M2C3 Python Assignment

1.Ejercicio práctico de Python:
checkpoint3.py
https://github.com/Izas-coder/checkpoint3

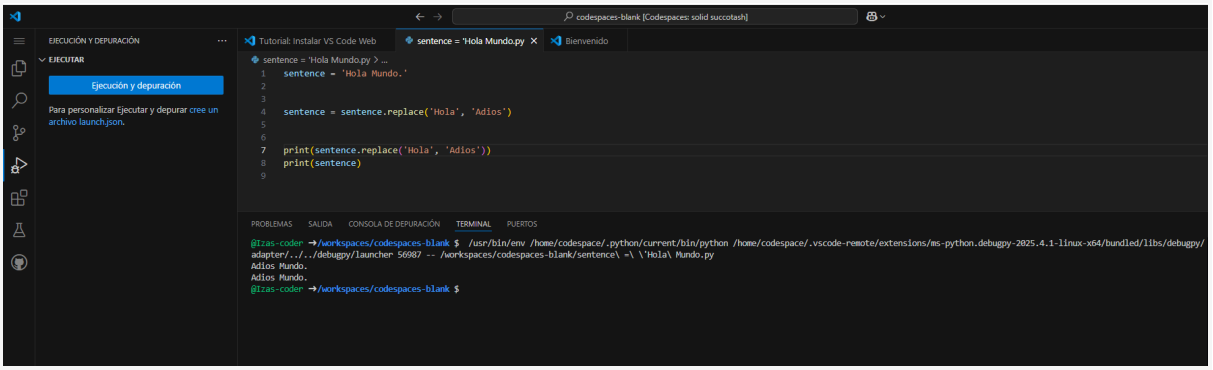
2.Crear una cadena que contenga la palabra "Hola". Usando la palabra clave en el método de búsqueda o el índice, busque y seleccione "Hola" en su cadena. Y usando la función de reemplazo, reemplace "Hola" en su cadena con "adiós". Este ejercicio de Python debes subirlo a tu Git-Hub o Replit para poder revisarlo
https://github.com/Izas-coder/checkpoint3/blob/main/checkpoint3_ejercicio2.py

```
sentence = 'Hola Mundo.'

sentence = sentence.replace('Hola', 'Adios')

print(sentence.replace('Hola', 'Adios'))

print(sentence)
```



3. Preguntas teóricas son:
¿Cuáles son los tipos de datos en Python?

Boolean: mediante este tipo de dato, nos devuelve si es verdadero (true) o falso(false).
Number: es un dato muy amplio. Dentro de este tipo de dato tenemos desde números enteros(int) hasta números con gran cantidad de decimales, fracciones o float-point(números reales). Los números tienen una gran variedad de elementos diferentes.

String: puede ser desde un nombre hasta un documento HTML. Hay diferentes modos de integrar string en el programa. Los strings van entre doble comilla. (“ “)
Bytes: representan una secuencia de bytes.
Bytearray: similar al byte, pero así como bytes no permite modificar su contenido, bitearray sí.
None: representa la ausencia de valor, define una variable a la que todavía no se ha asignado un valor.
Lists: permite almacenar múltiples elementos. Se definen en corchetes [ ].
Tuples: es similar a una lista, pero no se puede modificar. Se define entre paréntesis. ( )
Set: es una colección desordenada de elementos únicos. Se define entre corchetes { }
Dictionary: maneja colecciones.

¿Qué tipo de convención de nomenclatura deberíamos utilizar para las variables en Python?

Las nomenclatura recomendables en Python:
La nomenclatura que se utiliza para las variables en Python son las minúsculas dividiendo las palabras con guiones bajos. Por ejemplo:

nombre_completo = “Jon Lasa”

Nomenclaturas no recomendables en Python:
Evitar utilizar en solitario la minúscula de la ele(l) ya que es igual a la mayúscula de la letra i (I). Por lo tanto, evitar también la i mayúscula en solitario y letra o/O, ya que puede crear confusión con el número 0.

¿Qué es un Heredoc en Python?
Es una forma de llamar a un string multi lineal. Para llamar a más de una línea, podemos utilizar Heredoc, que sería añadir tres grupos de dobles comillas al principio y otras tres al final del grupo de líneas. (“”” “””).

¿Qué es una interpolación de cadenas?

La interpolación de cadenas es una forma **rápida y cómoda** de meter valores dentro de un texto. Para ello antes de las comillas la f nos dice que es una cadena formateada y las variables se meten entre corchetes { }.

```
name = 'Kristine'
greeting = f'Hi {name}'
print(greeting)
Devolvería: Hi Kristine
```

¿Cuándo deberíamos usar comentarios en Python?
Deberíamos utilizar los comentarios para hacer anotaciones en el código que utilizamos: con propósito de explicar el código utilizado, para aclarar partes complicadas o poco intuitivas, para marcar mejoras o para desactivar código temporalmente.

¿Cuáles son las diferencias entre aplicaciones monolíticas y de microservicios?

Las aplicaciones monolíticas tienen todos los componentes en una sola estructura con la ventaja de desarrollarlos más fácilmente y es más fácil probar y depurar. Pero a medida que crecen, son más difíciles de mantener.

La aplicación de microservicios se divide en servicios independientes y se comunican entre sí. Cada aplicación tiene su propia lógica y puede funcionar por separado. Como ventaja, decir que escala los servicios utilizando solo los que necesita y es más fácil de mantener, pero son más complejas de gestionar y necesitan una buena comunicación entre si.