UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

INTRODUÇÃO A TÉCNICAS DE PROGRAMAÇÃO

MARIA LUIZA DE ARAÚJO

***RELATÓRIO DE PROJETO: FINANÇASPRO - GERENCIADOR DE FINANÇAS PESSOAIS E GERADOR DE TABELAS DE INVESTIMENTO***

NATAL

2025

**Introdução e Contexto**

A cultura consumista brasileira evidencia que grande parte da população não possui o hábito de poupar ou investir, frequentemente gastando mais do que suas receitas permitem. Essa realidade resulta em altos índices de endividamento e limita o crescimento financeiro e pessoal dos indivíduos. Diante desse cenário, ferramentas que auxiliem na organização das finanças pessoais tornam-se cada vez mais necessárias.

O projeto, intitulado “FinançasPro – Gerenciador de Finanças Pessoais e Gerador de Tabelas de Investimento”, tem como objetivo principal auxiliar o usuário a ter controle sobre suas finanças diárias. Inicialmente, o programa permite o acompanhamento detalhado de receitas e despesas, fornecendo uma visão clara de lucros e déficits. Em uma segunda etapa, o projeto visa incentivar a poupança e o planejamento financeiro, oferecendo tabelas de investimento personalizadas que consideram metas específicas de valores e prazos, adaptando-se à realidade financeira de cada usuário.

O projeto busca enfrentar diretamente a dificuldade que muitas pessoas têm em gerenciar suas finanças pessoais, evitando gastos excessivos e promovendo hábitos de poupança e investimento. Ao oferecer uma visão estruturada das finanças e ferramentas para planejamento, o programa reduz o risco de endividamento e promove maior segurança e organização financeira.

O projeto foi escolhido por afinidade com a área de finanças pessoais e pelo reconhecimento da necessidade de um programa que vá além do simples monitoramento diário. Ao unir o acompanhamento das movimentações financeiras com a criação de tabelas de investimento baseadas em metas reais do usuário, o FinançasPro proporciona não apenas controle, mas também incentivo ativo à poupança, ajudando o usuário a alcançar objetivos financeiros concretos de maneira prática e adaptável.

**Análise Técnica**

O projeto foi desenvolvido utilizando uma abordagem iterativa e modular, priorizando clareza e aprendizado em programação na linguagem C. O desenvolvimento iniciou-se com a criação de um menu principal que direciona o usuário para duas funcionalidades centrais, sendo o gerenciador de finanças e as tabelas de investimento. Cada funcionalidade foi organizada em funções separadas, o que permitiu a reutilização do código e a melhor legibilidade do programa. Laços de repetição e validações foram implementados com o objetivo de manter a interação contínua do usuário, garantindo que o programa só seja encerrado quando o usuário assim desejar.

O projeto foi desenvolvido utilizando o editor Visual Studio Code e compilado com o GCC conforme orientações do professor, sendo o Visual Studio Code a ferramenta escolhida por proporcionar facilidade na edição do código, na depuração e na organização dos arquivos.

As estruturas condicionais foram aplicadas no programa através de comandos if e else if para validar as escolhas do usuário nos menus e submenus. Essas estruturas foram utilizadas para tratar condições específicas, como verificar se uma tabela de investimento já foi criada antes de exibir seus valores, evitando erros durante a execução. As estruturas de repetição foram implementadas com laços while, mantendo o menu principal e os submenus ativos até que o usuário optasse por sair, permitindo que múltiplas operações fossem realizadas consecutivamente sem a necessidade de reiniciar o programa. Dentro desses laços, também foram aplicadas validações para assegurar que o usuário apenas selecionasse opções válidas, o que confere maior robustez à aplicação.

Os vetores foram aplicados no projeto com o objetivo de armazenar os valores planejados para cada mês na criação de tabelas de investimento, sendo o vetor tabelainvestimento[100] utilizado como armazenamento global. Essa abordagem permite o acesso direto a cada posição do vetor, facilitando cálculos, atualizações e futuras implementações, como o registro de depósitos realizados pelo usuário. As funções criadas no programa possuem responsabilidades específicas, sendo mostrarSaldo responsável por exibir o saldo atual, atualizarRendimento por adicionar rendimentos diários ao saldo, atualizarDespesa por registrar despesas diárias e atualizar o saldo, criarTabelaInvestimento por gerar a tabela de investimento com valores mensais e armazená-los no vetor global e mostrarTabelaInvestimento por exibir os valores planejados da tabela. O uso de funções trouxe maior organização ao código, permitindo a separação de responsabilidades e simplificando a manutenção e futuras expansões do programa.

As estruturas de dados utilizadas no projeto envolvem principalmente variáveis simples e vetores. O vetor tabelainvestimento[100] armazena os valores planejados para cada mês, permitindo a manipulação e exibição dos dados de forma estruturada. A variável mesestabela armazena a quantidade de meses definida pelo usuário para atingir seu objetivo financeiro, enquanto a variável saldoatual mantém o registro do saldo do usuário durante a execução do programa. Variáveis auxiliares, como menu, menugerenciador e menutabela, são utilizadas para controlar a navegação pelos menus e submenus. O uso de variáveis e vetores globais facilita o acesso a informações essenciais do programa sem a necessidade de passar dados entre funções, o que simplifica a lógica e torna o código mais adequado para um projeto de nível iniciante.

**Implementação e Reflexão**

**Dificuldades Encontradas: Principais desafios técnicos**

Durante a implementação do projeto, as dificuldades iniciais foram relativamente menores, pois a lógica básica de programação, incluindo a definição de variáveis, estruturas condicionais e laços de repetição, foi compreendida com facilidade. No entanto, surgiram desafios técnicos ao lidar com funções e vetores, devido à menor prática com esses conceitos, o que exigiu atenção extra para entender como declarar, inicializar e manipular esses elementos dentro do código. Outra dificuldade encontrada foi a instalação do compilador GCC, que apresentou alguns problemas no início, tornando a configuração do ambiente de desenvolvimento um obstáculo adicional.

**Soluções Implementadas: Como foram superados os desafios**

As soluções implementadas envolveram estratégias de aprendizado e adaptação. Por ter mais facilidade com a lógica básica, o projeto foi iniciado focando na construção de menus, condicionais e laços de repetição, permitindo que o programa apresentasse resultados funcionais desde os primeiros testes. Paralelamente, foram estudadas as bases de funções e vetores para que pudessem ser aplicadas ao código de maneira inicial, com a perspectiva de melhorias futuras à medida que o aprendizado avançasse. Quanto à questão do ambiente de desenvolvimento, a instalação do Visual Studio Code foi adotada como alternativa eficiente, permitindo contornar os problemas iniciais com o GCC e possibilitando a escrita e execução do código de forma organizada.

**Organização do Código: Justificativa da estrutura escolhida**

A organização do código foi estruturada com funções separadas para cada operação principal, como mostrar saldo, atualizar rendimento, registrar despesas e criar ou exibir a tabela de investimento. Essa escolha buscou separar responsabilidades e facilitar a manutenção e futura expansão do projeto. O objetivo inicial do código era apresentar resultados funcionais e permitir a interação do usuário de forma clara, e à medida que o conhecimento sobre funções, vetores e boas práticas em programação se consolidar, a organização poderá ser aprimorada, podendo incluir a separação do código em arquivos distintos e uma estrutura modular ainda mais eficiente.

**Conclusão: Aprendizados obtidos e possíveis melhorias**

A conclusão do trabalho destaca o aprendizado significativo que ocorreu durante a retomada da programação, após um período sem prática. O projeto proporcionou a fixação de conhecimentos fundamentais em C, incluindo a comunicação com o usuário, uso de condicionais e laços de repetição, além de introduzir conceitos novos, como funções e vetores, que começaram a ser compreendidos na prática. Além da parte técnica, houve aprendizado em planejamento da implementação, organização das etapas do projeto e definição de objetivos claros, o que contribuiu para um desenvolvimento mais estruturado. Entre as melhorias pessoais previstas, está o aprimoramento do cronograma de implementação, permitindo que, conforme os estudos avancem, o código se torne mais funcional, enxuto e organizado, refletindo uma evolução contínua do aprendizado em programação.

**Perguntas Orientadoras**

**Quais conceitos da Unidade 1 foram aplicados e onde?**

As variáveis com tipos bem-definidos foram usadas para armazenar informações essenciais, como o saldo atual do usuário, valores de rendimentos e despesas diárias, quantidade de meses para atingir uma meta de investimento e os índices de navegação nos menus. Foram aplicadas operações aritméticas, como soma e subtração, para atualizar o saldo, além de operações relacionais nas validações dos menus e nas verificações de entrada do usuário. Vetores foram empregados para armazenar os valores mensais planejados na tabela de investimento, permitindo a criação de uma estrutura organizada para acessar e exibir os valores de cada mês. Os comandos condicionais if/else foram aplicados em todos os menus e submenus para validar escolhas do usuário e direcionar corretamente a execução do programa. Já os comandos de repetição while foram utilizados para manter os menus ativos até que o usuário optasse por sair, permitindo múltiplas interações consecutivas e garantindo que apenas entradas válidas fossem aceitas. Além disso, foram criadas funções separadas para mostrar o saldo, atualizar rendimentos, registrar despesas e criar ou exibir a tabela de investimento, totalizando mais de três funções além da função principal main.

**Como a organização em funções facilita a manutenção do código?**

A organização do código em funções facilitou a compreensão do programa, com cada função projetada para desempenhar uma tarefa específica, como atualizar o saldo ou gerar a tabela de investimento, permitindo que modificações futuras sejam feitas em uma única parte do código sem impactar outras funcionalidades.

**Quais foram os principais desafios na implementação das estruturas de repetição?**

Os principais desafios na implementação das estruturas de repetição envolveram a criação de menus que permanecessem ativos enquanto o usuário desejasse interagir e a validação das entradas para evitar opções inválidas. Foi necessário planejar os laços while de forma que os submenus e o menu principal não entrassem em conflito e sem existência de erros na compilação.

**Como os vetores foram utilizados para resolver o problema proposto?**

Os vetores foram utilizados para resolver o problema proposto armazenando os valores mensais que o usuário precisaria depositar para atingir um objetivo financeiro. Essa estrutura permitiu criar uma tabela organizada, em que cada posição do vetor corresponde a um mês, possibilitando calcular parcelas iguais e futuras melhorias, como registrar depósitos reais ou ajustar valores de acordo com alterações no planejamento do usuário.

**Que melhorias poderiam ser implementadas nas próximas unidades?**

Entre as melhorias que poderiam ser implementadas nas próximas unidades estão a inclusão de persistência de dados, para que as informações do usuário sejam armazenadas mesmo após o programa ser fechado, a implementação de funcionalidades adicionais para manipular valores já depositados na tabela de investimento, o aprimoramento da interface de interação para torná-la mais intuitiva e a expansão do código para permitir visualização dos saldos de forma diária. Outra melhoria é a organização do código mais refinada com a separação em múltiplos arquivos e funções adicionais.