

2K DESARROLLO DE APLICACIONES INFORMÁTICAS

14/11/2025

UT1: "GIT, pseudocódigo, diagramas de flujo"

Modelo A

Tiempo: 1h40

Nombre y apellidos:

Calificación:

/10

1. Test (6 puntos)	3. Ejercicio 1 (2 puntos)	4. Ejercicio 2 (2 puntos)

Utiliza bolígrafo azul o negro.

TEST

Respuesta correcta = 0,30 puntos. Respuesta incorrecta = -0,10 puntos. Marcar solo una opción. Todos los resultados que no estén en esta tabla no se considerarán una respuesta corregible. Cualquier carácter que no pueda ser reconocible por el profesor se considerará cómo error.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	

1.1. ¿Qué instrucción GIT añade los cambios al repositorio remoto?

- a- add
- b- commit
- c- clone
- d- push

1.2. ¿Qué instrucción GIT confirma los cambios de la zona de intercambio y crea una nueva versión del repositorio?

- a- add
- b- commit
- c- clone
- d- push

1.3. Si quiero ejecutar por terminal un programa en VsCode, sea cual sea el tipo de archivo, tendré que usar la instrucción

- a- Exe
- b- Python3
- c- Python
- d- Py

1.4. La instrucción code . permite:

- a- Abrir un archivo desde terminal
- b- Abrir varios archivos desde terminal
- c- Abrir todos los archivos de la URL que hemos añadido en el explorador
- d- Todas son correctas

1.5. Visual Studio Code es:

- a- Un editor de código fuente
- b- Un controlador de versiones
- c- Un control de versiones en la nube
- d- Todas son correctas

1.6. Una función en Python

- a- Usa un espacio indentado
- b- No usa indentación
- c- Usa dos espacios indentados
- d- Se usa el tabulador para indentar

1.7. La instrucción assert en Python:

- a- Permite generar funciones
- b- Es una técnica de debugging
- c- Es un framework
- d- Permite comprobar si una condición es verdadera o no

1.8. Si quiero crear un bucle for, con un rango que vaya de 0 a 10 con incremento de 1, será de la siguiente forma:

- a- For i in range (1, 10, 1):
- b- For i in range (0, 10, 1):
- c- For i in range (0, 1, 10):
- d- For i in range (1, 0, 10):

1.9. Para definir una función:

- a- `def <nombre-funcion>:`
- b- `def (<parámetros>):`
- c- `def <nombre-funcion>(<parámetros>):`
- d- `def (<parámetros>)<nombre-funcion>:`

1.10. El siguiente programa devolverá:

```
def area_triangulo(base, altura):  
    #  
    return base * altura / 2  
#  
  
#  
area_triangulo(2, 3)  
#
```

- a- El área de un triángulo de altura 2 y base 3
- b- El área de un triángulo de base 2 y altura 3
- c- No devolverá nada
- d- Almacenará en una variable el valor del área, pero no lo mostrará por pantalla

1.11. La instrucción GIT ADD .

- a- Añade los cambios en el fichero <fichero> del directorio de trabajo a la zona de intercambio temporal
- b- Añade los cambios en todos los ficheros de la carpeta <carpeta> del directorio de trabajo a la zona de intercambio temporal
- c- Añade los cambios al repositorio remoto
- d- Añade todos los cambios de todos los ficheros no guardados aún en la zona de intercambio temporal

1.12. La instrucción <return> :

- a- Finalizar la ejecución de una función
- b- Devolver un valor
- c- Finalizar la ejecución de una función y devolver un valor
- d- Finalizar la ejecución de una función y devolver un valor (opcional)

1.13. La instrucción <break>:

- a- Termina de manera inmediata un bucle for, while o DO while
- b- Termina de manera inmediata bucles for y while
- c- Termina de manera inmediata un bucle for
- d- Termina de manera inmediata un bucle while o DO while

1.14. ¿Cuál de las siguientes frases NO es correcta?

Las listas son:

- a- Un tipo de dato estructurado
- b- Tienen orden.
- c- Son inmutables, es decir, no pueden alterarse durante la ejecución de un programa.
- d- Pueden contener elementos de distintos tipos.

1.15. ¿Cuál de las siguientes frases NO es correcta?

Las Tuplas:

- a- Tienen orden.
- b- Pueden contener elementos de distintos tipos.
- c- Son mutables, es decir, pueden alterarse durante la ejecución de un programa.
- d- Se usan habitualmente para representar colecciones de datos de una determinada estructura semántica, como por ejemplo un vector o una matriz.

1.16. ¿Cuál de las siguientes es una operación que NO modifica una lista?

- a- Append
- b- Index
- c- Extend
- d- Insert

1.17. Indica cuál de las siguientes afirmaciones es correcta

```
lista = ["Matemáticas", "Física", "Química", "Historia", "Lengua"]
```

- a- Es un conjunto mutable de datos de tipo string
- b- Es un conjunto de datos inmutable de tipo string
- c- Es un conjunto de datos estructurados sin orden
- d- Ninguna es correcta

1.18 El código siguiente devolverá:

```
def bienvenida(nombre, apellido):  
#  
    print('¡Bienvenido a Python', nombre, apellido + '!')  
#  
    return  
#  
  
#  
bienvenida(apellido='Ugarte', nombre='Ramón')  
#
```

- a- ¡Bienvenido a Python <nombre, apellidos>!
- b- ¡Bienvenido a Python nombre apellidos!
- c- ¡Bienvenido a Python Ugarte Ramón!
- d- ¡Bienvenido a Python Ramón Ugarte!

1.19 Cuando trabajamos con funciones anidadas en Python debemos usar para diferenciar los distintos niveles de anidamiento:

- a- la indentación
- b- los espacios
- c- las líneas en blanco
- d- los comentarios

1.20. La estructura fundamental de la programación estructurada que permite la repetición es:

- a- la secuencia
- b- la iteración
- c- la alternativa
- d- cualquiera de las tres

Ejercicio 1

Diseña un algoritmo que pida al usuario un número entero del 0 al 9 y:

- si el número no es correcto, vuelva a solicitar introducir un número,
- si el número es correcto, lo vaya sumando a los números previamente introducidos por el usuario, volviendo a solicitar introducir otro número,
- si el número es 99, salga del programa, devolviendo el valor de la suma total de los números correctos previamente introducidos.

Pseudocódigo (1 punto)

Diagrama de flujo (1 punto)

Ejercicio 2

Corrige los errores del código siguiente:

```
# Crear una lista vacía para almacenar los números
numeros_ganadores = ()

# Pedir los números al usuario
print("Introduce los 6 números ganadores de la lotería
primitiva.")
for i in range(6):
    while True:
        try:
            # Pedir un número y guardarlo como entero
            numero = int((f"Introduce el número {i+1}: "))
            # Añadir el número a la lista
            numeros_ganadores.append(numero)
            # Salir del bucle while si el número es válido
        except ValueError:
            print("Por favor, introduce un número válido.")

# Ordenar la lista de números
numeros_ganadores.sort()

# Mostrar los números ordenados
print("\nLos números ganadores ordenados de menor a mayor
son:")
print(numeros_ganadores)
```

Código corregido: