

Ejercicio de Investigación – Respuestas Técnicas y Detalladas

=====

SECCIÓN 1: PYTHON

=====

1. Defina una lista y una tupla en Python. Proporcione algunos ejemplos.

Una **lista** en Python es una estructura de datos mutable que permite almacenar múltiples elementos, los cuales pueden ser modificados, agregados o eliminados. Se define usando corchetes [].

Ejemplo:

```
```python
mi_lista = [1, 2, 3, "Python", 5.4]
mi_lista.append(10)
mi_lista[0] = 99
print(mi_lista)
```
```

Una **tupla** es similar a una lista, pero es inmutable. Una vez creada, sus elementos no pueden cambiarse. Se define con paréntesis ().

Ejemplo:

```
```python
mi_tupla = (10, 20, 30, "Python")
```

```
print(mi_tupla[2])
```

```
'''
```

## 2. ¿Qué es un espacio de nombres en Python?

Un **espacio de nombres (namespace)** es una colección de nombres (variables, funciones, clases) asociados a objetos. Python utiliza distintos namespaces: local, global, built-in y de módulos. Permiten evitar colisiones de nombres entre diferentes áreas del programa.

## 3. Diferencia entre variable local y variable global.

- **Variable global:** Declarada fuera de funciones. Accesible en cualquier parte del archivo.
- **Variable local:** Declarada dentro de una función. Solo existe dentro de ella.

Ejemplo:

```
```python
```

```
x = 100 # global
```

```
def funcion():
```

```
    y = 50 # local
```

```
    print(y)
```

```
funcion()
```

```
print(x)
```

```
'''
```

4. ¿Qué es un IDE? Ejemplos.

Un **IDE (Integrated Development Environment)** es un programa que integra herramientas como editor, depurador, consola y autocompletado.

IDE populares para Python:

- PyCharm
- Visual Studio Code
- Spyder
- Jupyter Notebook
- Thonny

5. ¿Qué son los módulos en Python? Ejemplos.

Un **módulo** es un archivo .py que contiene funciones, variables o clases reutilizables. Se puede importar en otros programas.

Ejemplo:

```
```python
import math
print(math.sqrt(25))
```
```

Módulo propio:

```
```python
archivo operaciones.py
def sumar(a, b):
 return a + b
```

'''

6. Diferencia entre matriz y lista.

- **Lista:** estructura flexible, admite distintos tipos.
- **Matriz:** normalmente implementada con NumPy, diseñada para cálculos numéricos.

Ejemplo con NumPy:

```
```python
import numpy as np
matriz = np.array([[1, 2], [3, 4]])
'''
```

7. ¿Qué son los operadores? Ejemplos.

Son símbolos que ejecutan operaciones.

Ejemplos:

```
```python
a = 10
b = 5

print(a + b) # aritmético
print(a > b) # comparación
print(a == b and a > 0) # lógico
'''
```

=====

## SECCIÓN 2: BASES DE DATOS

=====

### 1. Diferencia entre base relacional y no relacional.

- **Relacional (SQL):** usa tablas, columnas y filas (MySQL, PostgreSQL).
- **No relacional (NoSQL):** usa documentos, grafos, clave-valor (MongoDB, DynamoDB).

### 2. ¿Qué son los índices?

Estructuras que mejoran la velocidad de búsqueda en una tabla mediante árboles B+ o hashes.

### 3. Claves principales y secundarias.

- **Clave primaria (PRIMARY KEY):** identifica un registro de forma única.
- **Clave foránea (FOREIGN KEY):** crea relaciones entre tablas.

### 4. Uniones internas y externas.

- **INNER JOIN:** devuelve registros coincidentes en ambas tablas.
- **LEFT/RIGHT OUTER JOIN:** devuelve todos los registros de un lado, aunque no haya coincidencia.

## 5. Diferencia entre DROP TABLE y TRUNCATE TABLE.

- DROP: elimina totalmente la tabla.
- TRUNCATE: elimina todos los registros, pero mantiene la estructura.

## 6. Tipos de datos SQL.

Incluyen: INT, VARCHAR, TEXT, BOOLEAN, DATE, FLOAT, DECIMAL.

## 7. WHERE vs HAVING.

- WHERE: filtra antes del agrupamiento.
- HAVING: filtra resultados después del GROUP BY.

=====

## SECCIÓN 3: FUNDAMENTOS AWS – PARTE 1

=====

### 1. Defina IaaS, PaaS y SaaS.

- IaaS: hardware virtualizado (EC2).
- PaaS: plataformas administradas (Elastic Beanstalk).
- SaaS: software completo (WorkSpaces).

### 2. Ventajas de la nube:

Escalabilidad, elasticidad, alta disponibilidad, pago por uso, seguridad, gestión simplificada.

### 3. Región y Zona de Disponibilidad.

- Región: área geográfica global.
- AZ: centros de datos dentro de la región.

### 4. Regiones AWS: us-east-1, us-west-1, eu-west-1, ap-south-1, etc.

### 5. Categorías de servicios AWS: cómputo, almacenamiento, seguridad, redes, analítica, bases de datos.

### 6. Objetos vs bloques:

- Objetos: S3.
- Bloques: EBS.

### 7. Dos servicios de cómputo:

- EC2: máquinas virtuales.
- Lambda: serverless.

### 8. Dos servicios de almacenamiento:

- S3: objetos.
- EFS: archivos.

=====

## SECCIÓN 4: FUNDAMENTOS AWS – PARTe 2

=====

1. Modelo de responsabilidad compartida:

AWS se encarga de la infraestructura; el cliente de la configuración y datos.

2. IAM Roles:

Otorgan permisos a entidades sin credenciales permanentes.

3. Políticas IAM:

JSON que define permisos.

4. AMI:

Imagen para lanzar instancias EC2.

5. Tipos de instancias EC2:

T3 (general), C5 (cómputo), R5 (memoria), P3 (GPU).

6. VPC:

Red virtual privada dentro de AWS.

7. Subred pública vs privada:

- Pública: acceso directo a internet.

- Privada: sin acceso, usada para bases de datos.

=====

SECCIÓN 5: AWS WELL-ARCHITECTED FRAMEWORK



=====

1. 5 pilares: seguridad, fiabilidad, costo, eficiencia del rendimiento, excelencia operativa.

2. 3 áreas de excelencia operativa:

Monitoreo, automatización, documentación.

3. Principios de seguridad:

Mínimo privilegio, trazabilidad, automatización.

4. Principios de fiabilidad:

Escalabilidad, tolerancia a fallos.

5. Eficiencia del rendimiento:

Uso adecuado de recursos, análisis de requerimientos.

6. Optimización de costos:

Instancias reservadas, Spot, apagado automático.

=====

## SECCIÓN 6: CLOUDFORMATION

=====

1. Aprovisionamiento: despliegue automatizado de recursos.

2. Administración de configuración: mantener estados homogéneos. Herramientas: Ansible, Puppet, Chef.

3. Integración continua: pipeline automatizado de pruebas.

4. Entrega continua: despliegue automatizado a producción.

5. CloudFormation + 3 ventajas:

- Infraestructura como código
- Versionamiento
- Repetibilidad de despliegues

6. JSON vs YAML:

JSON usa llaves, YAML usa indentación; YAML es más legible; JSON no soporta comentarios.

7. Pila:

Conjunto de recursos desplegados juntos.

=====

## SECCIÓN 7: FACTURACIÓN AWS

=====

1. Tipos de soporte:

Basic, Developer, Business, Enterprise.

## 2. Calculadora de costos:

Herramienta para estimar costos mensuales.

## 3. Modelos EC2:

On-Demand, Reserved, Spot, Savings Plans.