

Descrição do Teste

Você será responsável por criar uma aplicação de gerenciamento de tarefas usando a linguagem que desejar no backend (de preferência PHP (7.4 ou superior)) e HTML, CSS, JavaScript (de preferência Vanilla, sem uso de frameworks), e Bootstrap no frontend. A aplicação será executada em um ambiente Docker e deve incluir as seguintes funcionalidades:

Requisitos Técnicos

1. Backend (PHP):

- Crie uma API RESTful com as seguintes rotas:
 - `POST /tasks`: Cria uma nova tarefa.
 - `GET /tasks`: Retorna a lista de tarefas.
 - `PUT /tasks/:id`: Atualiza uma tarefa existente.
 - `DELETE /tasks/:id`: Exclui uma tarefa.
- Cada tarefa deve conter:
 - `id`: Identificador único (gerado automaticamente).
 - `title`: Título da tarefa.
 - `description`: Descrição da tarefa.
 - `status`: Status da tarefa (`pendente`, `em andamento`, `concluída`).
 - `created_at`: Data de criação.
 - `updated_at`: Data de atualização.
- Utilize um arquivo MySQL para persistir os dados.

2. Frontend:

- Use **HTML**, **CSS**, **JavaScript** e **Bootstrap** para construir uma interface simples e responsiva.
- Funcionalidades:
 - Listar as tarefas em uma tabela.
 - Formulário para criar e editar tarefas.
 - Botão para excluir uma tarefa.
 - Alterar o status da tarefa através de botões (ex.: "Concluir" ou "Reabrir").
- Utilize AJAX para comunicação com a API.

3. Docker:

- Crie um ambiente Docker para executar a aplicação.
- Inclua os serviços necessários (ex.: PHP, servidor web como Apache ou Nginx, banco de dados).
- Forneça um arquivo `docker-compose.yml` para subir toda a aplicação com um único comando.
- Execute o projeto na porta local 9000

4. Extras (Diferenciais):

- Autenticação básica no backend usando tokens JWT.
- Validação de dados no frontend e backend.
- Organização do código frontend utilizando módulos ES6.
- Boas práticas de segurança, como sanitização de entrada de dados.
- Testes unitários ou de integração para o backend.

Entrega

- Disponibilize o código em um repositório Git público (ex.: GitHub ou GitLab).
 - Inclua um README.md com as seguintes informações:
 - Descrição do projeto.
 - Instruções para configurar e rodar o ambiente (usando Docker).
 - Instruções para executar os testes (se aplicável).
-

Critérios de Avaliação

1. **Qualidade do Código:**
 - Clareza, organização e aderência a boas práticas.
 - Uso de padrões de projeto (se necessário).
2. **Funcionalidade:**
 - Todas as funcionalidades descritas estão implementadas corretamente.
3. **Design e Usabilidade:**
 - Interface responsiva e intuitiva.
4. **Configuração do Ambiente:**
 - Facilidade para configurar e rodar a aplicação usando Docker.
5. **Extras:**
 - Implementação dos diferenciais sugeridos.