

Facultad De Ingeniería En Electricidad Y Computación
CCPG1043 / CCPG1801 - Fundamentos De Programación
Primera Evaluación / PAE 2026-2027/ Mar 27, 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	Team 4	Total	Firma Revisé mi calificación

Funciones y propiedades de referencia en Python.

random as rd :	para listas :	para cadena s:
rd.randint (inicio,fin) rd.randrange (inicio,fin, salto) rd.choice (lista) rd.choices (lista,k=cant) rd.sample (lista,cant) rd.shuffle (lista) # Estructuras de control if condicion : instrucciones elif condicion : instrucciones else : instrucciones for iterador in secuencia : instrucciones	lista.append (...) lista.extend (...) lista.count (...) lista.index (...) lista.pop () lista.insert (pos, elem) elemento in lista range (inicio,fin, salto) separador.join (lista) len (lista) sum (lista) max (lista) Lista [idx] Lista [ini : fin] Lista [ini : fin : salto]	cadena.islower () cadena.isupper () cadena.isdigit () cadena.isalpha () cadena.isalnum () cadena.lower () cadena.upper () cadena.split (separador) cadena.capitalize () cadena.title () cadena.count (subcadena) cadena.replace (actual, nueva) int (cadena) float (cadena)

TEMA 1

[25 puntos] Lea detenidamente el código python a continuación e identifique 5 errores de sintaxis existentes en dicho código. Además, **explique** de forma sencilla **cómo corregir** cada error encontrado.

```
1 import random as rnd
2
3 numeros = []
4
5 for i in range(1, 10):
6     .numeros.append(rnd.randint(1, 50))
7
8 pares = []
9 impares = []
10 for n in numeros:
11     if n % 2 = 0:
12         if n > 10
13             pares.append[n]
14         else:
15             impares.append(n)
16     else:
17         if n % 2 != 0:
18             impares.append(n)
19
20 print("Lista original:", Numeros)
21 print("Pares mayores a 10:", pares)
22 print(f"Impares: {impares}")
23
```

TEMA 2

[25 puntos] Implemente la función **palabras_repetidas(texto)** que recibe una cadena de texto y retorna una lista con las palabras que aparecen más de una vez en el texto, sin repetirlas en la lista de salida y respetando el orden en que se detectan por primera vez como repetidas.

Considere lo siguiente:

1. La cadena que se recibe puede contener letras mayúsculas y minúsculas, comas, puntos y espacios.
2. Antes de procesar, debe convertir toda la frase a minúsculas.
3. Debe eliminar comas y puntos.
4. Debe comparar palabras exactas.
5. La lista resultante no debe contener palabras duplicadas.
6. Asuma que la frase no contiene ningún otro signo de puntuación aparte de comas y puntos.

```
# Ejemplo de entrada
texto = "Casa, perro, gato, casa, Perro, árbol, gato, casa."
# Ejemplo de salida
['casa', 'perro', 'gato']
```

TEMA 3

3.1 [20 puntos] Se desea sugerir una contraseña a partir de una frase, para esto, deberá crear la función `genera_password(frase, excluidas)` que recibe una frase y una lista de palabras prohibidas (que deberán ser excluidas de la contraseña).

La función debe construir y retornar una contraseña formada de la siguiente forma:

1. Convierta la frase en una lista de palabras.
2. Use las dos primeras letras de cada palabra para formar la contraseña.
3. Al formar la contraseña evite el uso de las palabras prohibidas.
4. Convierta la contraseña a mayúsculas.
5. Seguida de un número aleatorio entre 25 y 75, ambos incluidos.

Considere lo siguiente:

1. Todas las palabras de la frase y de la lista excluidas están en minúsculas.
2. La frase no contiene signos de puntuación.
3. Si una palabra tiene solo una letra, se debe excluir de la contraseña.
4. Debe conservarse el orden original de las palabras válidas.
5. Al final debe tener un número aleatorio como parte de la contraseña.

```
#Ejemplo de entrada
frase = "instituto superior de tecnologia y desarrollo"
excluidas = ["de", "la", .....] # Pueden haber varias cadenas

#Ejemplo de salida
"INSUTEDE57"
```

3.2 [5 puntos] Desarrolle un programa que, haciendo uso de la función `genera_password(frase, excluidas)` cree y muestre una contraseña a partir de la frase ingresada por el usuario y una lista de palabras prohibidas. Asuma que la lista de palabras prohibidas ya está definida en su programa.

TEMA 4

[25 puntos] Se necesita **implementar un programa** que valide claves de licencia para usar software especializado. Se debe solicitar al usuario que ingrese por teclado la clave de **una licencia** y verificar si es correcta o no.

Si la licencia es correcta, debe mostrar el mensaje: **"Licencia válida."**

En caso contrario, debe mostrar: **"Error: Licencia inválida."**

Restricción importante: No se permite el uso de estructuras repetitivas como for o while.

Una licencia correcta debe **cumplir con todas** las siguientes condiciones:

1. Debe tener una longitud exacta de 16 caracteres.
2. Debe tener exactamente 3 bloques separados por guiones "-".
3. Debe respetar el siguiente formato estandarizado: **99999-AAAAA-yyyy**. Donde:
 - a. **Primer bloque** (5 caracteres): únicamente números
 - b. **Segundo bloque** (5 caracteres): únicamente letras en mayúsculas.
 - c. **Tercer bloque** (4 caracteres): únicamente dígitos que representan el año entre 2010 y 2025 ambos incluidos.

```
#Ejemplo1
Ingrese la clave de licencia: 12345-ABCDE-2023
Licencia válida.
- - -
#Ejemplo2
Ingrese la clave de licencia: 98760-XYZWV-2010
Licencia válida.
- - -
#Ejemplo3
Ingrese la clave de licencia: 123AA-ABC-2008
Error: Licencia inválida.
- - -
#Ejemplo4
Ingrese la clave de licencia: 17348-jdhls-2021
Error: Licencia inválida.
```