

## Grupo SYM – Teste para Full Stack Developer

Você está participando da rodada do processo de contratação do Grupo SYM, esse teste nos permite avaliar seu código e habilidades para propor arquiteturas e solucionar problemas.

### Sem tempo para realizar testes técnicos?

Bem, você pode enviar o link de uma Pull Request com uma contribuição sua para qualquer projeto Open Source ou algum projeto próprio que você acredita que demonstre o seu nível técnico e a qualidade do seu código. Projetos próprios sem cobertura de testes, sem documentação ou sem integração contínua não serão considerados.

Mesmo assim considere fazer o nosso teste técnico, com ele teremos mais informações para poder avaliar o seu trabalho. No nosso teste técnico, você pode utilizar qualquer lib ou framework, mas quanto mais código seu for utilizado (leia-se não gerado automaticamente por frameworks) melhor você poderá ser avaliado.

### Pré requisitos

O seu teste deve ter um README com os passos necessários para:

- Rodar o projeto;
- Instalar as dependências;
- Rodar os testes automatizados;

### Tecnologia

Aceitamos preferencialmente soluções escritas utilizando as tecnologias Microsoft .NET / .NET Core para o BackEnd e Angular para o FrontEnd.

O teste deve utilizar como persistência de dados um banco MySQL.

Caso você não domine nenhuma das tecnologias acima citadas, pode utilizar outras linguagens/frameworks para a construção da solução, mais daremos prioridade na avaliação dos testes com os requisitos acima citados.

## Instruções

- Crie um repositório no Gitlab, GitHub ou Bitbucket e nos envie o link para o e-mail **julio\_borges@lifesys.com.br** com cópia para o e-mail **mariana\_couto@lifesys.com.br**;
- O prazo para a entrega do projeto é de uma semana (7 dias corridos), mais o quanto antes nos passar o link, mais rápida será nossa avaliação, portanto mesmo que não tenha concluído o trabalho, nos envie o link para que possamos avaliar todas as submissões (commits) no repositório;

## Aplicação

A solução proposta trata-se de um aplicativo Web para gerenciamento de consultórios/médicos, com a seguinte especificação:

- Deve ter um cadastro de médicos com as seguintes informações do médico:
  - CRM (Texto, 10 dígitos);
  - Nome do médico (Texto, 100 dígitos);
  - Telefone (Texto, 20 dígitos);
  - Valor da consulta (Decimal com duas casas após a vírgula).
- Deve ter um cadastro de consultórios com as seguintes informações:
  - Nome do consultório (Texto, 100 dígitos);
  - Endereço do consultório (Texto, 200 dígitos);
  - Telefone (Texto, 20 dígitos).
- Deve ter um meio para realizar o vínculo Médico/Consultório:
  - É uma forma de ligar um consultório a vários médicos, pode ser uma rotina/página separada ou pode ser um recurso embutido no cadastro de consultórios.
- Deve ter uma tela inicial com menu lateral ou superior para acesso às rotinas da aplicação;
- Um médico pode estar vinculado a no máximo 2 consultórios;
- Não é permitido cadastrar consultórios sem endereço e telefone;
- Não é permitido cadastrar médicos sem CRM;
- A aplicação precisa persistir os dados em um banco de dados MYSQL.

## Requisitos obrigatórios

Você precisará criar duas aplicações:

1. A primeira aplicação irá expor uma API seguindo os padrões REST para:
  - i. Criar/listar/remover/alterar os médicos;

- ii. Criar/listar/remover/alterar os consultórios;
  - iii. Vincular/desvincular médicos e consultórios.
2. A segunda aplicação será o FrontEnd que irá consumir a API acima listada e exibir os dados ao usuário.
  - i. Autenticação / login não é obrigatória, mais é considerada um *"Must Have"* para o projeto de teste.
3. Utilize boas práticas de programação e um código legível para uma melhor avaliação;

### **Critérios de avaliação:**

- Qualidade da abstração da solução;
- Estrutura do código, e qualidade dos testes unitários;
- Padronização de código;
- legibilidade do código;

### **Bonus points:**

- Ter configurado no projeto qualquer serviço de integração contínua;
- Conseguir levantar toda a solução com um simples docker-compose up (caso utilize containers);
- Seguir o guia de estilo padrão de mercado para a linguagem que você escolher usar;
- Um histórico do git (mesmo que breve) com mensagens claras e concisas.

***Enjoy your test and good lucky !!!***