

1. Gestión de Cuentas Bancarias

Preguntas:

- **¿Qué atributos debería tener la clase?**

Número de cuenta y Saldo

- **¿Cómo se asegura que el saldo no se vuelva negativo?**

El saldo de una cuenta no puede ser negativo para evitar errores lógicos, por lo tanto, validamos que el saldo de la cuenta sea mayor a cero.

2. Control de Temperatura

Preguntas:

- **¿Cómo se aplica la encapsulación en este problema?**

Se aplica al declarar el atributo `temperaturaCelsius` como privado. Esto asegura que la temperatura solo pueda ser accedida y modificada a través de los métodos `getTemperaturaCelsius` y `setTemperaturaCelsius`. De esta manera controlamos el acceso y la modificación de la clase.

- **¿Cómo se realiza la conversión de temperatura en los métodos?** La conversión de temperatura se realiza en los métodos `convertirAFahrenheit` y `convertirAKelvin`:

Fahrenheit: La fórmula para convertir de Celsius a Fahrenheit es $(\text{temperaturaCelsius} \times 9/5) + 32$.

Kelvin: La fórmula para convertir de Celsius a Kelvin es $\text{temperaturaCelsius} + 273.15$.

3. Registro de Productos en un Inventario

Preguntas:

- **¿Cómo se encapsula la información del producto?** Esto pasa cuando agregamos los atributos `codigo`, `nombre` y `precio` como privados. Esto asegura que estos atributos solo puedan ser accedidos y modificados a través de los métodos públicos `getPrecio`, `setPrecio`, `getCodigo` y `getNombre`.
- **¿Por qué es importante validar los valores ingresados?** Es importante validar los valores ingresados para asegurar que los datos del producto sean consistentes y válidos. En este caso, no permitir precios negativos evita errores lógicos y asegura que los productos tengan precios realistas.

4. Temporizador con Alarma

Preguntas:

- **¿Cómo interactúan las clases Temporizador y Alarma?** Temporizador tiene un atributo alarma que puede ser asociado mediante el método asociarAlarma. Durante la ejecución del método iniciar, Temporizador verifica si la alarma debe activarse llamando al método debeActivarse de la instancia Alarma.
- **¿Cómo se asegura que los atributos sean accesibles solo mediante métodos específicos?** La encapsulación sea segura al declarar los atributos tiempoActual y alarma en Temporizador, y tiempoObjetivo en Alarma como privados. Esto garantiza que estos atributos solo puedan ser accedidos y modificados a través de métodos específicos, controlando así el acceso y modificación del estado interno de los objetos

5. Control de notas de un curso

Preguntas:

- **¿Cómo se encapsulan los datos de los estudiantes dentro del curso?**
Los datos de los estudiantes (nombre, carnet y nota) están encapsulados en la clase Estudiante. La clase Curso maneja una lista de objetos Estudia Estudiante.
- **¿Cómo se garantiza que las notas sean válidas?**
Las notas se validan dentro del setter de notaFinal,

