

# Relatório do Projeto de Programação WEB

*Guilherme Azevedo Rodrigues - 13096*

## 1. Introdução Geral

O projeto **Outcast Skate Shop** foi desenvolvido com o objetivo de simular uma aplicação web comercial real. Desde a sua concepção, a prioridade foi criar uma aplicação **robusta, modular e escalável**, onde cada decisão técnica estivesse alinhada com boas práticas de engenharia de software e com problemas reais enfrentados por sistemas web modernos.

A filosofia adotada parte do princípio de que uma aplicação sólida não é definida apenas pela sua interface visual, mas principalmente pela **qualidade da sua arquitetura interna**, pela integridade dos dados e pela capacidade de evolução futura. Dessa forma, o backend foi tratado como o elemento central do sistema, responsável por garantir consistência, segurança e desempenho, enquanto o frontend atua como uma camada de apresentação desacoplada e flexível.

## 2. Modelação da Base de Dados e Normalização

A base de dados foi projetada como o pilar estrutural do sistema. Embora o enunciado inicial sugerisse uma implementação simples, foi abordado uma base de dados comum e funcional, prevenindo anomalias de inserção, atualização e remoção, e garantindo maior coerência dos dados ao longo do tempo. As decisões que definiram o rumo do projeto foi a separação de diferentes tabelas para a organização do projeto, entre elas está a entidade **Category** que foi criada para classificar os produtos de forma estruturada. Foram definidas oito categorias distintas: Skates, Shapes, Rodas, Trucks, Rolamentos, Lixas, Acessórios e Roupas. Como o enunciado não especificava uma hierarquia formal entre categorias, decidiu-se atribuir uma ordem lógica aos identificadores, refletindo o processo real de montagem de um skate, desde o produto principal até aos elementos complementares.

## 3. Estrutura do Backend e Organização do Código

O backend foi desenvolvido utilizando o framework Express, adotando uma estrutura modular desde as fases iniciais do desenvolvimento. Em vez de concentrar toda a lógica num único ficheiro, o código foi dividido em:

- Ficheiro principal (**app.js**) responsável pela configuração do servidor e dos middlewares.
- Pasta **/routes**, onde cada ficheiro gere um conjunto específico de rotas.
- E entre outras pastas como Controllers e Models.

## 4. Segurança, Autenticação e Autorização

A segurança foi tratada como um elemento estrutural do sistema, e não como um simples complemento funcional.

### 4.1 Controlo de Acesso Baseado em Funções

Foi implementado um modelo de **Role-Based Access Control**, permitindo diferenciar claramente utilizadores comuns de administradores. A tabela **User** inclui um campo **role**, com restrições que limitam os valores possíveis. Esta decisão impede estados inválidos e reforça a integridade dos dados.

Um middleware de autenticação foi desenvolvido para verificar:

- A existência de uma sessão ativa.
- A compatibilidade do papel do utilizador com o recurso solicitado.

Esta abordagem impede acessos indevidos ao painel administrativo, mesmo que o utilizador tente aceder diretamente ao URL.

### 4.2 Gestão de Sessões

A persistência de sessões foi configurada com um tempo de vida de 24 horas. Esta decisão equilibra segurança e experiência do utilizador, permitindo sessões contínuas durante um período de trabalho sem comprometer o controlo de acesso.

## 5. Gestão de Contactos e Sistema de Mensagens

Em vez de um formulário de contacto estático, foi desenvolvido um sistema de gestão de mensagens inspirado em conceitos básicos de CRM. As mensagens são armazenadas numa tabela dedicada (**ContactMessage**), permitindo consulta, gestão e eliminação pelo administrador. A inclusão de uma chave estrangeira opcional para **Product** permite contextualizar cada mensagem, associando-a diretamente a um produto específico. Esta decisão melhora significativamente a eficiência da gestão de pedidos e dúvidas dos clientes.

## 6. Considerações Finais

O projeto **Outcast Skate Shop** destaca-se pela sua abordagem estruturada, pela preocupação com integridade de dados e pela aplicação de conceitos que ultrapassam os requisitos mínimos do enunciado. As decisões tomadas ao nível da arquitetura, segurança, modelação da base de dados e organização do código transformam este trabalho numa simulação realista de uma aplicação web comercial. Em vez de privilegiar apenas o aspeto visual, o foco foi colocado na infraestrutura, na escalabilidade e na manutenção a longo prazo, refletindo uma abordagem madura e alinhada com práticas profissionais de desenvolvimento de software.