

Objetivo

Construir uma pequena carteira digital utilizando as tecnologias propostas.

Temática

Uma carteira digital é uma aplicação web, responsiva, que permita que um usuário:

- Acesse sua conta com um login e senha
- Verifique um saldo na tela inicial
- Tenha a opção de ver o histórico de transações de entrada e saída (débito e crédito) de dinheiro da sua conta
- Consiga transferir créditos para a conta de um amigo, usando o nome de usuário/login dele
- Deve ser possível criar um novo cadastro, em um formulário com login (obrigatório) senha (obrigatório) e nome (opcional)
- Quando uma nova conta for criada, deve ser creditado um valor de 100 reais.
 - Deve haver esse registro na tabela SALDO (abaixo), sem uma movimentação correspondente.
- Toda transferência para um amigo deve deduzir do saldo da conta de origem e creditar o valor na conta de destino, além de registrar a transferência na tabela MOVIMENTACAO.

Estrutura em banco

TABELA: USUARIO

coluna	tipo	obrigatória	chave primária	default
login	varchar	sim	sim	
senha	varchar	sim	não	
nome	varchar	não	não	

TABELA: SALDO

coluna	tipo	obrigatória	chave primária	default
login	varchar	sim	sim	
saldo	float8	sim	não	

TABELA: MOVIMENTACAO

coluna	tipo	obrigatória	chave primária	default
id_transacao	int	sim	sim (auto-gerada)	(usar sequence)
data	timestampz	sim	(auto-gerada)	now()
login_origem	varchar	sim		
login_destino	varchar	sim		
valor_transferido	float8	sim		

Tecnologias e critérios esperados

Fundamental:

- Frontend utilizando VueJs ou ReactJs (à escolha) para gerenciar os componentes visuais.
- Alguma biblioteca rest (ex.: Axios) para realizar chamadas ao backend
- Backend construído em NodeJs, utilizando ExpressJs separado em camadas que representem o controller da API, uma camada de negócio e uma camada de persistência
- Gravação dos dados em um PostgreSQL
- A API deve seguir um padrão Rest
 - Documente a API, indicando que endpoints e estruturas foram criadas.
- Instruções completas de execução, do download do repositório até o último passo necessário para conseguir acessar e usar a aplicação. Inclua qualquer questão de ambiente relevante e simplifique/automatize ao máximo esses passos.
 - Queries necessárias para criar tabelas, popular registros ou qualquer outra atividade de inicialização do banco devem estar documentadas no repositório e devem fazer parte das instruções de execução.
- O apontamento do banco postgres deve ser feito por meio de um arquivo de configuração, que permita indicar host, porta, schema, usuário e senha do banco.

Extra:

- A subida da aplicação ocorrer utilizando docker.
- A inicialização das tabelas do banco ser automatizada e não depender de uma pessoa acessar e executar CREATE TABLE e outros comandos similares para preparar o ambiente.
- A senha estar encriptada/com hash no banco usando SHA-256 ou algoritmo similar ou superior que impeça a leitura dela.

Super Extra:

- Se além do repositório, o ambiente for entregue em algum lugar online já executando para avaliação.
 - Nesse caso devem ser fornecidas instruções e credenciais para acesso ao banco para examinação
 - Não é necessário utilizar um DNS nesse caso. Basta apresentar um IP:PORTA que permita acesso.

Para todos os casos:

- Documente quais dos itens acima conseguiu seguir e executar e quais houve alguma tentativa, mesmo que falha. Compartilhe conosco até onde conseguiu chegar e o que por ventura o impediu.

Formato de entrega

- Um repositório público no Github ou Gitlab (para qualquer cenário)
- Opcionalmente, um endereço onde a aplicação já resida com as informações de acesso à banco solicitadas acima
- Um único repositório deve conter frontend, backend e recursos de banco de dados, além de toda a documentação.
- Utilize o documento README.md na raiz do projeto para expor os itens de documentação solicitados.
 - Esse formato permite links entre diferentes arquivos .md (markdown). Utilize-os na medida que entender necessário para organizar os tópicos documentados.