

Desafio

Objetivo: Desenvolver uma API CRUD com autenticação JWT e interfaces de usuário em Vue.js para um sistema de gerenciamento de usuários e concessionárias, tudo executado dentro de containers Docker.

Tecnologias Usadas:

- Laravel para o back-end
- MySQL para banco de dados
- JWT para autenticação
- Vue.js para o front-end
- Docker para containerização

Requisitos:

1. Configuração Inicial
 - Configurar um novo projeto Laravel e um projeto Vue.js.
 - Criar um ambiente Docker que inclua containers para Laravel, MySQL e Vue.js.
 - Configurar o Docker Compose para gerenciar múltiplos contêineres como uma única aplicação.
2. Dockerização
 - Criar um `Dockerfile` para Laravel e outro para Vue.js.
 - Definir serviços no `docker-compose.yml`, incluindo Laravel, Vue.js, e MySQL.
 - Garantir que todos os serviços estejam interconectados e possam se comunicar conforme necessário.
3. Modelos e Migrações
 - Criar modelos para Usuário e Concessionária com suas respectivas migrações.
 - Usuário deve ter: `id`, `nome`, `email`, `senha`.
 - Concessionária deve ter: `id`, `nome`, `sigla`, `CNPJ`.
4. Autenticação de Usuário
 - Implementar rotas de registro e login de usuários no back-end.
 - Usar JWT para gerenciar a sessão do usuário após o login.
5. CRUD de Concessionária no Back-End
 - Implementar rotas para o CRUD de concessionárias no Laravel:
 - Cadastro: `POST /concessionarias`
 - Listagem: `GET /concessionarias`
 - Detalhe: `GET /concessionarias/{id}`
 - Edição: `PUT /concessionarias/{id}`

- Exclusão: DELETE /concessionarias/{id}
 - Implementar validação adequada para todos os inputs.
6. Interfaces de Usuário no Vue.js
- Tela de Login: Criar uma interface de login que interaja com a API de autenticação para validar usuários.
 - CRUD de Concessionária:
 - Criar componentes Vue.js para as operações de CRUD:
 - Formulário de cadastro e edição de concessionárias.
 - Lista de concessionárias com opções para editar e excluir.
 - Integrar as chamadas API utilizando Axios para conectar o front-end ao back-end.
7. Middlewares
- Implementar um middleware no Laravel para verificar se o usuário está autenticado através do token JWT antes de permitir acesso às rotas de CRUD de concessionária.
8. Testes
- Escrever testes unitários e de integração para verificar a funcionalidade tanto da API quanto das interfaces de usuário.
9. Versionamento e Repositório Git
- Inicializar um repositório Git e realizar commits regulares para documentar o progresso do desenvolvimento.
 - Subir o projeto final no GitHub, incluindo todos os branches e commits, e fornecer um README detalhado para instalação e execução do projeto usando Docker.

Critérios de Avaliação:

- Qualidade do código: clareza, uso de padrões de projeto, comentários.
- Segurança: implementação correta de autenticação, validação de dados.
- Usabilidade das interfaces de usuário.
- Cobertura de testes: completude dos testes em relação aos requisitos.
- Configuração e uso de Docker para desenvolvimento e produção.
- Organização do repositório Git: estrutura de branches, commits, e documentação.

Entrega:

- O projeto deve ser submetido em um repositório Git (público ou privado) com instruções adequadas no README para configurar e rodar o projeto, incluindo a execução dos testes.