Desafio: Sistema de Controle de Despesas Pessoais com PHP e MySQL

Objetivo Geral:

Crie um sistema de controle financeiro básico, onde será possível gerenciar **despesas** e **receitas**, utilizando PHP puro e MySQL. O sistema deve permitir que o usuário registre suas transações e as classifique por **categorias** (no caso das despesas), além de visualizar um resumo financeiro em uma **dashboard**.

O que será avaliado:

- Compreensão de conceitos de CRUD (Create, Read, Update, Delete).
- Uso de PHP para interagir com um banco de dados MySQL.
- Implementação de uma dashboard com totais de despesas, receitas e saldo.
- Relacionamento entre despesas e categorias no banco de dados.
- Aplicação de boas práticas de desenvolvimento orientado a objetos (organização de código, classes, métodos, e separação de responsabilidades).
- Utilização de segurança básica (sessões, proteção de páginas restritas) opcional para quem fizer o desafio extra.

Passo a Passo do Desafio:

1. Configuração Inicial

- Banco de Dados: Crie um banco de dados MySQL com as tabelas necessárias para armazenar despesas, receitas e categorias de despesas. As tabelas sugeridas são:
 - o Tabela categorias: contendo id e nome da categoria.
 - Tabela transacoes: contendo id, tipo (despesa ou receita),
 descricao, valor, data e categoria_id (somente para despesas, que será uma chave estrangeira ligando à tabela de categorias).

2. CRUD de Despesas e Receitas com Categorias

Cadastro de Transações:

- Ao adicionar uma nova despesa, o usuário deve ser capaz de escolher uma categoria entre as disponíveis (ex: Alimentação, Transporte, Saúde).
- Cada despesa deve ser registrada com tipo, descrição, valor, data e categoria.
- o Para receitas, não será necessário associar uma categoria.
- **Listagem:** Crie uma página onde o usuário possa **ver todas as suas transações**, com uma tabela mostrando o tipo, descrição, valor, data e categoria (somente para despesas).
- **Edição:** Permita que o usuário **edite** uma transação existente, podendo alterar a categoria da despesa.
- Exclusão: Adicione a possibilidade de excluir uma transação.
- Gerenciamento de Categorias: Implemente a possibilidade de o usuário criar, editar e excluir categorias de despesas (item extra).

3. Dashboard Financeira

- Crie uma dashboard simples, onde o usuário possa visualizar os totais:
 - o Total de despesas, filtrado por categoria (o filt).
 - Total de receitas.
 - Saldo atual, que deve ser a diferença entre o total de receitas e despesas.

O objetivo da dashboard é dar ao usuário uma visão geral de suas finanças em tempo real.

4. Validações e Segurança

Validação: Valide os dados do usuário, garantindo que não sejam inseridos valores inválidos (como valores negativos para receitas, campos vazios, etc.).

Desafio Extra: Sistema de Login (Opcional)

Se você deseja um **desafio adicional**, pode implementar um **sistema de login** para autenticar os usuários no sistema.

Requisitos do Sistema de Login:

• Banco de Dados:

- o Crie uma tabela **usuarios** para armazenar os dados dos usuários:
 - id: identificador único.
 - nome: nome do usuário.
 - email: e-mail do usuário (único).
 - senha: senha do usuário, devidamente criptografada.

Registro e Login:

- o O usuário deve se **registrar** com nome, e-mail e senha.
- Utilize hashing de senhas com password_hash() para garantir que as senhas sejam armazenadas de forma segura no banco de dados.
- Após o registro, o usuário deve ser capaz de fazer login com suas credenciais.

Segurança e Sessões:

- Sessões: Utilize sessões para manter o usuário logado e garantir que apenas usuários autenticados possam acessar páginas protegidas do sistema.
 - Após o login, uma sessão deve ser iniciada para garantir que apenas o usuário autenticado possa acessar suas próprias despesas e receitas.
 - Neste caso, você deverá adicionar um "usuario_id" em todas as tabelas utilizadas (transações, categorias, etc).
- Proteção de Páginas Restritas: As páginas que manipulam despesas, receitas e categorias devem estar protegidas. Se o usuário não estiver logado, ele não deve ter acesso a essas páginas.

Logout:

 Implemente a funcionalidade de logout, permitindo que o usuário encerre a sessão de forma segura.

Obs.: Esse sistema de login e segurança não é obrigatório, mas será considerado um diferencial se implementado corretamente!

Regras do Desafio:

1. Repositório GitHub:

- Crie um repositório no GitHub com o nome sugerido: controlefinanceiro-desafio.
- O repositório deve ser público e conter um arquivo README.md
 com as instruções de instalação e execução do projeto.

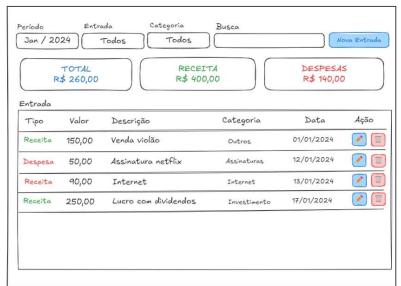
2. Prazo de Entrega:

- A data limite de entrega é 20/10. Se preferir, o desafio pode ser entregue antes dessa data.
- o Enviar o link do repositório assim que finalizar o projeto.

IMPORTANTE:

- Focar nas funcionalidades do backend é o mais importante, mas uma boa apresentação visual do frontend será valorizada.
- Seguir modelo das telas conforme imagens abaixo
- Implementar o filtro por período, os outros filtros entram fica como extra.

Lista de Despesas / Receitas



listagem

exibir o total de acordo com os filtros
 confirmação ao clicar em deletar,
 ou ir para uma tela de confirmação separada
 ordenar por padrão pela data, da mais
 antiga para mais atual

filtros

- no período, vir automaticamente selecionado o mês atual caso vazio
 no período, criar um select com as opções de acordo as datas das entradas existentes e a opção Todos
- entrada opções: Receita /Despesa / Todos
- categoria opções: Todos e as categorias que você estipular

Nova entrada ou editar entrada existente

