



CENTRO DE CIENCIAS BÁSICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
LENGUAJES INTELIGENTES
5° "A"

ACTIVIDAD PARA 1 Y 2 DE SEPTIEMBRE

Profesor: Alejandro Padilla Díaz

Alumno: Joel Alejandro Espinoza Sánchez

Fecha de Entrega: Aguascalientes, Ags., 5 de septiembre de 2020

Actividad para 1 y 2 de septiembre

Debido a las características de los sistemas expertos, podemos plantear acertijos lógicos preparados para el refuerzo del aprendizaje humano y artificial. Abordando el tema, hemos tocado un acertijo matemático para dar ejemplo a la afirmación anterior que después se programó.

El Acertijo

Para este acertijo tenemos un mago que conseguirá sorprender a los espectadores, junto con una persona inicial y unos cuantos participantes. La persona inicial otorgará un primer número al mago y él, con base en ese número va a comentarle otro número distinto a esta persona que sólo él sabrá diciéndole que lo tenga muy presente a lo largo del acertijo.

El acertijo se lleva a cabo, el mago ahora va a preguntarle a estos otros participantes números que ellos vayan proponiendo con la misma cantidad de dígitos que la cantidad del número que la persona inicial propuso mientras que el mago va proponiendo números para cada uno de los participantes.

Una vez que termina de preguntar a todos los participantes, le pide a la primera persona sumar su número junto con todos los propuestos por los participantes, así como los propuestos por el mago también.

Sorpresa: El mago siempre adivinará y el número que le dio en un principio a la persona inicial será la suma de todos los otros mencionados durante el truco, pero ¿cómo lo hizo?

El Análisis Matemático

Comencemos nombrando los roles del truco de magia. Tenemos al mago, a la persona inicial y al conjunto de demás participantes, establezcamos que existen n participantes jugando además de la persona inicial y el mago.

Así también, llamaremos como a al número dado por la persona inicial con $|a|$ como la cantidad de dígitos que a tiene. Llamaremos como x al número que el mago “mágicamente” sabe que será la suma de todos sus números y finalmente llamaremos dos conjuntos de números.

Llamaremos el conjunto P como el conjunto de todos los números propuestos por los participantes y que está definido bajo la igualdad $P = \{p_1, p_2, p_3, \dots, p_n\}$ donde claramente cada uno de estos elementos del conjunto son los números de cada participante y también definiremos al conjunto $Q = \{q_1, q_2, q_3, \dots, q_n\}$ que es el conjunto de valores que el mago va proporcionando para cada número de los participantes.

Finalmente, también definimos a un número *tope* el cual únicamente es un número conformado de 9 con la misma cantidad de dígitos que *a* y todos los demás.

El “mágico truco” está en que, el mago sabe de antemano el valor de *x* debido a que éste siempre tendrá el valor:

$$x = a + n(\textit{tope})$$

¿Y esto por qué es así? Debido a que el conjunto *Q* que proporciona el mago no es aleatorio (aunque pudiera parecer que sí). El mago proporciona estos números porque el quiere generar la situación:

$$\begin{aligned} p_1 + q_1 &= \textit{tope} \\ p_2 + q_2 &= \textit{tope} \\ p_3 + q_3 &= \textit{tope} \\ &\dots \\ p_n + q_n &= \textit{tope} \end{aligned}$$

De modo que tenemos *n* veces el valor de *tope* y por ello este término aparece en cómo el mago encuentra *x*. Simplemente al mago le queda sumarle *a* y él conocerá *x* desde antes que incluso los participantes den sus números.

El Programa

El programa sigue la misma lógica. En un principio se pide *a* (en el programa llamado *x1*) e inmediatamente también calculamos *|a|* (en el programa llamado *lenx1*):

```
### Pedimos el número inicial
print("===== COMENCEMOS =====")
x1 = input("Escriba un número: ")
print("")
```

Ahora con esa información, podemos obtener *tope* que es el siguiente paso en el programa (en el programa también llamado *tope*)

```
### Calculamos tope
tope = 0
for i in range(lenx1):
    tope = (tope*10) + 9
```

Con el ciclo anterior, se obtiene un número de 9 con una cantidad de dígitos igual al primer número otorgado.

Ahora lo siguiente que hará el programa es preguntar por la cantidad de participantes que habrá en el juego y conocer el valor de n (en el programa llamado p):

```
### Pedimos la cantidad de participantes en el juego
print("===== PARTICIPANTES =====")
print("")
p = input("Escriba la cantidad de participantes que nos proporcionarán un número: ")
print("")
```

Y para que parezca mágico también en el programa, el programa ya en este momento con la información que tiene, puede calcular el número final x (en el programa llamado total):

```
### ¡Qué comience la magia!
total = x1 + (p*tope)
print("===== A PREDECIR =====")
print("")
print("Yo sé que llegaremos al siguiente número: " + str(total))
print("")
```

Y ahora pediremos a todos los participantes que otorguen sus números para formar el conjunto P (en el programa llamada la lista numeros) y en el mismo procedimiento podemos calcular el conjunto Q (en el programa llamada la lista complementos):

```
### Pediremos el número de cada participante
print("===== HORA DE PARTICIPAR =====")
numeros = []
complementos = []
for i in range(p):
    print("Participante " + str(i + 1) + ":")
    xnew = input("Escriba un número que tenga " + str(len(x1)) + " dígitos: ")
    print("")
    numeros.append(xnew)
    complementos.append(tope - xnew)
```

Cabe destacar que tenemos que restringir a los participantes para que den los números correctamente. Ellos tienen que ingresar números forzosamente con la misma cantidad de dígitos que el primer número ingresado. Aprovechando que se están restringiendo, también se desarrolló un manejo de excepciones para evitar que un usuario tecleara alguna opción inválida como letras o rellenar el número con ceros a la izquierda. Para ello, cada vez que se piden números al usuario se tiene las siguientes piezas de código:

```

##### Verificamos que el número sea válido
while True:
    try:
        x1 = int(x1)
        lenx1 = len(str(x1))
        if not (lenx1 > 1):
            print("El número que introdujo no es un número entero válido")
            raise Exception
        break
    except:
        print("El número a introducir debe ser uno tal que tenga por lo menos dos dígitos")
        x1 = input("Introduzca un nuevo número: ")
        print("")

```

Ejemplo cuando se pide x1

```

##### Verificamos que el número sea válido
while True:
    try:
        p = int(p)
        if not (p > 0):
            print("La cantidad de participantes que introdujo no es un número entero válido")
            raise Exception
        break
    except:
        print("La cantidad de participantes debe ser un número entero mayor a 0")
        p = input("Introduzca una nueva cantidad de participantes: ")
        print("")

```

Ejemplo cuando se pide p

```

while True:
    try:
        xnew = int(xnew)
        dim = xnew // 10 ** (lenx1 - 1)
        if (dim < 1) or (dim > 9):
            print("El número que introdujo no es un número entero válido")
            raise Exception
        break
    except:
        print("El número otorgado debe ser un número entero positivo con " + str(lenx1) + " dígitos")
        xnew = input("Introduzca un nuevo número: ")
        print("")

```

Ejemplo cuando se pide cada elemento de la lista numeros

Finalmente, es momento de enseñarle al usuario el valor inicial, los números de los participantes y los propuestos por la computadora para que el usuario haga la suma y puede verificar que el número dado casi al principio sí es la suma de todos estos:

```

# %% Desplegamos los valores al usuario
print("===== HAGAMOS MAGIA =====")
print("")
print("Voy a proponer algunos números para cada número de los participantes")
print("y veamos cuánto suman los números siguientes:")
print("")
print("El primer número indicado fue " + str(x1))
print("")
for i in range(p):
    print("El participante " + str(i + 1) + " ha dado el número " + str(numeros[i]))
    print("Y yo propongo el número " + str(complementos[i]))
    print("")

print("Ahora comprueba si los anteriores números suman " + str(total))
print("")
print("¡VERDADERA MAGIA!")

```

Podemos hacer una prueba:

Prueba con $|a| = 4$

Pedimos el primer número:

```

===== COMENCEMOS =====

Escriba un número: 1234

```

Pedimos la cantidad de participantes:

```

===== PARTICIPANTES =====

Escriba la cantidad de participantes que nos proporcionarán un
número: 2

```

El programa hace su "magia":

```

===== A PREDECIR =====

Yo sé que llegaremos al siguiente número: 21232

```

Cada participante dará un número:

```
===== HORA DE PARTICIPAR =====  
Participante 1:  
  
Escriba un número que tenga 4 dígitos: 4560  
  
Participante 2:  
  
Escriba un número que tenga 4 dígitos: 6300
```

Vemos los resultados:

```
===== HAGAMOS MAGIA =====  
  
Voy a proponer algunos números para cada número de los  
participantes y veamos cuánto suman los números siguientes:  
  
El primer número indicado fue 1234  
  
El participante 1 ha dado el número 4560  
Y yo propongo el número 5439  
  
El participante 2 ha dado el número 6300  
Y yo propongo el número 3699  
  
Ahora comprueba si los anteriores números suman 21232  
  
¡VERDADERA MAGIA!
```

Prueba con $|a| = 6$

Pedimos el primer número:

```
===== COMENCEMOS =====  
  
Escriba un número: 439765
```

Pedimos la cantidad de participantes:

```
===== PARTICIPANTES =====  
  
Escriba la cantidad de participantes que nos proporcionarán un  
número: 2
```

El programa hace su “magia”:

```
===== A PREDECIR =====  
  
Yo sé que llegaremos al siguiente número: 2439763
```

Cada participante dará un número:

```
===== HORA DE PARTICIPAR =====  
Participante 1:  
  
Escriba un número que tenga 6 dígitos: 298610  
  
Participante 2:  
  
Escriba un número que tenga 6 dígitos: 392817
```

Vemos los resultados:

```
===== HAGAMOS MAGIA =====  
  
Voy a proponer algunos números para cada número de los  
participantes y veamos cuánto suman los números siguientes:  
  
El primer número indicado fue 439765  
  
El participante 1 ha dado el número 298610  
Y yo propongo el número 701389  
  
El participante 2 ha dado el número 392817  
Y yo propongo el número 607182  
  
Ahora comprueba si los anteriores números suman 2439763  
  
¡VERDADERA MAGIA!
```