

DESAFIO TÉCNICO

PROCESSO SELETIVO (DESENVOLVEDOR JÚNIOR)

Regras gerais

- O prazo final para a entrega do projeto será 17/03/2025 até às 17h;
- O código deverá estar em um repositório público (no github, bitbucket ou qualquer ferramenta GIT do seu desejo, desde que seja público e analisável);
- O sistema desenvolvido deverá estar hospedado na internet;
 - Não precisa comprar domínio, ou pagar por hospedagem. Existem meios de fazer tudo gratuitamente.
- Na entrega, deverá ser enviado:
 - Link do site;
 - Link do repositório;
 - Uma explicação dos principais problemas encontrados durante o desenvolvimento e como você os soluciona;
 - Uma explicação de como você colocou sua aplicação no ar;
- Será analisado:
 - Se todos os requisitos funcionais foram atendidos;
 - A qualidade do código (boas práticas);
 - A estrutura dos commits;
 - Responsividade;
- Acreditamos que a IA é uma ferramenta poderosa para nossa profissão, mas pedimos que as evitem, pois nosso intuito é conhecer seu trabalho;
 - Se houver qualquer detecção suspeita no código, ou na operação de entrega, o candidato será eliminado automaticamente;
 - Posteriormente, haverá uma entrevista, onde poderá haver perguntas específicas acerca do código, e caso não tenha sido você que o fez, não saberá responder

Projeto

- Abaixo temos duas opções de projeto, das quais o candidato deverá escolher apenas **uma** das opções;
- O projeto deverá ser feito nas linguagens postas como opção, não será aceito outra linguagem;
- Todos os requisitos funcionais deverão ser atendidos;
- Usar de boas práticas e padrões de projetos de maneira intencional, você poderá ser questionados acerca disso na etapa seguinte;

- Você tem liberdade para escolher qual visual aplicar no seu projeto, porém, boas práticas de UX/UI também serão analisadas;
 - Não precisa ser algo muito profundo, mas ao menos o básico precisará ser feito em relação aos elementos na tela, cores, etc.;
- O intuito do projeto é conhecer o conhecimento de vocês, então, fique à vontade para demonstrar da melhor forma possível;
- A estrutura de banco de dados deverá ser a mais otimizada possível;

Propostas

PROPOSTA 1 - Sistema de Gerenciamento de Tarefas

Descrição: Um pequeno sistema web para gerenciamento de tarefas, no qual usuários podem se cadastrar, criar tarefas, definir prazos e marcar tarefas como concluídas.

Tecnologias:

- Backend: PHP (Laravel ou CodeIgniter)
- Frontend: Angular (ou pode ser um template simples com Bootstrap + JS no framework PHP)
- API: Criar uma API para gerenciar tarefas e usuários (se usar o Angular, a API precisa ser RESTful)
- Banco de Dados: MySQL
- Autenticação: JWT, Sanctum ou Session para autenticação de usuário

Funcionalidades:

- Cadastro e login de usuários (com autenticação);
- CRUD de tarefas (criação, edição, listagem e exclusão);
 - As tarefas precisarão ter título, descrição, responsável, data limite para realização, data de registro, data de exclusão;
- Marcar tarefas como concluídas;
- Permitir filtro de tarefas (pendentes/concluídas);
- Validação de entrada de dados;

Boas Práticas:

- Separação clara entre backend e frontend;
- Utilização de rotas protegidas por autenticação;
- Organização do código seguindo padrões MVC;
- Uso de migrations para estruturação do banco de dados;
- Utilização de soft delete;
- Todos os campos de data, precisarão ter hora

- E exibição no formato brasileiro;

PROPOSTA 2 - Sistema de Gestão de Contatos

Descrição: Criar um pequeno sistema para que usuários possam armazenar contatos, como uma agenda digital.

Tecnologias:

- Backend: PHP (Laravel ou CodeIgniter)
- Frontend: Angular ou um simples frontend em Bootstrap + JS
- API: Criar uma API para gerenciar contatos (se usar o Angular, a API precisa ser RESTful)
- Banco de Dados: MySQL
- Autenticação: JWT, Sanctum ou Session para proteger os endpoints

Funcionalidades:

- Cadastro e login de usuários;
- CRUD de contatos (nome, telefone, email, observações, data de registro, data de exclusão);
- Listagem de contatos do usuário autenticado;
- Permitir excluir contato;
- Filtro para contatos ativos/excluídos/todos juntos;
- Buscar contatos por nome;
- Exportação da lista de contatos para CSV;

Boas Práticas:

- Separação de responsabilidades no código;
- Implementação de soft deletes para recuperação de contatos apagados;
- Validar formato do e-mail;
- Utilizar máscara no formato brasileiro para o telefone;