

## Objetivo do Case — *Mini Service Desk (Console App)*

Criar uma aplicação de console em C# que permita **abrir, listar, atualizar status e encerrar chamados de TI**, com **validações, tratamento de exceções e uso de interfaces/abstrações**.

O foco é aplicar **POO na prática**: modelagem de classes, encapsulamento, herança/polimorfismo, interfaces e boas práticas de erros.

---

## Requisitos Funcionais (MVP)

1. **Criar Chamado**
    - Campos mínimos: **Id** (auto), **Titulo**, **Descricao**, **Prioridade** (Baixa/Média/Alta), **Solicitante**, **Status** (Aberto, Em Andamento, Resolvido, Fechado), **DataAbertura**.
    - Regra: **Titulo** e **Solicitante** são obrigatórios; **Prioridade** default = **Média**.
  2. **Listar Chamados**
    - Listar todos e permitir filtros por **Status** e/ou **Prioridade** (pode usar LINQ).
  3. **Atualizar Status**
    - Transições válidas:
      - Aberto → Em Andamento
      - Em Andamento → Resolvido
      - Resolvido → Fechado
    - Invalidar transições fora dessa ordem com exceção amigável.
  4. **Detalhar Chamado por Id**
    - Mostrar todos os campos e histórico de mudanças de status (mínimo: status + data/hora).
  5. **Encerrar/Excluir Chamado** (opcional no MVP)
    - Só permitir **excluir** se o status for **Aberto**; caso contrário, lançar exceção de domínio com mensagem clara.
- 

## Requisitos de Engenharia (POO)

- **Modelagem**
  - **Chamado** (entidade) com **encapsulamento** (ex.: **set** privados quando fizer sentido).
  - **Prioridade** e **Status** como **enum**.
- **Herança/Polimorfismo**
  - Crie uma hierarquia simples para diferentes tipos de chamados (ex.: **ChamadoInfra** e **ChamadoAplicacao**) que **especializam** algum comportamento (p. ex., cálculo de SLA ou texto de resumo).
- **Interface / Abstração**

- `IRepositorioChamados` (métodos: `Criar`, `ObterPorId`, `Listar`, `Atualizar`, `Excluir`).
  - Implementação inicial **em memória**: `RepositorioChamadosMemoria`.
  - **Exceções**
    - Use `try/catch` onde faz sentido e crie ao menos **uma exceção de domínio** (ex.: `TransicaoStatusInvalidaException`).
    - Mensagens claras e não técnicas para o usuário final (no console).  
(Esses pontos reforçam os pilares da POO e boas práticas de tratamento de erros vistos nos slides.) [[Bootcamp\\_S...ana1\\_dia02](#) | [PowerPoint](#)]
- 

## Critérios de Aceite

- Ao tentar uma transição de status inválida, o sistema **não quebra** e exibe mensagem amigável.
- `Listar` com filtros funciona (ex.: “listar por **Alta** e **Em Andamento**”).
- `Criar` não aceita chamado sem `Titulo` ou `Solicitante`.