

Sistema para Gerenciamento de Aluguel de Automóveis

Flávio Mota Gomes¹, Rafael Antunes Vieira¹

¹Instituto de Matemática e Computação – Universidade Federal de Itajubá (UNIFEI)
Itajubá – MG – Brasil

flaviomotagomes@gmail.com, rafaelantunesvieira@outlook.com

Abstract. *This paper presents the configuration of a database server to accommodate the database of a car rental management system. The creation of the server is presented; configuring it to allow external access via the internet, which includes the Firewall configuration on the server device; the operation of the NAT protocol. The entity-relational model of the database is also presented. An application was also developed using HTML, CSS, JavaScript and PHP. This application is responsible for the functionalities of the rental system, which includes maintaining employee and customer records, as well as all vehicle and rental management.*

Resumo. *Este trabalho apresenta a configuração de um servidor de banco de dados para acomodar o banco de dados de um sistema para gerenciamento de aluguel de automóveis. Apresenta-se a criação do servidor; a configuração do mesmo para permissão de acessos externos via internet, o que inclui a configuração de Firewall no dispositivo do servidor; o funcionamento do protocolo NAT. Apresentam-se também o modelo entidade-relacional do banco de dados. Desenvolveu-se também uma aplicação utilizando HTML, CSS, JavaScript e PHP. Essa aplicação é responsável pelas funcionalidades do sistema de locação, o que inclui manutenção dos cadastros dos funcionários e clientes, bem como todo o gerenciamento de veículos e de suas locações.*

1. Introdução

Este relatório descreve a montagem de um ambiente de redes para confecção de um sistema de gerenciamento de aluguel de carros. A ideia central deste desenvolvimento consiste em uma aplicação web desenvolvida a partir de HTML, CSS, JavaScript e PHP, que permite o controle de alocação de recursos em uma empresa de aluguel de veículos.

O sistema permite login de funcionários para administração dos veículos - o que inclui criá-los, editá-los, tirá-los de circulação temporariamente para irem à oficina - além de permitir cadastro ao público para reservar os veículos desejados no período desejado. Para isso, criou-se um servidor de banco de dados e liberou-se o acesso desse banco na internet para permitir a inserção desses pedidos por qualquer pessoa que acesse a plataforma.

O objetivo desta atividade foi conhecer diferentes ambientes de redes de computadores, explorando de forma prática os conceitos vistos em sala de aula.

Este relatório está estruturado em: Instalação de aplicações, onde se descreve o processo de instalação do Apache, do PostgreSQL e do PHP; Configuração do servidor

de Banco de Dados, onde se discorre sobre o servidor, o modelo relacional e o mapeamento do modelo relacional para as tabelas do banco; Conexão entre aplicação e banco; Interfaces, onde são apresentadas características da aplicação e algumas de suas telas; Segurança, onde são feitas algumas considerações sobre segurança na aplicação desenvolvida. Por fim, são apresentadas as considerações finais.

2. Instalação de aplicações

Nos tópicos a seguir, são apresentadas algumas considerações sobre as aplicações necessárias para configuração do sistema. Neles, são descritas informações sobre a instalação e a configuração dessas aplicações, a saber: Apache, PostgreSQL e PHP.

2.1. Instalação do Apache

Segundo [Clemente 2019],

O Apache é um software de código aberto de propriedade da Apache Software Foundation (ASF) que garante desempenho, estabilidade e segurança para um servidor web. Ele se tornou muito popular entre os programadores devido à sua modularidade e constante atualização pela comunidade, o que contribui ativamente para a alta capacidade dessa ferramenta.

Criado em 1995, esse software, junto de outros componentes como PostgreSQL e PHP, tem forte atuação no fornecimento de conteúdos na internet. O Apache se comunica na rede por meio do protocolo TCP/IP ou HTTP/S [Clemente 2019].

A atuação do Apache funciona de maneira simples. Basicamente, sua execução consiste na obtenção do endereço IP - que todos os sites possuem - de um domínio na internet a partir do servidor DNS (*Domain Name System*). O navegador realiza a solicitação da URL e o servidor se responsabiliza por devolver ao navegador as páginas solicitadas por ele [Clemente 2019].

A instalação do Apache parte, inicialmente, do download do instalador no site <https://httpd.apache.org/download.cgi>. Clica-se no arquivo de instalação e prossegue-se, clicando em "Next", até a seção "Server Information". Nessa sessão, devem ser preenchidos os dados sobre o servidor e configurada a porta padrão (por padrão, a 80). Segue-se. Na tela de "Setup type", deve-se manter a opção "Typical". Na tela que se segue, há a opção de definir a pasta de instalação. Seguindo com o instalador, a aplicação será instalada no computador.

Para verificar se a instalação teve sucesso, basta abrir algum navegador e digitar o endereço do localhost. Uma página com a informação "It works!" deve aparecer.

2.2. Instalação do PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional. Baseado no Postgres versão 4.21, o PostgreSQL suporta uma grande parte do padrão SQL, além de oferecer características modernas. Consiste em um processo de servidor que lê e grava os arquivos de banco de dados reais, e um conjunto de programas cliente que se comunicam com o servidor [Dionísio 2015].

Cliente e servidor podem estar em diferentes hosts, e a comunicação entre eles é feita por meio de uma conexão de rede TCP/IP. O processo do servidor principal estará sempre correndo, esperando por conexões de cliente [Dionísio 2015].

Para instalar o SGBD PostgreSQL, é necessário partir pelo acesso ao site <https://www.postgresql.org/download/>, selecionar o sistema operacional correspondente. Assim, a página realizará o download do instalador. Dentro do instalador, basta avançar nas opções. É possível selecionar a localização do banco. O instalador solicitará a configuração para criação do serviço de banco de dados. Já há um padrão configurado, e ele pode ser mantido ou alterado. É necessário criar uma senha para o usuário. Importante frisar que essa conta e senha não é a do super usuário do banco, mas uma conta responsável por dar o start. A seguir, solicita-se a configuração inicial do banco. Todos os valores podem ser mantidos como padrão, à exceção do valor de Encoding, no qual deve ser colocada a opção UTF-8, que confere um padrão de símbolos mais completos. Preenche-se com a senha criada anteriormente e mantém-se o padrão até o final da instalação.

2.3. Instalação do PHP

O PHP consiste em uma linguagem de programação em código aberto, criada para o desenvolvimento web [Dev a].

A definição contida no site da aplicação é a seguinte [PHP]:

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

O PHP caracteriza-se pela sua simplicidade e vasta oferta de recursos avançados. O código PHP é executado no servidor, gerando o HTML que é enviado para o navegador que, apesar de exibir os resultados da execução do script, não conhece o código fonte [PHP].

A instalação do PHP é feita a partir de um instalador disponível no site <https://www.php.net/downloads.php>. É necessário acessar o site e selecionar o instalador que é compatível ao computador e seu sistema operacional. Após realizado o download, clica-se no arquivo de instalação e avança-se. Escolhe-se a pasta onde será instalado o PHP e, em seguida, informa-se qual tipo de servidor web está rodando no servidor. Nesse caso, é o Apache instalado. A seguir, deve-se informar ao instalador do PHP o local onde se encontra o arquivo de configuração do servidor web Apache. Ele instalará na pasta "conf", contida no diretório onde o Apache foi instalado. Por fim, o instalador pergunta sobre a instalação de extensões. Deve-se expandir esse menu e clicar no item "PostgreSQL" para instalar o suporte ao Banco de Dados PostgreSQL, que é o SGBD utilizado na aplicação desenvolvida. Segue-se com a instalação. O instalador questionará se deve-se configurar o Apache e, para esta pergunta, a resposta deve ser sim. Desse modo, o instalador adicionará as linhas de configuração no arquivo de configuração do Apache necessárias para que o PHP funcione. Finaliza-se aqui a instalação [Machado].

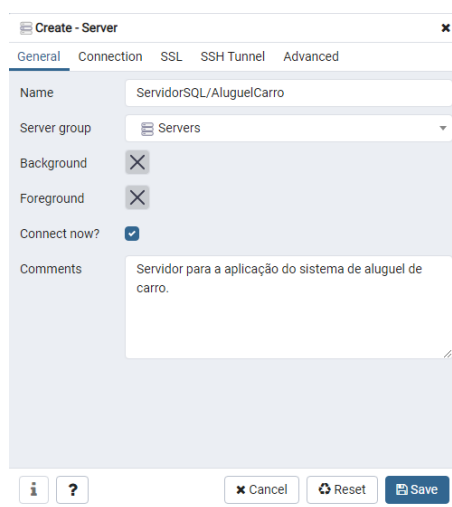
3. Configuração do servidor de Banco de Dados

O primeiro passo para construção do ambiente é a configuração do servidor de banco de dados. Para este trabalho, o SGBD PostgreSQL foi utilizado. No subtópico a seguir, descreve-se a configuração do servidor que hospedará o banco.

3.1. Servidor

Inicialmente, é preciso acessar um arquivo do PostgreSQL para configurar o modo de autenticação entre o servidor e o cliente. O arquivo que será modificado é o `pg_hba.conf`. Recomenda-se a leitura da documentação do PostgreSQL para que a configuração seja feita de acordo com a versão instalada. Lembrando que recomendamos sempre a versão mais atualizada do SGBD para que a segurança do sistema esteja assegurada. O link da documentação para alterar o arquivo `pg_hba.conf` é: <https://www.postgresql.org/docs/9.1/auth-pg-hba-conf.html>.

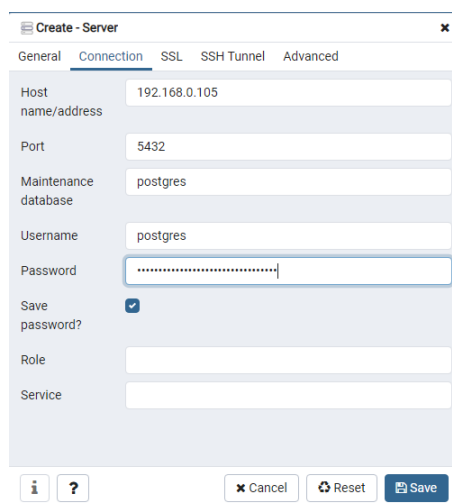
Após a configuração do arquivo `pg_hba.conf`, cria-se um servidor utilizando a interface do PostgreSQL. Veja na Figura 1.



The screenshot shows the 'Create - Server' dialog box with the 'General' tab selected. The 'Name' field contains 'ServidorSQL/AluguelCarro'. The 'Server group' dropdown is set to 'Servers'. The 'Background' and 'Foreground' checkboxes are both unchecked. The 'Connect now?' checkbox is checked. The 'Comments' field contains 'Servidor para a aplicação do sistema de aluguel de carro.' At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save'.

Figura 1. Criação do servidor na Interface do PostgreSQL

A seguir, na Figura 2, mostra-se a configuração de conexão do servidor. Atente-se ao número da porta.



The screenshot shows the 'Create - Server' dialog box with the 'Connection' tab selected. The 'Host name/address' field contains '192.168.0.105'. The 'Port' field contains '5432'. The 'Maintenance database' field contains 'postgres'. The 'Username' field contains 'postgres'. The 'Password' field is masked with dots. The 'Save password?' checkbox is checked. The 'Role' and 'Service' fields are empty. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save'.

Figura 2. Configuração de conexão do servidor de Banco de Dados

Após a criação do servidor, é necessário permitir conexões externas a ele a partir da internet. Para isso, dado que foi utilizado um sistema Windows, fez-se necessário permitir conexões no Firewall do sistema. Essa configuração dá-se da seguinte forma:

- "Painel de Controle";
- "Todos os itens do Painel de Controle";
- "Windows Defender Firewall";
- "Opções Avançadas";
- No menu à esquerda, entre em "Regras de Entrada";
- No menu à direita, entre em "Nova Regra";
- Assinale a opção "Porta" e clique em "Avançar";
- Selecione o protocolo "TCP", que é o protocolo utilizado pelo PostgreSQL; abaixo, preencha com o número da porta onde está criado o servidor (veja Figura 2);
- Assinale "Permitir a conexão" e avance;
- Dê um nome à regra e clique em concluir.

Após a execução dos passos descritos acima, a tela do Firewall ficará como na Figura 3.

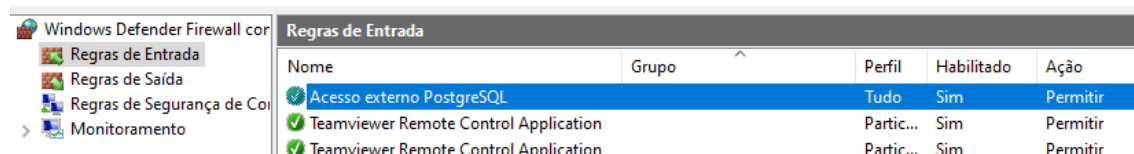


Figura 3. Firewall após configuração.

O próximo passo é configurar o roteador para permitir acessos externos. Para isso, acessa-se a página de configuração do seu roteador (siga as instruções da fabricante). Procure pela opção de "Servidor Virtual", clique em "Adicionar". Deixe a opção "Servidor Comum" como padrão. Em "Porta Externa", coloque a porta que será aberta ao mundo externo, a sua escolha. Em "Porta Interna", coloque a porta onde está o banco. No campo de endereço IP, coloque o IP da máquina onde está o servidor. Dentro da rede interna onde está conectado o servidor, existe uma tradução de IP pelo protocolo NAT. O protocolo NAT converte o IP de origem (interno) no endereço IP verdadeiro da rede (público), conforme ilustra a Figura 4. Marque o protocolo TCP e confirme. Veja na Figura 5. Após isso, a porta estará aberta na internet e será possível conexões externas ao banco por meio do IP público. A própria rede interna utilizará do IP interno para direcionar os pacotes ao servidor.

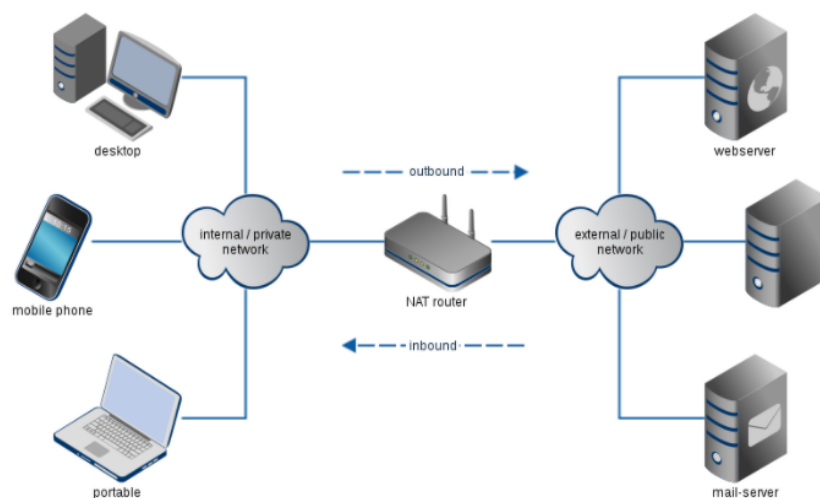


Figura 4. Funcionamento do protocolo NAT.

Servidores Virtuais



<input type="checkbox"/>	Servidor Comum	Porta Externa	Porta Interna	Endereço IP	Protocolo	Editar
<input type="checkbox"/>	Selecionar ▼	1998	5432	192.168.0.105	TCP ▼	Cancelar Salvar

Figura 5. Roteador após configuração.

3.2. Modelo Entidade-Relacional

A Figura 6 apresenta o Modelo Entidade-Relacional utilizado neste trabalho. Ele apresenta as entidades "automóvel", "locação" e "usuário". A entidade "usuário" possui especialização em "funcionário" e "cliente". Essa especialização é do tipo "d,t". Isso significa que um usuário desta aplicação é, obrigatoriamente, cliente ou funcionário, não havendo margem para que não seja de um desses tipos. Os relacionamentos demonstram a capacidade dos usuários do tipo "funcionário" em gerenciar os automóveis e dos clientes realizarem a locação dos veículos, bem como a capacidade dos usuários do tipo "cliente" em efetuarem reservas de locações de veículos.

Sistema de gerenciamento de aluguel de carros
Banco de dados relacional

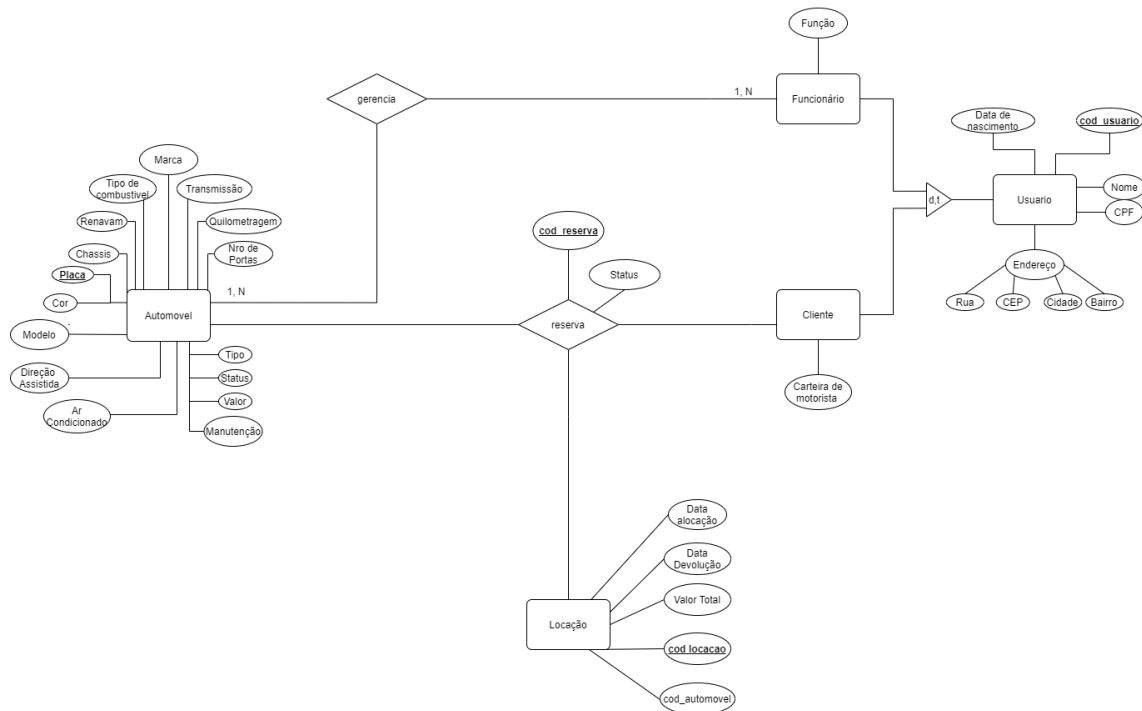


Figura 6. Modelo Entidade-Relacionamento

3.3. Tabelas do Banco

As tabelas do banco estão descritas a seguir. Elas foram construídas a partir do mapeamento do Modelo Entidade-Relacionamento descrito no tópico anterior. Os atributos assinalados com * indicam obrigatoriedade. Os atributos escritos em letras maiúsculas são as chaves primárias das tabelas. Esse mapeamento foi implementado utilizando o PostgreSQL. O banco de dados gerado foi utilizado no desenvolvimento da aplicação.

- Automovel(PLACA, cor*, Chassis*, Modelo*, Direção_Assistida*, Ar_condicionado*, Manutenção*, Nro_de_Porta*, Quilometragem*, Transmissão*, Marca*, Tipo_de_combustivel*, Renavam*,Tipo*,Status*,Valor*)
- Locacao(COD_LOCACAO,cod_automovel, Valor_total*,Data_devolução*,Data_alocação*)
- Usuario(COD_USUARIO, Data_nascimento* ,Nome* ,CPF*, Rua*,CEP*, Cidade*, Bairro*)
- Funcionario(COD_FUNCIONARIO, Função*)
COD_FUNCIONARIO referencia COD_USUARIO
- Cliente(COD_CLIENTE, Carteira_de_motorista*)
COD_CLIENTE referencia usuario(COD_USUARIO)
- Reserva(COD_RESERVA, COD_LOCACAO, COD_CLIENTE, COD_AUTOMOVEL,status)
COD_CLIENTE referencia cliente(COD_USUARIO)
COD_LOCACAO referencia locacao(COD_LOCACAO)
COD_AUTOMOVEL referencia automovel(PLACA)

3.4. Aplicações utilizadas no desenvolvimento

Para construir a aplicação, foi feito uso das seguintes aplicações: para a escrita de código, utilizou-se o VSCode; para o banco de dados, utilizou-se o PgAdmin e o PostgreSQL. Adicionalmente, este trabalho foi hospedado no GitHub, utilizando o seguinte link de acesso: https://github.com/Rav98/Sistema_aluguel_de_carros.

4. Conexão entre aplicação e banco

Para conectar a aplicação com o banco de dados PostgreSQL é utilizado um drive de conexão. O drive de conexão é responsável por fornecer a conexão entre o banco de dados e a aplicação, implementando um protocolo para transferir a consulta e o resultado entre cliente e banco de dados.

4.1. Configuração da conexão

Como a parte de *back-end* da aplicação é feita pela linguagem PHP, o drive de conexão usado é PDO, que significa PHP Data Object. O PDO é apenas orientado para objetos e suporta uma variedade de tipos de bancos de dados diferentes que usam PHP, como MySQL, MSSQL, Informix e o PostgreSQL.

A conexão é simples e é feita em poucas linhas de código. Primeiramente, é preciso passar os detalhes do servidor, como IP Host, a porta de conexão, usuário e senha que foram configurados na criação do servidor, como foi dito no tópico 2 deste artigo. Após reunir todos os parâmetros, é preciso colocá-los dentro da função PDO como mostra na Figura 7 e pronto, a aplicação está conectada com o banco de dados.

```
<?php
$datasource = "pgsql:host= IP DO SERVIDOR ; port= PORTA DO SERVIDOR; dbname= NOME DO BANCO DE DADOS";
$user = "USUARIO";
$pass = "SENHA";
$db = new PDO($datasource, $user, $pass);
?>
```

Figura 7. Drive de conexão com o banco.

5. Interfaces

A aplicação desenvolvida fornece 3 tipos de acessos. Um deles é o acesso "standart", ou seja, o acesso do usuário não logado ao sistema. Nessa interface, cuja página inicial é ilustrada na Figura 8, apresentam-se os menus da aplicação e descrições sobre o sistema e suas funcionalidades.



Sistema de Gerenciamento de Aluguel de Carros

PRINCIPAL LOGOUT ECONÔMICOS UTILITÁRIOS SUV LUXO CONTATO

Bem-vindo(a) ao sistema de aluguel de automoveis

Este é um sistema desenvolvido para facilitar o gerenciamento de aluguel de veículos.

Figura 8. Tela Inicial da aplicação.

Os menus apresentados na Figura 8 repetem-se quando o usuário faz login no sistema. A página principal, por sua vez, altera-se de acordo com o tipo de usuário logado na aplicação. Usuários do tipo cliente têm a possibilidade de alterar seus dados, alugar carro e alterar suas reservas. O aluguel de carros também poderá ser feito através das páginas que categorizam os veículos. Isso será demonstrado a seguir. Note também que quando um usuário está logado no sistema, o menu "Login" torna-se "Logout". Clicar em "Logout" desvincula o usuário da aplicação e a página inicial torna-se a página "standard" novamente. Veja essas informações na Figura 9.



Figura 9. Algumas funcionalidades do usuário cliente.

A Figura 10 apresenta a página inicial do usuário funcionário. Nela, constam-se as funcionalidades que podem ser executadas pelo usuário funcionário. Essas funcionalidades são específicas do papel do funcionário, e incluem a manutenção de veículos (incluindo enviá-los ao mecânico e retirá-los de lá), a criação de novos funcionários, manutenção de dados do cliente e de seus próprios dados. Um usuário funcionário é o único capaz de cadastrar novos usuários do tipo funcionário no sistema, mas é incapaz de editar os dados de funcionários que não sejam ele próprio. Em um uso comercial desta aplicação, naturalmente já constaria um usuário funcionário cadastrado para que esse seja o responsável por criar novos funcionários e abastecer o banco de dados com os veículos da empresa utilizadora desta aplicação.



Figura 10. Algumas funcionalidades do usuário funcionário.

No que tange ao menu da aplicação, afirmou-se anteriormente que a página principal varia conforme o tipo de usuário logado. A implementação disso baseou-se na questão de Sessões, em PHP, conforme mostra abaixo na Figura 11.

Com base na sessão, montou-se o menu da forma indicada na Figura 12: com base no tipo de usuário logado na aplicação, direciona-se para a página correspondente. Dessa forma, um usuário "cliente", que possui menos privilégios que um usuário "funcionário", é incapaz de acessar as funcionalidades que não lhe dizem respeito.

```

3  <?php
4  // Inicia a sessão.
5  session_start();
6  if (isset($_SESSION['cpf'])) {
7      $cpf = $_SESSION['cpf'];
8      $tipo = $_SESSION['tipo'];
9  }
10 else{
11     $tipo = '';
12 }
13
14 ?>

```

Figura 11. Sessão em PHP.

```

<div id="menu">
<ul>
<?php
    if($tipo == 'F'){
        echo "<li><a href='../indexFuncionario.php' accesskey='1' title='Principal1'>Principal</a></li>";
    }
    if ($tipo == 'C'){
        echo "<li><a href='../indexCliente.php' accesskey='1' title='Principal2'>Principal</a></li>";
    }
    if ($tipo == ''){
        echo "<li><a href='../index.php' accesskey='1' title='Principal3'>Principal</a></li>";
    }
?>
<?php
    if (isset($_SESSION['cpf'])) {
        print "<li><a href='../login/logout.php' accesskey='2' title=''>Logout</a></li>";
    }
    else {
        print "<li><a href='../login/login.php' accesskey='2' title=''>Login</a></li>";
    }
?>
<li class="current_page_item"><a href="economicos.php" accesskey="3" title="">Econômicos</a></li>
<li><a href="utilitarios.php" accesskey="4" title="">Utilitários</a></li>
<li><a href="suv.php" accesskey="5" title="">SUV</a></li>
<li><a href="luxo.php" accesskey="6" title="">Luxo</a></li>
<li><a href="../contato.php" accesskey="7" title="">Contato</a></li>
</ul>

```

Figura 12. Sessão em PHP.

A questão das sessões também é utilizada nas páginas relativas às categorias dos veículos. A Figura 13 ilustra essa distinção. Nela, note que o veículo é exibido para ambos os usuários, mas com funcionalidades distintas. Nesta figura, apresenta-se 3 visualizações de um mesmo veículo. Na visualização da esquerda, tem-se a visão desse veículo pelo funcionário. Nela, apresenta-se um botão que permite que o funcionário altere dados desse veículo. Na visualização do meio, apresenta-se a visão do usuário "cliente". Ele tem um botão que lhe permite alugar o veículo. Já na visualização da direita, tem-se a tela "standart", ou seja, o que é mostrado quando não há usuário logado no sistema. Essa visão é interessante para um sistema desse tipo porque permite que potenciais clientes, que ainda não têm login no sistema, visualizem os carros que existem para ser alugados. Isso serve para chamar a atenção desse visualizador da página e torná-lo um potencial cliente.

Visão como funcionário	Visão como cliente	Visão sem efetuar login
		
MODELO FIAT TORO PLACA: Ild9819 COR: VERMELHO DIREÇÃO ASSISTIDA: SIM AR CONDICIONADO: SIM NUMERO DE PORTA: 5 COMBUSTIVEL: DIESEL QUILOMETRAGEM: 12000 TRANSMISSÃO: AUTOMÁTICA	MODELO FIAT TORO PLACA: Ild9819 COR: VERMELHO DIREÇÃO ASSISTIDA: SIM AR CONDICIONADO: SIM NUMERO DE PORTA: 5 COMBUSTIVEL: DIESEL QUILOMETRAGEM: 12000 TRANSMISSÃO: AUTOMÁTICA	MODELO FIAT TORO PLACA: Ild9819 COR: VERMELHO DIREÇÃO ASSISTIDA: SIM AR CONDICIONADO: SIM NUMERO DE PORTA: 5 COMBUSTIVEL: DIESEL QUILOMETRAGEM: 12000 TRANSMISSÃO: AUTOMÁTICA
<input type="button" value="Editar"/>	<input type="button" value="Alugar"/>	

Figura 13. Diferentes visualizações de veículo de acordo com o usuário logado.

5.1. Alugando veículos

Um usuário do tipo "cliente" logado no sistema pode solicitar o aluguel de um determinado veículo. A solicitação pode partir de formas diferentes. Uma dessas formas é seguindo diretamente às páginas que categorizam os carros, presentes no menu. A segunda forma é acessando a página principal e clicando no botão "Alugar automóvel", conforme foi ilustrado na Figura 9. Nesse segundo caso, há um redirecionamento às páginas de categorias. Conforme ilustrou a visualização central da Figura 13. Ao clicar em "Alugar", o sistema redireciona para a página ilustrada na Figura 14, onde o cliente seleciona a data de início e de fim do aluguel do automóvel. A placa do carro já vem selecionada a partir do carro escolhido.



Sistema de Gerenciamento de Aluguel de Carros

ALOCAR CARRO

Placa:

Data Locação: (A-M-D)

Data Devolução: (A-M-D)

Figura 14. Menu aluguel de carros.

Após o cliente selecionar as datas desejadas, o sistema puxa os dados passados de carro, data inicial de locação e data final de locação. Além disso, também seleciona os dados do usuário logado na sessão, que selecionou o carro desejado para locação. A seguir, realiza-se uma checagem de disponibilidade do carro nas datas desejadas. Para isso, o sistema tenta selecionar no banco de dados da aplicação os códigos de locação referentes às datas de início e de fim. Se o retorno é igual a um código, isso indica que o veículo selecionado já está locado no período total ou parcial desejado. Nesse caso, o sistema indica ao usuário cliente a necessidade de trocar as datas de locação ou trocar o veículo escolhido. A seguir, estão os códigos SQL que demonstram essa checagem.

```
SELECT cod_locacao FROM locacao WHERE cod_automovel =  
$placaAutomovel AND data_devolucao BETWEEN $dataLocacao AND  
$dataDevolucao;
```

```
SELECT cod_locacao FROM locacao WHERE cod_automovel =  
$placaAutomovel  
AND data_alocacao BETWEEN $dataLocacao AND  
$dataDevolucao;
```

Na Figura 15, pode-se visualizar como é feito o cálculo do valor da locação. Cada veículo possui, em sua tabela, um campo com o valor da diária (que pode ser alterado por usuários do tipo "funcionário"). O sistema calcula a diferença de tempo entre a data de devolução e a data de locação; em seguida, por meio de um cálculo matemático, transforma essa diferença em dias inteiros; por fim, multiplica o valor da diária do carro (recebido, nesta página, via SQL) pela quantidade de dias de locação. Realizados os cálculos, o sistema armazena em duas tabelas: na tabela locação, ficam armazenados os dados relativos ao automóvel: código do automóvel, data de locação, data de devolução e valor total gasto pelo cliente. Na tabela reserva, ficam armazenados o código do automóvel, o código do cliente, o código da locação e o status. O status é uma variável booleana que, em todas as execuções, é setada como "true". A variável status torna-se "false" se houver um cancelamento dessa reserva, atendendo a uma regra de negócio que consiste em não apagar dados do banco, apenas torná-los "inúteis" se necessário for. Após as inserções na tabela, está concretizada a reserva de locação. Esses valores são visíveis ao usuário futuramente para eventual cancelamento ou alteração até a data de início da locação. Após esse período, continuam visíveis a ele, mas não são manipuláveis mais.

```
#Calculando quantidade de dias de aluguel:  
$diferenca = strtotime($dataDevolucao) - strtotime($dataLocacao);  
$dias = floor($diferenca / (60 * 60 * 24));  
  
#Calculando valor total do aluguel nesses dias:  
$aluguel = $valor * $dias;
```

Figura 15. Cálculo de diárias.

6. Segurança

Nesta seção são feitas algumas considerações sobre segurança na aplicação. Primeiro, trata-se da questão do SQL Injection, uma invasão que pode acontecer no momento do login de usuários. Em seguida, aborda-se também a questão das funcionalidades permitidas a cada tipo de usuário.

6.1. Evitando SQL Injection no login dos usuários

O SQL Injection consiste em uma técnica de injeção de código com capacidade de acessar ao banco de dados, o que representa um risco [W3S]. Ocorre, geralmente, durante a entrada de um usuário. Quando solicita-se ao usuário a entrada no sistema, como um nome/senha, o usuário fornece, em vez disso, uma instrução em SQL que será executada no banco de dados sem o conhecimento do administrador do banco.

A principal forma de evitar o SQL Injection se dá a partir de uma validação dos dados digitados pelo usuário. Há duas abordagens, segundo [Dev b].

A primeira abordagem consiste em rejeitar dados que são conhecidamente inválidos. O problema dessa abordagem dá-se no fato de que o desenvolvedor não necessariamente terá conhecimento acerca dos que sejam dados inválidos. É difícil conhecê-los e entender como devem ser modificados, visto que novas formas de “dados inválidos” são sempre descobertos.

A segunda é aceitar somente os dados que são conhecidamente válidos: É considerada a melhor estratégia usada, pois o programador sabe como deve ser a entrada e assim, é possível gerar estratégias para que a aplicação consiga aceitar somente a entrada adequada.

Na hora de realizar o login, uma aplicação geralmente realiza a consulta do usuário e da senha digitados pelo usuário. Essa consulta é realizada no banco de dados e, portanto, utiliza um código SQL para isso. Imaginando uma consulta do gênero cujo SQL fosse: “SELECT * FROM usuario WHERE cpf=’\$cpfUsuario’ AND senha=’\$senhaUsuario’”, uma entrada que deixasse o campo de login em branco e digitasse no campo senha os caracteres: ‘or’=’, conseguiria logar na aplicação. Isso aconteceria porque o jogo de aspas simples validaria o código SQL de select. Esse, ficaria da seguinte forma: “SELECT * FROM usuario WHERE cpf=’ ’ AND senha = ’or’=’ ’”. Usuário e senha são campos em branco, e campo em branco é igual a campo em branco. Logo, valida-se a query e uma invasão é realizada.

Com o intuito de impedir que isso aconteça, a aplicação implementou uma verificação de senha sem utilizar diretamente a variável do programa na query SQL, conforme ilustra a Figura 16. Dessa forma, ao preparar a variável primeiro para depois executá-la na query, uma invasão como a descrita anteriormente não ocorrerá nesses termos.

```
18 $query = "SELECT * FROM usuario WHERE cpf=? AND senha=?";
19 $stm = $db->prepare($query);
20 $stm->bindParam(1, $usuario);
21 $stm->bindParam(2, $senha);
22 $stm->execute();
```

Figura 16. Evitando o SQL Injection.

6.2. Permissões de usuário

Outra questão importante para a segurança da aplicação é que usuários de um tipo não interfiram em funções de usuários de outro tipo. Seria um problema de segurança se, por exemplo, um usuário do tipo cliente tivesse acesso às funcionalidades de um usuário funcionário.

A aplicação trata da resolução desse problema a partir já do login do usuário. Um usuário da aplicação só pode ser cadastrado por um funcionário da locadora. Esse funcionário é o responsável por categorizar esse novo usuário como cliente ou outro funcionário. Ao realizar um login, o sistema realiza uma busca naqueles dados informados e identifica o tipo de usuário que acabou de realizar login. Após essa varredura, salva-se na sessão do usuário logado um atributo que realiza essa identificação. É a partir disso que o sistema direciona o usuário às páginas cujo acesso dele é permitido. Um usuário cliente, por exemplo, não terá acesso à página que realiza manutenções no cadastro de veículos, já que o sistema não redireciona, em nenhum momento, a ela.

Além disso, as páginas possuem layouts diferentes conforme o tipo do usuário. Um exemplo disso foi o abordado na Figura 13 da seção 4.1, que mostrou como um funcionário, um cliente e um usuário não logado visualizam um carro cadastrado no sistema. O funcionário tem o botão para editar disponível; o cliente, o alugar; um usuário não logado, nenhum. Além desse exemplo, temos também a página principal com suas três versões.

6.3. Sistema operacional do servidor

Para manter uma boa segurança no sistema operacional que gerencia os recursos para o servidor, foram tomadas algumas ações:

- Instalação de um Antivírus: o antivírus é uma ferramenta que ajuda a evitar que possíveis ataques de malwares e spyware aconteçam no sistema operacional em que o servidor está rodando. Isso aumenta a segurança e melhora a confiabilidade e estabilidade do servidor. No caso do servidor utilizado nesta aplicação, foi instalado o antivírus Kaspersky;
- Atualização do sistema operacional: de tempos em tempos, a empresa por trás do sistema operacional lança atualizações que corrigem falhas de segurança do sistema. É sempre bom mantê-lo atualizado para que o sistema seja mais seguro contra ataques.
- Evitar a instalação de programas que não agregam nada ao funcionamento do servidor: para evitar perdas de desempenho e de segurança no sistema operacional onde o servidor está funcionando, deve-se evitar que sejam instalados programas que não agregam melhorias ao servidor. Além dos programas roubarem desempenho do servidor, pois rodam juntos no sistema, os programas podem ter algum problema e travar o sistema operacional da máquina, fazendo com que o servidor pare de funcionar.

7. Considerações Finais

Considera-se que o objetivo deste estudo foi atingido. Nos tópicos anteriores, foi descrito todo o processo envolvido na confecção do sistema para gerenciamento de aluguel de

carros, desde a criação e configuração de um servidor de banco de dados até a completa criação de uma aplicação utilizando HTML, CSS, JavaScript e PHP.

A aplicação, cujas funcionalidades foram aqui apresentadas, mostrou-se capaz de realizar todas as tarefas que se propôs a fazer: mantém os cadastros de veículos, funcionários e clientes e é capaz de realizar todas as etapas de gerenciamento do aluguel dos veículos. Tudo isso é feito de maneira satisfatória, ou seja, a aplicação mantém conexão plena com o servidor configurado.

Por sua vez, o servidor também atende às suas obrigações, mantendo armazenados os dados relativos à aplicação e sendo responsivo às solicitações da mesma.

Como trabalhos futuros, sugere-se o aprimoramento de questões de design das páginas, incluindo adaptações às diretrizes de design de empresas que eventualmente utilizem desse sistema.

Dado o cenário atual da internet, com seu enorme fluxo de usuários, e a busca dessas pessoas por serviços que facilitam a vida e poupam tempo, encerra-se este trabalho ressaltando a aplicabilidade real desta aplicação para empresas do ramo de aluguel de automóveis, que, de porte de uma ferramenta como essa, tendem a maximizar a base de clientes e, por consequência, de lucros.

Referências

Linguagem php. <https://www.devmedia.com.br/guia/linguagem-php/38780>. Acesso em: 22 out. 2020.

Linguagem php. https://www.php.net/manual/pt_BR/intro-what-is.php. Acesso em: 22 out. 2020.

Sql injection. https://www.w3schools.com/sql/sql_injection.asp. Acesso em: 28 set. 2020.

Sql injection. <https://www.devmedia.com.br/sql-injection/6102>. Acesso em: 28 set. 2020.

Clemente, M. (2019). O que é e por que usar o servidor apache. <https://rockcontent.com/br/blog/apache/>. Acesso em: 22 out. 2020.

Dionísio, E. J. (2015). Postgresql tutorial. <https://www.devmedia.com.br/postgresql-tutorial/33025>. Acesso em: 22 out. 2020.

Machado, E. M. Tutorial de instalação apache 2.2.4 + php 5.2.1 + postgresql 8.2. <http://www.simonson.br/aulasonline/tutorialapache.pdf>. Acesso em: 22 out. 2020.