# **Fluxo de Caixa - Documentação**

## **Sumário**

1. Pré-requisitos
2. Clone o repositório
3. Configuração do Banco de Dados
4. Compilar e Executar o Projeto
5. Executar os Testes
6. Instruções do Postman
7. Considerações Finais

## **Pré-requisitos**

Antes de começar, certifique-se de ter as seguintes ferramentas instaladas:

* [.NET 8 SDK](https://dotnet.microsoft.com/download/dotnet/8.0)
* [SQL Server](https://www.microsoft.com/pt-br/sql-server/sql-server-downloads) ou SQL Server Express
* Postman
* [Git](https://git-scm.com/downloads)

## **Clone o repositório**

Para clonar o repositório do projeto, utilize o seguinte comando:

git clone <https://github.com/gilgrap/testeArquitetura.git>

Este repositório está organizado da seguinte forma, no raiz o readme e duas pastas Documentação e FluxoDeCaixa.

Na pasta documentação estão os arquivos png do Diagrama e o Docx com a descrição dele.

Na pasta FluxoDeCaixa esta a solução em .Net 8, o projeto nnso comtempla toda a solução para simplificar e exemplificar. Importante configurar a string de conexao.

O script de banco de dados está no tópico abaixo deste documento.

## **Configuração do Banco de Dados**

1. **Script de Banco de Dados**: Execute o script SQL fornecido para criar as tabelas necessárias no seu banco de dados.

sql

Copiar código

CREATE DATABASE FluxoDeCaixaDB;

USE FluxoDeCaixaDB;

Go;

CREATE TABLE Lancamentos (

Id INT PRIMARY KEY IDENTITY(1,1),

Tipo VARCHAR(10) NOT NULL, -- 'Debito' ou 'Credito'

Valor DECIMAL(18,2) NOT NULL,

Data DATETIME NOT NULL DEFAULT GETDATE()

);

Go;

CREATE TABLE SaldoDiario (

Id INT PRIMARY KEY IDENTITY(1,1),

Data DATE NOT NULL,

Saldo DECIMAL(18,2) NOT NULL

);

Go;

1. **Configuração da String de Conexão**: Edite o arquivo appsettings.json no projeto api para configurar a string de conexão do banco de dados:

json

Copiar código

"ConnectionStrings": {  
 "DefaultConnection": "Server=seu\_servidor;Database=seu\_banco;User Id=seu\_usuario;Password=seu\_senha;"  
}

## **Compilar e Executar o Projeto**

1. Abra o projeto no Visual Studio.
2. Certifique-se de que o projeto api está definido como o projeto de inicialização.
3. Compile o projeto.
4. Execute o projeto para iniciar a API.

## **Executar os Testes**

1. No Visual Studio, abra o **Test Explorer** (Menu Test -> Windows -> Test Explorer).
2. Clique em **Run All** para executar todos os testes unitários.
3. Verifique se todos os testes passam.

## **Instruções do Postman**

Para interagir com a API, você pode usar o Postman. Aqui estão alguns exemplos de requisições que você pode fazer:

### **1. Adicionar Lançamento**

* **Método**: POST
* **URL**: <https://localhost:5001/api/lancamentos>
* **Body** (JSON):

json

Copiar código

{  
 "Tipo": "Credito",  
 "Valor": 100.00,  
 "Data": "2023-10-25T00:00:00"  
}

### **2. Consultar Saldo Diário**

* **Método**: GET
* **URL**: <https://localhost:5001/api/saldo>

### **Considerações Finais**

* Certifique-se de que todas as dependências e pacotes NuGet estejam instalados e atualizados.