

Relatório Técnico – Unidade 1

Projeto: Simulador de Independência Financeira

Autor: Bruno César de Oliveira Rodrigues

Matrícula: 20250052728

Disciplina: Introdução a Técnicas de Programação (2025.2)

Data: 25 de Setembro de 2025

1. Introdução

1.1. Objetivo do Projeto

Este projeto, chamado de "Simulador de Independência Financeira", é uma aplicação desenvolvida em linguagem C e tem como objetivo principal dar ao usuário uma ferramenta capaz de projetar resultados futuros a partir de investimentos em ativos financeiros como Ações e Fundos Imobiliários. O programa calcula o tempo necessário para alcançar uma meta de renda passiva mensal, estabelecida pelo usuário, através de aportes recorrentes e do reinvestimento dos proventos.

1.2. Problema que o Projeto Resolve

Devido à falta de investimentos em educação financeira, aquele que resolve tentar criar um pé de meia visando o futuro com ativos mais voláteis, por vezes, termina desistindo, seja pelo medo, pela falta de conhecimento ou mesmo por não enxergar o potencial efeito “bola de neve” neste tipo de ativo. Este projeto visa solucionar esse problema ao oferecer uma calculadora de fácil utilização que responde a uma questão central: "Em quanto tempo meus investimentos podem gerar a renda que desejo?".

Apesar de simples, o fato de o usuário conseguir enxergar uma prospecção futura através dessa calculadora, já serve como um motivador para manter a consistência nos aportes e para manter-se atualizado sobre o assunto.

1.3. Fonte de Dados para o Usuário

Para utilizar o simulador, o usuário precisa de informações públicas e de fácil acesso sobre os ativos financeiros. Os dados necessários são o **preço atual** da cota/ação e o seu **Dividend Yield (DY) anual**. Essas informações podem ser obtidas gratuitamente em diversos portais de finanças consolidados no mercado, como o [Status Invest](#) e o [Fundamentus](#), garantindo que as simulações sejam baseadas em dados realistas e atualizados.

1.4. Justificativa da Escolha do Projeto

A escolha deste projeto foi baseada pelo interesse pessoal e certo conhecimento na área, bem como sua aplicabilidade prática e alinhamento com os objetivos da disciplina. O tema de finanças pessoais é motivador e a ferramenta possui uma utilidade real.

2. Análise Técnica

2.1. Metodologia: Ferramentas Utilizadas

O desenvolvimento do projeto foi realizado utilizando as seguintes ferramentas:

- **Linguagem:** C (padrão C99).
- **Compilador:** GCC (MinGW-w64 via MSYS2), versão 15.2.0, em ambiente Windows.
- **Editor de Código:** Visual Studio Code (com a extensão C/C++ da Microsoft).
- **Sistema de Versionamento:** Git, com o código-fonte versionado em um repositório público no GitHub.

2.2. Aplicação dos Conceitos da Unidade 1

- **Variáveis e Tipos:** Foram utilizados tipos de dados adequados para cada informação: `int` para contadores e escolhas de menu, `char[][]` para o vetor de strings `tickers`, e `double` para valores financeiros, a fim de garantir a precisão nos cálculos. A variável “quantidadeUnidades” foi declarada como `long long int` para evitar erros do programa em caso de suportar números muito grandes (como em caso de criptomoedas com valores muito baixos) em simulações de longo prazo.
- **Estruturas Condicionais (if/else, switch):** O comando `switch` na função `main` é o controlador central do fluxo do programa. O `if/else` foi aplicado para validações cruciais, como verificar se a carteira está cheia, se a escolha do usuário é válida, e para processar o “aporteInicial” apenas se o valor for maior que zero.
- **Estruturas de Repetição (do-while, while, for):** O `do-while` na função `main` é o cérebro do programa, é através dele que as demais funções são chamadas e é ele que garante a execução contínua do menu. O `for` é usado em “listarAtivos” para percorrer os vetores e exibir os dados. O `while` é o motor da função “simularInvestimento”, executando a lógica mês a mês enquanto a meta de renda não for atingida.
- **Vetores:** Foram criados três vetores (`tickers`, `precos`, `dysAnual`) que armazenam os dados digitados pelo usuário e é a partir deles que o cálculo é feito.
- **Funções:** O projeto foi dividido em 6 funções com responsabilidades únicas. Essa decomposição foi essencial para a clareza e manutenibilidade. Se for necessário

alterar o texto do menu, por exemplo, sabe-se exatamente qual função modificar (“exibirMenu”), sem impactar a lógica de cálculo (“simularInvestimento”).

2.3. Estruturas de Dados

As estruturas de dados do projeto são os três vetores e a variável `totalAtivos`. `tickers[5][7]` foi dimensionado para conter 5 strings, cada uma de até 6 caracteres mais um espaço de segurança. Os vetores `precos[5]` e `dysAnual[5]` armazenam os dados financeiros correspondentes. A variável `totalAtivos` indica o número de ativos cadastrados e a próxima posição livre nos vetores.

3. Implementação e Reflexão

3.1. Dificuldades e Soluções Implementadas

O maior desafio foi a modelagem matemática realista dentro do laço `while`. Quando compramos uma Ação, obviamente, ela é vendida em valores inteiros. Quando implementado a primeira vez, o programa considerava a compra de frações de uma Ação e desconsiderava o “troco” de cada compra.

A solução implementada foi refinar a lógica interna do laço:

- **Compra de Unidades Inteiras:** O cálculo de novas unidades foi modificado para usar uma conversão explícita para `long long int`, o que trunca a parte decimal do resultado.
- **Gerenciamento de Sobra de Caixa:** Foi criada a variável `caixaSobra`. Dentro do laço, a diferença entre o valor disponível e o custo real da compra é calculada e armazenada. Este valor é então somado ao poder de compra do mês seguinte, garantindo que nenhum capital seja perdido na simulação.

3.2. Conclusão e Aprendizados

O desenvolvimento deste projeto permitiu a aplicação prática e integrada de todos os conceitos fundamentais da Unidade 1. O principal aprendizado foi a importância da decomposição de um problema em funções menores e a necessidade de modelar cuidadosamente a lógica de negócio antes de traduzi-la para o código.

3.3. Melhorias para as próximas unidades

A intenção é evoluir o programa para que fique mais robusto e completo com funcionalidades como:

- **Expansão da Simulação:** A simulação poderia ser expandida para analisar o desempenho da carteira como um todo, permitindo ao usuário definir estratégias de alocação de aportes entre os diferentes ativos cadastrados. Atualmente, analisamos um ativo por vez.
- **Gerenciamento completo da carteira:** Seria possível acrescentar funcionalidades para que o usuário possa gerenciar seu portfólio de forma mais dinâmica, como editar as informações de um ativo já existente ou até mesmo removê-lo da carteira.
- **Salvar e carregar dados:** Implementação de um sistema para salvar e carregar a carteira de ativos sempre que quiser.
- **Número de ativos da carteira flexível:** Futuramente, o limite fixo de ativos cadastrados poderia ser removido, tornando a ferramenta mais flexível e permitindo que o usuário construa um portfólio de tamanho ilimitado.