**TP3: Introducción a los patrones de Diseño clase 4**

*Entrega: 20/09*

Integrantes:

- Quinteros, Mariana

- Bazán Azargado, Ma. Emilia.

- Torres Urzua, Siro Ezequiel

- Esteche, Yamila Eliana

- Balmaceda, Adriana Verónica

- Martinez, Fabricio

- Juan Beresiarte

ACTIVIDAD:

-Realizar la actividad de Polimorfismo: Imaginemos un programa que gestiona diferentes tipos de animales utilizando polimorfismo para representar comportamientos comunes como comer y sonido.

# Clase base

class Animal:

def comer(self):

raise NotImplementedError("Este método debe ser implementado por la subclase")

def sonido(self):

raise NotImplementedError("Este método debe ser implementado por la subclase")

# Subclase 1: Perro

class Perro(Animal):

def comer(self):

return "El perro está comiendo croquetas."

def sonido(self):

return "El perro ladra: ¡Guau!"

# Subclase 2: Gato

class Gato(Animal):

def comer(self):

return "El gato está comiendo atún."

def sonido(self):

return "El gato maúlla: ¡Miau!"

# Subclase 3: Vaca

class Vaca(Animal):

def comer(self):

return "La vaca está comiendo hierba."

def sonido(self):

return "La vaca hace: ¡Muu!"

# Función que recibe cualquier animal

def interaccion\_animal(animal: Animal):

print(animal.comer())

print(animal.sonido())

# Ejemplo de uso

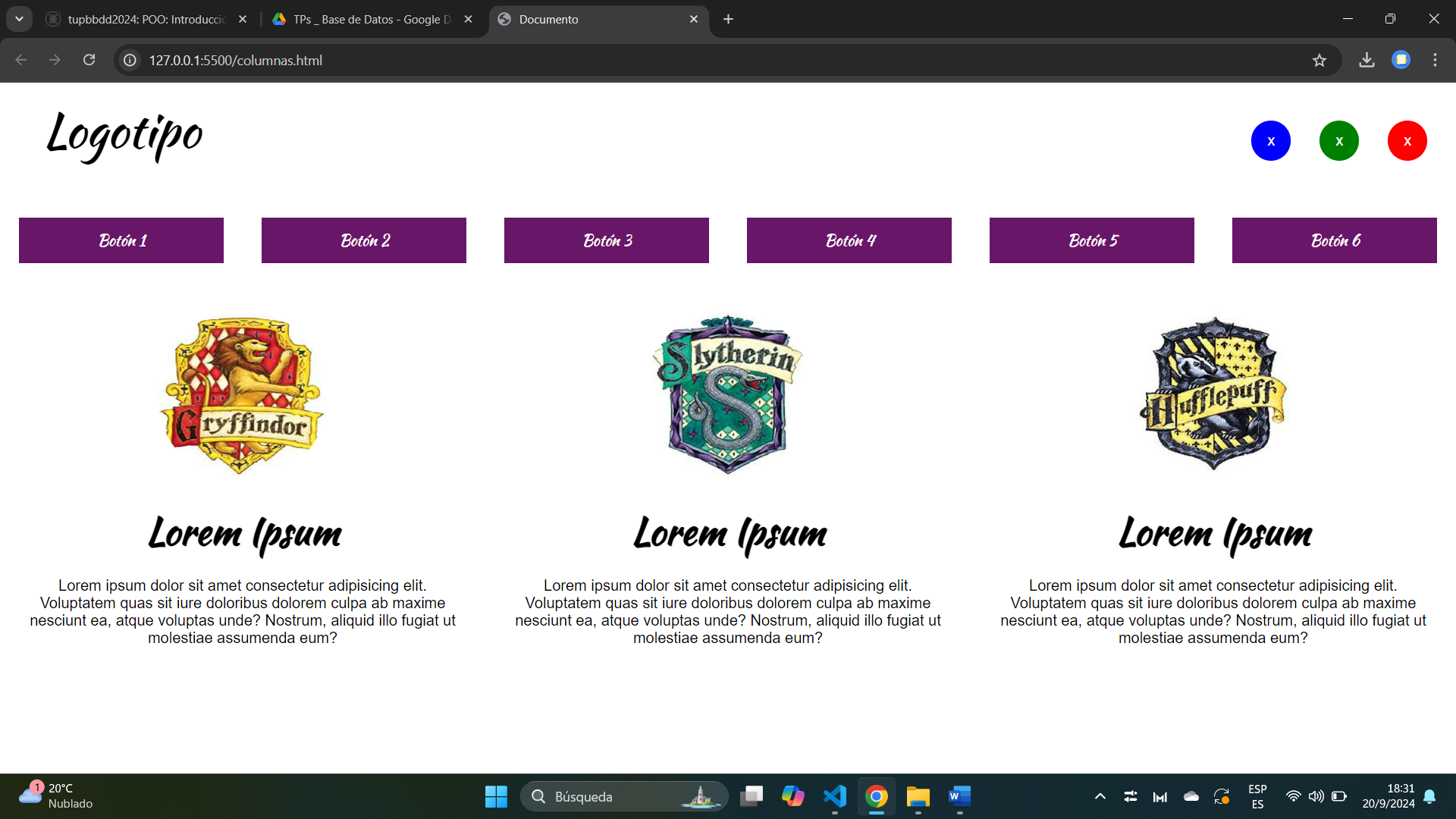
animales = [Perro(), Gato(), Vaca()]

for animal in animales:

interaccion\_animal(animal)

print("-" \* 30)

-Añadir los cambios realizados en nuestro diseño responsivo, compartir captura de visualización.



Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente