

Objetivo

Este projeto tem como objetivo criar um aplicativo onde o usuário consiga cadastrar suas despesas mensais e acompanhar se no mês ele economizou ou não conforme um objetivo/limite registrado.

Fluxo de sessão:

O aplicativo deve ter um controle de sessão, ou seja, o usuário deve poder criar uma conta (**signup**), iniciar uma sessão com base na conta criada (**signin**) e encerrar sessão após iniciada a sessão (**signout**).

Campos obrigatórios: data de nascimento, nome, email (chave primária) e senha

Se o usuário fechar o aplicativo e estiver autenticado, ele deve ser redirecionado para a tela autenticada de **Home**. Caso contrário, deve ser redirecionado para a tela de **Signin**.

Fluxo da home:

O aplicativo deve ter uma tela de Home. Nela deve ter:

- Card com feedback
 - Cenário que o usuário economizou;
 - Cenário que ele não economizou (gastou a mais do limite);
 - Cenário que nenhum limite foi registrado.
 - Adicionar um botão que redireciona para a tela de limite.
- Barra de soma das despesas mensais conforme o limite mensal cadastrado
 - Caso o usuário tenha cadastrado despesas, mas não cadastrou um limite, deve-se mostrar a barra com a soma de despesas mesmo assim, porém sem limite.

Na mesma tela, deve ser possível filtrar por data (mês). Caso o mês já tenha sido finalizado, além de mostrar a barra de soma, deve mostrar se o usuário atingiu o objetivo de gastar menos que o limite cadastrado. Em caso de sucesso, mostrar uma imagem de forma positiva com um texto explicativo. Em caso de falha, mostrar uma imagem de forma negativa com um texto motivador para o mês seguinte.

Fluxo meus dados:

Na tela meus dados devem conter as informações de usuário cadastradas no signup.

Campos que devem ser exibidos: nome, email e data de nascimento.

Além das informações do usuário, deve haver um botão para o usuário encerrar a sessão (signout).

Fluxo de cadastro de despesa:

Deve ter uma tela/modal para o usuário conseguir cadastrar uma despesa.

- Não pode ser possível criar/editar/cadastrar despesas dos meses anteriores ao mês corrente;

Campos obrigatórios: descrição, valor e mês referência.

Deve ser possível buscar as despesas cadastradas por mês (Histórico).

Deve ser possível editar e excluir uma despesa criada do mês corrente ou superior.

Fluxo de cadastro de limite:

Deve ter uma tela/modal para o usuário conseguir cadastrar um limite mensal.

- Só deve ser possível criar um limite por mês

- Não pode ser possível criar/editar/cadastrar um limite dos meses anteriores ao mês corrente;

Campos obrigatórios: valor e mês referência.

Deve ser possível buscar o limite por mês (Consulta).

Deve ser possível editar e excluir um limite criado do mês corrente ou superior.

Requisitos técnicos:

Requisitos back-end:

- Pode ser utilizada qualquer linguagem de programação;
- Rotas privadas só podem ser acessadas com um token válido;
- Deve haver validações conforme requisitos (criação de usuário com email já existente, criação de limite mês passado, criação de despesa mês passado e etc);
- O banco de dados deve ser MySql ou Postgres.

Requisitos front-end:

- O token de autenticação de ser um JWT;
- O token de autenticação deve ser armazenado no local storage e repassado nas chamadas REST para os endpoints privados;
- Deve-se validar se o usuário está autenticado nas trocas de rota;
- Deve haver validações antes de fazer requisição para o back-end (campos obrigatórios preenchidos e etc).

Regras e entrega:

Regras:

O trabalho deve ser feito em grupo de no máximos 3 pessoas;

Data de entrega: 22/06/2025 (23:59);

Data da apresentação: 23/06/2025 (19:00);

A entrega deve possuir os seguintes materiais:

- Código back-end
- Código front-end
- Slide de apresentação
- Vídeo interativo do app rodando acessando as principais telas de forma previa

Requisitos gerais:

Deve-se utilizar obrigatoriamente React Native;

Deve ser entregue um aplicativo de forma funcional conforme requisitos descritos (front-end e back-end);

O aplicativo deve ser apresentado em aula;

A aplicação deve estar funcional e rodando no dia da apresentação pois será solicitado fluxos de usuário para serem apresentados.

Pontos que serão avaliados:

- Demonstração de evolução em cada aula;
 - Toda aula, de forma presencial, irei na mesa de cada grupo e comparar as evoluções do projeto, logo, o grupo tem que apresentar previamente.
- Apresentação do trabalho;
- Aplicação de forma funcional como um todo;

- Código React e React Native;
- Código React;
- Validações de campos e interação;
- Utilização excessiva de IA;
- Estrutura do código (componentes, duplicidade de código, código não utilizado e etc)
- Design do e UI;
- Data de entrega;
 - Entregar após a data e horário definido, será zerado

Protótipo:

[Figma](#)

OBS: O visual descrito acima no Figma é um exemplo base. Pode-se ser seguido ou não a parte visual e estrutura de tela, porém a ideia das funcionalidades, propósito e requisitos solicitados devem ser cumpridos.