

INSITITUTO TECNOLOGICO DE IZTAPALAPA

INTELIGENCIA ARTIFICIAL

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

LÓPEZ CORONA MARÍA DE LOS ANGELES

MARTÍNEZ HERNANDEZ ANDRES

MENDOZA OLMEDO BERENICE JIMENA

REPORTE FINAL DE PROYECTO

ISC-8AV

REPORTE FINAL PROYECTO

Dado las investigaciones realizadas durante estas semana de trabajo hemos terminado la idea principal que se dio desde un principio para poder llegar un al resultado final; el cual consiste en un juego de adivinar el número, donde el usuario tiene que hacer una serie de intentos en adivinar el número que la maquina tiene guardada.

Conforme pasaron esta semanas de trabajo, fuimos modificando, realizando pruebas al código para obtener buenos resultado, implementando cada vez más este proyecto con el algoritmo de machine learning.

Se estructuró el código basándonos en uno de los algoritmos de machine learning, que es el algoritmo de aprendizaje automático estudia los datos para identificar patrones. No hay una clave de respuesta o un operador humano para proporcionar instrucción. En cambio, la máquina determina las correlaciones y las relaciones mediante el análisis de los datos disponibles.

En un proceso de aprendizaje no supervisado, deja que el algoritmo de aprendizaje automático interprete grandes conjuntos de datos y dirija esos datos en consecuencia. Así, el algoritmo intenta organizar esos datos de alguna manera para describir su estructura. Esto podría significar la necesidad de agrupar los datos en grupos u organizarlos de manera que se vean más organizados.

A medida que se evalúan más datos, su capacidad para tomar decisiones sobre los mismos mejora gradualmente y se vuelve más refinada.

El aprendizaje no supervisado se puede subdividir en:

Agrupamiento

Asociación

Agrupación: Agrupación significa agrupar datos inherentemente. Por ejemplo, puede clasificar los hábitos de compra de los consumidores y utilizarlos para publicidad al dirigirse a los consumidores en función de sus compras y hábitos de compra.

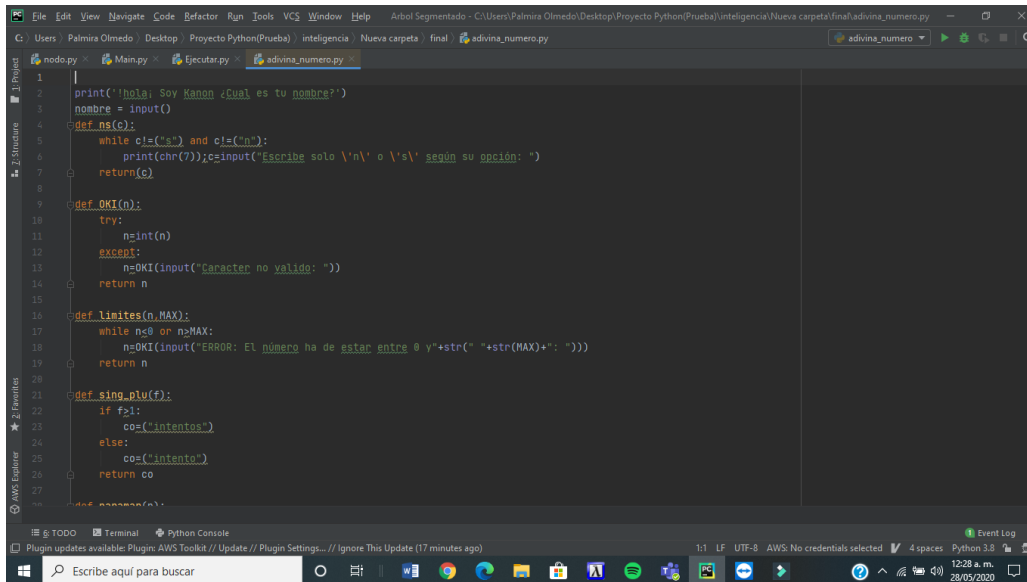
Asociación: Asociación es donde identifica las reglas que describen grandes conjuntos de sus datos. Este tipo de aprendizaje puede ser aplicable al asociar libros basados en el autor o la categoría, ya sean libros motivacionales, ficticios o educativos.

Aquí se muestra la estructura del código, donde empezamos darle estructura a la base del código, para poder ir dándole forma al proyecto y poner las líneas correspondientes ir desarrollando la sintaxis del código donde imprime la interacción, donde te pregunta el tu nombre y te explica como es el juego y el nivel de dificultad que quieras jugar .

Cada línea que se estructuró, tiene un grado de dificultad ya que no es fácil entender su funcionamiento que es lo que nos costó trabajo pero conforme se fue trabajando entendíamos más y era un poco más fácil de seguir con la sintaxis del mismo.

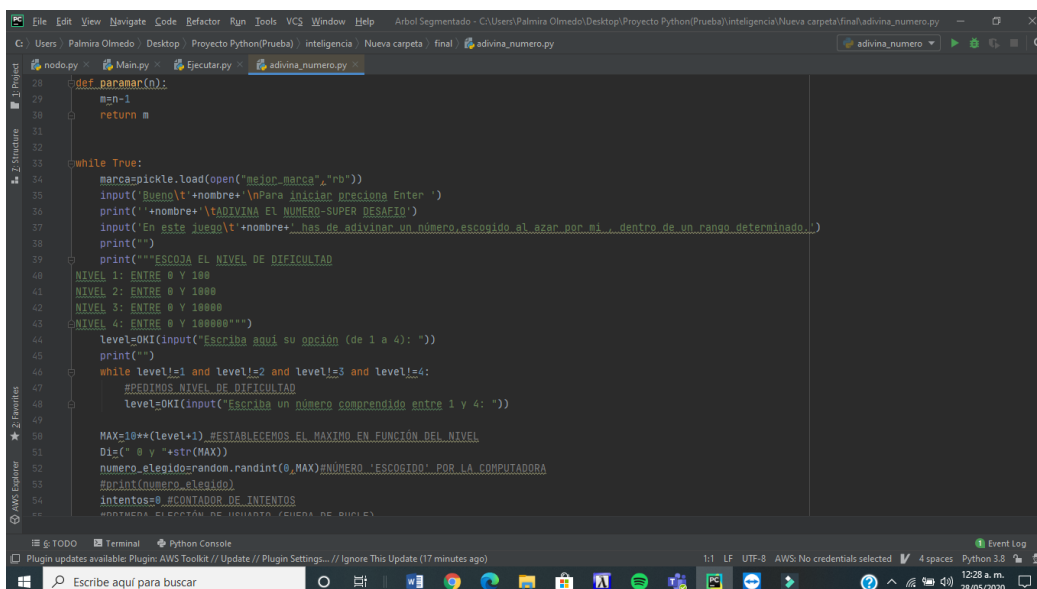
Como todos tuvimos algunos problemas técnicos, pero al final se supieron resolver para obtener el objetivo principal que es que el usuario interactúe y puede lograr adivinar el número con la menor cantidad de intentos posibles.

1. Captura del principio del código , donde la interacción con usuario donde se declara el cual es tu nombre , el que te de la opción de escribir de escribir tu nombre , que te conteste y explique como es el juego.



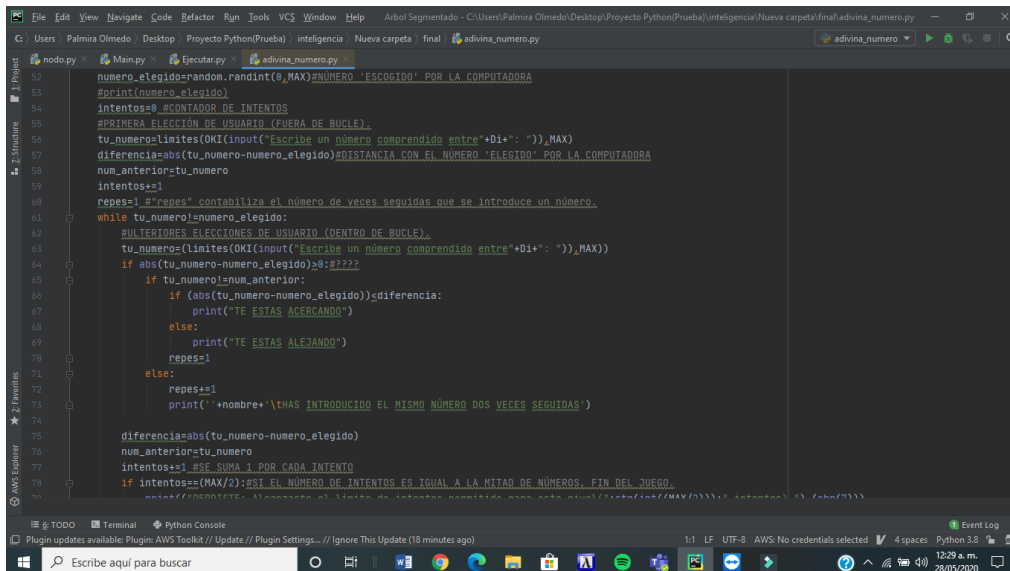
```
1 | print('¡Hola! Soy Kanon ¿Cual es tu nombre?')
2 | nombre = input()
3 | def ns(c):
4 |     while c!="s" and c!="n":
5 |         print(chr(7));c=input("Escribe solo '\n' o '\s' según su opción: ")
6 |     return(c)
7 |
8 | def OKI(n):
9 |     try:
10 |         nsint(n)
11 |     except:
12 |         nsOKI(input("Caracter no valido: "))
13 |     return n
14 |
15 | def limites(n,MAX):
16 |     while ns0 or nsMAX:
17 |         nsOKI(input("ERROR: El número ha de estar entre 0 y "+str(" "+str(MAX)+" : )))
18 |     return n
19 |
20 | def sing_plu(f):
21 |     if f>1:
22 |         co="intentos"
23 |     else:
24 |         co="intento"
25 |     return co
26 |
27 | def escape(s):
```

2.- En esta captura del código se empieza a programar el juego, donde primero se puso un menú donde tú eliges la dificultad del juego, se ingresaron 4 niveles del juego y los componentes que dará como respuesta al usuario dependiendo el nivel que escoja.



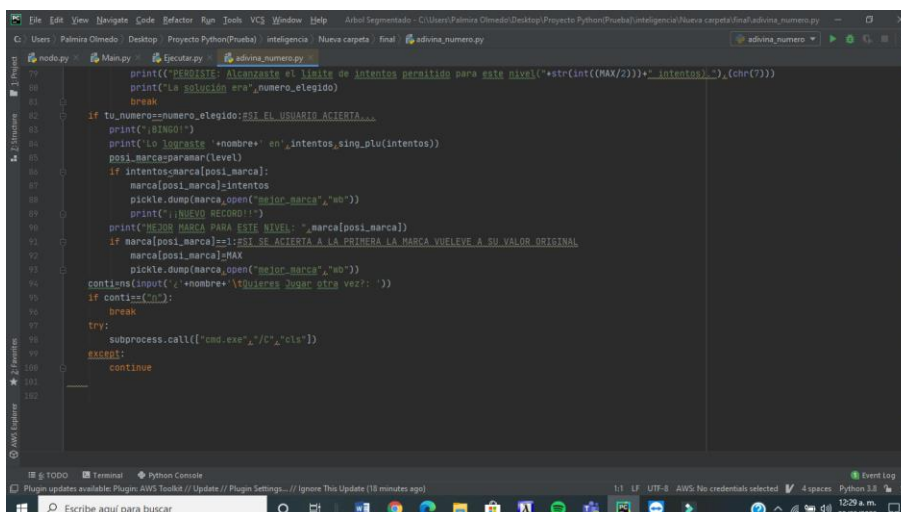
```
28 | def paramar(n):
29 |     m=n-1
30 |     return m
31 |
32 | while True:
33 |     marcas=pickle.load(open("mejor_marca","rb"))
34 |     input("¡Buena!t+nombre+ '\nPara iniciar presiona Enter ' ')
35 |     print('t+nombre+ '\tADIVINA EL NÚMERO-SUPER DESAFIO')
36 |     input("En este juego t+nombre+ 'has de adivinar un número escogido al azar por mí , dentro de un rango determinado.")
37 |     print("")
38 |     print("ESCOGA EL NIVEL DE DIFICULTAD")
39 |     NIVEL 1: ENTRE 0 Y 100
40 |     NIVEL 2: ENTRE 0 Y 1000
41 |     NIVEL 3: ENTRE 0 Y 10000
42 |     NIVEL 4: ENTRE 0 Y 100000
43 |     level=OKI(input("Escriba aquí su opción (de 1 a 4): "))
44 |     print("")
45 |     while level!=1 and level!=2 and level!=3 and level!=4:
46 |         #PEDIMOS NIVEL DE DIFICULTAD
47 |         level=OKI(input("Escriba un número comprendido entre 1 y 4: "))
48 |
49 |     MAX=10**(level+1) #ESTABLECEMOS EL MAXIMO EN FUNCIÓN DEL NIVEL
50 |     Dic= { "0" : " "+str(MAX)}
51 |     numero_elegido=random.randint(0,MAX)#NÚMERO 'ESCOGIDO' POR LA COMPUTADORA
52 |     #print(numero_elegido)
53 |     intentos=0 #CONTADOR DE INTENTOS
54 |     #MOTIVACIÓN DE JUGARLO / FUERA DE BUCLE
```

3.- En esta captura viene la parte de la sintaxis donde pusimos el contador de intentos , la primera elección del usuario y que esta fuera un bucle, la distancia del numero escogido por el computador ; la contabilización de veces seguidas que se introduce el numero, las elecciones dentro del bucle



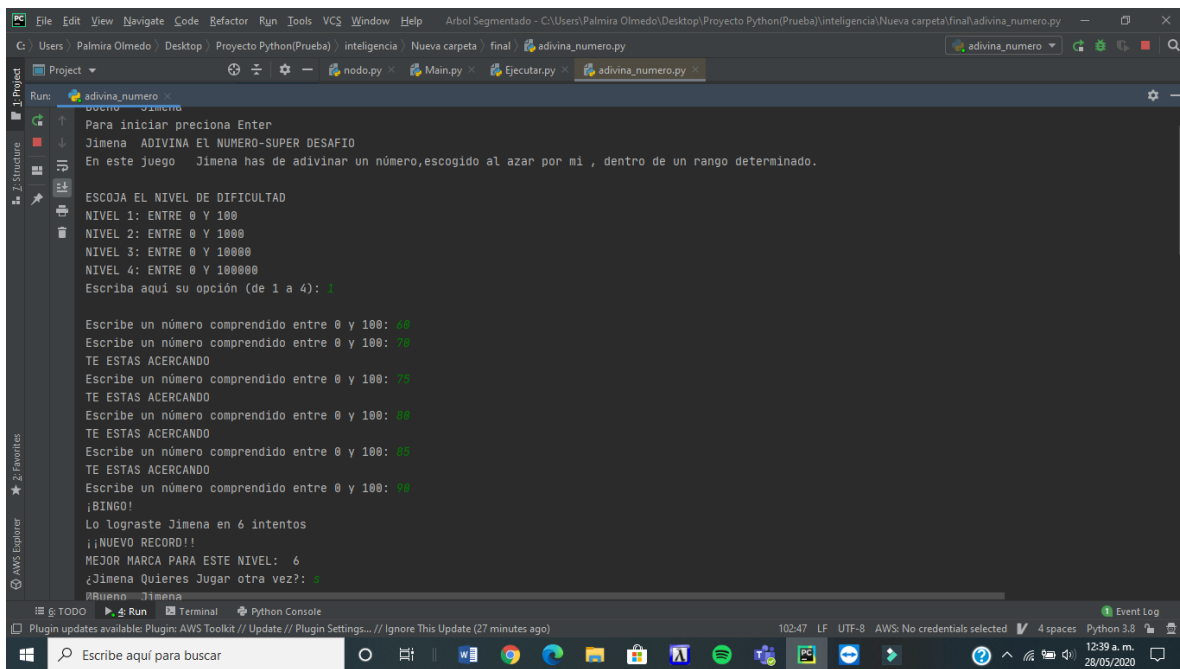
```
52 numero_elegido=random.randint(0,MAX)#NUMERO "ESCOGIDO" POR LA COMPUTADORA
53 #print(numero_elegido)
54 intentos=0 #CONTADOR DE INTENTOS
55 #PRIMERA ELECCIÓN DE USUARIO (FUERA DE BUCLE).
56 tu_numero=limites(OKI(input("Escribe un número comprendido entre "+D1+" : ")),MAX)
57 diferencia=abs(tu_numero-numero_elegido)#DISTANCIA CON EL NUMERO "ELEGIDO" POR LA COMPUTADORA
58 num_anterior=tu_numero
59 intentos+=1
60 repes=1 #repetes contabiliza el número de veces seguidas que se introduce un número.
61 while tu_numero!=numero_elegido:
62     #ULTIMOS ELECCIONES DE USUARIO (DENTRO DE BUCLE).
63     tu_numero=(limites(OKI(input("Escribe un número comprendido entre "+D1+" : ")),MAX))
64     if abs(tu_numero-numero_elegido)>8:#2222
65         if tu_numero!=num_anterior:
66             if (abs(tu_numero-numero_elegido)< diferencia):
67                 print("TE ESTAS ACERCANDO")
68             else:
69                 print("TE ESTAS ALEJANDO")
70             repes+=1
71         else:
72             repes+=1
73             print("¡+nombre+! \tHAS INTRODUCIDO EL MISMO NÚMERO DOS VECES SEGUIDAS")
74     diferencia=abs(tu_numero-numero_elegido)
75     num_anterior=tu_numero
76     intentos+=1 #SE SUMA 1 POR CADA INTENTO
77     if intentos==(MAX/2):#EST EL NÚMERO DE INTENTOS ES IGUAL A LA MITAD DE NÚMEROS, FIN DEL JUEGO.
78         #print("FIN DEL JUEGO. El número de intentos permitidos para este nivel es: "+str(int((MAX/2))+1)+" intentos")
79         #break()
```

4.- En esta captura es parte del final de la sintaxis de nuestro proyecto donde nos dice cuando encontramos el numero guardado y te dice también el número de intentos en el que lo lograste, si obtuviste una mejor marca , si deseas jugar de nuevo .



```
70 print("¡FELICIDADES! ¡Alcanzaste el límite de intentos permitido para este nivel! "+str(int((MAX/2))+1)+" intentos")
71 print("La solución era",numero_elegido)
72 break
73 if tu_numero==numero_elegido:#EST EL USUARIO ACIERTA...
74     print("¡BINGO!")
75     print("Lo lograste "+nombre+ " en "+intentos_sing_plu(intentos))
76     posi_marca=paramar(level)
77     if intentos<marca[posi_marca]:
78         marca[posi_marca]=intentos
79         pickle.dump(marca,open("marca_marca","wb"))
80     print("¡NUEVO RECORD!")
81     print("MEJOR MARCA PARA ESTE NIVEL: ",marca[posi_marca])
82     if marca[posi_marca]==1:#EST SE ACIERTA A LA PRIMERA LA MARCA VUELVE A SU VALOR ORIGINAL
83         marca[posi_marca]=MAX
84     pickle.dump(marca,open("marca_marca","wb"))
85     continue(input("¡+nombre+! ¿QUIERES JUEGAR otra vez? : "))
86     if continue():
87         break
88     try:
89         subprocess.call(["cmd.exe","/C","cls"])
90     except:
91         continue
```

5.- En esta captura es nuestro código ,funcionando con los requerimientos que le dimos , la interacción con el usuario, con el menú para escoger el grado del juego y los intentos que obtuvimos hasta encontrar el número , tu record y la marca que se tiene en ese nivel y te vuelve a preguntar si desea volver a jugar o no.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C:\Users\Palmira Olmedo\Desktop\Proyecto Python(Prueba)\inteligencia\Nueva carpeta\final\adivina_numero.py
adivina_numero.py
Run: adivina_numero.py
Python Console
Para iniciar presiona Enter
Jimena Adivina El NUMERO-SUPER DESAFIO
En este juego Jimena has de adivinar un número,escogido al azar por mí , dentro de un rango determinado.

ESCOJA EL NIVEL DE DIFICULTAD
NIVEL 1: ENTRE 0 Y 100
NIVEL 2: ENTRE 0 Y 1000
NIVEL 3: ENTRE 0 Y 10000
NIVEL 4: ENTRE 0 Y 100000
Escriba aquí su opción (de 1 a 4): 1

Escribe un número comprendido entre 0 y 100: 10
Escribe un número comprendido entre 0 y 100: 70
TE ESTAS ACERCANDO
Escribe un número comprendido entre 0 y 100: 75
TE ESTAS ACERCANDO
Escribe un número comprendido entre 0 y 100: 80
TE ESTAS ACERCANDO
Escribe un número comprendido entre 0 y 100: 85
TE ESTAS ACERCANDO
Escribe un número comprendido entre 0 y 100: 90
¡BINGO!
Lo lograste Jimena en 6 intentos
¡¡NUEVO RECORD!!
MEJOR MARCA PARA ESTE NIVEL: 6
¿Jimena Quieres Jugar otra vez?:
Buena Jimena
```