# **Evaluación n°4**

# Crud mongo flask

NOMBRE: Benjamin Martinez, Brandon Marín

CARRERA: ingeniería en informática

ASIGNATURA: base de datos no estructuradas

PROFESOR: Paulo Leonardo Taípe Ortega

FECHA:9/07/2023

Contenido

[1 Introducción 3](#_Toc139825742)

[2 Consulta de datos (Find) 4](#_Toc139825743)

[3 Insertar datos (insert) 5](#_Toc139825744)

[4 Activar/desactivar(delete) 6](#_Toc139825745)

[5 Actualizar datos (update) 7](#_Toc139825746)

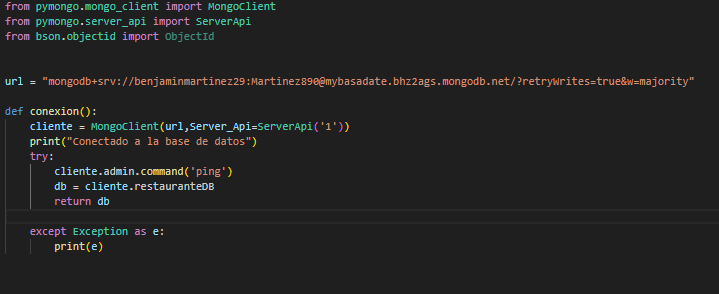
# Introducción

Este proyecto es un Crud donde se utilizó el framework flask y la base de datos mongodb y el conocido lenguaje de programación Python con un front de HTML para la interfaz. donde los datos son almacenados en la base de datos de mongodb que es una base de datos no estructural y el framework flask nos otorga

La conexión entre mongodb y el lenguaje de programación de Python para extraer los datos de la base de datos y renderizarlo en el front con HTML y el framework boostrap para el diseño de las páginas y con esto se creará un sistema llamado crud (Create, Read, Update, Delete)

# Consulta de datos (Find)

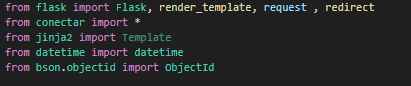
Primero creamos el archivo conexión.py para utilizar Python e instalamos pymongo, flask



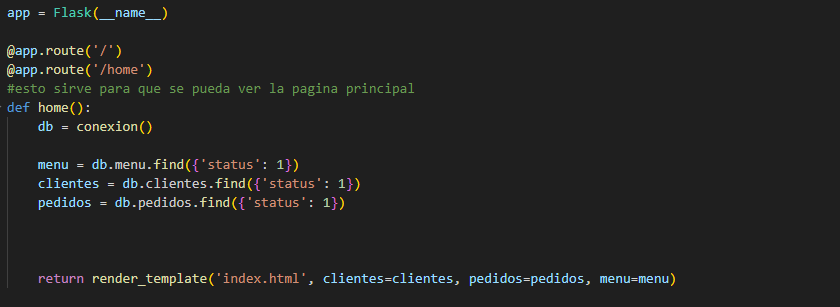
Y realizamos la conexión en Python e importamos las librerías de pymongo

En la url ponemos nuestras credenciales de nuestra base de datos como la contraseña y el usuario

Creamos otro archivo llamado app.py y e importamos el archivo conexión y las librerías a ocupar



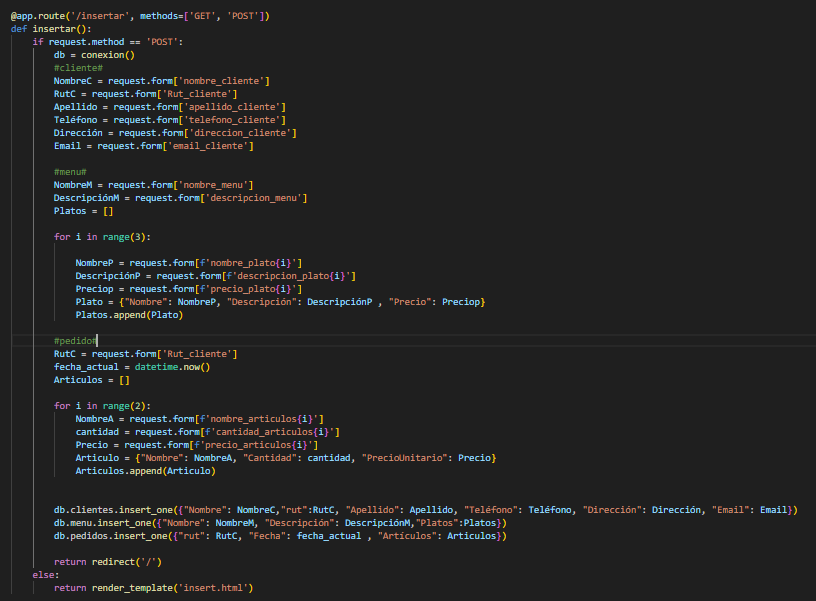
Ahora se programa una función para dirigirnos a la pagina por defecto y realizamos las consultas con los comandos de pymongo



Para renderizarlo y se pueda ver en el html

# Insertar datos (insert)

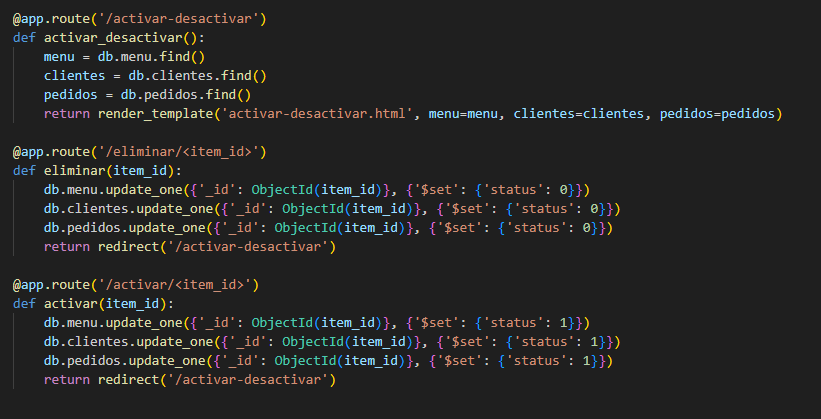
Para insertar datos se creó una función insertar con una ruta independiente



Asignamos variables para que solicite los atributos para insertar los datos y un if si el método es post Se redirecciona a la página por defecto sino se queda en la misma ruta y traemos la función db que es la conexión a la base de datos. en las colecciones pedido y menú Se hiso un for para que recorra y solicite los atributos que están dentro de las listas artículos y platos. Al final se implementa los comandos de pymongo para finalmente insertar los datos a las respectivas colecciones usando un insertOne

# Activar/desactivar(delete)

Para activar y desactivar hicimos una función para consultar las colecciones para que cuando se desactive un documento siga en pantalla



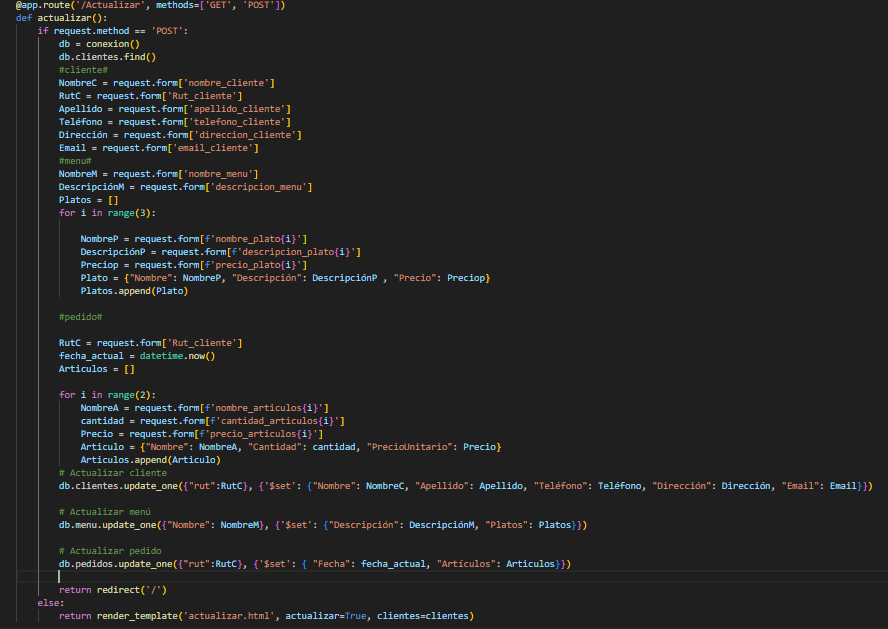
Luego se creó otra función para desactivar y que obtenga el id del documento para desactivar un documento en específico donde ocupamos en comando updateONE y se le agrego otro atributo llamado status en cero y para que reconozca el id del documento llamamos el objectid

Se crea otra función para activar el documento que también se obtiene el id del documento para activar un documento en especifico con un updateONE y el atributo status en 1

Las 3 funciones se renderizan en la ruta activar-desactivar para administrar en una pagina independiente

# Actualizar datos (update)

Para actualizar datos creamos una función llamada actualizar donde se asignaron variables que solicitan los atributos del formulario



Se implemento un if para que si el método era post se redireccionara a la pagina por defecto si no se mantendría en la ruta asignada y al final se ocupa un updateONE con los atributos y las variables anteriormente asignadas y se renderiza en actualizar.html