ACTIVIDAD EVALUATIVA EJE 3

Laura Juliana Ramírez Barrera

Fundación Universitaria del Área Andina

Ingeniería De Sistemas – Virtual

[Modelos De Programación II - IS - 202460-6a - 61](https://areandina.instructure.com/courses/51454)

Deivys Morales

15 de agosto de 2024

INTRODUCCIÓN

Durante el desarrollo de diversas aplicaciones y sistemas de red, los hilos y sockets juegan un papel fundamental para garantizar un gran rendimiento y la capacidad de comunicación entre diferentes procesos. Mientras los hilos permiten la ejecución de múltiples tareas al tiempo los sockets establecen la comunicación entre distintos dispositivos en una red, logrando facilitar el intercambio de datos entre un cliente y un servidor. En este trabajo lograremos ver como funcionan mediante lenguaje de programación Python.

OBJETIVOS

El objetivo principal de este trabajo es entender cómo se implementan los hilos y sockets en lenguaje de programación Python mediante la construcción de un juego, entre el cliente y el servidor donde el cliente enviara números al servidor de ac

TAREA

Construir una aplicación en python que permite aplicar el tema de hilos y socket, la aplicación consiste en:

• Construir un juego en el que se generan números aleatorios a través de hilos por parte del cliente.

• Establecer una comunicación entre el cliente y el servidor para que el servidor adivine los números generados por el cliente.

• Contabilizar los aciertos y desaciertos cuando finalice la aplicación.

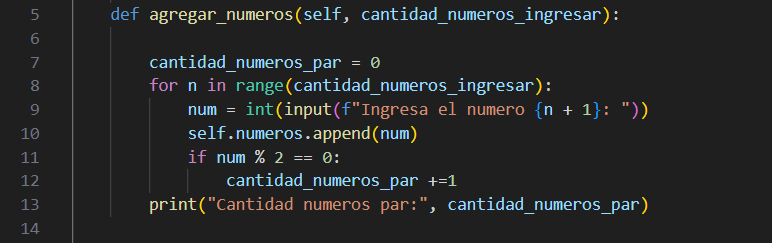
• Cuando el servidor lleve 3 desaciertos seguidos debe salir un mensaje de “Perdiste”.

• Terminar la aplicación cuando el cliente envíe la palabra “terminar”

SOLUCIÓN

Primero creamos dos archivos, client\_socket.py quien enviara los mensajes y server\_socket.py quien recibirá los mensajes mediante la implementación del socket.

Continuamos definiendo el método agregar\_numeros donde se le solicitara al usuario que nos indique la cantidad de números que desea ingresar. Dentro de este método definimos la variable cantidad\_numeros\_par que por medio del ciclo for se cuenten cuantos números se están ingresando y con ayuda de la condición if determine cuantos de estos son pares.



Texto

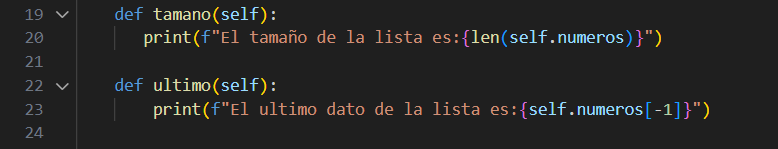
Descripción generada automáticamente con confianza media

Así mismo definimos dentro de la clase los demás métodos que son: def vacia que verifica si la lista esta vacia por medio de la condición if.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

El método def ultimo que nos imprime el último dato de la lista utilizando la notación [-1] que trae Python para dicho caso. El método def tamaño que nos devuelve la cantidad de números que fueron ingresados en la variable self.numeros mediante la función len().

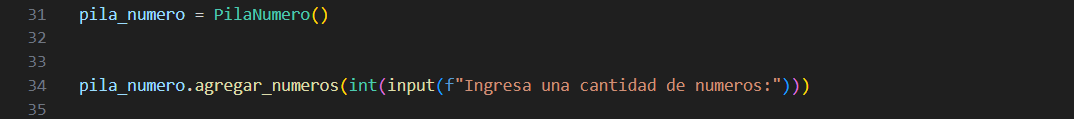


El método def eliminar que con ayuda de la función .pop elimina el ultimo dato ingresado en la lista

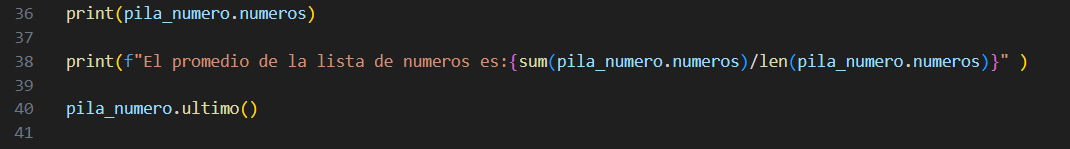
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego de haber definido todos los métodos creamos la variable pila\_numero que nos ayudara a utilizar los métodos de nuestra clase, en este caso solo se llaman a los solicitados en el eje.



Mostramos los datos que fueron ingresados en la lista y para sacar el promedio se realiza una pequeña operación sumando todos los números que se ingresaron dividiendo el resultado por la cantidad de datos que hay en la lista. Así mismo se solicita que muestre el último dato almacenado.





CODIGO COMPLETO ESTRUCTURA DE DATOS PILA Y SOLUCION

class PilaNumero:

    def \_\_init\_\_(self) -> None:

        self.numeros = []

    def agregar\_numeros(self, cantidad\_numeros\_ingresar):

        cantidad\_numeros\_par = 0

        for n in range(cantidad\_numeros\_ingresar):

            num = int(input(f"Ingresa el numero {n + 1}: "))

            self.numeros.append(num)

            if num % 2 == 0:

                cantidad\_numeros\_par +=1

        print("Cantidad numeros par:", cantidad\_numeros\_par)

    def vacia(self):

        if len(self.numeros) == 0:

            print("La lista esta vacia:")

    def tamano(self):

       print(f"El tamaño de la lista es:{len(self.numeros)}")

    def ultimo(self):

        print(f"El ultimo dato de la lista es:{self.numeros[-1]}")

    def eliminar(self):

        self.numeros.pop()

pila\_numero = PilaNumero()

pila\_numero.agregar\_numeros(int(input(f"Ingresa una cantidad de numeros:")))

print(pila\_numero.numeros[::-1])

print(f"El promedio de la lista de numeros es:{sum(pila\_numero.numeros)/len(pila\_numero.numeros)}" )

pila\_numero.ultimo()

pila\_numero.tamano()

Texto

Descripción generada automáticamente

EJERCICIO #2

Crea una lista enlazada simple de tipo Pila y Cola que permita almacenar los n datos de una persona como son: código, nombre, teléfono y edad.

• Mostrar la lista de los elementos

• Eliminar el primer elemento de la lista y mostrar nuevamente la lista

• Contar los elementos que quedan en la lista.

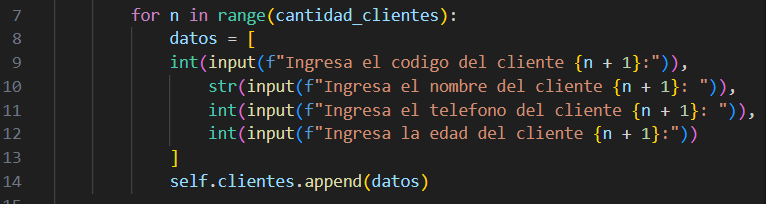
SOLUCIÓN

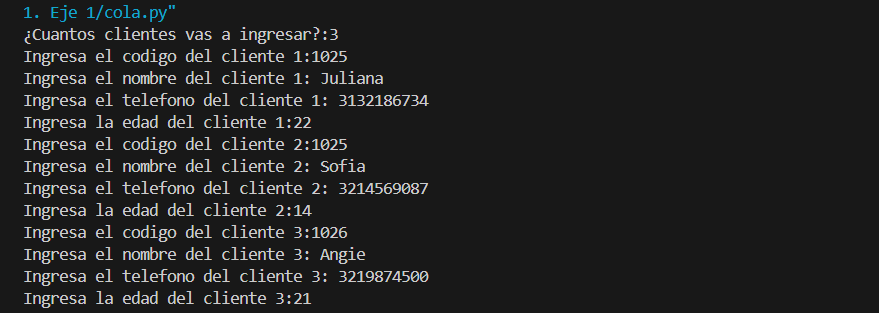
Definimos la clase Cliente y creamos la variable self.clientes para almacenar los datos que se le solicitaran al usuario, por medio del método agregar\_clientes se añadirán los datos a la lista anidada de clientes.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Con el ciclo for contaremos la cantidad de clientes que se ingresaron y por medio de la lista datos = [] se almacenaran los datos del cliente: código, nombre, teléfono y edad.



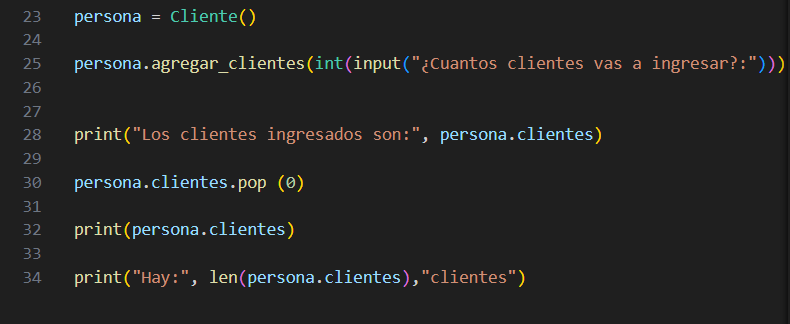


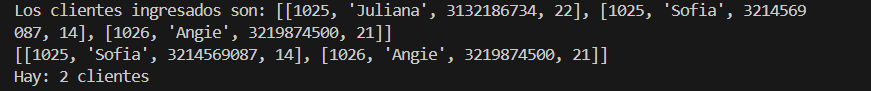
Definimos el método mostrar\_primer\_cliente que obtiene el primer dato teniendo en cuenta que en Python los índices comienzan desde 0 por eso al colocarlo en realidad le estamos pidiendo que nos muestre el primer dato en la lista. Lo mismo pasa al definir la función mostrar\_ultimo\_cliente solo que al usar la notación -1 recoge la última posición de la lista.

Texto

Descripción generada automáticamente

Por último definimos la variable persona para poder llamar los métodos que posterior mente fueron incluidos dentro de la clase Cliente. Mostramos los datos de la lista, por medio de la función pop() le pedimos que elimine el primer elemento de la lista, mostramos de nuevo los datos que quedan y finalmente con la función len nos mostrara cuantos datos quedan en la lista.





CODIGO COMPLETO ESTRUCTURA DE DATOS COLA Y SOLUCION

class Cliente:

    def \_\_init\_\_(self) -> None:

        self.clientes = []

    def agregar\_clientes(self, cantidad\_clientes):

        for n in range(cantidad\_clientes):

            datos = [

            int(input(f"Ingresa el codigo del cliente {n + 1}:")),

                str(input(f"Ingresa el nombre del cliente {n + 1}: ")),

                int(input(f"Ingresa el telefono del cliente {n + 1}: ")),

                int(input(f"Ingresa la edad del cliente {n + 1}:"))

            ]

            self.clientes.append(datos)

    def mostrar\_primer\_cliente(self):

        return self.clientes[0]

    def mostrar\_ultimo\_cliente(self):

        return self.clientes[-1]

persona = Cliente()

persona.agregar\_clientes(int(input("¿Cuantos clientes vas a ingresar?:")))

print("Los clientes ingresados son:", persona.clientes)

persona.clientes.pop (0)

print(persona.clientes)

print("Hay:", len(persona.clientes),"clientes")

Texto

Descripción generada automáticamente

CONCLUSIONES

El lenguaje Python es muy sencillo de entender y gracias a esto se puede entender de una manera más simple las estructuras de datos tipo pila y cola que se utilizaron para el desarrollo del eje. De igual manera estas estructuras de datos nos ayudan a entender de una manera mas realista el como funciona el mundo hoy en día, desde cómo funciona pedir un turno en una fila hasta como se organizan los libros en un estante.

REFERENCIAS

Marco Mendoza. *Aprende Como un Profesional Python 3.12.1 el Lenguaje de Programación Más Demandado del Mundo.* <https://www.udemy.com/course/curso-de-python-310-aprende-desde-cero-a-pro-2022-parte-3/> . Agosto de 2024

Python. *Estructuras de datos.* <https://docs.python.org/es/3.12/tutorial/datastructures.html#using-lists-as-stacks> . Agosto de 2024