

Estratexia para Resolver Exercicios en Java

1 ENTENDER (5-10 minutos)

- ✓ Le o enunciado **lentamente** e subliña os puntos clave.
 - ✓ Identifica claramente:
 - **Clases principais** e as súas relacións (Vehiculo, Cliente, Almacen...).
 - **Métodos obrigatorios** (toString(), equals(), compareTo(), alugar(), devolver()...).
 - **Interfaces e enumeracións necesarias** (Alugable, TipoMotor...).
 - ✓ **Escribe un esquema rápido nun papel** coas clases e funcionalidades necesarias.
-

2 PLANIFICAR (10 minutos)

- ✓ Define a **estrutura do proxecto** antes de programar.
- ✓ Crea un **bosquexo rápido** co IDE ou en papel:

css

CopiarEditar

- Clases abstractas/interfaces: Vehiculo, Alugable
 - Subclases concretas: Coche, Moto, Furgoneta
 - Xestión de datos: Almacen<T>
 - Métodos principais: add(), del(), search(), update()
- ✓ **Prioriza:** Primeiro a estrutura básica, logo os detalles.
-

3 IMPLEMENTAR A ESTRUCTURA BÁSICA (20-30 minutos)

- ✓ Define **todas as clases e interfaces** necesarias.
 - ✓ **Só escribe os atributos e constructores** sen implementar os métodos.
 - ✓ Define **atributos básicos** con protected ou private.
 - ✓ Engade implements en interfaces (Vehiculo implements Alugable).
-

4 IMPLEMENTAR OBRIGATORIOS (1-1.5 horas)

- ✓ Implementa os métodos fundamentais:
 - toString() para a representación correcta.
 - equals() para establecer igualdade (Matrícula en Vehiculo, DNI en Cliente).
 - compareTo() para definir a orde natural.
 - ✓ Implementa Alugable en Vehiculo e sobrescribe alugar(), devolver() e prezo().
-

5) IMPLEMENTAR A LÓXICA DA APP (1 hora)

- ✓ Implementar Almacen<T> cos métodos correctamente definidos.
 - ✓ Probar os métodos cunha clase main() sinxela.
 - ✓ Implementar correctamente os métodos:
 - add(): Engadir elementos ao Almacen.
 - del(): Eliminar elementos.
 - search(): Buscar con Filtro<T>.
 - update(): Modificar datos correctamente.
-

6) FUNCIONALIDADES E MENU (30-45 minutos)

- ✓ Crear un **menú sinxelo** en AluguerEnterprise.
 - ✓ Engadir opcións para **xestionar vehículos e clientes**.
 - ✓ Permitir alugueiros e devolucións correctamente.
 - ✓ Asegurar que **manexa erros** (try-catch para AlmacenException).
-

7) REVISAR DETALLES (últimos 30 min)

- ✓ Comprobar que se cumpren **todos os requisitos** do enunciado.
 - ✓ Verificar que toString(), equals(), compareTo() e Alugable están ben implementados.
 - ✓ Probar casos límite:
 - Intentar engadir un cliente duplicado.
 - Intentar alugar un vehículo xa alugado.
 - Intentar devolver un vehículo que non estaba alugado.
-

8) REVISIÓN FINAL (15-20 minutos)

- ✓ **Código limpo**: nomes claros e organizados.
 - ✓ Engadir **comentarios explicativos** en métodos clave.
 - ✓ Revisar e formatear o código (Ctrl + Shift + F en NetBeans).
-

◆ CONSELLOS FINAIS PARA EVITAR BLOQUEOS

- ✓ Divide o problema en partes pequenas e resolve unha cada vez.
- ✓ Proba o código constantemente (non deixes as probas para o final).
- ✓ Se algo non funciona, depura con System.out.println() ou usa o depurador de NetBeans.
- ✓ Mantén a calma e segue este esquema paso a paso.