Universidad Nacional de Colombia - sede Bogotá
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas
Curso: Ingeniería de Software

Estudiantes:

- Andres Felipe Arias Gonzalez
- Juliana Alejandra Nieto Cárdenas
- Misael Jesús Florez Anave
- Nicolas Betancur Sanchez

Grupo 1: Formula 1 - (2025-1) Martes y Jueves.

DOCUMENTACIÓN Y ELECCIÓN DEL PATRÓN DE DISEÑO.

Imaginemos que cada aplicación es como un pequeño café de la Zona G. Está quién prepara el café (el barista), el que atiende a los clientes en la barra y quien lleva la contabilidad en la caja. Si cada uno hace bien su trabajo, el negocio fluye, el café sale a tiempo, la caja cuadra y los clientes toman un buen cafesito .

Eso llevado al código, es exactamente lo que hacemos usando los elementos del patrón Modelo-Vista-Controlador (MVC).

¿Por qué elegir MVC?

Cuando todo se mezcla en un único archivo --datos, pantallas y reglas--, cualquier cambio se siente como echar azúcar directamente en la cafetera: nada vuelve a ser igual y el sabor se arruina.

MVC separa esas tareas:

- Modelo: el contable del café. Guarda los números y vigila que cada pedido tenga sentido.
- Vista: la barra y la vitrina. Ahí es donde los clientes ven (y eligen) su bebida.
- Controlador: El típico barista todero multitarea. Escucha lo que pide cada persona, revisa con el contable lo que hay disponible y, por último, saca el pedido a la barra.

Con ese reparto es mucho más fácil arreglar el molino, pintar la pared o cambiar la receta sin detener el servicio.

¿Cómo lo vivimos en VeciRun?

Cuando montamos VeciRun queríamos justo eso: un “café” bien sabroso y listo para ser servido. Así lo dividimos:

Modelo (models.py + SQLAlchemy)

Aquí viven nuestros Users, Bicycles, Stations y Loans. Son algo más que tablas: se aseguran, de que una bicicleta no aparezca en dos estaciones a la vez (igual que el cajero evita vender dos veces el mismo pan).

Vista (pantallas hechas con Flet)

Desde el “Home” hasta el formulario “Crear usuario”, todo se pinta aquí. La vista sólo se encarga de verse bonita y recoger los clics.

Controlador / Servicios (VeciRunApp, UserService, etc.)

Son los baristas del sistema: reciben la orden, la validan y la pasan al modelo. Cuando todo está listo, envían el resultado de vuelta a la vista para que el usuario vea su “cafecito” servido.

Un ejemplo cotidiano: alguien llena el formulario para registrarse → la vista nos pasa los datos → el UserService comprueba que no exista ya ese correo → el modelo guarda la nueva persona → la vista muestra el mensaje. Ni una línea de más, ni un paso de menos.

¿Por qué fue la mejor elección?

Podemos rediseñar la interfaz sin tocar una sola regla del negocio; o ajustar las reglas sin romper la interfaz.

1. No matarse la cabeza

¿Queremos lanzar una app móvil o una API? Solo se Reemplaza la “barra” (la vista) y usamos el mismo contable y el mismo barista.

2. Las pruebas no asustan

Al estar todo separado, podemos probar los servicios y modelos en solitario. Los errores cantan más rápido y se corrigen antes.

3. Si la casa está ordenada, no se pierden las medias

Cualquiera que se una al equipo sabe dónde encontrar las cosas. Entonces uno, dentro de un año, vuelve al repositorio y entenderá todo sin mayor estrés. Por eso MVC nos permitió

servir cada taza de código caliente y sin posos, mantener el mostrador limpio y mostrar la mejor versión de nuestra idea.