

# Proposta de Trabalho

MAC0499 - Trabalho de formatura supervisionado

Abril / 2024

**Aluna:** Briza Mel Dias de Sousa

**Responsável:** Prof<sup>a</sup>. Dra. Nina S. T. Hirata

**Orientador:** Prof. Dr. Alfredo Goldman

**Co-Orientador:** M.Sc. Renato Cordeiro Ferreira

## Introdução

Os paradigmas de programação reúnem uma série de regras independentes das linguagens que determinam uma forma de programar. Atualmente, três paradigmas ditam os padrões que delineiam a produção de um software: a programação estruturada, a programação orientada a objetos, e a programação funcional. Escolher um paradigma significa abraçar seus pontos fortes e fracos integralmente — a escolha considera os requisitos do software a ser desenvolvido, de forma que o resultado reflita mais as qualidades que as deficiências.

A programação funcional é um paradigma derivado dos trabalhos do matemático Alonzo Church sobre o cálculo lambda — um sistema formal construído a partir de funções recursivas e computáveis — e pelo posterior esforço do matemático Alan Turing em provar que o cálculo lambda de Church é computacionalmente universal. Esse resultado corroborou a ideia de que uma linguagem baseada no cálculo lambda poderia restringir *como* os programas eram escritos sem restringir *o que* eles poderiam expressar. LISP, a primeira linguagem funcional de alto-nível, foi criada por volta de 1950.

As consequências de adotar a disciplina funcional tem sido percebida pela indústria da tecnologia e é vista como uma solução elegante para problemas tradicionalmente encontrados em outros paradigmas. O modo como a programação funcional repensa o conceito de valores indefinidos, por exemplo, sem recorrer ao uso de referências nulas ("*The Billion-Dollar Mistake*"), é apenas uma das alternativas poderosas que o paradigma oferece à indústria. Algumas histórias de sucesso do uso do paradigma funcional na indústria incluem o Nubank, startup pioneira no segmento de tecnologia financeira no Brasil, e a Netflix, empresa responsável por revolucionar a indústria de entretenimento.

Dado o crescimento de popularidade e relevância, é fundamental dar visibilidade ao paradigma funcional para que as pessoas desenvolvedoras de software se sintam equipadas para encarar os desafios da indústria tecnológica. Para tanto, este projeto visa traçar uma comparação entre a programação funcional e a programação orientada a objetos. Ao evidenciar as assimetrias entre os dois paradigmas, os profissionais terão capacidade de eleger a abordagem com mais potencial de simplificação do desafio a ser enfrentado.

## **Objetivos**

O trabalho de conclusão de curso terá como objetivo principal a aplicação da programação funcional em um sistema que soluciona um problema do mundo real. O sistema servirá como objeto de estudo dos principais conceitos abraçados pelo paradigma, fornecendo exemplos práticos dos benefícios garantidos pela disciplina funcional. Além disso, o mesmo sistema deverá ser reescrito em outro paradigma, permitindo a comparação entre a programação funcional e outras formas de escrever programas. Nesse sentido, este projeto busca avaliar as consequências de restringir o desenvolvimento ao paradigma funcional, comparando-o com uma segunda versão escrita em outro paradigma. O objetivo é explicitar como o paradigma funcional repensa problemas clássicos de engenharia de software, analisar comparativamente o desempenho da solução proposta, e evidenciar os cenários nos quais a abordagem funcional pode ser desejável.

## **Metodologia**

O objeto de estudo será um sistema que simula uma carteira digital. O sistema deve receber transações financeiras, criar um registro contábil, e performar a movimentação de dinheiro entre as carteiras mencionadas na transação, se houver fundos suficientes e os demais requisitos forem verificados. Além disso, se for conveniente para os objetivos do projeto, é possível adicionar novas responsabilidades à carteira digital, como o processamento de transações sob demanda.

Alguns aspectos serão observados durante o desenvolvimento do sistema, como a facilidade em lidar com o tratamento de erros, tolerância a falhas, testabilidade, e a fluidez em construir a lógica de negócio da aplicação. Esse processo será realizado duas vezes: uma com base na programação funcional, e uma com base na programação orientada a objetos. Os dois paradigmas usarão técnicas distintas para implementar os requisitos funcionais da carteira digital, produzindo os exemplos necessários para traçar uma análise comparativa que evidencie os conceitos da programação funcional.

## **Requisitos Não-Funcionais**

### **Confiabilidade**

O sistema deve ser tolerante a falhas, garantindo que uma transação só será processada uma vez. Caso a transação seja válida, carteiras envolvidas na transferência só devem ser atualizadas uma vez.

### **Legibilidade**

O sistema deve ser legível, de forma que seja fácil de observar como o paradigma está sendo usado para escrever o fluxo que processa as transações e as transferências.

### **Extensibilidade**

