**Teste lógico - Imóvel Guide**

**Contextualização do problema**

Quando um anúncio imobiliário fica indisponível, deve-se fornecer aos usuários três opções similares do mesmo tipo de imóvel, na mesma cidade e bairro, utilizando apenas o link do anúncio.

Pede-se para elaborar uma solução para resolver o problema levando em considerações as seguintes **regras**:

- 1 - Em TODO link de anúncio sempre terá a palavra “em” e sempre depois do “em” temos a junção do bairro e cidade respectivamente em um slug único, exemplo: no link anterior o imóvel se encontra no bairro Vila Andrade na Cidade de São Paulo, no link será descrito da seguinte maneira “em-vila-andrade-sao-paulo”.

- 2 - Em TODO link de anúncio sempre terá a palavra “com” e antecedendo essa palavra teremos sempre o tipo de imóvel, podendo ser: apartamento, casa, terreno, sala comercial, casa de condomínio, edifício residencial e etc..., sendo tipos de imóveis com duas ou mais palavras em formato de slug, exemplo: sala-comercial, casa-de-condominio, edificio-residencial.

**Descrição da solução e apresentação dos cenários**

Para resolver esse problema, considerarei a linguagem da vaga: PHP e MySQL.

Cenário 1: Utilizando uma consulta direta ao banco de dados

Primeiro devemos **identificar os links indisponíveis**, para isso criar um mecanismo que detecte automaticamente quando o link do anúncio está indisponível ou não será necessário.

Isso pode ser feito utilizando o monitoramento do **status HTTP**, essa é a maneira mais direta para conseguir isso. Podemos utilizar a função **get\_headers( ),** que retorna um array com os cabeçalhos enviados pelo servidor em resposta a uma requisição HTTP. Se o status de ‘200’ significa que o anúncio está disponível caso ao contrário está indisponível.

Exemplo do código:

<?php

function verificarStatusLink($link) {

$headers = @get\_headers($link);

if ($headers && strpos($headers[0], '200') !== false) {

return true; //Link disponível (status 200 OK)

} else {

return false; //Link indisponível

}

}

?>

Assim que identificar um link indisponível devemos **extrair as informações** relevantes do link, como o tipo do imóvel, bairro e cidade, para isso podemos utilizar **expressões regulares**, que são padrões de buscas de texto, e com base nas regras extrair as informações necessárias

Exemplo do código:

<?php

//Função para extrair informações relevantes

function extrairInformacoesLink($link) {

//Extrair o tipo de imóvel

preg\_match('/imovel\/(.\*?)-com/', $link, $matches);

$tipo\_imovel = str\_replace("-", " ", $matches[1]); //Substituir "-" por " " para o tipo de imóvel

//Extrair o bairro e cidade

preg\_match('/em-(.\*?)\//', $link, $matches);

$bairro\_cidade = str\_replace("-", " ", $matches[1]); // Substituir "-" por " " para o bairro e cidade

//Retornar as informações extraídas

return [

'tipo\_imovel' => ucfirst($tipo\_imovel), // Converter a primeira letra para maiúscula

'bairro\_cidade' => ucwords($bairro\_cidade) // Converter a primeira letra de cada palavra para maiúscula

];

}

//Exemplo

$link\_indisponivel = "https://imovelguide.com.br/imovel/apartamento-com-3-quartos-a-venda-110-m2-em-vila-andrade-sao-paulo/1317073";

$info\_link = extrairInformacoesLink($link\_indisponivel);

echo "Tipo de imóvel: " . $info\_link['tipo\_imovel'] . "<br>";

echo "Bairro/Cidade: " . $info\_link['bairro\_cidade'];

?>

Saída:

Tipo de imóvel: Apartamento

Bairro/Cidade: Vila Andrade Sao Paulo

Extraídas as informações necessárias faremos uma \*\*consulta no banco de dados\*\* para encontrar outros imóveis semelhantes e apresentar ao usuário

Exemplo do código:

<?php

//Extrair informações do link

$link\_indisponivel = "https://imovelguide.com.br/imovel/apartamento-com-3-quartos-a-venda-110-m2-em-vila-andrade-sao-paulo/1317073";

$info\_link = extrairInformacoesLink($link\_indisponivel);

//Consulta SQL

$sql = "SELECT \* FROM imoveis WHERE tipo\_imovel = '{$info\_link['tipo\_imovel']}' AND bairro\_cidade = '{$info\_link['bairro\_cidade']}' LIMIT 3";

//Executar a consulta

$result = $conn->query($sql);

//Verificar se há resultados

if ($result->num\_rows > 0) {

//Exibir os resultados

echo "Opções similares para o imóvel indisponível:<br>";

while($row = $result->fetch\_assoc()) {

echo "Tipo de imóvel: " . $row["tipo\_imovel"] . "<br>";

echo "Bairro/Cidade: " . $row["bairro\_cidade"] . "<br>";

echo "<br>";

}

} else {

echo "Nenhuma opção similar encontrada para o imóvel indisponível.";

}

?>

Com isso já temos a base para que possamos achar imóveis semelhantes caso o anúncio esteja fora do ar.

**Cenário 2: Utilizando o framework PHP, Laravel**

Para este cenário utilizaremos a mesma lógica, porém usando o**Laravel** ele simplifica muito o processo de consulta ao banco de dados e apresentação dos resultados.

Tanto para identificar os links indisponíveis quanto para a extração de informações dos links continuaremos a usar a mesma abordagem.

Já para a consulta do banco de dados temos junto ao Laravel o **Eloquent ORM** para criar as consultas de forma mais fácil e intuitiva. Podendo criar o modelo da tabela e utilizar os métodos presentes

Exemplo do código:

<?php

//Consulta ao banco de dados usando Eloquent

$imoveisSimilares = Imovel::where('tipo\_imovel', $tipoImovel)

->where('bairro\_cidade', $bairroCidade)

->take(3)

->get();

return view('imoveis\_similares', ['imoveisSimilares' => $imoveisSimilares]);

}

}

Para a apresentação dos dados usaremos o **Blade**, é um mecanismo de template que vem junto ao Laravel, podemos criar uma view para apresentar as opções similares aos usuários.

Exemplo do código:

<!DOCTYPE html>

<html>

<head>

<title>Imóveis Similares</title>

</head>

<body>

<h1>Opções similares para o imóvel indisponível:</h1>

@if ($imoveisSimilares->count() > 0)

<ul>

@foreach ($imoveisSimilares as $imovel)

<li>

<strong>Tipo de imóvel:</strong> {{ $imovel->tipo\_imovel }}<br>

<strong>Bairro/Cidade:</strong> {{ $imovel->bairro\_cidade }}

</li>

@endforeach

</ul>

@else

<p>Nenhuma opção similar encontrada para o imóvel indisponível.</p>

@endif

</body>

</html>