

NOMBRE DE LA FACULTAD:

**UNIVERSIDAD INTERNACIONAL DEL
ECUADOR**

AGSINATURA:

LÓGICA DE PROGRAMACIÓN

TRABAJO:

PROYECTO INTREGADOR

CARRERA:

INGENIERIA EN CIBERSEGURIDAD

AUTOR:

BRITHANY KARIME BROWN (BK)

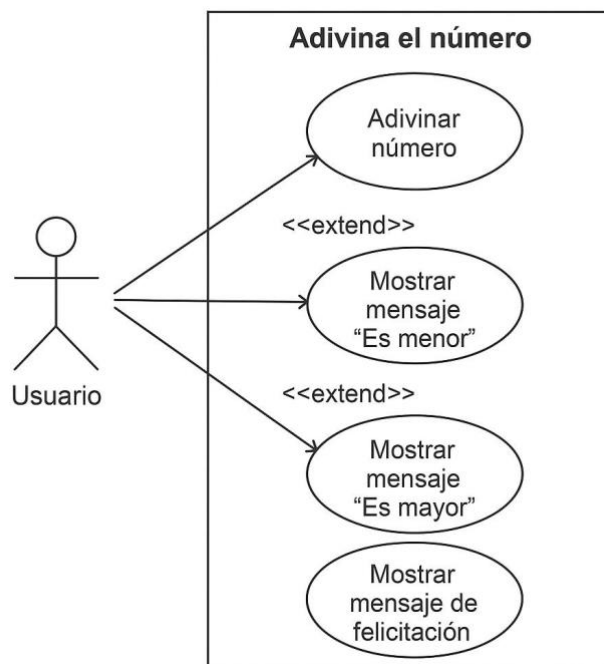
Nombre del proyecto: Impacto de las nuevas tecnologías en la sociedad: visualización del futuro.

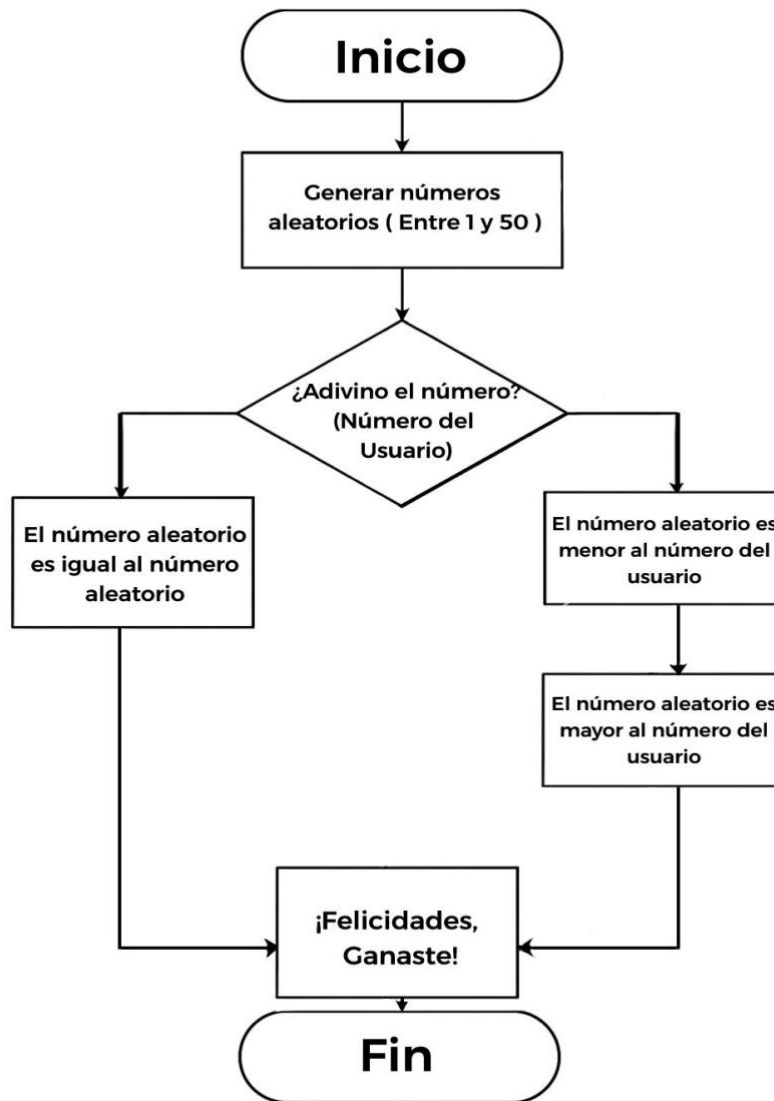
Objetivo: Reflexionar sobre como las tecnologías impactan a la sociedad actual y como podrían hacerlo en el futuro. Integrando las actividades autonomas.

- Actividad autónoma 1.-
 - a) Investigar tipos de diagramas funcionales y arquitectura de software.
 - b) Detalles del software a desarrollar y el uso de los tipos de diagramas.
- Actividad autónoma 2
 - a) Configurar entorno de desarrollo (GitHub, IDE, etc.).
 - b) Subir avances a GitHub.
 - c) Iniciar codificación básica del programa.
 - d) Resultado: Repositorio creado + configuración lista + primeros códigos.

1. Tipos de diagramas y el uso de estos.

Para mostrar los casos de estudio del software, Se utilizó diagramas de uso. Sirven para mostrar de una forma gráfica los distintos casos usos de nuestro software por parte del usuario.

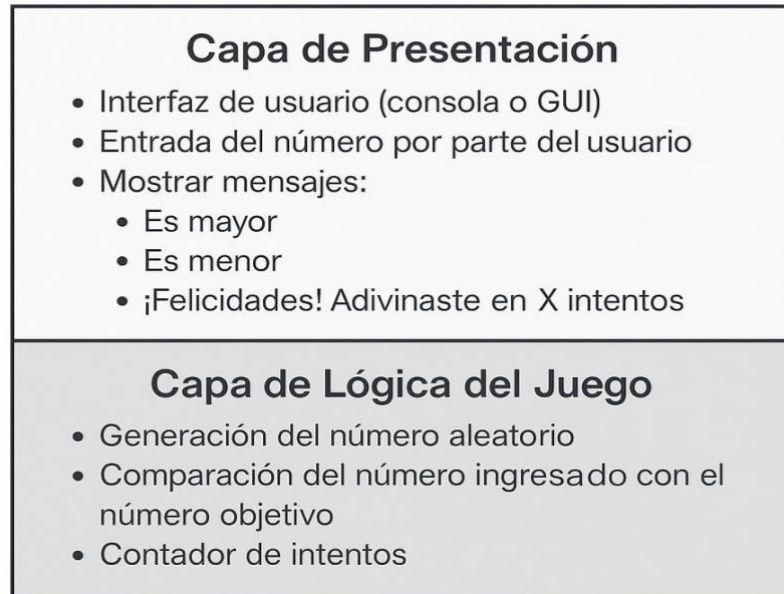




Y también se utilizó un diagrama de arquitectura en capas, que nos ayuda a graficar los componentes necesarios para la presentación del software (juego), y también el apartado lógico del software.

¿Qué Hace? Se encarga de indicar al usuario por medio de las señales visuales todo con lo puede interactuar en el juego.

Diagrama de Arquitectura en Capas



a) Capa de Presentación (Interfaz de Usuario).

- Indica “Ingresa Un Número”.
- Permite al usuario elegir entre sus opciones.
- Evalúa y muestra resultados dependiendo el caso.

b) Capas de Lógica del Juego

- Genera un número aleatorio.
- Pide un número al usuario.
- Compara los datos.
- Decide el resultado de acuerdo a las reglas del juego.

2. Análisis del software:

2.1 El juego: se divide en distintas etapas, comenzando con:

- La computadora genera un número aleatorio oculto
- El usuario intenta adivinar de que cantidad se trata (En este caso un número entre 1 y 50)
- Se compara el número del usuario con el número aleatorio.

- ❖ If (Si): El Número Usuario es mayor al Número Random. Se presenta un mensaje que le informa al usuario.
- ❖ Elif (Si/No): El Número Usuario es menor al Número Random. Se presenta un mensaje que le informa al usuario.
- ❖ Else (No): El Número Usuario es el mismo al Número Random. Se presenta un mensaje felicitando al usuario.

2.2 Requisitos Funcionales:

- Generar un número aleatorio (Computadora)
- Presentación del juego (Petición de ingresar una cantidad)
- Número del Usuario
- Comparación
- Determinar un nuevo intento (La cantidad ingresada fue mayor o menor al NumRandom)
- Número del usuario (nuevo intento)
- Comparación
- Determinar resultado correcto
- Mostrar el número correcto y los intentos totales

3. Entorno de programación.

En el autónomo N°2 personalizamos nuestro entorno de programación, con esto, personalizamos nuestro perfil y repositorio de Github. Yo le puse un gatito.



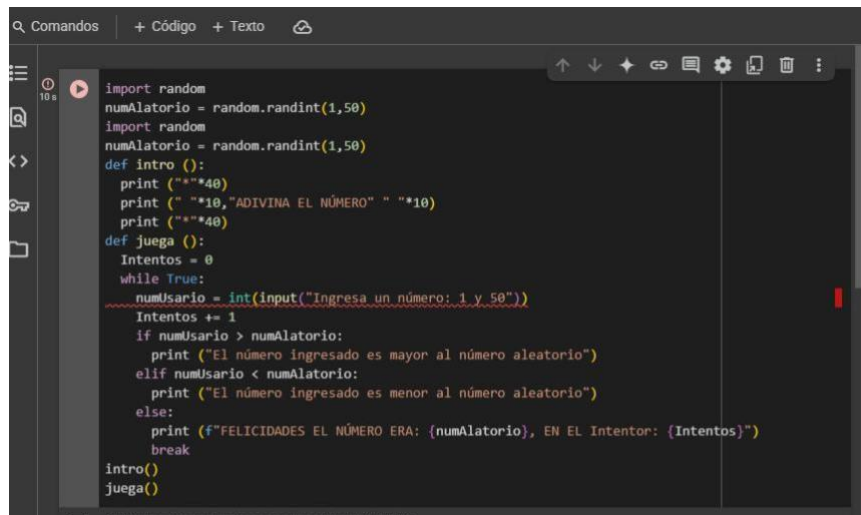
4. Desarrollo del software.

A lo largo de este módulo, el código ha sufrido cambios, se fue expandiendo gracias a los conocimientos aprendidos en cada semana

Como agregando [Break] para romper distintos bucles. Aprender la función del [While True], también agregando un (except ValueError) y aplicado los conocimientos de tuplas, lista y diccionario en el caso del software desarrollado, agregando lista de los intentos realizados.

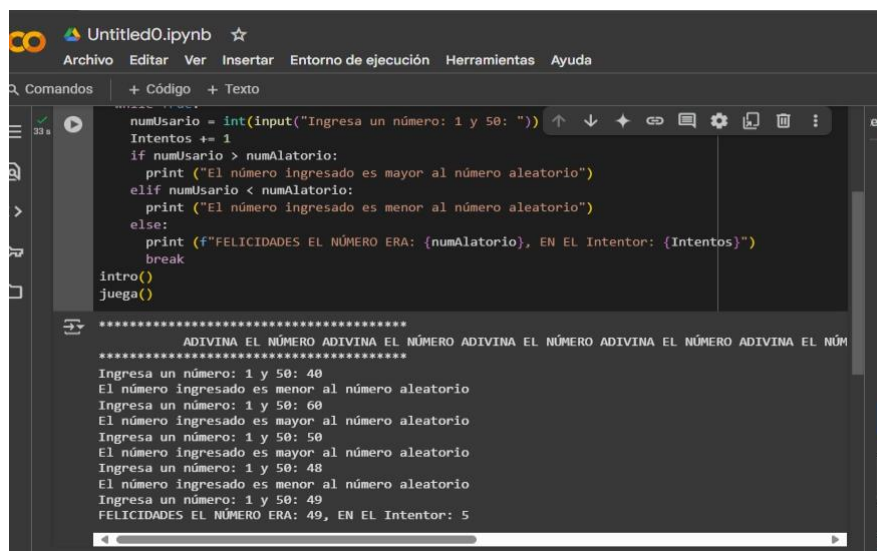
4.1 Primer Avance.

En el primer avance del software, (El juego de adivina el número), utilizando el lenguaje de programación Python y la página web de Colab de Google.



```
import random
numAlatorio = random.randint(1,50)
import random
numAlatorio = random.randint(1,50)
def intro():
    print ("***40")
    print (" *10,ADIVINA EL NÚMERO " " *10)
    print ("***40")
def juega():
    Intentos = 0
    while True:
        numUsuario = int(input("Ingresa un número: 1 y 50"))
        Intentos += 1
        if numUsuario > numAlatorio:
            print ("El número ingresado es mayor al número aleatorio")
        elif numUsuario < numAlatorio:
            print ("El número ingresado es menor al número aleatorio")
        else:
            print (f"FELICIDADES EL NÚMERO ERA: {numAlatorio}, EN EL Intentor: {Intentos}")
            break
    intro()
juega()
```

Realicé el cuerpo de mi código, tenía errores Semánticos, tanque el “Adivina el número” se repetía infinitamente porque no estaba bien definido.



```
numUsuario = int(input("Ingresa un número: 1 y 50: "))
Intentos += 1
if numUsuario > numAlatorio:
    print ("El número ingresado es mayor al número aleatorio")
elif numUsuario < numAlatorio:
    print ("El número ingresado es menor al número aleatorio")
else:
    print (f"FELICIDADES EL NÚMERO ERA: {numAlatorio}, EN EL Intentor: {Intentos}")
    break
intro()
juega()

*****
ADIVINA EL NÚMERO ADIVINA EL NÚMERO ADIVINA EL NÚMERO ADIVINA EL NÚMERO ADIVINA EL NÚM
*****
Ingresa un número: 1 y 50: 40
El número ingresado es menor al número aleatorio
Ingresa un número: 1 y 50: 60
El número ingresado es mayor al número aleatorio
Ingresa un número: 1 y 50: 50
El número ingresado es mayor al número aleatorio
Ingresa un número: 1 y 50: 48
El número ingresado es menor al número aleatorio
Ingresa un número: 1 y 50: 49
FELICIDADES EL NÚMERO ERA: 49, EN EL Intentor: 5
```

4.2. Segundo avance

En el segundo avance del código del juego. Gracias a nuestra clase de bucles, agregue (break), para cuándo el usuario ingrese un número mayor a 50 o cuando se ingrese una letra o símbolo. Cuando esté escenario sucede, el ciclo se rompe automáticamente.

```
import random

def intro():
    print("*" * 40)
    print(" " * 12 + "ADIVINA EL NÚMERO" + " " * 12)
    print("*" * 40)

def juega():
    num_aleatorio = random.randint(1, 50)
    intentos = 0

    while True:
        entrada = input("Ingresa un número entre 1 y 50: ")

        if not entrada.isdigit():
            print("El número ingresado No es válido. Fin del Juego :(")
            break

        num_usuario = int(entrada)

        if num_usuario > 50:
            print(" El número ingresado No es válido. Fin del Juego :(.")
            break

        intentos += 1

        if num_usuario > num_aleatorio:
            print(" ▲ El número ingresado es mayor al número misterioso.")
```

```
        intentos += 1

        if num_usuario > num_aleatorio:
            print(" ▲ El número ingresado es mayor al número misterioso.")
        elif num_usuario < num_aleatorio:
            print(" ▼ El número ingresado es menor al número misterioso.")
        else:
            print(f"\n🎉 ¡FELICIDADES! El número era {num_aleatorio}.")
            print(f" Lo adivinaste en {intentos} intento(s).")
            break

intro()
juega()
```

4.3 Tercer avance

En tercer avance del código del juego, decidí agregar una lista para contabilizar el número de intentos, esto de nuestra clase de Tuplas, Listas y Diccionarios.

```

import random

def intro():
    print("*" * 40)
    print(" " * 12 + "ADIVINA EL NÚMERO" + " " * 12)
    print("*" * 40)

def juega():
    num_aleatorio = random.randint(1, 50)
    intentos = 0

    while True:
        entrada = input("Ingresa un número entre 1 y 50: ")

        if not entrada.isdigit():
            print("El número ingresado No es válido. Fin del Juego :(")
            break

```

```

        entrada = input("Ingresa un número entre 1 y 50: ")

        if not entrada.isdigit():
            print("El número ingresado No es válido. Fin del Juego :(")
            break

        num_usuario = int(entrada)

        if num_usuario > 50:
            print(" El número ingresado No es válido. Fin del Juego :(.")
            break

        intentos += 1

        if num_usuario > num_aleatorio:
            print(" ▲ El número ingresado es mayor al número misterioso.")
        elif num_usuario < num_aleatorio:
            print(" ▼ El número ingresado es menor al número misterioso.")
        else:
            print(f"\n 🎉 ¡FELICIDADES! El número era {num_aleatorio}.")
            print(f" Lo adivinaste en {intentos} intento(s).")
            break

    lista = [intentos]
    print("El número de intentos es: ")
    return lista

intro()
juega()

```

```

lista = [intentos]
print("Lo adivinaste en el intento número: ")
return lista

intro()
juega()

*****
ADIVINA EL NÚMERO
*****
Ingresa un número entre 1 y 50: 20
▼ El número ingresado es menor al número misterioso.
Ingresa un número entre 1 y 50: 30
▲ El número ingresado es mayor al número misterioso.
Ingresa un número entre 1 y 50: 28
▲ El número ingresado es mayor al número misterioso.
Ingresa un número entre 1 y 50: 27
🎉 ¡FELICIDADES! El número era 27.
Lo adivinaste en el intento número:
[4]

```


5. Conclusión.

En todo el módulo, he aprendido muchas cosas para continuar mis pasos en la programación, desde los errores Sintácticos y Semánticos, (If, Elif, Else), (While True), (Break), (For, Or, And), (except ValueError). Entre muchas, para terminar con las Tuplas, Listas, y Dicionarios.

Además, también de cómo funciona el lenguaje de programación Python, los distintos diagramas de uso y como implantarlos.

Pero lo más importante, aprendimos a como pensar de manera crítica y los pasos para resolver problemas. Esto siendo los primeros pasos para convertimos en los nuevos líderes tecnológicos del mañana.

Con esto, profesora, le deseo lo mejor, le agradezco por estar con nosotros hasta este momento. Espero volver a verla pronto con esto finalizo

Muchas gracias.

B.k-Arte (Brithany Brown)

Bibliografía.

- Diagramas de casos de uso: Definición:

PMOinformatica.com. <https://www.pmoinformatica.com/2021/02/diagrama-de-casos-de-uso-definicion-elementos-ejemplos-como-hacer.html>

- ¿Qué es la arquitectura en capas: Descubre sus ventajas y ejemplos:

LATAM|TECH. Software-Cloud-Support. <https://latamtech-sac.com/que-es-la-arquitectura-en-capas-descubre-sus-ventajas-y-ejemplos/>