

Orientações:

- **Ler com atenção os requisitos;**
- **Dúvidas Enviar por e-mail:** carlos.braz@aplicativo.net e avisar a recrutadora via WhatsApp;
- **Prazo de execução do teste são de 48 horas corridas;**
- **Qualidade da implementação, boas práticas de padrões de código serão diferenciais.**

1. Documentação dos Requisitos Funcionais e Não Funcionais

- **Objetivo:** Descrever as funcionalidades principais e os requisitos que a aplicação deve atender.
- **Requisitos Funcionais:**
 - i. Cadastro de Pacientes com os campos: nome, telefone, sexo e e-mail.
 - ii. Geração de um número sequencial de atendimento para a fila de espera.
 - iii. Registros de triagem com sintomas, pressão arterial, peso, altura e direcionamento para uma especialidade.
 - iv. Chamada do paciente de acordo com a ordem de chegada.
- **Requisitos Não Funcionais:**
 - v. Interface amigável e responsiva (React ou Angular).
 - vi. Desempenho adequado com um tempo de resposta de até 2 segundos por requisição.
 - vii. Armazenamento seguro dos dados de pacientes.

2. Modelagem de Dados

- **Objetivo:** Criar um esquema de banco de dados que suporte as funcionalidades descritas.
- **Tabelas Principais:**
 - i. **Pacientes:** ID, Nome, Telefone, Sexo, Email.
 - ii. **Atendimentos:** ID, NumeroSequencial, PacienteID, DataHoraChegada, Status.
 - iii. **Triagem:** ID, AtendimentoID, Sintomas, PressaoArterial, Peso, Altura, EspecialidadeID.

3. Diagramas UML

- **Diagrama de Caso de Uso:** Representar o fluxo principal de cadastro, triagem e atendimento.

4. DFD (Diagrama de Fluxo de Dados)

- **Objetivo:** Visualizar o fluxo de informações entre os processos do sistema.
- **Principais Processos:**
 - i. Registro de Paciente na Recepção.
 - ii. Geração de Número de Atendimento.
 - iii. Processo de Triagem.

5. Implementação

- **Objetivo:** Desenvolver uma aplicação simples usando .NET Core com Entity Framework e um front-end em React ou Angular.
- **Tarefas de Desenvolvimento:**
 - i. Criação da API REST para o gerenciamento de pacientes e atendimentos.
 - ii. Implementação de endpoints para cadastro de pacientes, geração da fila de atendimento e triagem.
 - iii. Front-end com interface para registro de pacientes, visualização da fila e registro de triagem.
 - iv. Uso de uma base de dados SQL Server com as tabelas definidas anteriormente.

6. Critérios de Avaliação

- Qualidade da Documentação: Clareza nos requisitos e diagramas.
- Modelagem do Banco de Dados: Estrutura correta e normalização das tabelas.
- Implementação: Funcionalidade, qualidade do código, uso correto das tecnologias.
- Front-end: Responsividade, usabilidade e integração com a API.

Entrega

- O candidato deve entregar a documentação (requisitos, diagramas UML, DFD e modelo de dados).
- Implementação completa da aplicação (código-fonte da API e front-end)
- Um vídeo explicando a aplicação rodando, com todas as funcionalidades e passo a passo;
- Scripts SQL DDL de criação das tabelas e objetos de banco de dados;
- Scripts SQL DML de listagem, inclusão, alteração e exclusão de registros;
- O vídeo e fontes devem ser entregues via link compartilhado do drive, não enviar em anexo no e-mail.
- No corpo do e-mail enviar um resumo do processo de implantação, se não conseguir fazer algum processo, descrever o motivo e como implementaria em português estruturado;