

Refactoring Ejemplo 6

Por: Juan Sebastián Sánchez L.

Refactoring: Uso de ArrayList en lugar de un array primitivo

- Descripción: Cambiar el uso de un array primitivo *Día[]* por una colección *ArrayList List<Dia>* para manejar la lista de días de manera más flexible.
- Razón de aplicación: Utilizar ArrayList permite agregar y quitar días del calendario dinámicamente sin preocuparse por el tamaño fijo del array primitivo.
- Consecuencia: Facilita la manipulación y gestión de los días en el calendario.

Refactoring: Eliminación de cálculo en índice del día

- Descripción: Eliminar la resta de 1 al índice del día seleccionado.
- Razón de aplicación: La resta de 1 al índice del día no es necesaria si se utiliza un índice basado en 1, lo que simplifica el código.
- Consecuencia: Simplifica el código y lo hace más legible.

Refactoring: Uso de switch en lugar de múltiples if-else

- Descripción: Cambiar la estructura de múltiples if-else para la selección de opciones por una declaración switch.
- Razón de aplicación: La estructura switch es más adecuada para manejar múltiples opciones y hace que el código sea más legible.
- Consecuencia: Mejora la claridad del código.

Refactoring: Organizar las clases Dia y Evento en un paquete de modelos

- Descripción: Organizar las clases Dia y Evento en un paquete llamado model dentro del paquete *refactoring.problema6*.
- Razón de aplicación: Organizar las clases relacionadas en un paquete dedicado ya que mejora la estructura del proyecto.
- Consecuencia: Mejorar la organización y la estructura del código del proyecto.