

**Objetivo:**

Implementar um sistema de controle de gastos residenciais.

Deixar claro qual foi a lógica/função do que foi desenvolvido, através de **comentários** e **documentação** no próprio código.

**Especificação:**

Em linhas gerais, basta que o sistema cumpra os requisitos apresentados. É importante que o sistema seja separado entre webApi e front.

**Tecnologias (obrigatório):**

**Back-end:** C# e .Net.

**Front-end:** React com typescript.

**Persistência:** Fica a sua escolha, mas os dados devem se manter após reiniciar o sistema.

**Funcionalidades:****Cadastro de pessoas:**

Deverá ser implementado um cadastro contendo as funcionalidades básicas de gerenciamento: criação, deleção e listagem.

Em casos que se delete uma pessoa, todas a transações dessa pessoa deverão ser apagadas.

O cadastro de pessoa deverá conter:

- Identificador (deve ser um valor único gerado automaticamente);
- Nome (texto);
- Idade (número inteiro positivo);

**Cadastro de categorias:**

Deverá ser implementado um cadastro contendo as funcionalidades básicas de gerenciamento: criação e listagem.

O cadastro de categoria deverá conter:

- Identificador (deve ser um valor único gerado automaticamente);
- Descrição (texto);
- Finalidade (despesa/receita/ambas)

### **Cadastro de transações:**

Deverá ser implementado um cadastro contendo as funcionalidades básicas de gerenciamento: criação e listagem.

Caso o usuário informe um menor de idade (menor de 18), apenas despesas deverão ser aceitas.

O cadastro de transação deverá conter:

- Identificador (deve ser um valor único gerado automaticamente);
- Descrição (texto);
- Valor (número decimal positivo);
- Tipo (despesa/receita);
- Categoria: restringir a utilização de categorias conforme o valor definido no campo finalidade. Ex.: se o tipo da transação é despesa, não poderá utilizar uma categoria que tenha a finalidade receita.
- Pessoa (identificador da pessoa do cadastro anterior);

### **Consulta de totais por pessoa:**

Deverá listar todas as pessoas cadastradas, exibindo o total de receitas, despesas e o saldo (receita – despesa) de cada uma.

Ao final da listagem anterior, deverá exibir o total geral de todas as pessoas incluindo o total de receitas, total de despesas e o saldo líquido.

### **Consulta de totais por categoria (opcional):**

Deverá listar todas as categorias cadastradas, exibindo o total de receitas, despesas e o saldo (receita – despesa) de cada uma.

Ao final da listagem anterior, deverá exibir o total geral de todas as categorias incluindo o total de receitas, total de despesas e o saldo líquido.

### **CrITÉrios de avaliação:**

A avaliação do teste técnico será baseada nos seguintes pontos:

- Aderência às regras de negócio;
- Atenção aos detalhes;
- Qualidade e legibilidade do código;
- Boas práticas em .NET e React.

**Formato:**

Você deve subir o código-fonte da sua solução para um repositório no GitHub e nos enviar o link no campo abaixo.

**Observações:**

1. Solicitamos gentilmente que quaisquer referências à \*\*\*\*\* sejam removidas de todos os arquivos antes da publicação.
2. Caso se sinta à vontade, pode-se implementar recursos adicionais desde que os mesmos não afetem o funcionamento dos já especificados nesse formulário.