.

3.2 [5 puntos] Agregue una nueva columna llamada **Ingreso_promedio** utilizando operaciones vectorizadas de pandas, donde, el ingreso promedio es el ingreso dividido para la cantidad de clientes.

3.3 [8 puntos] Filtre y muestre los registros que cumplen ambas condiciones:

- El **Ingreso_promedio** es mayor al promedio de esa misma columna
- El número de ventas es mayor a 800

3.4 [10 puntos] Utilizando **groupby** sobre la columna **Ciudad**, calcule el total de Ingresos.

Muestre:

- El resultado del agrupamiento.
- La ciudad con el mayor ingreso total.

3.5 [12 puntos] Permita que el usuario ingrese el nombre de una categoría, y a partir de esa categoría:

- Si la categoría existe:
 - Filtre los registros correspondientes
 - Ordene los resultados por Ventas en forma descendente
 - Muestre los 5 registros con mayor número de ventas
- Si no, muestre un mensaje de que dicha categoría no existe

3.6 [5 puntos] Finalmente, utilice el resultado de las 5 mayores ventas por categoría y realice lo siguiente:

- Obtenga únicamente las columnas: **Ciudad**, **Ventas**, **Clientes**
- Guarde el resultado en un archivo CSV con el nombre: `top_5_ventas_[categoria].csv`. Donde indique la categoría correspondiente. Para este literal asuma que la categoría si existe.