3. Ejercicio de Programación
Intento de Fuerza Bruta

ESTUDIANTES:

Juan Sebastian Londoño Mosquera – 1065871157

Keimer Duvan Moreno Murillo – 1077436457

MATERIA:

Herramientas de Programación I

PROFESOR:

JUAN DAVID OSORIO SANCHEZ

FECHA:

6/09/2025

Código En Virtual Studio (Python)

```python
#Juan Londoño
# Y
#Keimer Moreno

#Codigo Fuerza Bruta con validaciones ----------------------------------------
----------------------

# Paso 1: Solicitar numero de usuarios (validando que sea un numero)
while True:
    num_usuarios = input("Cuantos usuarios desea registrar: ").strip()
    if num_usuarios.isdigit() and int(num_usuarios) > 0:
        num_usuarios = int(num_usuarios)
        break
    else:
        print("Por favor, ingrese un numero valido mayor que 0.")

# Paso 2: Crear diccionario para almacenar usuarios
usuarios = {}

for i in range(num_usuarios):
    print(f"\nRegistro del usuario {i+1}")

    # Validacion nombre (solo letras y espacios)
    while True:
        nombre = input("Ingrese el nombre: ").strip()
        if nombre.replace(" ", "").isalpha():
            break
        print("Nombre invalido. Solo se permiten letras y espacios.")

    # Validacion numero de cuenta bancaria (solo digitos y minimo 10)
    while True:
        cuenta = input("Ingrese el numero de cuenta bancaria: ").strip()
        if cuenta.isdigit() and len(cuenta) >= 10:
            break
        print("Numero de cuenta invalido. Debe contener solo numeros y minimo
10 digitos.")

    # Validacion clave de 3 digitos
    while True:
        clave = input("Ingrese clave de 3 digitos: ").strip()
        if clave.isdigit() and len(clave) == 3:
            print("Clave valida")
            break
        print("Clave invalida, intente nuevamente")

    # Guardamos en el diccionario
    usuarios[nombre] = {
        "cuenta": cuenta,
        "clave": clave
    }
    print("\nUsuario registrado correctamente\n")

# Paso 3: Solicitar el usuario a consultar
usuario_buscar = input("\nIngrese el nombre del usuario a consultar: ").strip()
```

```
if usuario_buscar in usuarios:
    clave_real = usuarios[usuario_buscar]["clave"]

    # Paso 4: Simular fuerza bruta
    print("\nIniciando ataque de fuerza bruta para encontrar la clave...")
    encontrada = False
    intentos = 0

    for intento in range(0, 1000):  # claves de 000 a 999
        intentos += 1
        clave_intento = str(intento).zfill(3)  # asegura formato 3 digitos

        if clave_intento == clave_real:
            print(f"Intento {intentos}: {clave_intento}  <-- ESTA AQUI ")
            encontrada = True
            print(f"\nClave encontrada: {clave_intento}")
            print(f"Total de intentos realizados: {intentos}")
            break
        else:
            print(f"Intento {intentos}: {clave_intento}")

    if not encontrada:
        print("No se encontro la clave (error en los datos).")

    # Paso 5: Mostrar la informacion completa
    print("\nInformacion del usuario consultado:")
    print(f"Nombre: {usuario_buscar}")
    print(f"N Cuenta: {usuarios[usuario_buscar]['cuenta']}")
    print(f"Clave: {usuarios[usuario_buscar]['clave']}")

else:
    print("\nEl usuario no esta registrado.")
```

Pruebas de ejecución

Intento 1: Error al ingresar letras en vez de número de usuarios

Entrada:
Cuantos usuarios desea registrar: tres

Salida esperada en consola:

Por favor, ingrese un numero valido mayor que 0.

Intento 2: Error en número de cuenta bancaria demasiado corto

Entrada:
Ingrese el número de cuenta bancaria: 12345

Salida esperada en consola:

Número de cuenta invalido. Debe contener solo numeros y minimo 10 digitos.

Intento 3: Error en clave no numérica

Entrada:
Ingrese clave de 3 digitos: ab1

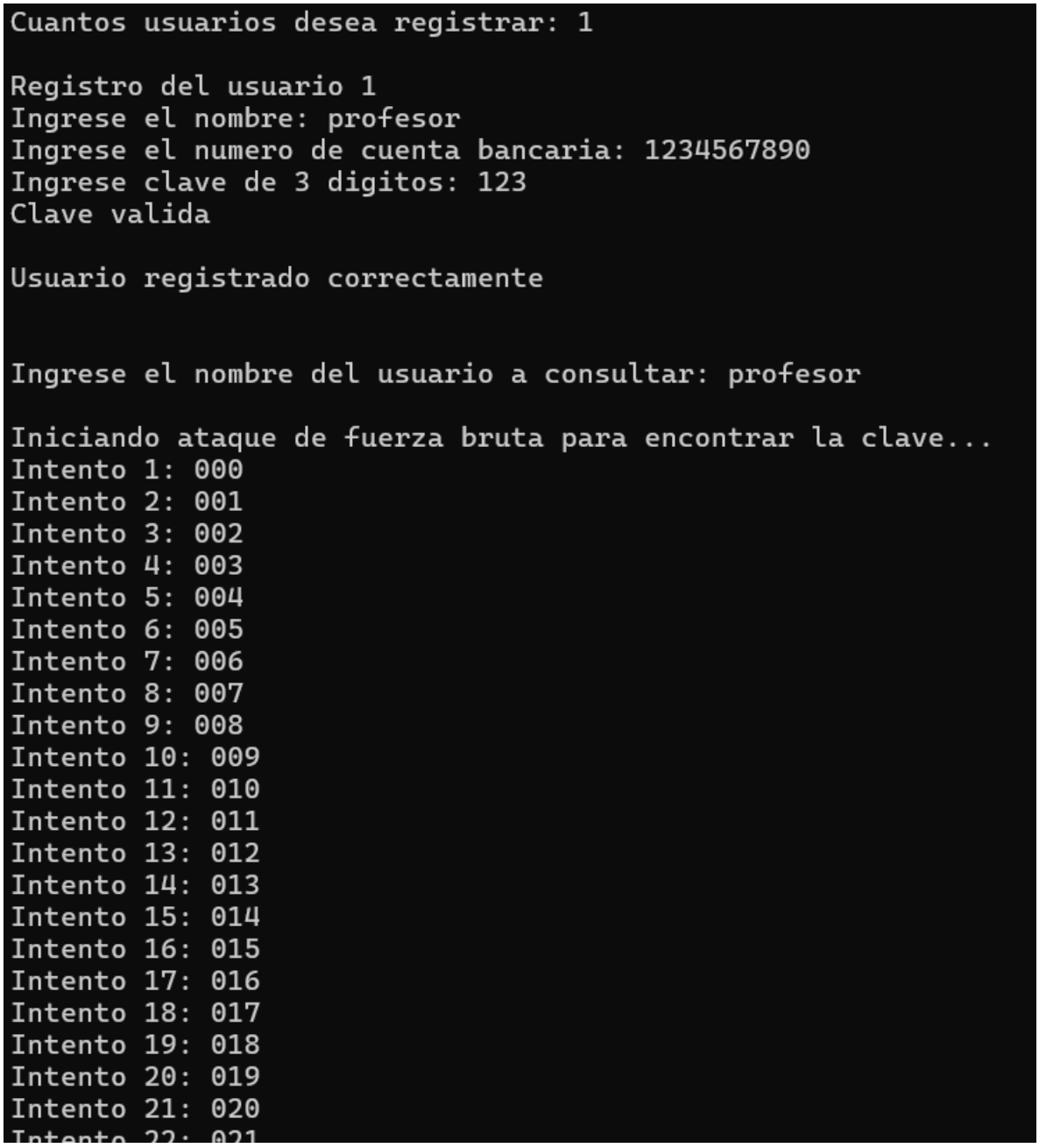
Salida esperada en consola:

Clave invalida, intente nuevamente

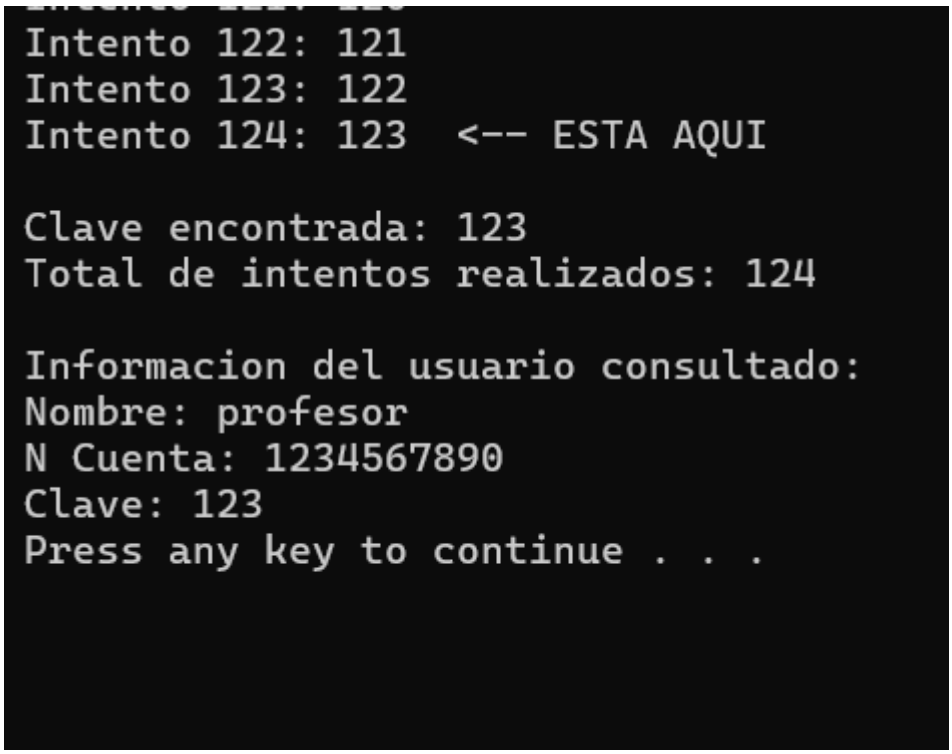
OUTPUT

```
Registro del usuario 1
Ingrese el nombre: 123
Nombre invalido. Solo se permiten letras y espacios.
Ingrese el nombre: h O L a
Ingrese el numero de cuenta bancaria: Pedro
Numero de cuenta invalido. Debe contener solo numeros y minimo 10 digitos.
Ingrese el numero de cuenta bancaria: 12345
Numero de cuenta invalido. Debe contener solo numeros y minimo 10 digitos.
Ingrese el numero de cuenta bancaria: 1234577880
Ingrese clave de 3 digitos: 5554
Clave invalida, intente nuevamente
Ingrese clave de 3 digitos: dsf
Clave invalida, intente nuevamente
Ingrese clave de 3 digitos: 123
Clave valida

Usuario registrado correctamente


Registro del usuario 2
Ingrese el nombre: juanita
Ingrese el numero de cuenta bancaria: 123123
Numero de cuenta invalido. Debe contener solo numeros y minimo 10 digitos.
Ingrese el numero de cuenta bancaria: 1231212312321
Ingrese clave de 3 digitos: 123
Clave valida

Usuario registrado correctamente


Ingrese el nombre del usuario a consultar: Pedro

El usuario no esta registrado.
Press any key to continue . . .
```

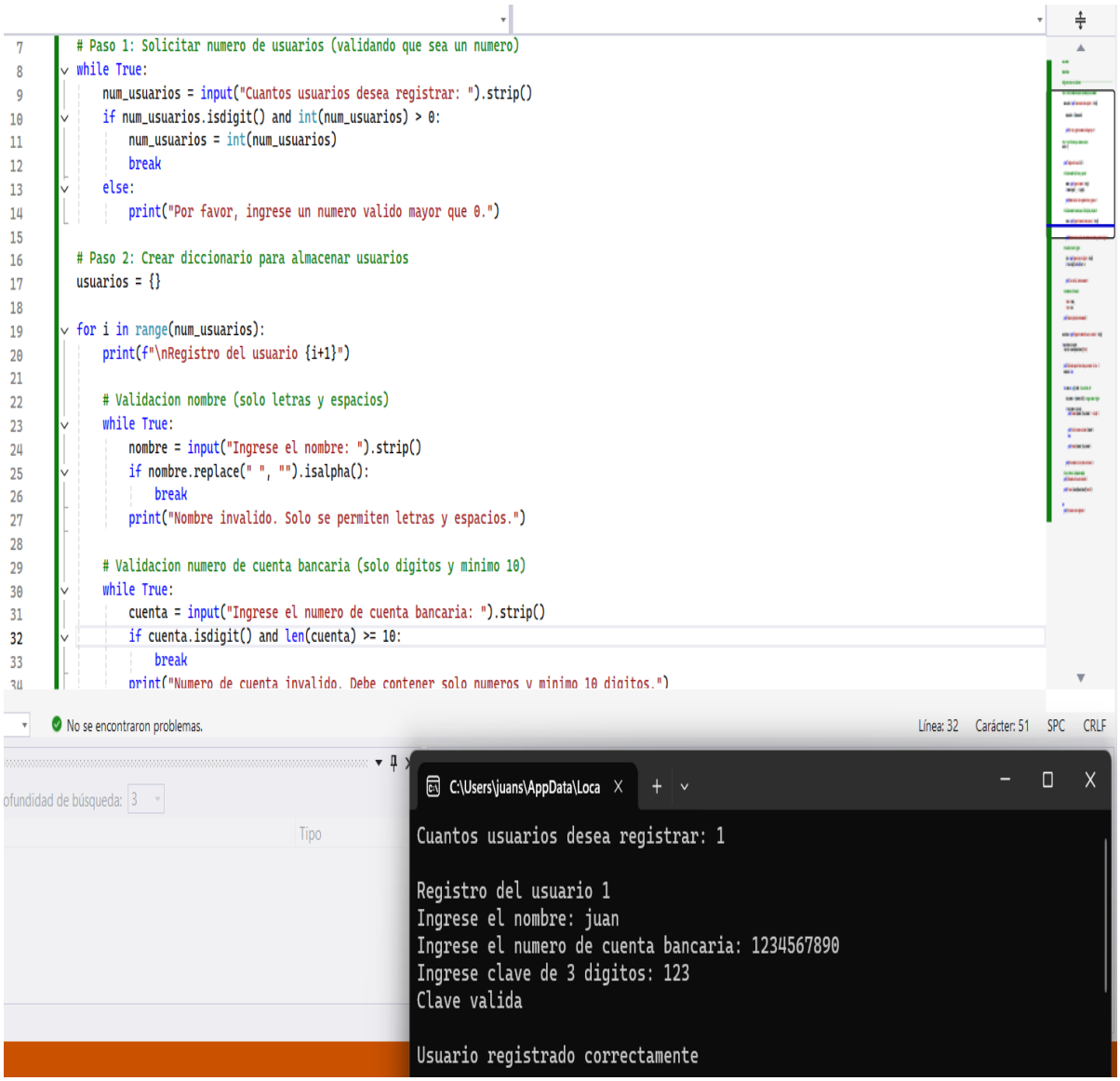
Como se logra apreciar en esta imagen, vemos que valida bien los errores por cada atributo/dato del sistema creado y verifica si lo ingresado es correcto, sino, vuelve a repetírtelo con un mensaje incluido. Al ingresar lo correcto, prosigue con su función y después tienes que escoger el usuario a atacar , si no ingresas el correcto el programa se termina.



Aquí vemos una versión con solo un usuario registrado en la que esta vez si ingresamos el nombre correcto que pusimos arriba “Profesor” y empezó el ataque en menos de un segundo ya tenia la clave encontrada.



Por ultimo como logramos apreciar, en el intento 124 ya que el 1 es el 0, encontró la clave exitosamente y te lo remarca con un texto para lograr apreciar la clave correctamente y mas abajo se muestran los datos exactos exitosamente encontrados.



```python
# Paso 1: Solicitar numero de usuarios (validando que sea un numero)
while True:
    num_usuarios = input("Cuantos usuarios desea registrar: ").strip()
    if num_usuarios.isdigit() and int(num_usuarios) > 0:
        num_usuarios = int(num_usuarios)
        break
    else:
        print("Por favor, ingrese un numero valido mayor que 0.")

# Paso 2: Crear diccionario para almacenar usuarios
usuarios = {}

for i in range(num_usuarios):
    print(f"\nRegistro del usuario {i+1}")

    # Validacion nombre (solo letras y espacios)
    while True:
        nombre = input("Ingrese el nombre: ").strip()
        if nombre.replace(" ", "").isalpha():
            break
        print("Nombre invalido. Solo se permiten letras y espacios.")

    # Validacion numero de cuenta bancaria (solo digitos y minimo 10)
    while True:
        cuenta = input("Ingrese el numero de cuenta bancaria: ").strip()
        if cuenta.isdigit() and len(cuenta) >= 10:
            break
        print("Numero de cuenta invalido. Debe contener solo numeros y minimo 10 digitos.")
```

No se encontraron problemas.

Línea: 32 Carácter: 51 SPC CRLF

ofundidad de búsqueda: 3

Tipo

C:\Users\juans\AppData\Loca

```
Cuantos usuarios desea registrar: 1

Registro del usuario 1
Ingrese el nombre: juan
Ingrese el numero de cuenta bancaria: 1234567890
Ingrese clave de 3 digitos: 123
Clave valida

Usuario registrado correctamente
```

Visual Studio