

Processo de Seleção - 3^a Etapa

Construir um mini-sistema utilizável (com interface) para controlar entrada/saída de veículos em um estacionamento, com persistência em banco, relatórios e regras de negócio, que será comercializado no Brasil e Argentina, seguindo as seguintes premissas:

- O projeto deve ser feito em ASP.NET Core Web API;
- Toda persistência deve ser feita em SQLite;
- Front-end: Angular 17+

Entrega Base

- 1) Campos mínimos: Placa (Obrigatório), Modelo, Cor e Tipo
 - Placa é unica (não pode ter 2 veículos cadastrados com a mesma placa)
 - Validações mínimas de formulário
- 2) Movimentação de Entrada e Saída
 - Um veículo não pode entrar se já estiver “no pátio” (sessão aberta).
 - Um veículo não pode sair se não tiver sessão aberta.
 - A saída deve registrar: data/hora de saída e valor cobrado.
- 3) Precificação
 - Primeira hora: R\$ X
 - Demais horas: R\$ Y por hora (arredondamento a seu critério, explique no README)
- 4) Interface de Operação (Front)
 - Tela “Pátio agora”: lista dos veículos com sessão aberta com busca por placa
 - Ação “Registrar saída” (mostrando valor antes de confirmar)
 - Tela “Cadastro de veículo”: criar/editar/listar
- 5) Relatórios (SQL / Consultas)
 - Query de Faturamento por dia (últimos 7 ou 30 dias)
 - Query Top 10 veículos por tempo estacionado em um período
 - Query Ocupação por hora (ex.: quantos veículos estavam no pátio por hora do dia, em um período)

Achou fácil demais?

Caso tenha achado a primeira parte fácil demais, você pode avançar no seu desenvolvimento escolhendo até 3 aprofundamentos abaixo:

- Autenticação e Autorização
 - Usuários com perfis: Admin e Operador
 - Admin gerencia usuários
 - Operador faz entradas/saídas

- Concorrência e consistência
 - Tratar condição de corrida (duplo clique, duas requisições simultâneas)
 - Garantir que não abre/fecha sessão duplicada
- Paginação, filtros e ordenação
 - Endpoints com filtros por período/placa/status (aberta/fechada)
 - UI com paginação real
- Testes automatizados
 - Unit tests (regras de domínio)
 - Integration tests (principais endpoints)
- Observabilidade
 - Logging estruturado
 - Healthcheck
 - Trace/correlation id (bônus)
- UX caprichada
 - Estados de loading/empty
 - Validação amigável
 - Layout minimamente profissional

Use todas as suas habilidades para projetar e desenvolver o sistema da melhor forma possível. Tudo será avaliado!

O escopo é aberto propositalmente para que você invista todas as ferramentas, tecnologias e criatividade que achar necessário.

Entrega

As soluções devem ser compartilhadas por algum serviço de armazenamento (FTP, Dropbox, GIT, Google Drive, etc.) e os links para download devem ser enviados para o e-mail careers@envvio.com até as **23:59 horas do dia 22/02/2026**.

No e-mail deve conter o tempo levado para desenvolver o sistema e as dificuldades encontradas.

Atenção: Soluções e projetos que nem se quer estejam compilando serão reprovados de imediato! Faça a análise tudo antes de entregar!

Boa sorte!