

Laboratorio 2 técnicas

Punto 1

1. Sin el uso de la interfaz, cada modo de transporte tendría que ser gestionado explícitamente por el sistema de control, sin contrato general. Esto significa que el sistema de control tendría que definir una función diferente para gestionar cada modo de transporte del trayecto. Gracias a la interfaz, solo hay que acceder a un único método general que funcionará con el tipo y con el sistema, y el método se ejecuta sin preocuparse de qué clase concreta se está implementando.
2. Debido a la interfaz utilizada, no sería necesario alterar el sistema de control. Solo sería necesario establecer una nueva clase. La clase debería implementar una interfaz idéntica y el sistema principal podría comenzar a utilizarla de inmediato sin necesidad de cambiar nada más.
3. La interfaz especifica lo que hay que hacer, es decir, los métodos obligatorios que debe utilizar cualquier sistema de transporte. Las clases concretas definen cómo se hace, proporcionando la lógica específica de cálculo y presentación de cada medio de transporte.

Punto 2

1. Para cambiar la aplicación y que funcione con una base de datos real en lugar de un ArrayList, se modificaría principalmente el Modelo, porque es el encargado de gestionar los datos y la lógica de negocio. La Vista y el Controlador no necesitarían cambios significativos.
2. Para usar una GUI creada en otro lenguaje más elaborado, se modificaría principalmente la Vista, ya que es la responsable de la interfaz de usuario. Se puede reutilizar el Modelo, porque la lógica de negocio y el manejo de datos sigue igual, y también se puede reutilizar el Controlador, adaptando solo cómo se reciben los eventos de la nueva GUI.
3. El Modelo no debe llamar directamente a la Vista para mostrar errores, porque en MVC el Modelo no tiene conocimiento de la presentación. La forma correcta es que el Modelo lance excepciones o devuelva códigos de error, y el Controlador se encargue de capturarlos y mostrarlos a través de la Vista.