# 1 – Mundo Comum

A empresa Hona é uma startup brasileira voltada para a área da saúde. A empresa deseja criar uma plataforma com prontuários médicos eletrônicos. O nome da plataforma será M-REC e será lançado no mercado em meados de junho de 2019.

# 2 – Chamado para a Aventura

O objetivo da empresa é desenvolver um sistema web/mobile onde será possível para os pacientes, visualizar suas informações referentes a consultas e/ou exames e os médicos terão acesso a todos os registros do paciente, incluindo consultas antigas e/ou feitas por outros profissionais da saúde.

## Revisão e JavaScript

A empresa solicitou o desenvolvimento das páginas de apresentação (landing page), cadastro de usuário e login. Você deverá elaborar o layout de baixa/alta fidelidade e produzir a página web. Além disso a página de cadastro deve ter validação de entrada.

* Entrega 1: produção de layout de baixa/alta fidelidade das páginas;
* Entrega 2: Páginas web Login e Cadastro;
* Entrega 3: Validação na página de cadastro (cliente);

## Consumir/Enviar REST via JavaScript

A empresa contratou uma outra empresa para gerenciar a base de todos os seus clientes. Sendo assim, ela quer que você, como desenvolvedor, não mais mantenha a sua página estática, mas que ela mantenha interação com os dados do back-end.

Para isto, todos os registros que os clientes manipularem, devem ter interação com o back-end. Ou seja, todos os seus registros, serão enviados para o back-end.

Etapas:

* Entender o que é API;
* Entender JSON;
* Entender endpoints;
* Verbos HTTP;
* Os cadastros devem ter interação com o back-end;

## Banco de Dados

Infelizmente, o serviço que a empresa utilizava, não está mais no ar. A empresa parceira declarou que a partir do próximo mês, ela não mais irá prover o seu serviço e que você deve agora, construir o seu próprio sistema para que os dados dos usuários que já utilizaram a plataforma, continuem funcionando.

Para que isto ocorra, ela apenas te disponibilizou a base atual de cadastro de seus usuários. A empresa disponibilizou um arquivo \_\_\_\_\_\_\_.

* Entrega 1: Importar os dados para o banco de dados;
* Entrega 2: Realizar o teste manual de crud na base;

## Desenvolvimento da API

Desenvolver a API de maneira que todos os salvos armazenados passem a ser no SQL Server e toda a sua interação com o Front ou outras plataformas, seja por JSON. Além disso, deve-se restringir o acesso a cadastros de novos pacientes/médicos/exames/consultas somente a usuários autenticados. Deve-se trabalhar com autorização e autenticação.

* Entrega 1: Criar um sistema para que realize o crud de seus pacientes e/ou agendamentos/serviços;
  + Obs.: o front-end deve continuar integrado e funcionando depois de sua construção.
* Entrega 2: Os dados devem ser persistidos no banco;
* Entrega 3: Autorização e autenticação de usuários;
* Entrega 4: Áreas de cadastros e login devem ser totalmente funcionais e sua interação deve ser por API;

## Framework Vue

Para facilitar o seu trabalho, percebemos que, ao construir toda uma estrutura de front-end, apesar de ser leve e mais simples de trabalhar, a partir do momento em que a sua aplicação começa a crescer, aumenta a dificuldade e a sua manutenção no código e refatoração.

Pensando nisso, você poderá/deverá utilizar um framework para que facilite o seu processo de desenvolvimento.

* Entrega 1: Implementar o VueJS no projeto construído;

## Integração API/Vue

Mesmo o desenvolvedor trabalhando com um framework ou sua linguagem em seu código nativo, a interface com o usuário permanecerá a mesma. Sendo assim, toda sua integração e todas as suas funcionalidades deverão permanecer funcionando para o usuário e todas as mudanças deverão ser transparentes para o usuário da aplicação (médicos e pacientes).

* Entrega 1: Seu sistema, mesmo utilizando um framework, deverá continuar funcionando e interagindo com o projeto do back-end;

## Mobile

A empresa quer que você desenvolva os novos layouts de tela para o mobile, pois ela percebeu que a maioria de seus pacientes consultam seus registros através da versão mobile de seu site. Para eles, a consulta acaba se tornando bem mais rápida, diferente da visão do médico que deve conter todas as informações dos pacientes.

* Entrega 1: Layout de baixa/alta fidelidade;
* Entrega 2: Projeto desenvolvido no mobile;

Como a sua API já foi construída e os seus dados já estão sendo armazenados no banco de dados, consuma a API para visualizar as informações sobre os pacientes.

* Entrega 3: Seu aplicativo deve interagir com a API do back-end;

Como o seu aplicativo consome as informações do back-end, mas as informações do usuário serão buscadas através do token, armazene as informações sobre o usuário localmente em sua aplicação. Como nome e tipo de usuário.

* Entrega 4: Seu aplicativo deve armazenar as informações sobre o usuário localmente;