Desafio Técnico de Programação

Bem-vindo ao desafio técnico da Brasyst! Queremos avaliar suas habilidades com base em um problema real do dia a dia de um programador. O desafio consiste em criar um sistema de autenticação utilizando JWT (JSON Web Token), seguindo as especificações abaixo.

Desafio: Sistema de Autenticação JWT

Você deverá desenvolver um sistema de autenticação completo, com as seguintes funcionalidades:

1. **Cadastro de Usuário:** Permitir que novos usuários se cadastrem no sistema, armazenando os dados de login em um banco de dados SQL.
2. **Login:** Implementar o processo de login, onde, ao fornecer as credenciais corretas (e-mail e senha), um token JWT é gerado e retornado ao usuário.
3. **Verificação da Sessão:** Após o login, o sistema deverá validar o token JWT em cada requisição protegida, garantindo que o usuário esteja autenticado.
4. **Logout:** Implementar a funcionalidade de logout, invalidando o token JWT.

Requisitos Técnicos

Back-end:

- A API deverá ser desenvolvida em PHP, utilizando boas práticas de segurança para manipulação de senhas e tokens JWT.
- O sistema deverá conectar-se a um banco de dados SQL para armazenar as informações dos usuários (como credenciais e outros dados pertinentes).
- Você pode utilizar qualquer framework ou bibliotecas PHP que achar necessário, mas a aplicação deve ser funcional e bem estruturada.

Front-end:

- O front-end deve ser desenvolvido separadamente do back-end, com uma interface simples para cadastro e login de usuários.
- A comunicação com o back-end deverá ser feita através de chamadas HTTP (via fetch, Axios, etc.).
- O front-end pode ser desenvolvido utilizando HTML, CSS, e JavaScript (ou bibliotecas/frameworks de sua escolha).

Critérios de Avaliação

1. Funcionalidade: O sistema deve estar completo e funcionando conforme descrito.
2. Organização do Código: Será avaliado o uso de boas práticas de programação, organização e estrutura dos arquivos.
3. Segurança: É importante que o sistema tenha uma abordagem segura no tratamento de senhas, tokens JWT, e outras vulnerabilidades comuns.
4. Separação entre Front-end e Back-end: O front-end e o back-end devem ser desenvolvidos de forma separada, mas integrados corretamente.
5. Documentação: Explique brevemente como configurar e rodar a aplicação.