



Planeación de aula.

Identificación

Grado/Grupo:10	Área/Asignatura: Informática	Fecha : 04-06-23
Docente / C.D.A.: JAIME CABALLERO		
Sede: PRINCIPAL	Periodo Académico: DOS	
Eje temático : Adquirir conocimientos sobre estructuras de datos: Los estudiantes deben entender las estructuras de datos básicas, como matrices, listas enlazadas, pilas y colas. Deben ser capaces de elegir la estructura de datos adecuada para un problema determinado y aplicarla en su código.		
Tiempo de Ejecución: 1 MES		

Aprendizajes

Objetivos de aprendizajes
Adquirir conocimientos sobre estructuras de datos: Los estudiantes deben entender las estructuras de datos básicas, como matrices, listas enlazadas, pilas y colas. Deben ser capaces de elegir la estructura de datos adecuada para un problema determinado y aplicarla en su código.
Desarrollar habilidades de trabajo en equipo: La programación a menudo se realiza en equipos, especialmente en entornos profesionales. Los estudiantes deben aprender a colaborar con otros programadores, utilizar sistemas de control de versiones y participar en el desarrollo de proyectos conjuntos.
Referentes curriculares (EBC, DBA, Matriz de Referencia, Mallas de Aprendizaje)
Familiarizarse con un lenguaje de programación: Los estudiantes deben aprender y practicar al menos un lenguaje de programación, como Python, Java o C++. Deben comprender la sintaxis del lenguaje y ser capaces de escribir programas simples.
Evidencias de Aprendizajes / Desempeños Esperados
Fomentar la creatividad y la innovación: La programación ofrece oportunidades para la creatividad y la innovación. Los estudiantes deben ser alentados a pensar de manera creativa y a buscar soluciones innovadoras a los problemas planteados.
Recursos y materiales
Se utilizará dentro del aula proyector audiovisual, sobre el cual se estará pausando frecuentemente en caso de dudas y profundizar explicaciones.



Momentos de la clase

Inicio /exploración de saberes previos

Estos objetivos de aprendizaje pueden adaptarse y ampliarse según el nivel y la duración del curso, así como los recursos disponibles

Contenido / Estructuración

Los comando de Python son una base que debes de tener muy sólida, porque son algo que necesitas para poder seguir en tu camino de programador. Como todo debes de entenderlo y saber qué le están diciendo al usuario y lo que la computadora esta leyendo, les voy a dejar la lista de comandos usada en esta vídeo y una descripción breve de para qué sirven.

Print(): este comando sirve para escribir una indicación en el sistema de programación, v.g. cuando pide una introducción de variables numéricas. También es para mostrar resultados.

Cuando se quiere escribir un texto se debe de escribir con comillas, así:

Print("Buenos días")

Int(): este comando es utilizado para que Python tome la entrada como un número entero.

Input(): este comando es usado para obtener una respuesta por parte del usuario que está usando el programa. Si no introduce números se marcará un error.

float(): interpreta la entrada como un número decimal. Si no introduce números se marcará un error.

While: es un ciclo que tiene una repetición dominada por la elección del usuario, esta se repite cada que el usuario cumpla con la condición dictada por el ciclo while.

Por si esta breve descripción no es suficiente, les adjunto un vídeo que yo hice, con un ejemplo simple para que puedan, ver lo que hace cada comando en acción.

```
# función sin parámetros o retorno de valores
```

```
def diHola():
    print("Hello!")
```

```
diHola() # llamada a la función, 'Hello!' se muestra en la consola
```

```
# función con un parámetro
```



```
def holaConNombre(name):
    print("Hello " + name + "!")


holaConNombre("Ada") # Llamada a la función, 'Hello Ada!' se muestra en la consola

# función con múltiples parámetros con una sentencia de retorno
def multiplica(val1, val2):
    return val1 * val2

multiplica(3, 5) # muestra 15 en la consola


print(max(2, 3)) # Devuelve 3 ya que 3 es el mayor de los dos valores
print(max(2, 3, 23)) # Devuelve 23 ya que 23 es el mayor de todos los valores

list1 = [1, 2, 4, 5, 54]
print(max(list1)) # Devuelve 54 ya que 54 es el valor más grande de la lista

list2 = ['a', 'b', 'c' ]
print(max(list2)) # Devuelve 'c' ya que 'c' es el valor más grande en la lista porque 'c' tiene un valor ascii mayor que 'a' , 'b'.


list3 = [1, 2, 'abc', 'xyz']
print(max(list3)) # Se genera TypeError ya que los valores en la lista son de tipos diferentes


#Soluciona el TypeError mencionado anteriormente antes de continuar con el siguiente paso
```

```
list4 = []
print(max(list4)) # Se genera ValueError ya que el argumento está vacío
```

Práctica / Transferencia

Se desarrollarán actividades donde el estudiante por medio de la observación de su entorno podrá establecer criterios claros de recursos tecnológicos, su funcionamiento, fallas y requerimientos, explicando al frente de sus compañeros obteniendo expresión oral y manejo del tema.

Descripción de la Evaluación y Valoración/cierre

Todos los estudiantes realizarán escritos investigativos sobre los temas dados, los harán saber a sus padres de familia para que en conjunto realicen evaluaciones desde sus hogares.