

Ricardo Parmezan de Almeida Castro

DESENVOLVEDOR FULL STACK

☎ [telefone: +55 \(71\) 99290 - 7777](tel:+55(71)99290-7777)



<https://github.com/Rickccastro>



Salvador, Bahia

✉ ricardo.castro.linkedin@gmail.com



www.linkedin.com/in/rickccastro

HABILIDADES

- Backend Development: C# .NET, Java Springboot, JavaScript Express
- Frontend Development: TypeScript Angular, JavaScript React, Bootstrap, HTML, CSS
- Nuvem e ferramentas: **AWS**, Azure DevOps Server, Postman, Imsonia, Microsoft SQL Profiler
- Banco de Dados: Microsoft SQL Server, MySQL, PostgreSQL
- Soft skills: Trabalho em equipe, Comunicativo, Gestão de tempo, Ética e responsabilidade.

EXPERIÊNCIA PROFISSIONAL

TecnoTrends - Desenvolvedor

- Atuo no desenvolvimento de aplicações para o setor educacional, utilizando **C#** com **.NET**. Onde minhas principais funções são colaborar na **implementação de novas funcionalidades**, na **resolução de problemas técnicos**, na manutenção dos sistemas e na criação da **modelagem de dados**. Além disso, ofereço suporte aos analistas de QA em consultas ao banco de dados, contribuindo para a eficiência na validação e garantia da qualidade dos sistemas. Participo também de reuniões para melhorias na empresa, onde nossa equipe está sendo pioneira na adoção de práticas como **commits semânticos** e **TDD (Test-Driven Development)**.

Projetos

• CashFlow

Esta API **RESTful** foi desenvolvida em **C#** com **.NET 8** para rastrear de forma segura as despesas pessoais dos usuários. Seguindo os princípios do **Domain-Driven Design (DDD)**, ela utiliza o Entity Framework com **MySQL** para persistência de dados. A aplicação adota Design Patterns como o Builder e bibliotecas como AutoMapper e FluentValidation e o **Swagger**. Para garantir a qualidade do código, são utilizados **testes unitários** com FluentAssertions e Moq.

Repositorio: <https://github.com/Rickccastro/CashFlow>

• CleanArchCatalogApi

Esta **API** segue os princípios da **Clean Architecture**, com camadas bem definidas que separam responsabilidades e facilitam a manutenção. Os **controllers** apenas recebem requisições e delegam aos **services**, que concentram a lógica de negócio e utilizam **repositórios** para acessar os dados. A **inversão de dependência** é aplicada por meio da **injeção de interfaces**, aumentando a flexibilidade e testabilidade. O padrão **Repository** abstrai o acesso ao banco, enquanto o **AutoMapper** facilita a conversão entre entidades e **DTOs**, mantendo o domínio limpo e desacoplado.

Repositorio: <https://github.com/Rickccastro/CleanArchCatalogApi.git>

• CoffeeShop

Coffee Shop é um e-commerce em desenvolvimento com **Angular** e **TypeScript**, que utiliza conceitos como componentização, **injeção de dependência**, services, resolvers e a função **inject** para um gerenciamento moderno de dependências. A aplicação emprega **Reactive Forms** e **signals** para controle de formulários e estado reativo. Durante o desenvolvimento, o **JSON Server** está simulando momentaneamente o back-end, facilitando testes completos de forma prática.

Repositorio: <https://github.com/Rickccastro/CoffeShop>

Vídeo Demo: <https://www.youtube.com/watch?v=8lnJV7l2zxU>

CERTIFICAÇÕES E EDUCAÇÃO

- Engenharia de Software na Universidade Católica do Salvador (UCSAL) — 2022–2026.1 (previsão)
- **Certificação AWS** Academy Cloud Foundations
- Monitor da matéria de Programação Orientada a Objetos I (POO) na UCSAL — 2024.1
- Cursos em andamento: C# .NET, TypeScript Angular, JavaScript React

IDIOMAS

- Português (Fluente) e Inglês (Avançado)
- Curso de Inglês da Oxford — 2014–2020