 Maratona Tech

*Tecnologia da Informação*

*Backend – P2*

# Backend

## Introdução

A imobiliária Viver um Sonho possui uma página estática que exibe os imóveis disponíveis para venda ou locação e sempre que precisa efetuar a inserções de novos imóveis, ela necessita contratar um desenvolvedor para inserir as informações de seus novos imóveis e seus respectivos dados em sua página. Pensando nisso, o dono da empresa deu a ideia de solicitar que fosse implementado em seu site, um painel de gestão de seus imóveis, no qual a empresa terá autonomia para inserir seu próprio conteúdo deixando de forma dinâmica. Foi solicitado que a tela principal do site possua um sistema de busca com filtros avançados para facilitar a busca de forma refinada com as informações mais especificas dos imóveis dos sonhos de seus clientes.

## Descrição do projeto e tarefas

A execução do projeto dura até 4 horas. Neste tempo você deve criar um sistema de gerenciamento para o site da imobiliária fazer o backend com uma base de dados MYSQL. Você pode utilizar a linguagem de backend a sua escolha disponibilizado na lista de infraestrutura para construção do sistema. Sua responsabilidade é desenvolver o backend e fazer a integração com o frontend disponibilizados, assim como quando necessário desenvolver trechos do frontend.

O sistema deve contemplar às seguintes funcionalidades:

* Gestão Imóveis.
* Pesquisa de imóveis
* Listagem indivídual
* Gestão administrativa

## Gestão imóveis

* A gestão dos imóveis deve contemplar a inserção, alteração, visualização e remoção dos imóveis.
* Cada imóvel deve contemplar seguintes informações:
  + Endereço
  + Tipo de anúncio
    - Aluguel
    - Venda
  + Estado (UF)
  + Cidade
  + Bairro
  + Quantidade de vagas
  + Quantidade de quartos
  + Quantidade de Banheiros
  + Quantidade de m² – Informar a área útil.
  + Valor do imóvel
  + Foto de capa
  + Nome
* O cadastro de imóveis é uma funcionalidade fundamental para o sistema e necessita que os dados sejam inseridos de forma normalizada, sendo assim antes de efetuar o cadastro de um imóvel o administrador deve cadastrar previamente os Tipo de imóvel
* Além disso, se ao informar o estado, cidade ou bairro do imóvel, alguma dessas opções ainda não estiver cadastrada no sistema, o sistema deve criá-las no banco de dados antes de cadastrar o imóvel. Isso permitirá associar o ID de cada informação ao imóvel. Após o cadastro dessas novas informações, elas devem estar disponíveis para os próximos cadastros ou atualizações de imóveis.

O projeto possui os seguintes **Requisitos Funcionais**:

# Página de Login:

* 2 campos de formulários que permitam que o usuário insira seu nome de usuário e senha.
* Validar as informações de login fornecidas pelo usuário.
  + Na validação crie um teste CAPTCHA que deve gerar um valor numérico entre 1000 e 9999 aleatório que deve ser validado pelo usuário, garantindo que o mesmo não seja um robô.
* Autenticar o usuário com sucesso e redirecioná-lo para a tela de administração.
  + Exibir uma mensagem de sucesso ao conseguir realizar o login: **Bem vindo administrador!**
* Exibir uma mensagem de erro em caso de falha na autenticação.
  + Indicar o campo com erro: **email ou a senha incorretos.**

## Página de Gestão de Imóveis

* Deve ser possível cadastrar e deletar imóveis e sua respectiva imagem, além de possuir uma listagem separada no painel administrativo
* Ao cadastrar e atualizar um imóvel, exibir campos para adicionar as propriedades mencionadas anteriormente.
* Local no formulário para que o usuário possa selecionar a imagem do imóvel a ser cadastrado.
  + Armazenar as imagens enviadas em um diretório no servidor.
  + Cada imóvel possui uma imagem
* O cadastro deve ser validado, só permitindo que seja realizado se todos os campos estejam preenchidos.
* Exibir uma mensagem de sucesso ao conseguir fazer o cadastro: **Cadastro realizado com sucesso!**
* Exibir uma mensagem de erro em caso de falha no envio, informando o motivo.

### Logout:

* Fornecer um botão de logout no cabeçalho de usuários logados, substituindo o botão de login e possuindo um fundo avermelhado.
* Ao clicar no botão de logout, encerrar a sessão atual do usuário e o usuário é redirecionado para a página inicial.

### Página inicial:

* Exibir a lista de todos os imóveis:
  + Os imóveis são exibidos em cards com as suas propriedades, sendo elas:
    - Imagem de capa
    - Tipo de imóvel
    - Endereço
    - Sigla do Estado
    - Qt. de vagas
    - Área do imóvel
    - Qt. de banheiros
    - Qt. de dormitórios
    - Preço (Colocar um identificador de “por mês” caso o imóvel seja para alugar)
* Os imóveis devem ser exibidos em três carrosséis diferentes, alterando sua listagem de acordo com o carrossel. São eles:
  + Carrossel ‘Destaques’ (Exibe 3 imóveis, ordenando-os pelos imóveis com mais acessos)
  + Carrossel ‘Alugar’ (Exibe 3 imóveis disponíveis para alugar, ordenando-os pelos mais recentes)
  + Carrossel ‘Comprar’ (Exibe 3 imóveis disponíveis para compra, ordenando-os pelos mais antigo)
* Ao clicar em um imóvel, o usuário deve ser redirecionado à página de listagem individual do imóvel.

### Página de pesquisa

* Ao clicar nos links de ‘Aluguel ou ‘Venda’ no cabeçalho, o usuário deve ser redirecionado à página de pesquisa de imóveis, com o campo de Tipo de anúncio (Alugar ou Comprar) já selecionado
* A página de pesquisa exibe os imóveis em cards iguais aos da página inicial, mas com o diferencial de poder filtrá-los de acordo com sua necessidade.
* Os filtros disponíveis são:
  + Endereço: Inserir uma string e localizar imóveis com endereços similares
  + Tipo: Alugar, Comprar
  + Estado, Cidade, Bairro
  + Vagas: Campo para inserir o número de vagas
  + Quartos: Campo para inserir o número de quartos
  + Banheiros: Campo para inserir o número de banheiros
  + Área: Campos para inserir a área mínima e máxima
  + Preço: Campos para inserir o preço mínimo e máximo
* Após clicar no botão ‘Aplicar filtros’, os filtros devem ser aplicados à página de pesquisa.
* Caso filtros não sejam selecionados, serão exibidos todos os imóveis disponíveis
* Após a listagem, deve ser possível ordenar a listagem pela área, preço e data, em ordem crescente ou decrescente.

### Listagem individual

* Exibe todas as informações de um imóvel em detalhes
  + Endereço
  + Cidade
  + Estado
  + Bairro
  + Vagas
  + Área
  + Banheiros
  + Quartos
  + Observações
* Após ser acessada, deve aumentar o número de acessos do imóvel em +1

### Página de Administração

Na página de administração deve ser exibido na parte superior os 3 imóveis com a maior quantidade de acesso,

Além disso deve ser exibido uma seção contendo:

* Listagem de imóveis com:
  + Imagem
  + Quantidade de acessos
  + Botões com opção de visualizar, editar e excluir
* Mensagens enviadas para contato do mais antigo até o mais novo

### Página de Contato

Na página de contato deve permitir que o usuário preencha os dados necessários disponíveis no arquivo layouts, esses dados devem ser salvos no banco de dados e serem possíveis de visualizar na página de administração

### Layout

* O layout que será exibido nas páginas Inicial, Pesquisa, Contato, Listagem individual e Login já foi construído e deve ser utilizado pelos competidores, sendo necessário a implementação da listagem dos dados fornecidos pelo backend e o envio dos dados nas páginas quando necessário.
* Já nas demais telas (referentes ao painel administrativo): O layout destas é livre e pode ser feito de acordo com o gosto do competidor, contanto que apresente as funcionalidades requeridas e seja minimamente intuitivo.

### Tecnologias:

* Utilizar qualquer linguagem de programação para desenvolver a aplicação.
* Utilizar MySQL como sistema de gerenciamento de banco de dados.
* Utilizar o básico de HTML, CSS e JS, podendo utilizar frameworks caso seja necessário ou facilite o desenvolvimento

## Base de dados

* Modele o BD necessário para resolver o problema da empresa descrita.
* Você é obrigado a usar o banco de dados para armazenar os dados referenciados.
* Como não criaremos uma tela de cadastro para usuários, utilize a instrução SQL adequada para adicionar um usuário padrão (**admin**) e uma senha padrão(**admin**), na tabela “Users”. É com este usuário e senha que poderemos fazer login no sistema.

Segue as tabelas que base para a implementação do sistema, juntamente de seus campos/colunas, os nomes dos atributos podem ter nomes diferentes contato que as informações armazenadas sejam as mesmas:

#### Tabela "Users":

* id (chave primária)
* name (nome do usuário)
* password (senha)

#### Tabela: “Realty”:

* id (chave primária)
* address (endereço do imóvel)
* offer\_type (tipo de oferta de imóvel (aluguel/venda))
* rooms\_qt (quantidade de quartos)
* toilet\_qt (quantidade de banheiros)
* views (quantidade de visualizações)
* parkingspace\_qt (quantidade de vagas)
* Image\_url (endereço para a imagem
* value (valor do imóvel)
* stateId (id do estado do imóvel)
* cityId (id da cidade do imóvel)
* neighborhoodId (id do bairro do imóvel)
* realtyTypeId (id do tipo de imóvel)

#### Tabela: “RealtyType”:

* id (chave primária)
* name (nome do tipo de imóvel)

#### Tabela "State":

* id (chave primária)
* name (nome do estado)

#### Tabela "City":

* id (chave primária)
* name (nome da cidade)

#### Tabela "Neighborhood":

* id (chave primária)
* name (nome do bairro)

#### Tabela "Contact":

* id (chave primária)
* name (nome do usuário)
* email (email do usuário)
* message (mensagem do usuário)

## Arquivos fornecidos

|  |  |
| --- | --- |
| **ITEM** | **DESCRIÇÃO** |
| Geral | Os arquivos de mídias fornecidas:   * Imagens * Arquivos html |

**Instruções ao competidor**

Ao final do processo deve ser enviado um

* Arquivo “README.txt” indicando como rodar a aplicação
* Código fonte com o backend e frontend
* Script sql para importar dados e tabelas do banco de dados
* Os arquivos devem ser salvos dentro da pasta “Backend – P2”