**ENUNCIADO DO PROJETO:**

Desenvolver uma **calculadora completa** via interface de linha de comando (CMD) em C#. O sistema deve implementar funções (métodos) para operações matemáticas e armazenar os resultados em três estruturas de dados: **Fila, Pilha e Lista**. O usuário poderá escolher imprimir os resultados acumulados conforme a estrutura desejada:

**1)** Imprimir em forma de Fila  
**2)** Imprimir em forma de Pilha  
**3)** Imprimir em forma de Lista (com opção de escolher o índice ou resultado específico)

O menu da aplicação deve seguir princípios de Interação Humano-Computador (**IHC**) e Experiência do Usuário (**UX**).

Documento de Planejamento de Projeto - Calculadora CMD em C#

# Nome do Projeto:

Calculadora

# Integrantes do Grupo:

Eduarda e Gabriel Oliveira

# 3. Análise dos Requisitos

3.1 Requisitos Funcionais:

- Realizar operações básicas: adição, subtração, multiplicação e divisão

- Amarzenar os resultados em três estruturas diferentes: Fila, pilha e lista

- Exibir menu principal com opções claras e acessíveis

- Fornecer mensagens de erro claras em caso de valores invalidos ou operações invalidas

- Registrar o historico de operações realizadas

3.2 Requisitos Não Funcionais:

- Desempenho: Velocidade de calculo e tempo de resposta

- Usabilidade: facil de usar e acessibilidade

- Confiabilidade: Precisão nos calculos

# 4. Definição da Arquitetura

Tipo de Arquitetura (assinale uma):

[ ] Monolítica

[ ] Camadas

[ ] Cliente-Servidor

[ X ] Microserviços

**Justificativa da escolha:**

Microserviços favorece tanto o crescimento gradual do sistema quanto a rápida resolução de problemas localizados, sem comprometer a operação global. Por meio de APIs bem definidas, os serviços comunicam-se uns com os outros mantendo o isolamento necessário para garantir segurança e escalabilidade.

# 5. Descrição das Camadas do Sistema

5.1 Camada de **Interface** (CMD com princípios de IHC):

Esta camada é responsável por intermediar a comunicação entre o utilizador e o sistema, para garantir uma boa experiência, a apresentação clara de menus, feedback imediato para ações do utilizador e mensagens de erro compreensíveis. Os comandos são organizados de forma intuitiva e estruturada, visando facilitar o fluxo de tarefas e reduzir a possibilidade de erro. Além disso, a camada contempla acessibilidade, permitindo navegação por teclado e textos descritivos, tornando o sistema amigável e inclusivo para diferentes perfis de utilizadores.

5.2 Camada de **Negócios**:

A camada de negócios garante a integridade e a consistência dos dados, realizando operações como validação de entradas, aplicação de políticas específicas do domínio, integração entre diferentes microserviços e gestão de transações. Além disso, esta camada facilita a escalabilidade e a manutenção do sistema, permitindo que alterações nas regras de negócio sejam feitas de maneira centralizada e sem impacto direto nas demais camadas.

5.3 Camada de **Dados** (uso de fila, pilha e lista):

A camada de dados utiliza fila para processar tarefas na ordem de chegada , pilha para comandos de desfazer/refazer e lista para armazenar conjuntos dinâmicos, como históricos e logs, garantindo organização e eficiência na gestão de informações.

# 6. Projeto Detalhado dos Componentes

Descreva os principais componentes e suas responsabilidades:

- Componente 1: Gestor de Sessão: Cuida da autenticação, autorização e mantém o estado das sessões dos utilizadores.

- Componente 2: Orquestrador de Microserviços: Gere a integração entre serviços, coordena pedidos e equilibra a carga.

- Componente 3: Gestor de Dados Transacionais: Processa, armazena e recupera dados usando filas, pilhas e listas, garantindo a integridade das transações.

# 7. Estratégia de Documentação

Como o grupo pretende documentar o projeto (ex: comentários no código, arquivos markdown, etc.):

Pretendemos comunicar no codigo e fazer um documento documentando cada passo

# 8. Validação do Projeto

Como o grupo pretende validar o funcionamento do sistema (ex: testes manuais, testes automatizados, revisão entre pares):

Fazendo teste, revisando entre pares e anotando a mudança para não haver mais erros

# 9. Diagramas UML

Espaço reservado para inserção de diagramas UML (caso desejado):

[ ] Diagrama de Casos de Uso

[ ] Diagrama de Atividades

[ ] Diagrama de Classes

(Anexe ou desenhe abaixo)

# 10. Observações Finais