



## DELOITTE BIG DATA - CASE TÉCNICO

Neste desafio você terá oportunidade de mostrar suas habilidades em programação. Fique à vontade para utilizar as linguagens listadas na [página da academia](#), o importante é a entrega final!

Teremos duas opções de desafios: [front-end](#) e [back-end](#). Escolha a modalidade que condiz com seus conhecimentos de programação. Caso você tenha conhecimento nas duas áreas, é livre para desenvolver o desafio completo, sendo as páginas renderizando conteúdo dinamicamente a partir da API.

### DESAFIO FRONT-END

A **Agência Cronos**, que presta serviços de tecnologia (desenvolvimento, marketing e UX design), contratou você para o desenvolvimento do site institucional com as seguintes páginas:

- **Página inicial:**
  - banner ilustrativo
  - 3 serviços + link para ver todos na página Serviços
  - 3 posts + link para ver todos na página Blog



- **Serviços:** lista de serviços
- **Quem somos:** lista de integrantes da equipe
- **Blog:** lista de posts

Utilize boas práticas para manter seu código organizado, e tags semânticas no HTML. É imprescindível que todas as páginas sejam responsivas e utilize React ou Angular para componentização.

## REQUISITOS MÍNIMOS FRONT-END

1. Todas as páginas devem ser responsivas
2. Utilize React ou Angular para organizar os componentes
3. Todas as páginas devem ter boa performance no carregamento de imagens e scripts
4. Todos os códigos deverão estar em um repositório no **Github** com README.md descrevendo tecnologias utilizadas no projeto.

## OPCIONAIS FRONT-END

1. Utilização de Cache
2. Uso de Logs
3. Testes automatizados
4. Deploy da aplicação

-----



## DESAFIO BACK-END

A **Agência Cronos**, que presta serviços de tecnologia (desenvolvimento, marketing e UX design), contratou você para o desenvolvimento de uma API para gerenciamento do conteúdo do site institucional.

Faça um **CRUD** de Serviços, Posts e Integrantes da Equipe, permitindo que o administrador do site consiga criar, editar, deletar e visualizar os dados através de um **painel administrativo**.

O gerenciamento dos dados deve ser feito com uma **API REST** conectada ao banco de dados relacional.

### REQUISITOS MÍNIMOS BACK-END

1. Todos os códigos deverão estar em um repositório no Github com README.md descrevendo tecnologias utilizadas no projeto.
2. API:
  - a. CRUD (Criar, ler, atualizar e deletar) de Serviços
  - b. CRUD (Criar, ler, atualizar e deletar) de Posts
  - c. CRUD (Criar, ler, atualizar e deletar) de Integrantes da Equipe
3. Banco de dados:
  - a. Utilizar banco de dados relacional
  - b. Adicionar dados para teste

### OPCIONAIS BACK-END

5. Autenticação de administrador para gerenciamento do conteúdo
6. Documentação da API
7. Testes automatizados
8. Deploy da aplicação