 Projeto Teste

*Tecnologias WEB*

*Módulo 8 - PHP*

### Submetido por:

Guibson Barros de Almeida Santana, PE

**Conteúdo**

[**PHP – 2 horas 3**](#_1fob9te)

[Introdução 3](#_d46xki4gywec)

[Descrição do projeto e tarefas 3](#_3znysh7)

[Base de dados 4](#_at02p47rrwsv)

[Arquivos fornecidos 4](#_rsmfapb40qsr)

[**Ficha de Avaliação Simplificada 5**](#_17dp8vu)

# PHP – 2 horas

## Introdução

A imobiliária Viver um Sonho possui uma página estática que exibe os imoveis 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 imoveis 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 imoveis, 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 2 horas. Neste tempo você deve criar um sistema de gerenciamento para o site da imobiliária WSBrasil utilizando PHP com uma base de dados MYSQL. Você pode utilizar o PHP puro ou um dos Frameworks PHP 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.
* Gestão Estados
* Gestão Cidades
* Gestão Bairros
* Gestão Comodidades
* Gestão usuários
* Exibir estatísticas do site

1. Gestão imoveis

A gestão dos imoveis deve contemplar a inserção, alteração, visualização e remoção dos imóveis.

Cada imóvel deve contemplar seguintes informações:

* + Endereço
    - Rua
    - Número
    - Complemento
  + Tipo de anúncio
    - Aluguel
    - Venda
  + Estado (UF)
  + Cidade
  + Bairro
  + Quantidade de vagas
  + Quantidade de quartos
  + Quantidade de suites
  + Quantidade de Banheiros
  + Comodidades
  + Quantidade de M2 – Informar a área útil.
  + Valor do imóvel
  + Foto de capa
  + Galeria de Fotos
  + Observações

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 em uma outra área os seguintes dados:

* Estados (UF)
* Cidades
* Bairros
* Comodidades
* Tipo de imovel

Conforme a necessidade o administrador deve ter a opção de inserir, alterar, visualizar e remover elementos das entidades mecionadas através do painel administrativo

Os outros dados devem ser inseridos no formulário de cadastro de imoveis.

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

1. **Tela 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 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.**
  + Após a segunda tentativa, o usuário deve ser comunicado que caso erre 3 vezes, ele ficará sem acesso ao sistema.
  + Ao errar 3 vezes o usuário deve ter o ip do usuário bloqueado por 1min impossibilitando o acesso a plataforma. Informe ao usuário: Acesso bloqueado temporariamente por excesso de erros.

1. **Tela de Gestão de Imóveis**

* Deve ser possível cadastrar e deletar imóveis e suas respectivas imagens, além de possuir uma listagem separada no painel administrativo
* Ao cadastrar e atualizar um imóvel, exibir campos para adicionar as propriedades mencionadas anteteriormente.
  + Os campos relacionados ao local do imóvel devem aparecer dinamicamente. O campo de cidade só deve aparecer após um estado ser selecionado e o campo de bairro só deve aparecer após uma cidade ser selecionada**.**
  + Cada imóvel pode ter mais de uma comodidade, estendendo-se em quantidade limitada apenas pela contagem total de comodidades cadastradas no sistema.
  + Validar os campos obrigatórios antes de permitir o envio do formulário.
    - Os campos de quantidade de quartos e faixa de valores devem só permitir valores numéricos.
    - O campo de valor não pode ser menor ou igual a zero.
* Local no formulário para que o usuário possa selecionar as imagens do imóvel a ser cadastrado.
  + Armazenar as imagens enviadas em um diretório no servidor.
  + Salvar as informações relevantes das imagens no banco de dados(**Nome do arquivo, Tipo de arquivo, Tamanho do arquivo, Data e hora de upload, e associar a(s) imagem(ns) ao imóvel cadastrado**).
  + O campo da galeria de fotos deve ter a opção de abrir um arquivo em um diretório do computador, enviar múltiplas fotos simultaneamente e a opção de arrastar e soltar no campo para enviar a foto.
  + Uma das imagens (apenas) pode ser definida como a capa do imóvel. Caso o imóvel não possua uma capa, a primeira imagem cadastrada para esse imóvel deve possuir o comportamento de capa.
  + Cada imóvel deve ter ao menos 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.

1. **Tela de Gestão de Estados, Cidades, Bairros, Comodidades e Tipos de Imóveis (CRUD)**

* Deve ser possível cadastrar, atualizar e deletar todo e qualquer registro das entidades acima, além de possuirem uma listagem separada no painel administrativo.
* Para cadastrar um Bairro é necessário que já tenha uma cidade cadastrada para fazer a relação e para cadastrar uma Cidade é necessário que tenha um Estado para efetuar a relação.
* As comodidades são os elementos que os imoveis podem possuir como por exemplo churrasqueira, piscina, sauna, playground entre outros.
* O tipo de imóvel é a categoria de imóvel que ele se encontra, exemplo casa, apartamento, sala comercial, terreno entre outras.
* 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.
* Caso um registro de algumas destas entidades seja deletado, todos os registros dependentes por chave secundária devem ser deletados também

1. **Tela de estatísticas**

* Exibir estatísticas referentes aos imóveis cadastrados, são elas:
  + Total de acessos do último mês
  + Acessos do dia
  + Total de imóveis cadastrados
  + Top 3 imóveis mais acessados no último mês
* Todas as estatísticas devem poder ser filtradas pelas propriedades de localização dos imóveis (Estado, cidade e bairro)

1. **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.
* Exibir uma mensagem de sucesso ao realizar o logout: **obrigado por utilizar nosso sistema**.
* Deve haver um logout automático após 20min de inatividade.

1. **Página inicial**

* Exibir a lista de todos imoveis:
  + 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 (somada com suítes)
    - Qt. de dormitórios (somada com suítes)
    - 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 todos os imóveis, ordenando-os pelos imóveis com mais visitas no último mês)
  + Carrossel ‘Alugar’ (Exibe todos os imóveis disponíveis para alugar, ordenando-os pelos mais recentes)
  + Carrossel ‘Comprar’ (Exibe todos os imóveis disponíveis para compra, ordenando-os pelos mais recentes)
* Ao clicar em um imóvel, o usuário deve ser redirecionado à página de listagem individual do imóvel.

1. **Página de pesquisa**

* Ao clicar nos links de ‘Locação’ 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 e Bairro: Selecionar um estado, cidade e bairro. O seletor de cidades só deve ser exibido após selecionar um estado, e o seletor de bairros só deve ser exibido após selecionar uma cidade
  + Vagas: Campo para inserir o número de vagas
  + Dormitórios: Campo para inserir o número de dormitórios/quartos (considerar suítes)
  + Banheiros: Campo para inserir o número de banheiros (considerar suítes)
  + Suítes: Campo para inserir o número de suítes
  + 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.

1. **Página de contato**

* Pode ser acessada pelo botão ‘Contato’ no cabeçalho da aplicação ou pelo botão ‘Envie uma mensagem’ na seção ‘Entre em contato’ das páginas inicial, pesquisa e de listagem individual.
* Deve-se abrir um tela simples com um formulário de contato com os seguintes campos:
  + Nome
  + Email
  + Mensagem
* Após a mensagem ser enviada, ela deve ser cadastrada no Banco de Dados e exibida ao administrador logado em uma tela dedicada

1. **Listagem individual**

* Exibe todas as informações de um imóvel em detalhes
  + Endereço
  + Cidade
  + Estado
  + Bairro
  + Vagas
  + Área
  + Banheiros
  + Quartos
  + Suítes
  + Observações
  + Comodidades disponíveis e indisponíveis
* Exibe todas suas imagens em um carrossel
* Após ser acessada, deve aumentar o número de acessos do imóvel em +1

1. **Layout**

* O layout que será exibido nas páginas Inicial, Pesquisa, Contato, Listagem individual e Login já foi contruído e deve ser utilizado pelos competidores, sendo necessário somente a implementação da listagem dos dados fornecidos pelo backend.
* 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.

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

1. **Tecnologias:**

* Utilizar PHP como linguagem de programação para desenvolver a aplicação.
* Utilizar framework PHP na construção: Laravel ou o CodeIgniter, para agilizar o desenvolvimento e garantir uma estrutura organizada para a aplicação.
* Comente o código para garantir um melhor entendimento de quem for verificar seu algoritmo.
* Utilizar MySQL como sistema de gerenciamento de banco de dados.
* Utilizar o básico de HTML e CSS, apenas para organizar o conteúdo conforme solicitado.

1. **Segurança:**

* Garantir que apenas usuários autenticados tenham acesso à tela de envio de imagens.
* Implementar medidas de segurança para proteger as informações de login dos usuários.
  + Implementar um mecanismo para limitar o número de 3 tentativas de login , e assim bloquear temporariamente uma conta após as 3 tentativas sem sucesso.
    - Ao realizar a 3º tentativa sem sucesso, o usuário deve ser notificado, informando o seguinte: **Acesso bloqueado temporariamente por excessos de erros.** 
      * + Implementar mecanismos de monitoramento e registro de atividades de login, permitindo a detecção e investigação de atividades suspeitas ou tentativas de acesso não autorizado. Guarde em sua base de dados o registro de logs de login, como endereços IP, horários de login, e outros que achar necessário.
* Proteger contra ataques de injeção de SQL.
* Criptografar o transporte. Utilize protocolos criptográficos, como o SSL (Secure Sockets Layer) ou o TLS (Transport Layer Security), para proteger a comunicação entre o cliente (por exemplo, o navegador da web) e o servidor de login. Esses protocolos criptografam os dados durante a transmissão, garantindo que eles não possam ser interceptados ou lidos por terceiros.
* Proteger os dados em trânsito e em repouso. Além da criptografia durante a transmissão, certifique-se de que os dados sensíveis sejam adequadamente protegidos quando estão em repouso, ou seja, armazenados no banco de dados. Isso pode incluir a criptografia dos dados sensíveis, como números de cartão de crédito, bem como a implementação de medidas de segurança para proteger o acesso físico aos servidores e bancos de dados.

1. **Desempenho:**

* Garantir que o sistema seja responsivo e rápido no processamento das operações.
* Otimizar consultas ao banco de dados para reduzir o tempo de resposta.

## 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.

Espera-se que seu esquema seja normalizado.

Segue as tabelas que base para a implementação do sistema, juntamente de seus campos/colunas:

### Tabela "Users":

* id (chave primária)
* username (nome do usuário)
* password (armazenada de forma segura, usando técnicas de criptografia, como hash)
* registration\_date (data e hora de registro do usuário)
* last\_login\_date (data e hora do último login bem-sucedido)

**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)
* suite\_qt (quantidade de suítes)
* parkingspace\_qt (quantidade de vagas)
* value (valor do imóvel)
* comments (observações 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 "Images":**

* id (chave primária)
* filepath (caminho da imagem)
* upload\_date (data de upload da imagem)
* isThumbnail (se a imagem é a capa ou não)
* realtyId (id do imóvel da imagem)

**Tabela "Amenity":**

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

**Tabela "State":**

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

**Tabela "City":**

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

**Tabela "Neighborhood":**

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

Como não criaremos uma tela de cadastro para usuários, utilize a instrução sql adequada para adicionar um usuário padrão (**competidor**) e uma senha padrão(**Modalidade#17**), na tabela “Users”. É com este usuário e senha que poderemos fazer login no sistema e ir para tela de cadastro de imagens.

Ao final, além de um banco de dados configurado, você deve fornecer o diagrama do banco de dados e o script sql.

## Arquivos fornecidos

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

# Ficha de Avaliação Simplificada

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SEÇÃO** | **CRITÉRIO** | **PONTOS DE JULGAMENTO** | **PONTOS DE MEDIÇÃO** | **TOTAL** |
| C1 | Acesso e Autenticação do Usuário |  |  |  |
| C2 | Tela de cadastro de imóveis |  |  |  |
| C3 | Exibição de imagens com as respectivas informações dos imóveis |  |  |  |
| Total | |  |  |  |