



HERNANDEZ GARDUÑO
MARIA FERNANDA

ACADEMIA JAVA

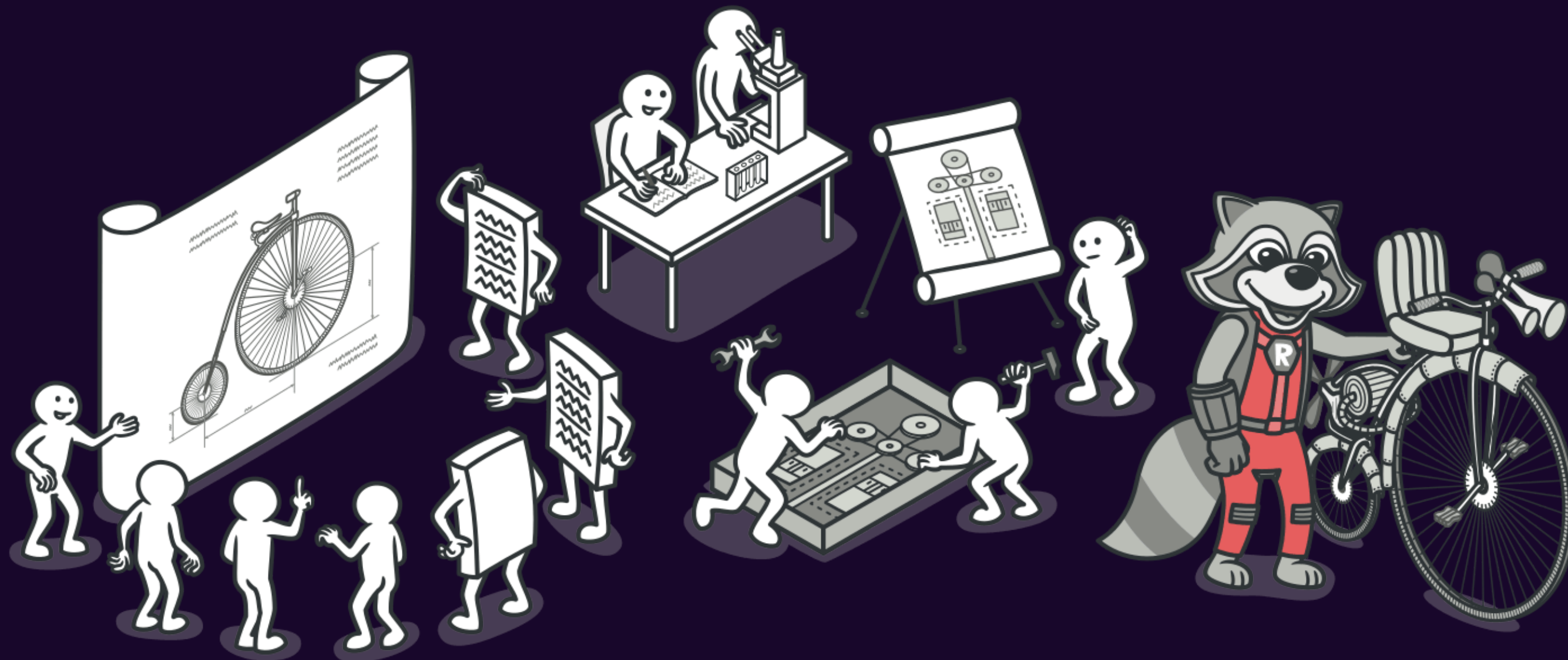
MODEL VIEW CONTROLLER

MVC



¿QUE ES MVC?

SE TRATA DE UN PATRON DE DISEÑO DE SOFTWARE, LO QUE QUIERE DECIR QUE ES UN SOLUCIÓN A PROBLEMAS REPETITIVOS O COMUNES, ALGO ASI COMO UNA PLANTILLA, Y SE PUEDE UTILIZAR EN MUCHAS CANTIDADES DE FRAMEWORKS, POR LO QUE SE CONSIDERA EL PATRON DE DISEÑO MAS UTILIZADO.



UNO DE LOS OBJETIVOS FUNDAMENTALES DEL MVC ES LA SEPARACIÓN DE RESPONSABILIDADES, LO CUAL SE LOGRA CON LA DIVISIÓN DE 3 GRANDES CAPAZ:

VIEW

Se encarga de mostrar los datos al usuario y de presentar la interfaz de usuario con la que el usuario interactúa. Su función principal es representar visualmente la información del modelo de una manera comprensible y atractiva. La vista obtiene los datos necesarios del modelo a través del controlador y los muestra de manera adecuada para que el usuario pueda interactuar con ellos.

CONTROLLER

Se encarga de manejar la interacción entre la vista (interfaz de usuario) y el modelo (datos y lógica de negocio). Su función principal es recibir las acciones del usuario desde la vista, procesarlas y actualizar el modelo en consecuencia. Esta separación permite que la vista y el modelo operen de manera independiente y promueve una arquitectura modular y mantenible.

MODEL

Es responsable de gestionar y procesar los datos, implementando la lógica de negocio y las operaciones de acceso y manipulación de datos. Actúa como el núcleo de la aplicación, asegurando una separación clara entre la presentación y el control, lo que facilita la modularidad y el mantenimiento del código.

Flujo de MVC

- 1.-El proceso comienza cuando el usuario interactúa con la vista, ya sea por su celular o cualquier dispositivo, y se manda la solicitud (consultar un saldo por ejemplo) al Controller
- 2.- El controlador es el intermediario entre la vista y el modelo. Recibe la acción, interpreta la solicitud y determina qué operación de modelo debe realizarse.
- 3.-El controlador interactúa con el modelo para realizar operaciones en los datos subyacentes. Esto podría implicar acceder a una base de datos, realizar cálculos o aplicar lógica de negocio específica.
- 4.- Una vez que el modelo completa la operación, informa al controlador sobre el resultado.
- 5.- Con la respuesta del modelo, el controlador decide cómo actualizar la vista para reflejar los cambios.
- 6.-La vista toma la información del controlador y actualiza la interfaz de usuario para mostrar los datos y cambios recientes. Puede ser tan simple como actualizar un elemento visual en la página web o mostrar una notificación al usuario.

