# CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍA DEPARTAMENTO DE CIENCIAS COMPUTACIONALES



#### SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE SISTEMAS OPERATIVOS BECERRA VELZQUEZ VIOLETA DEL ROCIO INGENIERIA EN COMPUTACIÓN SECCION D01

## PACHECO ROMERO VICTOR MANUEL 216589519

#### LINK VIDEO

https://drive.google.com/file/d/1SEWHWfl0ImiagXZUxXaJBeyXD5E6X-Mh/view?usp=sharing

## Actividad de aprendizaje 2

## Índice

## Contenido

Objetivo	4
Desarrollo	4
Conclusiones	6

## Índice de ilustraciones

Ilustración 1-Clase	4
Ilustración 2-Insertar datos	4
Ilustración 3-Validación de división y residuo	4
Ilustración 4-Validacion de id	5
Ilustración 5-Validacion de tiempo	5
Ilustración 6-Menú de ejecución	5
Ilustración 7-Consola	

#### Objetivo

El objetivo de esta actividad es la de desarrollar un programa que simule el procesamiento por lotes el cual deberá ingresar un numero de N procesos y mostrar en tiempo real esos procesos aplicando algunas características a cada proceso como lo son el tiempo, nombre id, y la operación a realizar, además de mostrar en tiempo real el número de lotes pendientes y un contador global de los procesos.

#### Desarrollo

Durante el desarrollo de la actividad comencé creando una clase con los atributos necesarios para la creación del programa.

```
class Lote:

def __init__(self, id, nombre, operacion, tiempo_estimado, num1, num2, num_lote):

self.id = id

self.operacion = operacion

self.operacion = operacion

self.tiempo_estimado = tiempo_estimado

self.num1 = num1

self.num2 = num2

self.num_lote = num_lote

def __str__(self):
    return ( str(self.id) + " " + self.nombre + " "

+ str(self.operacion ) + " " + str(self.tiempo_estimado)

+ " " + str(self.num1) + " " + str(self.num2) + " " + str(self.num_lote) + " ")
```

Ilustración 1-Clase

Seguidamente de esto cree los menus necesarios para que el usuario agregara los datos requeridos.

```
def obtener_datos(num_lotes):
    os.system("cls")
    nombre = input("Nombre: ")
    id = verificar_id()
    tiempo_estimado = validar_tiempo()
    operacion = menu_operaciones()
    num1 = int(input("Numero 1: "))
    num2 = int(input("Numero 2: "))
    continuar = True
```

Ilustración 2-Insertar datos

Tambien me centre en hacer las validaciones correspondientes como lo son la del tiempo mayor a 0, y que la division y el residuo no sea 0.

```
if((operacion == 4 or operacion == 5) and num2 == 0):

while(continuar):

print("Tiene que ser un numero mayor a 0")

num2 = int(input("Numero 2: "))

continuar = (num2==0)

lote = Lote(id,nombre,operacion,tiempo_estimado,num1,num2, num_lotes)

cantidad lotes.append(lote)
```

Ilustración 3-Validación de división y residuo

```
147 \( \times \text{def verificar_id():} \)
148 \( \text{continuar} = \text{True} \)
149 \( \text{while (continuar):} \)
150 \( \text{id} = \text{int(input("ID: "))} \)
151 \( \text{if (id} > 0 \) and buscar_id(id)==False):} \)
152 \( \text{continuar} = \text{False} \)
153 \( \text{else:} \)
154 \( \text{print("ID no se puede repetir ")} \)
155 \( \text{return id} \)
```

Ilustración 4-Validacion de id

Ilustración 5-Validacion de tiempo

Otras de las cosas necesarias fue crear una interfaz en la consola para observar los datos de manera ordena, en esta parte fue donde mas trabaje ya que el objetivo era simular un procesamiento con lotes asi que era necesario agreagar librerias como la de time.sleep para controlar el tiempo de ejecucion del ciclo while, en esta parte el codigo se seguia ejecutando hasta que se terminaran los procesos.

Ilustración 6-Menú de ejecución

Finalmente, el programa muestra todo de forma correcta, procesando el tiempo, lotes pendientes, lote en ejecución, operación, lotes terminados de manera asíncrona.

```
Numero de Lotes pendientes: 0
Lote en ejecución: 3
Contador Global: 90
************
Lotes en ejecución
Nombre: Melany
Operación: 12 / 4
ID: 12
TME: 12
TR: 0
Lotes terminados
       NL
             |Operacion y resultado
             2 / 4 = 0.5
             4 % 6 = 4
            | 52 / 52 = 1.0
            21 * 72 = 1512
            16 * 84 = 1344
       1
            411 - 22 = 389
       | 2
             21 - 12 = 9
              21 - 12 = 9
       | 2
              12 % 2 = 0
               23 + 411 = 434
              12 / 4 = 3.0
```

Ilustración 7-Consola

#### **Conclusiones**

Esta actividad me pareció bastante interesante inicialmente comencé desarrollando el programa en c++ pero creía más factible utilizar python debido al manejo más fácil que tiene con las listas, el único inconveniente que tuve fue a la hora de mostrar los datos, fue algo complicado debido a que eran muchos datos que debía tener en cuenta al mismo tiempo, considero que puede lograr el objetivo de la actividad, gracias a la implementación de esta práctica pude aprender un poco más sobre el manejo del procesamiento por lotes. Me gusto esta actividad y considero que fue de gran ayuda además de que me quedo mucho más claro el tema.