

# Relatório Individual - Vitor

---

## 1. Qual foi sua principal contribuição no trabalho?

Minha principal contribuição foi desenvolver toda a estrutura backend da aplicação, focando na arquitetura MVC. Fui responsável por criar a base sólida do projeto, implementando a separação de responsabilidades entre as camadas de dados, lógica de negócios e segurança.

## 2. Quais partes do projeto você desenvolveu?

---

Desenvolvi as seguintes partes:

- **Configuração inicial do projeto:** Criei o package.json, configurei as dependências e estruturei as pastas do projeto
- **Camada de dados:** Implementei os arquivos de dados em memória (usuarios.js, produtos.js, carrinhos.js) e todos os repositories
- **Camada de serviços:** Desenvolvi toda a lógica de negócios, incluindo autenticação, gerenciamento de produtos e operações do carrinho
- **Middlewares personalizados:** Criei os middlewares de autenticação e permissão para proteger as rotas privadas
- **Arquivo principal (index.js):** Configurei o Express, sessões e integração das rotas

## 3. Qual foi o trecho mais complicado do código? Explique o motivo.

---

O trecho mais complicado foi a implementação do `carrinhoRepository.js`, especialmente a função de adicionar itens ao carrinho. A dificuldade estava em gerenciar corretamente os casos onde:

1. O produto já existe no carrinho (precisa apenas incrementar a quantidade)
2. O produto é novo (precisa criar um novo item)
3. Manter o cálculo do total sempre atualizado após qualquer operação

Tive que pensar bastante na lógica para evitar duplicação de produtos e garantir que o total fosse recalculado automaticamente após cada operação (adicionar, remover ou atualizar quantidade).

## 4. Qual foi sua maior dificuldade técnica e como resolveu?

---

Minha maior dificuldade foi implementar o sistema de sessões do Express de forma segura e funcional. Inicialmente, tive problemas com a persistência da sessão entre requisições e com a configuração correta dos cookies.

Para resolver, pesquisei bastante sobre o express-session e descobri que precisava configurar corretamente as opções `resave` e `saveUninitialized`. Também aprendi sobre a importância de definir um `secret` forte e configurar o `maxAge` do cookie para controlar o tempo de expiração da sessão.

## 5. Você utilizou IA (como ChatGPT, Copilot, etc.) em alguma parte? Se sim, explique exatamente como.

---

Sim, utilizei IA em algumas partes do desenvolvimento:

- **Estrutura inicial:** Pedi ao ChatGPT sugestões de como organizar a arquitetura MVC de forma profissional e escalável
- **Lógica do carrinho:** Quando fiquei travado na função de adicionar produtos, usei o ChatGPT para entender melhor como trabalhar com arrays de objetos em JavaScript e encontrar itens existentes
- **Middlewares:** Consultei o Copilot para ver exemplos de middlewares de autenticação no Express, mas adaptei completamente para as necessidades do projeto
- **Debugging:** Quando tive erros com as sessões, usei o ChatGPT para entender melhor as configurações do express-session

Importante destacar que a IA foi usada como ferramenta de consulta e aprendizado, mas todo o código foi escrito e adaptado por mim para atender aos requisitos específicos do projeto.

## 6. O que você aprendeu de novo com este trabalho?

---

Aprendi muito sobre:

- **Arquitetura MVC:** Entendi na prática a importância de separar responsabilidades e como isso torna o código mais organizado e manutenível
- **Padrão Repository:** Aprendi a criar uma camada de abstração para acesso aos dados, facilitando futuras mudanças (como migrar para um banco de dados real)
- **Express-session:** Compreendi como funcionam sessões no Node.js e como gerenciar autenticação de usuários
- **Middlewares personalizados:** Aprendi a criar middlewares próprios para adicionar funcionalidades customizadas à aplicação
- **Gerenciamento de estado em memória:** Entendi os desafios de trabalhar com dados em memória e como estruturar objetos complexos

## 7. Se tivesse mais tempo, o que melhoraria no projeto?

---

Se tivesse mais tempo, eu melhoraria:

- **Banco de dados real:** Implementaria MongoDB ou PostgreSQL para persistir os dados de forma permanente
- **Validações mais robustas:** Adicionaria validações mais completas nos services, como verificar estoque de produtos
- **Testes automatizados:** Criaria testes unitários para os repositories e services usando Jest
- **Sistema de recuperação de senha:** Implementaria funcionalidade de “esqueci minha senha” com envio de e-mail
- **Logs da aplicação:** Adicionaria um sistema de logs para rastrear erros e ações dos usuários
- **Criptografia de senhas:** Usaria bcrypt para criptografar as senhas dos usuários ao invés de armazená-las em texto puro
- **API REST:** Separaria o backend em uma API REST completa, permitindo criar diferentes front-ends (mobile, web, etc.)