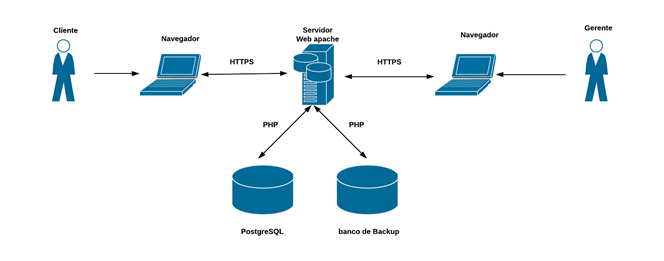
**Alunos:** Neimar Sausen, Matheus Leuck, Matheus Scheffer, Geraldo Fulgêncio

**Arquitetura do software**

**Framework utilizado**

O framework a ser utilizado é o Bootstrap que é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo.

Deste framework, serão utilizados os seguintes recursos:

1. Grid-layout;
2. Estilização de formulários;
3. Respostas assíncronas sobre as entradas dos formulários, avisando o que o usuário está fazendo de errado para auxiliá-lo no manuseio do sistema tornando-o mais intuitivo e;
4. Sistema de cores do framework para deixar as ações mais intuitivas. (danger para algum erro, success para uma ação bem feita, por exemplo).

**Padrões de Projeto**

Um dos padrões de projeto que foi adotado pela equipe foi o MVC. Este será de grande valia para a organização e a independência das camadas, possibilitando dividir a equipe para trabalhar em cada uma das partes de forma independente, segura e de melhor manutenção.

A organização das camadas seguirá as seguintes orientações:

**Camada Model:**

1. Function **ehCliente():**

Quando o cliente enviar seus dados através do formulário da parte da camada View responsável pelo login, será recebido o **usuario**, o qual, será feito uma **consulta no banco de dados** para saber se o mesmo já está cadastrado, assim, sabendo se seu dono já **é um cliente** transferindo-o para a página de encomendas**.** **Se** o usuário **não** for um cliente, retornará uma resposta para **login()** informando que não há este número cadastrado.

1. Function **realizarCadastro()**:

Será recebido os dados do formulário **cadastro** e separado seus dados em variáveis de fácil entendimento. Após isso, será verificado se os dados existem no **banco de dados**. **Se** existirem, enviará para a função **cadastro()** uma resposta negativa. **Caso contrário,** enviará uma resposta positiva.

1. Function **gerarOrcamento()**:

Receberá os dados da **verificarRequisitos()**, armazenando-os em variáveis de fácil entendimento e criando as variáveis respectivas para os cálculos de **área,custo\_material, custo\_desenho** e **valor\_placa.** Por fim, fará o pré-cadastro da encomenda com o campo de confirmação (na tabela de banco de dados) preenchido com o valor 0, indicando que a confirmação está pendente a pagamento e, após o cadastro da “pré-encomenda”, enviará os dados para a função **verificarRequisitos().**

1. Function **verificarCartão():**

Verifica, na tabela de cadastro do cliente, se há um cartão de crédito cadastro. **Se** houver, **então** enviará uma resposta para a camada **control** informando os 4 últimos dígitos do(s) cartão(s) de crédito para o usuário. **Caso contrário**, enviará uma resposta negativa quanto a existência de um cartão.

1. Function **fazerPgto( numero\_cartão)** :

Fará uma query, com os 4 últimos dígitos do cartão passados por parâmetro pela função **efetuarPgto(numeroCartao)**, pegará o número completo do cartão escolhido e, junto aos outros dados (do usuário e do pedido), enviará para a **api pagseguro** para efetuar o pagamento. **Se** houver **erro,** retornará para a função que fez a requisição. **Caso contrário**, enviará uma resposta positiva para a função que fez a requisição e alterará o **campo de confirmação** do banco de dados para valor **1.**

**Camada View:**

1)Página de login:  
O cliente fará contato com campos de formulários que fará operações de login, na primeira página o cliente vai digitar o usuário e senha, após clicar em “entrar” caso já tenha acesso. Caso ainda não tenha cadastro vai clicar em “clique aqui para se cadastrar”, sendo daí direcionado para a página 2.

2) Página de cadastro:  
Nessa página o usuário vai entrar com os dados pessoais, cadastrar um cartão de crédito(Caso o cartão não existir aparecerá uma mensagem informando) e criar o login e senha de acesso.

3)Página na qual serão digitados os requisitos para o cálculo da encomenda, esta página fica disponível quando o usuário, que já tem cadastro, acessar o sistema digitando o login e senha.

4)Essa página fica disponível depois da entrada dos requisitos na página 3 e ao clicar no botão comprar. A página em questão refere-se a forma de pagamento que poderá ser no cartão ou em dinheiro(no caixa) , caso o usuário não tenha ainda o cartão como forma de pagamento e desejar cadastrar poderá clicando na opção “clique aqui se não possuir um cartão cadastrado”.Nesse caso o usuário será direcionado para a página 5.

5)Nesta página o usuário entra com os dados do cartão de crédito e clica em cadastrar e após é direcionado de volta para a página 4 para continuar com a compra.

6)Na página 4 após selecionar uma forma de pagamento ele será direcionado para a página 6(página do recibo).

**Camada Control:**

1. Function **login()**:

Após a função **ehCliente()** verificar se o usuário existe e retornar a resposta, esta função redireciona-rá o usuário para a **página de encomendas** e gravará uma **sessão** com os dados do cliente. Se a função **ehCliente()** trouxer uma resposta negativa, será alterada a camada de ***View*** informando para o usuário se cadastrar.

1. Function **cadastro()**:

Cria um jSon contendo os dados cadastros e envia para a **realizarCadastro()** via Ajax, que retornará “ok” se o usuário não existe e foi cadastro, dando uma mensagem na camada ***View*** que o cadastro foi realizado com sucesso e, ainda nesta mensagem, um link para a página de **login. Caso contrário,** será retornado algum valor para identificar que o usuário já existe, alterando a camada **View** para informar que o usuário existe.

1. Function **verificarRequisitos():**

Verificará os **requisitos mínimos**

Placa: área = altura X largura;

Custo material: custo\_material = area X R$147,30

Desenho : custo\_desenho = número\_letras x R$ 0,32

Valor Placa: valor\_placa = custo\_material + custo\_desenho

Se ok, enviará uma requisição para a função **gerarOrcamento().** **Quando** receber os dados,preencherá todos os campos relativos a custo obtidos da requisição e fará com queo botão ***calcular*** seja renomeado para ***comprar*** e redirecionará para a página de **escolha de método de pagamento**. **Caso contrário,** mostrará um aviso no campo o qual não foi preenchido os requisitos.

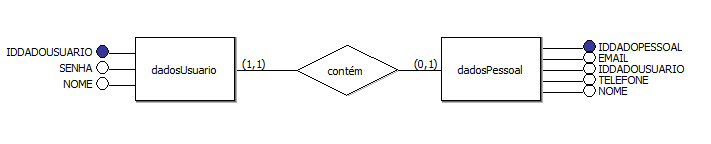
1. Function **escolherPgto()**:

Permite escolher entre **pagar no caixa** ou **pagar no cartão. Se** escolher **pagar no cartão**, **então** será acionada a função **verificarCartão(),** na camada **Model.** Quando receber a resposta da função requisita, verificará se houve retorno de resposta positivo e mostrará na tela os 4 últimos digitos do(s) cartão(s) do usuário, pedindo para clickar em qual será usado para o pagamento e, também, uma opção para **usar outro cartão,** que levará para a página de **cadastro de cartão.**

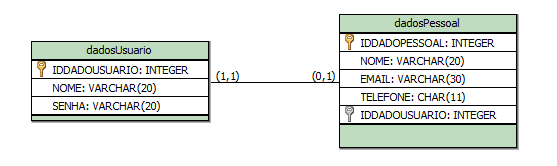
1. Function **efetuarPgto(numeroCartao)**:

Enviará para a função **fazerPgto( numero\_cartão)** o número cartão escolhido quando clicou nas opções mostradas anteriores para, então, finalizar a compra. Quando receber a resposta, fará o redirecionamento para a página contendo o documento de recibo em pdf, sendo possível fazer o download do mesmo, **se** for positiva. **Caso contrário**, informará que não foi possível realizar a compra pelo **erro** retornado pela função requisitada em forma de um **alert()**.

**ESQUEMA CONCEITUAL (BR MODELO)**

****

**ESQUEMA LÓGICO (BR MODELO)**

****

**CRIAÇÃO DAS TABELAS**

**Banco de dados: PostgreSQL**

*CREATE DATABASE PROJETO*

*USE PROJETO*

*CREATE TABLE dadosUsuario (*

*IDUSUARIO SERIAL,*

*NOME VARCHAR(20) UNIQUE NOT NULL,*

*SENHA VARCHAR(20) NOT NULL,*

*PRIMARY KEY (IDUSUARIO)*

*);*

*CREATE TABLE dadosPessoal (*

*IDDADOPESSOAL SERIAL NOT NULL,*

*NOME VARCHAR(20) NOT NULL,*

*EMAIL VARCHAR (30),*

*TELEFONE CHAR(11),*

*IDUSUARIO SERIAL,*

*PRIMARY KEY (IDDADOPESSOAL),*

*FOREIGN KEY (IDUSUARIO) REFERENCES dadosUsuario (IDUSUARIO)*

*);*

*Exemplos:*

*Insert:*

*INSERT INTO dadosUsuario( NOME, SENHA)*

*VALUES ('FUNALO...','\*\*\*\*');*

*INSERT INTO dadosPessoal( NOME, EMAIL, TELEFONE)*

*VALUES ('FUNALO...','FUNALO@...' '51999999999');*

*Select:*

*SELECT NOME FROM DADOSUSUARIO WHERE NOME = ‘Fulano...’;*

Criação Cadastro/Login php:

Cadastro:   
  
<?php

$login = $\_POST['login'];

$SENHA = MD5($\_POST['SENHA']);

$EMAIL = $\_POST['EMAIL'];

$TELEFONE = $\_POST['TELEFONE'];

$connect = mysql\_connect('dadospessoal','NOME','SENHA',’EMAIL’,’TELEFONE’);

$db = mysql\_select\_db('dadospessoal');

$query\_select = "SELECT login FROM dadosusuario WHERE login = '$login'";

$select = mysql\_query($query\_select,$connect);

$array = mysql\_fetch\_array($select);

$logarray = $array['login'];

if($login == "" || $login == null){

echo"<script language='javascript' type='text/javascript'>

alert('O campo login deve ser preenchido');window.location.href='

cadastro.html';</script>";

}else{

if($logarray == $login){

echo"<script language='javascript' type='text/javascript'>

alert('Esse login já existe');window.location.href='

cadastro.html';</script>";

die();

}else{

$query = "INSERT INTO dadospessoal (LOGIN,SENHA,EMAIL,TELEFONE) VALUES ('$LOGIN,'$SENHA',’$EMAIL’,’$TELEFONE’)";

$insert = mysql\_query($query,$connect);

if($insert){

echo"<script language='javascript' type='text/javascript'>

alert('Usuário cadastrado com sucesso!');window.location.

href='login.html'</script>";

}else{

echo"<script language='javascript' type='text/javascript'>

alert('Não foi possível cadastrar esse usuário');window.location

.href='cadastro.html'</script>";

}

}

}

?>

Login:

<?php

$login = $\_POST['login'];

$entrar = $\_POST['entrar'];

$senha = md5($\_POST['SENHA']);

$connect = mysql\_connect('dadosusuario','NOME','SENHA');

$db = mysql\_select\_db('dadosusuario');

if (isset($entrar)) {

$verifica = mysql\_query("SELECT \* FROM usuarios WHERE login =

'$login' AND SENHA = '$SENHA'") or die("erro ao selecionar");

if (mysql\_num\_rows($verifica)<=0){

echo"<script language='javascript' type='text/javascript'>

alert('Login e/ou senha incorretos');window.location

.href='login.html';</script>";

die();

}else{

setcookie("login",$login);

header("Location:index.php");

}

}

?>

**Login, cadastro e css**

**Login**

<!DOCTYPE HTML>

<html>

<head>

<title>Sistema de pedidos</title>

<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="estilo.css">

</head>

<body>

<main>

<div>

<form action="login.php">

<h3>Bem vindo ao sistema de pedidos!</h3>

<h2>Faça o seu login</h2>

<label>Usuário:</label><input style="background-color: #4391EB;" type="text" required />

<label>Senha:</label><input style="background-color: #4391EB;" type="password" required />

<input type="submit" value="Entrar"/>

<p>Não tem cadastro? Clique <a href="cadastrar.html">aqui</a> para se cadastrar!</p>

</form>

<p>Desenvolvido por: Neimar Sausen, Matheus Leuck, Matheus Scheffer, Geraldo Fulgêncio</p>

</div>

</main>

</body>

</html>

**Cadastrar**

<!DOCTYPE HTML>

<html>

<head>

<title>Cadastro de usuário</title>

<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="estilo.css">

</head>

<body>

<main>

<div>

<form action="cadastro.php">

<fieldset name="dadospessoais">

<h2>Dados pessoais</h2>

<label>Nome completo:</label><input style="background-color: #4391EB;" type="text" required />

<label>Telefone:</label><input style="background-color: #4391EB;" type="tel"required/>

<label>Email:</label><input style="background-color: #4391EB;" type="email"required/>

</fieldset>

<fieldset name="dadosdeusuario">

<h2>Dados de usuário</h2>

<label>Usuário:</label><input style="background-color: #4391EB;" type="text"required/>

<label>Senha:</label><input style="background-color: #4391EB;" type="password"required/>

<label>Confirmar senha:</label><input style="background-color: #4391EB;" type="password"required/>

<input type="submit" value="Cadastrar!"/>

</fieldset>

</form>

</div>

</main>

</body>

</html>

**css**

@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@500&display=swap');

body{

display: flex;

align-items: center;

justify-content: center;

font-family: 'Montserrat', sans-serif;

}

input[type=text], select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=tel], select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=email], select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=password], select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=number], select {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=submit] {

width: 100%;

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

font-family: 'Montserrat', sans-serif;

}

button[type=submit] {

width: 100%;

background-color: red;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type=submit]:hover {

background-color: #45a049;

}

div {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

width: 600px;

}