Solucionario de

ejercicios de autoevaluación

Ejercicios de autoevaluación Unidad de Aprendizaje 1

1. ¿Cuál es la función de la memoria RAM en la arquitectura de Von Neumann?

a. Ejecutar todos los cálculos y lógica del programa.
b. En esta arquitectura no se contempla el uso de una memoria.
c. Almacenar información durante mucho tiempo, incluso al apagar el ordenador.
d. Almacena los datos a los que accede la CPU mientras el ordenador está encendido.

2. Determina si la siguiente oración es verdadera o falsa: "Los tipos de datos *booleanos* nos permiten bifurcar el flujo de un programa".

- **Verdadero**
- Falso

3. ¿Qué imprimirá por pantalla el siguiente programa?

```
>>> frase = "MiNombre"
>>> for letra in frase:
        if (letra == "m"):
                continue
        print (letra)
```

a. El programa está mal escrito.
b. M, i, N, o, b, r, e.
c. No se puede usar un *if* con caracteres.
d. i, N, o, b, r, e.

4. ¿Es obligatorio usar la indentación en Python?

a. Sí, porque es una buena práctica de programación.
b. Solo es necesario si programamos de manera funcional.
c. Sí, porque es imprescindible para estructurar el código.
d. No, no es necesario usar la indentación en Python.

5. Determina si la siguiente oración es verdadera o falsa: "Estamos realizando un programa y nombramos una variable de este modo: *_Primera*, ¿está permitido en Python?".

- **Verdadero**
- Falso

6. ¿Cuáles de estos lenguajes son formales?

a. El lenguaje lógico
b. Los lenguajes de programación
c. La lengua española
d. Las opciones a y b son correctas.

7. ¿Cuál de las siguientes es una buena práctica a la hora de programar?

a. Procurar que el código esté bien estructurado.
b. Documentar bien el código.
c. Usar nombres para las variables que sean explícitos sobre su uso.
d. Todas las opciones son correctas.

8. ¿Pueden cambiar las constantes su valor en Python?

a. No, porque son constantes.
b. Sí, las constantes pueden cambiar su valor en Python y en todos los lenguajes de programación.
c. En Python no existe el concepto de constantes en el mismo sentido que en otros lenguajes.
d. Si son de tipo numérico pueden cambiar su valor.

9. ¿Cuáles son las principales características de un algoritmo?

a. Eficiente, estructurado, infinito.
b. Depende del problema que resuelva el algoritmo.
c. Finito, fácilmente modificable, eficiente, bien estructurado.
d. Todas las opciones son incorrectas.

10. ¿Qué imprimirá por pantalla el siguiente programa?

```
>>> x = 43
>>> while (x > 35):
        print (x)
        x = x - 1
        if (x == 39):
             break
```

a. No puede haber una sentencia *if* dentro de un bucle *while.*
b. 43, 42, 41, 40.
c. El programa da error.
d. 43, 42, 41, 40, 39, 38, 37, 36.

Ejercicios de autoevaluación Unidad de Aprendizaje 2

1. ¿Almacena de manera correcta el siguiente código los equipos dentro de sus respectivos países?

```
>>> Equipos = ("Bayern de Múnich", "Real Madrid",
"Manchester United", "Chelsea", "Arsenal",
"Juventus", "AC Milán")
>>> Alemania, España, *Inglaterra, Italia1, Italia2 =
Equipos
```

- **Verdadero**
- Falso

2. ¿Cuáles son las estructuras de datos inmutables en Python?

a. En Python no hay estructuras de datos inmutables.
b. Tuplas, conjuntos y diccionarios.
c. Cadenas de caracteres, tuplas y conjuntos.
d. Cadenas de caracteres, listas y conjuntos.

3. ¿Cuáles de los siguientes son métodos que poseen las listas en Python?

a. count, pop, sort, insert.
b. count, pop, sort, add.
c. count, pop, keys, values.
d. count, plus, sort, add.

4. Si imprimimos por pantalla el contenido de una estructura no ordenada, ¿se mostrará siempre el mismo resultado por pantalla? En este caso, ¿se comportan igual todas las estructuras no ordenadas?

a. Falso, falso
b. Verdadero, falso
c. Falso, verdadero
d. Verdadero, verdadero

5. Si en Python modificamos alguna de las estructuras mutables, ¿qué ocurre internamente en la memoria del ordenador?

a. En Python tenemos la opción de elegir si, cuando modificamos una estructura de este tipo, la almacenamos en una dirección de memoria diferente o la mantenemos en la misma dirección.
b. El nuevo cambio se produce en la estructura, y esta se almacena en una dirección de memoria diferente.
c. Ningún cambio se produce en la memoria.
d. El nuevo cambio se almacena en la memoria, pero la dirección donde estaba almacenada la estructura no cambia.

6. ¿Qué resultado mostrará por pantalla el siguiente código?

```
>>> texto = "España está al sur de Europa"
>>> texto[5:-2:3]
```

a. El comando está mal escrito y producirá un error del intérprete.
b. "as reu"
c. "ñeálueu"
d. [] (lista vacía porque el segundo índice es menor que el primero).

7. ¿Qué resultado producirá el siguiente código?

```
>>> conjun1 = {1, 2, 5, 9, 10, 15}
>>> conjun2 = {5, 10, 6, 8, 7}
>>> conjun2.difference(conjun1)
```

a. {1, 2, 6, 7, 8}
b. {1, 2, 9, 15}
c. {8, 6, 7}
d. {} (El conjunto vacío)

8. ¿Cómo se verá por pantalla el siguiente mensaje?

```
>>> print("Vamos a pasar las vacaciones: \n
Mis hermanos \tMis tíos \t\"Algún Primo que se
apuntará\"")
```

a. Producirá un error.
b. Vamos a pasar las vacaciones:
Mis hermanos Mis tíos "Algún Primo que se apuntará"
c. Vamos a pasar las vacaciones:
Mis hermanos Mis tíosAlgún Primo que se apuntará
d. Todas las opciones son incorrectas.

9. ¿Cómo podemos obtener mediante una *list comprehension* una lista de los múltiplos de 3 menores que 100?

a. [i for i in range (3, 100) if i /3 == 0]
b. [i for i in range (3, 100) if i % 3 == 0]
c. [i for i in range (3, 99) if i % 3 == 0]
d. [i for i in range (99) if i % 3 == 0]

10. Tras estas operaciones, ¿qué habrá dentro del diccionario medallas3?

```
>>> medallas1 = {'Brasil' : {'oro' : 1, 'plata' : 2,
'bronce' : 0}, 'Alemania' : {'oro' : 5, 'plata' : 0,
'bronce' : 3}, 'España' : {'oro' : 2, 'plata' : 7,
'bronce' : 5}}
>>> medallas2 = medallas1
>>> medallas2['Suiza'] = {'oro' : 0, 'plata' : 8,
'bronce' : 1}
>>> medallas3 = medallas2.copy()
>>> medallas3['Inglaterra'] = {'oro' : 2, 'plata' :
0, 'bronce': 0}
>>> medallas2 = medallas3
>>> medallas2['Australia'] = {'oro' : 2, 'plata' : 1,
'bronce' : 4}
>>> medallas2['Australia'] = {'oro' : 2, 'plata' : 2,
'bronce' : 5}
```

a. {'Brasil': {'oro': 1, 'plata': 2, 'bronce': 0}, 'Alemania': {'oro': 5, 'plata': 0, 'bronce': 3}, 'España': {'oro': 2, 'plata': 7, 'bronce': 5}, 'Suiza': {'oro': 0, 'plata': 8, 'bronce': 1}, 'Inglaterra': {'oro': 2, 'plata': 0, 'bronce': 0}, 'Australia': {'oro': 2, 'plata': 2, 'bronce': 5}}
b. {'Brasil': {'oro': 1, 'plata': 2, 'bronce': 0}, 'Alemania': {'oro': 5, 'plata': 0, 'bronce': 3}, 'España': {'oro': 2, 'plata': 7, 'bronce': 5}, 'Inglaterra': {'oro': 2, 'plata': 0, 'bronce': 0}, 'Australia': {'oro': 2, 'plata': 1, 'bronce': 4}}

c. {'Brasil': {'oro': 1, 'plata': 2, 'bronce': 0}, 'Alemania': {'oro': 5, 'plata': 0, 'bronce': 3}, 'España': {'oro': 2, 'plata': 7, 'bronce': 5}, 'Suiza': {'oro': 0, 'plata': 8, 'bronce': 1}, 'Inglaterra': {'oro': 2, 'plata': 0, 'bronce': 0}, 'Australia': {'oro': 2, 'plata': 1, 'bronce': 4}, 'Australia': {'oro': 2, 'plata': 2, 'bronce': 5}}
d. {'Brasil': {'oro': 1, 'plata': 2, 'bronce': 0}, 'Alemania': {'oro': 5, 'plata': 0, 'bronce': 3}, 'España': {'oro': 2, 'plata': 7, 'bronce': 5}, 'Suiza': {'oro': 0, 'plata': 8, 'bronce': 1}, 'Inglaterra': {'oro': 2, 'plata': 0, 'bronce': 0}, 'Australia': {'oro': 4, 'plata': 3, 'bronce': 9}}

Ejercicios de autoevaluación Unidad de Aprendizaje 3

1. ¿Qué es un decorador en Python?

a. Es una función que recibe a otra como parámetro, implementa en ella alguna funcionalidad nueva y devuelve la función modificada como resultado.
b. Permite convertir una función cualquiera en lo que se conoce como "función anónima", que se puede definir posteriormente en una sola línea.
c. En Python no pueden usarse decoradores. Estos existen solo en otros lenguajes de programación.
d. Al usarlo, permite que una función devuelva una secuencia completa de resultados que se pueden iterar.

2. ¿Qué nos permiten los *docstrings?*

a. Posibilitan que podamos pasar ciertos parámetros especiales a la hora de ejecutar el programa.
b. Nos permiten dar información sobre lo que las funciones hacen, sus parámetros y valores de retorno.
c. Sirven para indicar al intérprete de Python cómo comportarse cuando llega a esta parte del código.
d. Todas las opciones son incorrectas.

3. Determina si las siguientes oraciones son verdaderas o falsas:

a. Decimos que tenemos funciones anidadas cuando tenemos una función definida dentro de otra.

- **Verdadero**
- Falso

b. La función principal se conoce como la "función externa".

- **Verdadero**
- Falso

4. ¿Cuáles de las siguientes afirmaciones sobre las funciones recursivas son ciertas?

a. Este tipo de funciones se pueden utilizar en todos los lenguajes de programación.
b. Son útiles para resolver todo tipo de problemas a la hora de programar.
c. Son funciones que se llaman a sí mismas, necesitan un caso base que indique cuándo deben finalizar.
d. Todas las opciones son correctas.

5. Determina si la siguiente oración es verdadera o falsa: "Si introducimos parámetros por defecto a la hora de definir nuestras funciones, estos no se pueden modificar posteriormente a la hora de usarlas".

- Verdadero
- **Falso**

6. ¿Qué ocurrirá tras ejecutar estas líneas de código?

```
>>> def resta(valor1, valor2):
        print (valor1 - valor2)
>>> numero = resta(17, 12)
```

a. Se mostrará por pantalla el valor 5 y este se almacenará además en la variable *número.*
b. Tendremos almacenado dentro de la variable *número* el valor de la resta para usarlo cuando lo necesitemos.
c. Mostrará el valor 5 por pantalla, pero, al no tener nuestra función la palabra reservada *return*, la variable *número* no almacenará ningún valor.
d. Todas las opciones son incorrectas.

7. ¿Cuáles son las diferencias entre el paso de un argumento por valor y por referencia?

a. En el paso por valor se crea una copia de la variable dentro de la función y no se ve afectado el valor exterior de esta, este comportamiento ocurre con los tipos de datos simples. En el paso por referencia se maneja directamente el valor de la variable y por tanto los cambios que realicemos afectarán a esta a lo largo de todo el programa, este com-

portamiento ocurre generalmente con los tipos de datos compuestos.

b. En el paso por valor se maneja directamente la variable y los cambios que realicemos en ella afectarán a su valor a lo largo de todo el programa, este comportamiento suele ocurrir con los tipos de datos simples. En el paso por referencia se crea una copia de la variable dentro de la función y no se ve afectado el valor exterior de la variable, esto pasa usualmente con los tipos de datos compuestos.
c. En el paso por valor se crea una copia de la variable dentro de la función y no se ve afectado el valor exterior de esta, este comportamiento ocurre con los tipos de datos compuestos. En el paso por referencia se maneja directamente el valor de la variable y por tanto los cambios que se realicen afectarán a esta a lo largo de todo el programa, este comportamiento ocurre generalmente con los tipos de datos simples.
d. En el paso por valor se maneja directamente la variable y los cambios que realicemos en ella afectarán a su valor a lo largo de todo el programa, este comportamiento se da con los tipos de datos compuestos. En el paso por referencia se crea una copia de la variable dentro de la función y no se ve afectado el valor exterior de la variable, esto suele ocurrir con los tipos de datos simples.

8. ¿Se comportará del mismo modo la función *mostrar_nombre* si la invocamos pasando como argumentos los nombres Sara y Alejandra?

```
>>> def mostrar_nombre(nombre):

        if (len(nombre)) > 4:
            return

        return (nombre.upper())
```

a. Sí, en ambos casos se mostrarán por pantalla ambos nombres con la primera letra en mayúscula.
b. No, porque al ser la longitud del nombre Alejandra superior a cuatro caracteres, el código que se ejecuta es el del bloque *if,* y al tener este una palabra reservada *return* se finalizará el código de la función sin retornar nada.
c. Sí, porque en la última línea de código de la función se retorna el valor de la variable *nombre* con el método *upper()*.
d. No, tenemos dentro de la función dos veces la palabra reservada *return,* en el caso de introducir como argumento Sara

se ejecutará el bloque del código *if* y no retornará nada. Si introducimos Alejandra la función devolverá este nombre con todas las letras en mayúscula.

9. Determina si la siguiente oración es verdadera o falsa: "En Python podemos implementar el paradigma de la programación funcional mediante el uso de decoradores, generadores y funciones lambda".

- **Verdadero**
- Falso

10. ¿Cuáles son los beneficios de usar funciones en nuestros programas?

a. Nos permiten reutilizar el código ya escrito en otras partes de nuestros programas.
b. Nos facilitan escribir programas más claros y legibles.
c. Al dividir el código en pequeños bloques, es más fácil encontrar los errores del programa.
d. Todas las opciones son correctas.

Ejercicios de autoevaluación Unidad de Aprendizaje 4

1. ¿Cuál de las siguientes afirmaciones sobre el concepto de namespace es correcta?

a. Son una especie de contenedores de identificadores únicos que pueden pertenecer a variables, funciones, etc.
b. Al separar identificadores con el mismo nombre en diferentes *namespaces* se evita que entre ellos se produzca algún tipo de conflicto.
c. Nos permiten que cada identificador del *namespace* pueda tener asociadas unas características únicas.
d. Todas las opciones son correctas.

2. ¿Cuál es la jerarquía que usamos para organizar nuestro código en Python de menor a mayor complejidad?

a. Archivos < módulos < paquetes
b. Archivos < funciones < paquetes
c. Funciones < paquetes < módulos
d. Funciones < módulos < paquetes

3. Determina si la siguiente oración es verdadera o falsa: "Los *scripts* son archivos que almacenan en su interior una secuencia de comandos o líneas de código de programación".

- **Verdadero**
- Falso

4. ¿Cuál es el nombre del módulo que no puede faltar en un paquete de Python para que sea considerado como tal por el intérprete?

a. init.py
b. __path__.py
c. __init__.py
d. __modul.init__.py

5. Determina si la siguiente oración es verdadera o falsa: "Una expresión regular es una cadena de caracteres con un orden determinado que se usa para buscar coincidencias del mismo patrón en otras cadenas".

- **Verdadero**
- Falso

6. ¿Es una buena práctica importar todos los módulos que contiene un paquete?

a. Sí, no tenemos ningún problema al respecto puesto que los paquetes están creados por otros programadores y ya se han probado.
b. No, puesto que es posible que algunos paquetes oficiales sean pirateados por *hackers* y oculten virus.
c. No, porque hay algunos paquetes que ocupan mucha memoria en nuestro ordenador y quizá solo necesitemos utilizar ciertas funcionalidades que poseen.
d. Todas las opciones son incorrectas.

7. ¿Cuáles de los siguientes son algunos de los principales métodos del módulo random?

a. randint, random, choice
b. random, trunc, randint
c. choice, modf, randint
d. random, search, randint

8. ¿Cómo podemos usar los módulos de terceros en Python que no pertenecen a la biblioteca estándar?

a. Una vez encontrado el módulo que nos interesa utilizar, hay que contactar con el autor del mismo para que nos proporcione los permisos necesarios para poder usarlo.
b. Instalándolos con el comando *pip install* más el nombre del módulo.
c. Usando el comando *install* más el nombre del módulo.
d. Basta con buscar en internet el nombre del módulo y descargarlo del primer enlace disponible.

9. Determina si la siguiente oración es verdadera o falsa: "Los módulos que componen la biblioteca estándar no están disponibles al instalar el intérprete de Python y necesitamos descargarlos de internet".

- Verdadero
- **Falso**

10. ¿Cuáles son las ventajas de usar módulos para organizar nuestros programas?

a. Podemos usar funcionalidades creadas por otros programadores.
b. Hace posible adelantar tiempo en el desarrollo de los programas al usar trabajo creado de manera altruista por otros programadores.
c. Nos facilita usar módulos creados por terceros que, al haber sido usados por muchos programadores previamente, estarán limpios de errores de programación.
d. Todas las opciones son correctas.

Ejercicios de autoevaluación Unidad de Aprendizaje 5

1. ¿Cuáles son las características de las variables estáticas?

a. Están definidas dentro de la clase y de los métodos de estas. No se puede acceder a ellas directamente.
b. Están definidas dentro de la clase, pero solo se puede acceder a ellas al crear un objeto a partir de las mismas.
c. Están definidas dentro de la clase, pero no dentro de los métodos de esta. Esto propicia que se pueda acceder a ellas directamente mediante el nombre de la clase.
d. Todas las opciones son incorrectas.

2. ¿Cuáles son las principales características de la POO que hemos estudiado?

a. Encapsulamiento, herencia y polimorfismo.
b. Herencia, transmisibilidad y polimorfismo.
c. Herencia, instancia y encapsulamiento.
d. Atributos, métodos y *testing.*

3. Determina si la siguiente oración es verdadera o falsa: "Algunas de las grandes ventajas del paradigma de la programación orientada a objetos son la gran cantidad de código que nos permite reutilizar y el hacer más fácil encontrar errores".

- **Verdadero**
- Falso

4. ¿Qué nos permite el método especial conocido como método constructor?

a. En Python, este método nos permite construir clases de manera rápida y eficiente aportando una información mínima.
b. Nos permite inicializar objetos con unos atributos ya determinados.
c. Nos facilita usar módulos construidos por terceros que ya se han probado de manera continua.
d. Este método no existe en Python.

5. ¿Cómo se llama el proceso para crear un objeto a partir de una clase?

a. Proceso estático
b. Atributo
c. Instanciar
d. Herencia

6. ¿Para qué se usa en Python el parámetro *self*?

a. Añadido a los atributos, nos permite encapsularlos.
b. Es un modo de decirle a Python que nos estamos refiriendo al objeto en sí mismo.
c. En la herencia, nos permite acceder directamente a los métodos y atributos de la clase padre.
d. Nos sirve para pasarlo como parámetro e indicar que un método es de clase.

7. Los atributos son las características que pueden tener los objetos. Los métodos son las funcionalidades que pueden tener asociadas los objetos.

a. Verdadero, falso
b. Verdadero, verdadero
c. Falso, verdadero
d. Falso, falso

8. ¿Cuáles son las características de una clase?

a. En la POO nos sirven de molde para crear los objetos, que son abstracciones de entes reales.
b. Son una especie de contenedor abstracto de información que nos permite empaquetar juntos datos y funcionalidades.
c. Para crearlas en Python se usa la palabra reservada *class*.
d. Todas las opciones son correctas.

9. ¿Qué tres tipos diferentes de métodos existen en Python?

a. Heredados, estáticos y polimorfos.
b. Estáticos, decorados y encapsulados.
c. Estáticos, de clase y de instancia.
d. De clase, de instancia y constructores.

10. Determina si la siguiente oración es verdadera o falsa: "En Python, como en todos los lenguajes de programación modernos, no se permite la herencia múltiple para simplificar el código".

- Verdadero
- **Falso**

Ejercicios de autoevaluación Unidad de Aprendizaje 6

1. **Determina si la siguiente oración es verdadera o falsa: "La programación defensiva es una técnica de programación en la que corregimos los errores una vez que estos le han ocurrido al cliente para evitar que le vuelvan a suceder".**

   - Verdadero
   - **Falso**

2. **Algunos consejos para depurar el código son:**

   a. Introducir sentencias para que el lenguaje muestre por pantalla lo que está ocurriendo a medida que se ejecuta el código.
   b. Corregir los fallos de sintaxis que nos muestra el editor de código.
   c. Refactorizar el código.
   d. Todas las opciones son correctas.

3. **¿Cuándo se ejecuta la sentencia *finally*?**

   a. Se ejecuta siempre.
   b. Se ejecuta siempre que no se ejecuta la sentencia *except*.
   c. Se ejecuta solo si no se cumple la excepción.
   d. Se ejecuta cuando no se cumple la excepción y salta la sentencia *else*.

4. **Determina si la siguiente oración es verdadera o falsa: "El *traceback* de un error es una información que nos muestra el intérprete para ayudarnos a localizar el tipo de error y dónde se localiza".**

   - **Verdadero**
   - Falso

5. ¿Qué diferencias existen entre la depuración del código y las pruebas de *software?*

a. Las pruebas tratan de encontrar los errores que tiene el *software* y la depuración intenta comprobar si cumple con las tareas para las que fue diseñado.
b. La depuración trata de encontrar los errores que tiene el *software* y las pruebas tratan de comprobar si cumple con las tareas para las que fue diseñado.
c. Son sinónimos que podemos usar indistintamente en nuestras conversaciones con otros informáticos.
d. Todas las opciones son incorrectas.

6. Si, antes de publicar nuestra disruptiva aplicación móvil de mensajería, empezamos a utilizar la aplicación completa entre los miembros de nuestro equipo, y además vamos comprobando por separado los diferentes módulos de la aplicación para ver por ejemplo si la parte de conexión a la red de telefonía funciona adecuadamente, ¿de qué tipo de prueba de *software* estamos hablando?

a. Prueba de componentes
b. Prueba de aceptación
c. Prueba del sistema
d. Prueba de integración

7. ¿Cuáles son los tres tipos en los que podemos clasificar los errores en un lenguaje de programación?

a. Errores de sintaxis, errores lógicos y errores de compilación.
b. Errores de sintaxis, errores lógicos y errores en tiempo de ejecución.
c. Errores lógicos, excepciones y errores en tiempo de ejecución.
d. Errores unitarios, errores lógicos y errores de arquitectura del programa.

8. ¿Qué son las excepciones?

a. Una herramienta para comprobar que nuestro código cumple los requisitos para los que fue diseñado.
b. Un modo de manejar la información que fluye a través de los programas.

c. Un modo de controlar los errores que surgen en tiempo de ejecución de los programas.
d. Todas las opciones son correctas.

9. ¿Qué nos permite la palabra reservada *assert?*

a. Nos permite mostrar excepciones personalizadas.
b. Es una palabra que se usa siempre con *try* para manejar las excepciones.
c. Nos permite introducir condiciones en nuestro código, de manera que si la condición devuelve *True* el código se ejecuta.
d. Nos permite introducir condiciones en nuestro código, de manera que si la condición devuelve *False* el código se ejecuta.

10. Las pruebas unitarias son de tipo funcional y las pruebas de integración son de tipo no funcional.

a. Verdadero, falso
b. Verdadero, verdadero
c. Falso, verdadero
d. Falso, falso

Ejercicios de autoevaluación Unidad de Aprendizaje 7

1. Determina si la siguiente oración es verdadera o falsa: "El cursor de un fichero funciona de manera diferente según el tipo de información que contenga este".

- Verdadero
- **Falso**

2. Sobre los manejadores de contexto en Python podemos afirmar:

a. Se ocupan de controlar las excepciones.
b. Cierran los ficheros de forma automática.
c. Se usan con la palabra reservada *with*.
d. Todas las opciones son correctas.

3. Si abrimos un fichero con la opción de apertura "a", esta nos permite:

a. Abrir el fichero para escritura; se comenzará a escribir tras la información ya existente y se creará el fichero si este no existe.
b. Abrir el fichero para escritura; producirá un error si este no existe.
c. Abrir el fichero para escritura; creará el fichero si este no existe y sobrescribirá la información existente.
d. Permite la escritura y lectura simultáneas.

4. Son funciones relacionadas con el manejo de ficheros:

a. open(), close(), organizated() y std().
b. close(), append(), pop() y index().
c. open(), dumps(), seek() y writer().
d. open(), close(), type() y tell().

5. Las diferencias entre los ficheros binarios y de texto son:

a. Los ficheros de texto almacenan texto con tipografías y otros metadatos, los binarios almacenan imágenes y sonidos.
b. Los ficheros de texto almacenan texto plano sin ningún tipo de información complementaria, los binarios almacenan cualquier otro tipo de dato.
c. Los ficheros de texto almacenan texto plano y los binarios otros tipos de información. Dependiendo de qué tipo sean, se guardan en zonas de la memoria especialmente diseñadas para ellos.
d. Todas las opciones son incorrectas.

6. El principal uso de los ficheros tipo JSON es:

a. El intercambio y almacenamiento de datos.
b. Sirven para guardar datos de tipo tabular.
c. Son ficheros ejecutables del sistema operativo *Linux* que podemos usar con Python en otros sistemas operativos.
d. Sirven para guardar imágenes y son usados por el lenguaje JavaScript.

7. Son característica de los ficheros:

a. Poseen un nombre y extensión que los identifica.
b. Se guardan en una dirección de la memoria determinada.
c. La extensión nos indica el tipo de información que contienen.
d. Todas las opciones son correctas.

8. Determina si la siguiente oración es verdadera o falsa: "La serialización es un proceso para transformar la información en código binario que busca facilitar su transmisión y almacenamiento".

- **Verdadero**
- Falso

9. ¿Por qué es buena práctica cerrar un fichero cuando dejamos de trabajar con él?

a. Porque el microprocesador solo puede trabajar con un conjunto muy limitado de procesos a la vez.
b. No es necesario, puesto que todos los lenguajes de programación los cierran de forma automática al dejar de usarlos.
c. Porque mientras permanece abierto ocupa espacio en memoria, además, podrá ser modificado por error.
d. Todas las opciones son incorrectas.

10. Las operaciones básicas que podemos llevar a cabo con un fichero son:

a. Depende del tipo de fichero, de si este es binario o de texto plano.
b. Creación, apertura, manipulación y cierre.
c. Dependerá del tipo de sistema operativo y lenguaje de programación.
d. Creación, duplicado, borrado y apertura.

Ejercicios de autoevaluación Unidad de Aprendizaje 8

1. ¿Cuál es la diferencia entre las partes DDL y DML del lenguaje SQL?

a. DDL es la parte del lenguaje que se encarga de definir, entre otras cosas, cómo va a ser nuestra base de datos. DML es la parte del lenguaje que se encarga de la manipulación de los datos.
b. Esta división solo existe cuando usamos SQL con las bases de datos no relacionales.
c. En SQL no existe esta división del lenguaje.
d. DDL es la parte del lenguaje que se encarga de la manipulación de los datos. DML es la parte del lenguaje que define el esquema de las bases de datos.

2. ¿Qué son las transacciones ACID?

a. Son un tipo de transacciones que debemos evitar por ser poco seguras.
b. Son un tipo de transacciones que mantienen la integridad de los datos.
c. Son transacciones que hacemos cuando los datos que vamos a insertar en la base de datos no están estructurados.
d. Todas las opciones son incorrectas.

3. Indica si las siguientes oraciones son verdaderas o falsas:

a. Un buen modelado de datos hará que nuestra base de datos no contenga información redundante.

- **Verdadero**
- Falso

b. La acción de disminuir la redundancia en una base de datos se conoce como *normalización.*

- **Verdadero**
- Falso

4. ¿Qué es una clave foránea?

a. Es una clave introducida en una columna de nuestra tabla que nos la relaciona con la clave primaria de una tabla diferente.
b. Es una clave de la tabla que nos dice que esa información es redundante.
c. Es una clave única que nos permite identificar cada fila de la tabla de manera eficiente.
d. Todas las opciones son incorrectas.

5. Indica si la siguiente oración es verdadera o falsa: "El tipo de modelado de datos que realicemos a la hora de diseñar nuestra base de datos no afectará posteriormente al manejo de estos".

- Verdadero
- **Falso**

6. ¿Cuál de las siguientes afirmaciones sobre las bases de datos relacionales es cierta?

a. Son la mejor elección si se necesita un gran rendimiento y alta disponibilidad para un gran número de conexiones simultáneas.
b. Dado que están divididas en tablas, son las más idóneas para guardar imágenes y audios, puesto que podemos incorporar cada uno de esos datos en una columna diferente de la tabla.
c. No se recomienda usarlas si vamos a necesitar escalar rápidamente la capacidad de almacenar datos.
d. Todas las opciones son correctas.

7. ¿Qué es lo que caracteriza al modelo relacional de bases de datos?

a. El modelo relacional se basa en que los datos se organizan en tablas relacionadas entre sí, por lo que se puede acceder a los datos en relación con otro dato de la propia base de datos.
b. Este modelo se diseñó específicamente para guardar tipos de datos no estructurados.
c. Se caracteriza por que las diferentes tablas que forman la base de datos se pueden distribuir entre diferentes nodos y relacionarse entre sí mediante el entorno *cloud.*

d. Se caracteriza por que la estructura que presentan las bases de datos basadas en él es rígida y una vez definida no permite añadir nuevas tablas ni campos a estas. Esto ocurre por seguridad en las transacciones.

8. ¿Cuáles de los siguientes tipos de bases de datos NoSQL, que permiten distribuir los datos entre diferentes equipos, son los más parecidos a las bases de datos relacionales?

a. Bases de datos basadas en grafos
b. Bases de datos columnares
c. Bases de datos en formato de llaves y valores
d. Bases de datos basadas en documentos

9. Indica si la siguiente oración es verdadera o falsa: "Un sistema gestor de bases de datos es un programa específico que nos facilita el manejo de las bases de datos".

- **Verdadero**
- Falso

10. ¿Qué factores debemos tener en cuenta a la hora de elegir el tipo de base de datos para nuestra aplicación?

a. La velocidad de consulta sobre nuestros datos que vamos a necesitar.
b. El tipo de datos que insertar en la base de datos.
c. La seguridad de las operaciones que vamos a realizar.
d. Todas las opciones son correctas.

Ejercicios de autoevaluación Unidad de Aprendizaje 9

1. **Cuando trabajamos con entornos virtuales, ¿cómo sabemos que están activados?**

    a. Porque podemos consultarlo con el comando Python pip freeze desde la consola.
    b. Porque, cuando usamos el comando virtualenv, la consola nos devuelve True.
    c. Porque en la línea de comandos aparecerá el nombre del entorno entre paréntesis.
    d. Todas las opciones son correctas.

2. **¿En qué fichero tenemos que realizar las modificaciones para indicarle a Flask el archivo que queremos buscar que posibilite iniciar la aplicación mediante el comando *flask run?***

    a. flask_App.py
    b. activate.bat
    c. App.config
    d. flask_Env.bat

3. **¿Cuál es la diferencia entre un *framework* y una librería?**

    a. Una librería es un conjunto de *frameworks,* que además posee un conjunto de estándares sobre cómo debemos implementar el código.
    b. Un *framework* no es más que un conjunto de librerías que dotan de diferentes funcionalidades.
    c. Un *framework* es un conjunto de librerías, que además posee un conjunto de estándares sobre cómo debemos implementar el código.
    d. Ninguna, son dos términos para referirnos a un mismo concepto.

4. ¿Qué parámetro del decorador route() debemos modificar para poder pasar datos al servidor?

a. El parámetro POST
b. El parámetro GET
c. El parámetro methods
d. El parámetro url_for

5. Determina si la siguiente oración es verdadera o falsa: "Cuando desarrollamos con Flask, el parámetro que le pasamos a route() nos indicará la URL que visitar para ver el contenido de la ruta".

- **Verdadero**
- Falso

6. ¿Qué ventajas tiene usar Django como entorno de desarrollo frente a los demás *frameworks* para el desarrollo web que existen en Python?

a. Programar una API REST es mucho más simple que en los demás.
b. La documentación es breve y clara.
c. Incluye más opciones que los demás para dar protección a las aplicaciones web ante ataques informáticos.
d. Es más sencillo de aprender que los demás.

7. ¿En qué directorio principal debemos guardar los archivos CSS que usemos al desarrollar con Flask?

a. static
b. templates
c. stylesheet
d. Es indiferente.

8. ¿De qué clase debemos heredar para crear formularios de manera eficiente con Flask?

a. ConfigForm
b. FlaskForm
c. WTForm
d. FlaskRenderForm

9. Determina si la siguiente oración es verdadera o falsa: "Flask posee un sistema de autentificación de usuarios mucho mejor que Django".

- Verdadero
- **Falso**

10. ¿Cómo le indicamos al decorador route que nuestras URL tienen secciones variables?

a. Incluyendo un doble asterisco en la función asociada a la ruta.
b. Mediante corchetes angulares y un convertidor de datos opcional.
c. Incluyendo en el decorador las cadenas de las URL a las que va a dirigirse.
d. Todas las opciones son incorrectas.