ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
CCPG1043 / CCPG1801 - FUNDAMENTOS DE PROGRAMACIÓN
SEGUNDA EVALUACIÓN
I TÉRMINO 2025-2026/ Agosto 29, 2025

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: .....

Tema 1	Tema 2	Tema 3	Total	
				Firma Revisé mi calificación

Funciones y propiedades de referencia en Python.

para ***diccionarios***:	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)	**df** = **pd**.DataFrame(...) **df**.head() **df**.tail() **df**.info() **df**.describe() **df**["col"] **df**[lista] **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("col") **df**.groupby("col1")["col2"].sum() **df**["col"].mask(condicion, valor) **pd**.read_csv("file.csv") **df**.to_csv("file.csv")

TEMA 1

[15 puntos] Analice el siguiente programa:

```
texto = 'j8L.qiAS982'
tipos = {}
i = 0
c = 'r'
while i < len(texto) and c.isalnum():
    c = texto[i]
    if c.isalpha():
        if "letras" not in tipos:
            tipos["letras"] = 0
        tipos["letras"] += 1
    elif c.isdigit():
        if "digitos" not in tipos:
            tipos["digitos"] = 0
        tipos["digitos"] += 1
    else:
        print(f"Error: caracter no permitido '{c}' en posición {i}")
    i += 1
if i == len(texto):
    print('Texto válido')
    print("Resumen:", tipos)
```

Indique:

- **Qué imprime** en consola, **y**
- **El contenido** de la variable **tipos**.

Salida en consola	Contenido de variable tipos

Justifique su respuesta. Sin justificación no se calificará su respuesta.

TEMA 2

2.1 [30 puntos] Implemente un programa en Python que permita registrar los puntajes de los jugadores de una partida en un juego cooperativo.

El programa recibirá entradas en el formato:

`Nombre,puntaje`

Ejemplo:

```
Registro de puntajes (nombre,puntaje). Escriba FIN para terminar.
Ingrese puntos: ana,30
Ingrese puntos: pedro,20
Ingrese puntos: luis,25
Ingrese puntos: ANA,15
Ingrese puntos: pedro,40
Ingrese puntos: FIN
```

El programa debe incluir lo siguiente:

- La captura de datos termina cuando se ingresa la palabra **FIN**.
- Cada jugador puede ingresar sus puntos varias veces (porque se juegan varias rondas).
- Debe construirse un diccionario donde:
  - La clave es el nombre del jugador (la primera letra en mayúsculas).
  - El valor es la lista de números que contiene todos los puntajes obtenidos.

```
{'Ana': [30, 15], 'Pedro': [20, 40], 'Luis': [25]}
```

2.2 [15 puntos] Al final se debe mostrar el puntaje total de cada jugador y el puntaje total como equipo.

```
Totales
Ana: 45
Pedro: 60
Luis: 25
Total: Ganaron 130 puntos como equipo
```

TEMA 3

El DataFrame **df** contiene información de consumos nutricionales de pacientes. Cada fila representa el consumo mensual con las siguientes columnas:

Paciente: ID del paciente,
Sexo: F o M (Corresponde a femenino y masculino)
Edad: Edad en años
Mes: 1–12
Tipo: ej. Frutas, Verduras, etc
Calorías: kilocalorías consumidas
Protenas: kg de proteínas consumidas

Nota: La tabla adjunta muestra información de ejemplo.

	Paciente	Sexo	Edad	Mes	Tipo	Calorías	Proteinas
0	P001	F	35	1	Frutas	5000	63
1	P001	F	35	2	Verduras	4800	60
2	P001	F	35	3	Cereales	30000	750
3	P002	M	42	1	Frutas	5200	65
...							
85	P012	M	29	1	Carnes	2000	480
86	P012	M	29	7	Frutas	6200	74
87	P003	F	51	5	Cereales	22000	831
88	P005	F	60	1	Frutas	5800	73
...							

Escriba código para responder:

3.1. **[Bono 5 puntos]** Agregue una columna llamada **Edad_str** que clasifique a los pacientes en:
`'Joven'` si la edad es **menor que 35**
`'Adulto'` si la edad está entre **35 y 59** inclusive
`'Senior'` si la edad es **mayor o igual que 60**
Respuesta esperada: código para agregar la nueva columna de tipo `str`.

3.2. **[8 puntos]** Seleccione a los **pacientes más longevos** ('Senior') y calcule el promedio de las calorías consumidas por ese grupo. Si no resolvió el literal anterior, asuma que la columna **Edad_str** ya existe.
Respuesta esperada: un número (`float`) que representa el promedio, en la variable **`promedio`**.

3.3. **[10 puntos]** ¿Cuál es el **tipo** de alimento cuyo **promedio** de `Calorías` es el **más alto** en el df?.
Respuesta esperada: un `str` con el Tipo, en la variable **`resultado`.**

3.4. **[10 puntos]** Considerando únicamente los registros del primer trimestre (meses 1 a 3) y el tipo de Alimento `'Cereales'`, calcule la cantidad total de registros correspondientes al sexo Femenino.
Respuesta esperada: un `int` en la variable **`total`.**

3.5. **[12 puntos]** Cree la variable **`pacientes`** a partir de **df** que contenga:

- `Paciente`: El ID del paciente
- `Calorias`: El **total de calorías** consumidas por paciente

Considere que cada paciente tiene varios registros en el df original.

Finalmente, **ordene** los datos de **`pacientes`** por consumo de calorías (de mayor a menor) y guarde estos datos en un archivo **CSV** llamado **`'calorias_paciente.csv'`**.