

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE – CÂMPUS PELOTAS - VISCONDE DA GRAÇA  
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**Sistema de Informação para Reserva de Produtos de Informática da  
Loja Andrômeda**

**Guilherme Costa da Silva**

Pelotas, dezembro de 2019.

**Guilherme Costa da Silva**

**Sistema de Informação para Reserva de Produtos de Informática da  
Loja Andrômeda**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Técnico, do curso Técnico em Desenvolvimento de Sistemas, do Instituto Federal Sul-Rio-Grandense – Câmpus Pelotas - Visconde da Graça.

Orientadora: Profa. Dra. Verlani Timm Hinz

Pelotas, dezembro de 2019.

## LISTA DE FIGURAS

Figura 1: Diagrama de Casos de Uso.....	9
Figura 2: Modelagem Conceitual do Banco de Dados.....	11
Figura 3. Diagrama Relacional .....	12
Figura 4: Tela inicial do sistema .....	15
Figura 5: Alteração na Cor das palavras clicáveis .....	16
Figura 6: Página de Cadastro de Clientes. ....	16
Figura 7: Tela de login. ....	17
Figura 8: Tela de login com campos não preenchidos. ....	18
Figura 9 - Administrador. ....	19
Figura 10 - Tela do administrador.....	19

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	5
1.1 TEMA .....	5
1.2 MOTIVAÇÕES .....	5
1.3 OBJETIVOS .....	6
<b>2 ESPECIFICAÇÃO DE REQUISITOS</b> .....	7
2.1 MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS .....	7
2.2 REQUISITOS FUNCIONAIS .....	7
2.3 REQUISITOS NÃO FUNCIONAIS .....	8
<b>3 MODELAGEM</b> .....	9
3.1 MODELO DE CASOS DE USO .....	9
3.2 MODELAGEM CONCEITUAL DO BANCO DE DADOS .....	11
3.3 MODELAGEM LÓGICA DO BANCO DE DADOS.....	12
<b>4 TECNOLOGIAS UTILIZADAS</b> .....	14
<b>5 DESCRIÇÃO DO SISTEMA</b> .....	15
<b>6 CONSIDERAÇÕES FINAIS</b> .....	20
<b>7 REFERÊNCIAS</b> .....	21
<b>8 APÊNDICE A – Códigos do banco de dados.</b> .....	22

## **1. INTRODUÇÃO**

O sistema foi sugerido a empresa Andrômeda pois atualmente, o crescimento do uso de tecnologias traz a necessidade de realizar tarefas de maneira cada vez mais simples e informatizadas, ou seja, de modo que seja exigido o menor esforço possível por parte das pessoas.

As aplicações do tipo *web* são exemplos dessa informatização, já que cada vez mais tarefas, desde o envio de mensagens até o pagamento de contas, podem ser realizadas de maneira *online*, utilizando apenas um dispositivo eletrônico com acesso à Internet.

Diante disso, esse trabalho tem por objetivo desenvolver um sistema que permita acesso, visualização e reserva de produtos, de forma *online*, da loja de informática Andrômeda.

A Andrômeda é uma loja de informática de produtos de informática e assistência técnica, de pequeno porte, localizada no bairro Fragata, rua Av. Cidade de Lisboa, Nº 442.

Atualmente, para encomendar um dos produtos da loja é necessário ir até a o local ou entrar em contato através de redes sociais e posteriormente buscar o produto na loja.

Assim, este sistema busca facilitar e agilizar o atendimento e encomendas de produtos, através de uma interface *Web*.

### **1.1 TEMA**

O tema deste trabalho consiste no desenvolvimento de um sistema *web* que permita que os clientes efetuem a reserva de produtos de informática da Loja Andrômeda, que posteriormente serão adquiridos na loja física.

### **1.2 MOTIVAÇÕES**

A principal motivação para o desenvolvimento do sistema é agilizar, facilitar e organizar de maneira automática a reserva de produtos, auxiliando tanto os clientes como o proprietário da Andrômeda.

Por se tratar de uma loja de assistência técnica e vendas de produtos a fila de espera é alta, pois há poucos funcionários atendendo e os mesmos atendem vendas de produtos e serviços de manutenção. Assim, com o desenvolvimento e implantação do sistema pretendesse auxiliar e reduzir o tempo de espera na fila.

### **1.3 OBJETIVOS**

O objetivo geral é desenvolver um sistema para controle da reserva de produtos de informática da Loja Andrômeda.

Para que seja alcançado o objetivo geral os seguintes objetivos específicos foram definidos:

- Realizar uma pesquisa sobre os sistemas similares que existem na área.
- Definir os requisitos do sistema.
- Permitir o gerenciamento de categorias, subcategorias, marcas, produtos e usuários.
- Permitir o gerenciamento de reserva de produtos de maneira *on-line*.
- Permitir o acesso dos usuários somente às funcionalidades do sistema disponíveis para o tipo de perfil adequado.
- Treinar os funcionários da Loja Andrômeda para utilização do sistema.

## 2 ESPECIFICAÇÃO DE REQUISITOS

### 2.1 MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS

Os requisitos funcionais e não funcionais especificados para o desenvolvido deste sistema foram obtidos através de entrevistas. Uma entrevista é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados, visto que o entrevistador dá margens ao entrevistado para expor as suas ideias.

Essa entrevista foi realizada com o proprietário da loja que aceitou ser o cliente desse produto a ser produzido e as entrevistas foram realizadas tanto pessoalmente quanto pelos aplicativos *WhatsApp* e *Facebook*.

Além das entrevistas, para definir as funcionalidades do sistema, foram estudados alguns sistemas similares disponíveis na Web como a *Amazon*, *Kabum* e *Rocketz*.

### 2.2 REQUISITOS FUNCIONAIS

O sistema deve possuir os seguintes requisitos funcionais:

- **REF01 – Gerenciar Vendas** – O administrador tem o acesso ao histórico do que foi vendido, quando foi vendido e para quem foi vendido, o administrador será capaz de alterar uma reserva para o status de venda, assim como cancelar uma reserva.
- **REF02 - Consulta de Produtos** – Os usuários têm total acesso aos preços e características dos produtos disponíveis.
- **REF03 - Gerenciamento de Produtos** – O administrador tem a capacidade de alterar valores, imagens, descrições e disponibilidade de produtos, assim como inserir novos produtos.
- **REF04 – Cadastrar Usuários** – O usuário deve poder se autocadastrar, sem o auxílio de qualquer administrador.
- **REF05 – Gerenciamento de Categoria** – O administrador deve poder cadastrar, alterar e excluir uma categoria quando desejar.
- **REF06 – Gerenciamento de Subcategoria** – O administrador deve poder cadastrar, alterar e excluir subcategorias quando quiser.

- **REF07 – Gerenciamento de Marca** – O administrador deve ser capaz de cadastrar, alterar e excluir marcas quando lhe for conveniente.
- **REF08 – Gerenciar Reserva** – O administrador é capaz de cancelar uma reserva e tornar uma reserva uma venda. O usuário deve efetuar o *login* para fazer qualquer reserva no sistema.
- **REF9 – Gerenciamento de Perfil** – O administrador é capaz de transformar um usuário comum em um administrador e também transformar um administrador em um usuário comum.

### 2.3 REQUISITOS NÃO FUNCIONAIS

As características abaixo descrevem os requisitos não funcionais do sistema:

- **RNF01:** ser desenvolvido com tecnologias que permitam o acesso pela *Web*.
- **RNF02:** ter os dados armazenados em um Sistema Gerenciador de Banco de Dados (SGBD).
- **RNF03:** ser bem intuitivo para o usuário e o administrador.
- **RNF04:** ficar *online* 24 horas por dia, 365 dias por ano.
- **RNF05:** ser fácil de administrar.

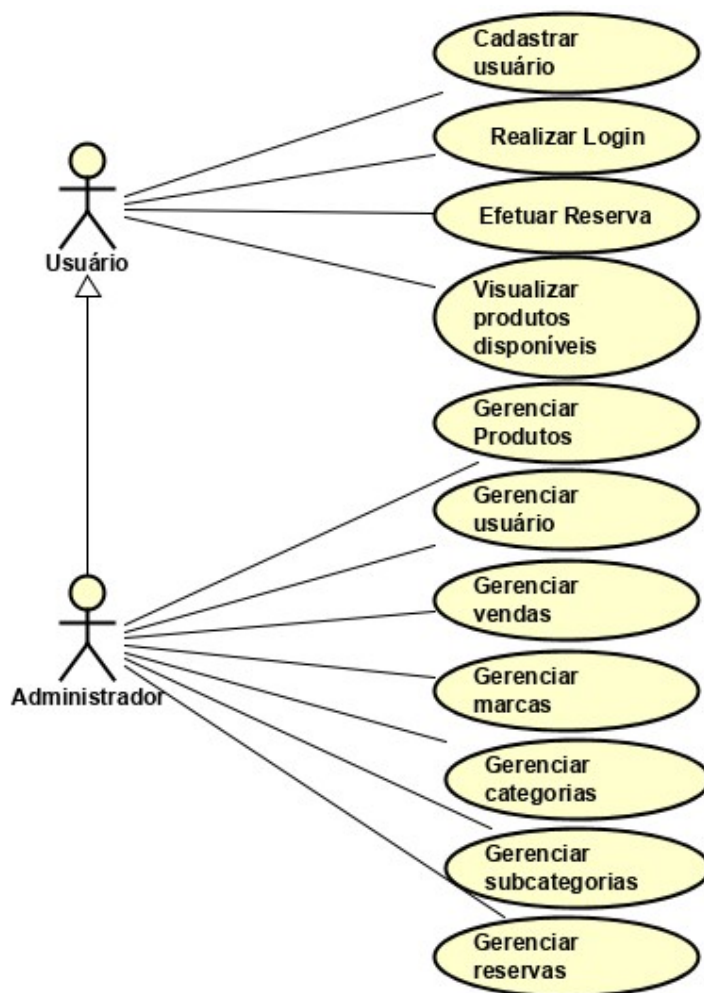


### 3 MODELAGEM

#### 3.1 MODELO DE CASOS DE USO

Na Figura 1, são descritos os casos de uso que foram levantados na fase de levantamento de requisitos do sistema.

Figura 1: Diagrama de Casos de Uso



Fonte: Elaborado pelo Autor.

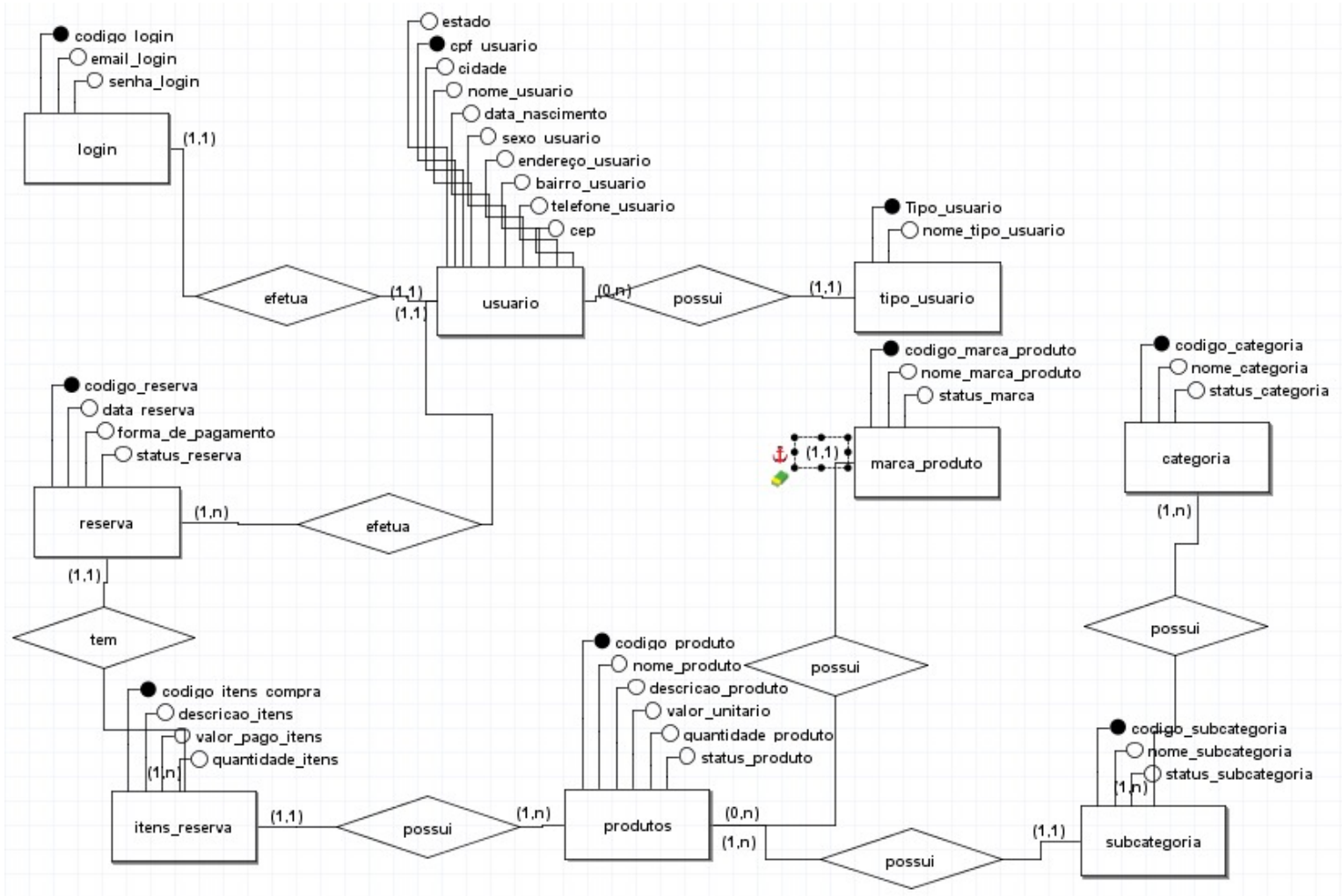
O ator usuário representa todas as pessoas que irão visualizar e efetuar as reservas, sendo que somente é necessário a realização do login caso seja efetuada uma reserva, do contrário ele poderá visualizar os produtos disponíveis.

- **Cadastro de perfil:** O usuário é cadastrado inicialmente como usuário padrão, o administrador pode tornar um usuário padrão em um administrador.
- **Realizar login:** O usuário deve se logar no sistema para realizar a reserva de produtos.
- **Efetuar reserva:** O usuário pode fazer a reserva somente dos produtos disponíveis.
- **Visualizar produtos disponíveis:** O usuário deve visualizar apenas os produtos considerados disponíveis pelo administrador.
- **Gerenciar produtos:** O administrador deve ser capaz de inserir, alterar e remover produtos sempre que necessário.
- **Gerenciar vendas:** O administrador deve ser capaz de tornar um registro de reserva em um registro de venda.
- **Gerenciar marcas:** O administrador deve ser capaz de inserir, alterar e desabilitar ou habilitar marcas, quando necessário, sendo que, produtos com marcas desabilitadas não serão visualizados pelo usuário.
- **Gerenciar categorias:** O administrador deve ser capaz de inserir, alterar e desabilitar ou habilitar categorias, quando necessário, sendo que, subcategorias “filhas” de categorias desabilitadas devem permanecer ocultas ao cliente.
- **Gerenciar subcategorias:** O administrador deve ser capaz de inserir, alterar e desabilitar ou habilitar subcategorias, quando necessário.
- **Gerenciar reserva:** O administrador deve ser capaz de cancelar reservas feitas pelos usuários.
- **Cadastrar usuário:** O usuário é capaz de se cadastrar sem ajuda do administrador.

### 3.2 MODELAGEM CONCEITUAL DO BANCO DE DADOS

A Figura 2 mostra a representação gráfica do modelo ER, o Diagrama de Entidade e Relacionamento. Este diagrama tem como objetivo descrever quais dados devem ser armazenados e quais desses dados se relacionam.

Figura 2: Modelagem Conceitual do Banco de Dados



Fonte: Elaborado pelo autor.

As entidades Usuário e Tipo\_Usuário caracterizam os usuários do sistema. Um usuário tem apenas um tipo de usuário e um tipo de usuário pode estar associado a mais de um usuário.

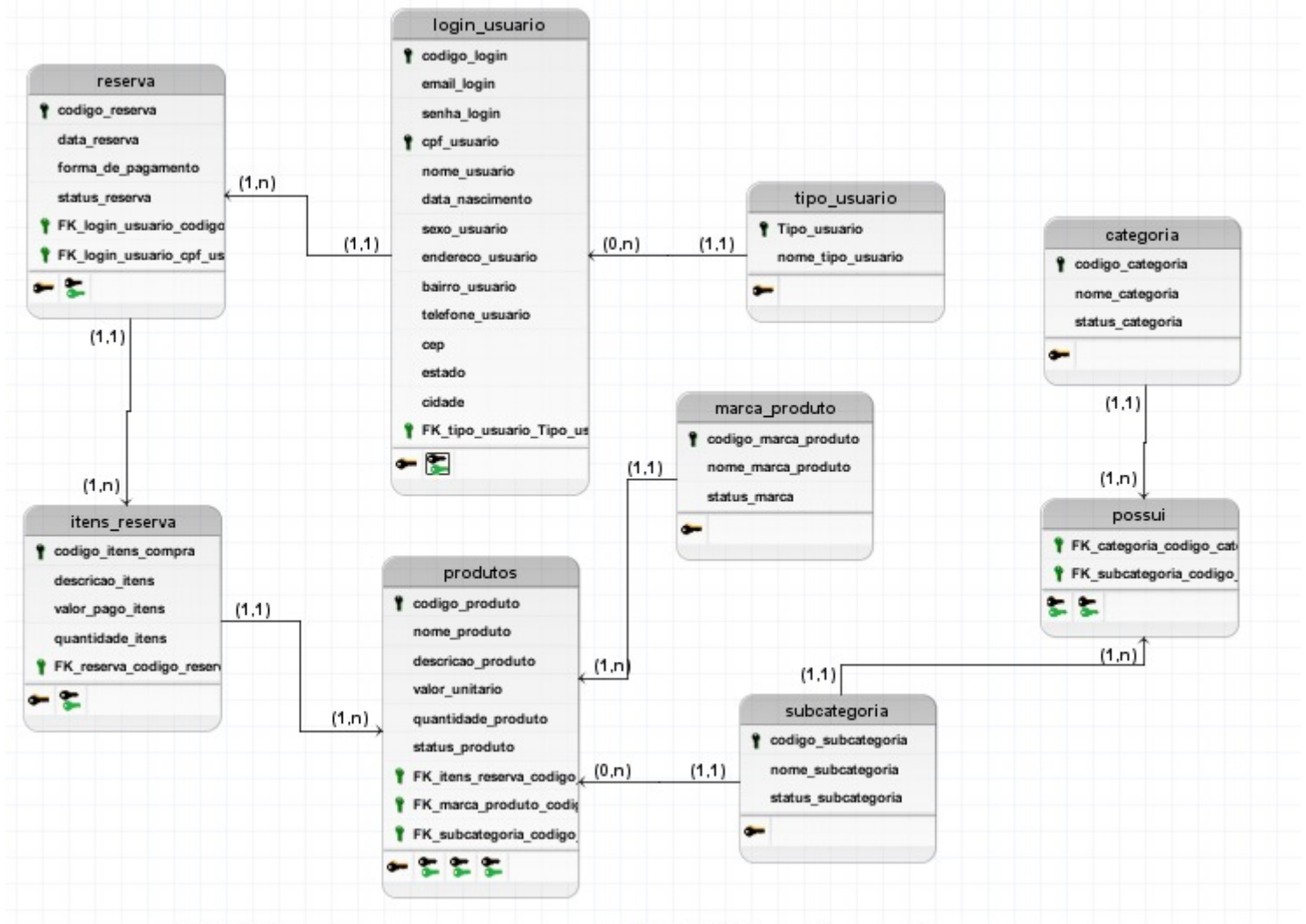
Existem dois tipos de usuário (Administrador e Usuário Padrão). O administrador tem total acesso a loja sendo capaz de gerenciar produtos, categorias, marcas e subcategorias, o usuário é capaz de visualizar os produtos disponíveis e realizar reservas.

A entidade Reserva é essencial para registrar as reservas feitas pelo usuário, com uma relação com Itens\_reserva a entidade armazena os valores dos produtos, o cpf do usuário e a data da reserva.

### 3.3 MODELAGEM LÓGICA DO BANCO DE DADOS

A Figura 3 mostra o diagrama relacional do sistema. Pode-se observar o conjunto de tabelas que forma a base de dados do sistema, com a identificação de todos os atributos e seus respectivos tipos, bem como as chaves primárias e as chaves estrangeiras que implementam os relacionamentos entre as tabelas.

Figura 3. Diagrama Relacional



Fonte: Elaborado pelo autor.

Com exceção da chave primária da tabela “Usuário” que armazena caracteres, as demais são do tipo inteiro e tem seus valores incrementados automaticamente (auto incremento).

Na tabela usuário existem duas chaves estrangeiras, uma da tabela login e a outra da tabela tipo\_usuario, sendo que, a chave estrangeira da tabela login serve para verificar o e-mail e senha do usuário, caso e-mail e senha estejam corretos é utilizado os dados cadastrados na tabela usuário para realizar reservas, a chave estrangeira tipo\_usuario é utilizada para verificar e fornecer permissões diferentes a tipos de usuários.

Na tabela reserva existe a chave estrangeira da tabela usuário (cpf\_usuario) que é utilizada para relacionar a reserva ao usuário que efetuou a reserva.

Já na relação itens\_reserva e reserva, conseguimos notar que existem duas chaves estrangeiras apenas, uma de ligação a tabela reserva e a outra a itens\_reserva que se relaciona diretamente com produtos.

A tabela produtos é utilizada para guardar informações dos produtos como imagens, descrições, valores, nomes, quantidade em estoque, marcas, subcategorias e categorias.

## 4 TECNOLOGIAS UTILIZADAS

Neste projeto foram utilizadas ferramentas para modelagem do sistema e para modelagem e criação do banco de dados, bem como tecnologias que possibilitam a implementação de sistemas para acesso através da Web.

Para desenvolver a interface foi utilizado as linguagens *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS), JavaScript e o framework *Bootstrap*. Para a programação *back-end* foi utilizado a linguagem PHP.

O HTML (em português Linguagem de Marcação de Hipertexto) é a linguagem base da Internet, utilizando marcações que são utilizadas para produção de páginas na web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela Internet.

Já CSS é uma ‘folha de estilo’ composta por ‘camadas’ e utilizada para definir a apresentação (aparência) em páginas da Internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML). Enquanto que o *JavaScript*<sup>1</sup> é uma linguagem de programação de alto-nível, permite ao desenvolvedor implementar diversos itens de alto nível de complexidade em páginas web, como animações, mapas, gráficos ou informações que se atualizam em intervalos de tempo padrão;

O *Bootstrap*<sup>2</sup> é um *framework* web com código-fonte aberto para desenvolvimento de componentes *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.

Como banco de dados a escolha foi *MySQL*<sup>3</sup>, sendo um dos sistemas de gerenciamento de bancos de dados mais populares da *Oracle Corporation*. O *MySQL* é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface.

---

<sup>1</sup> <https://www.javascript.com/>

<sup>2</sup> <https://getbootstrap.com/>

<sup>3</sup> <https://www.mysql.com/>

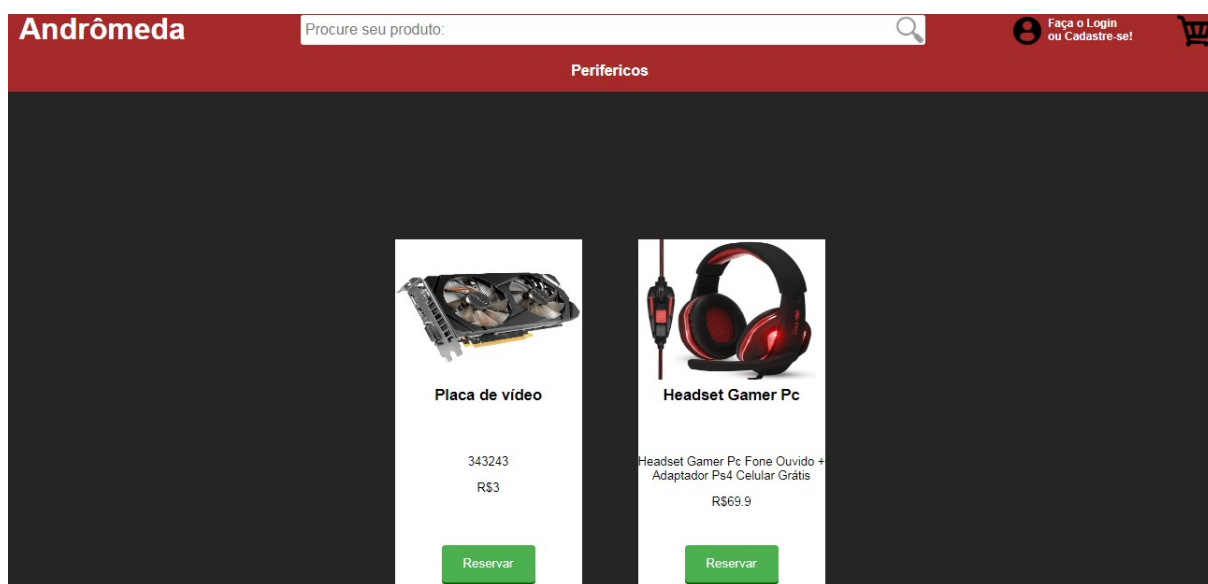
Para acessar o banco se utilizou o *XAMPP*, um servidor independente de plataforma e também sendo um software livre, sendo constituído pelo SGBD *MySQL*, o servidor *Web Apache* e o interpretador para a linguagem *PHP*.

## 5 DESCRIÇÃO DO SISTEMA

Neste capítulo são apresentadas as principais funcionalidades do sistema, sendo caracterizado um fluxo de utilização a partir da exibição das principais interfaces do sistema.

A Figura 4 demonstra a tela inicial do sistema, onde o usuário é capaz de fazer pesquisas através das opções localizadas no *header* da tela e também já é capaz de visualizar os produtos.

Figura 4: Tela inicial do sistema



Fonte: Elaborado pelo autor.

Na Figura 5 podemos observar que existe a seguinte frase próxima a um ícone, “Faça o **Login** ou **Cadastre-se!**” a o colocar o mouse em cima o usuário pode observar que a cor da palavra login e cadastre-se mudam.

**Figura 5: Alteração na Cor das palavras clicáveis**



Fonte: Elaborado pelo autor.

Se o usuário clicar no “Cadastre-se!”, será redirecionado para a página de cadastro como demonstra a Figura 6.

**Figura 6: Página de Cadastro de Clientes.**

\* Campo Obrigatório

Nome \*

CPF \*  Nascimento \*  Sexo \*  Feminino  Masculino

Telefone \*

Email de Cadastro \*

Senha de Cadastro \*

Confirmar Senha \*

CEP \*

Endereço: Rua  Nº \*  Bairro   
Cidade  Estado

Fonte: Elaborado pelo Autor.

Se o usuário clicar na palavra “Login” ou efetuar o cadastro como demonstra a figura 6 ele será redirecionado a uma nova página, para efetuar assim o *login* como demonstra a Figura 7.



**Figura 7: Tela de login.**

**Andrômeda**

**Email:**

**Senha:**

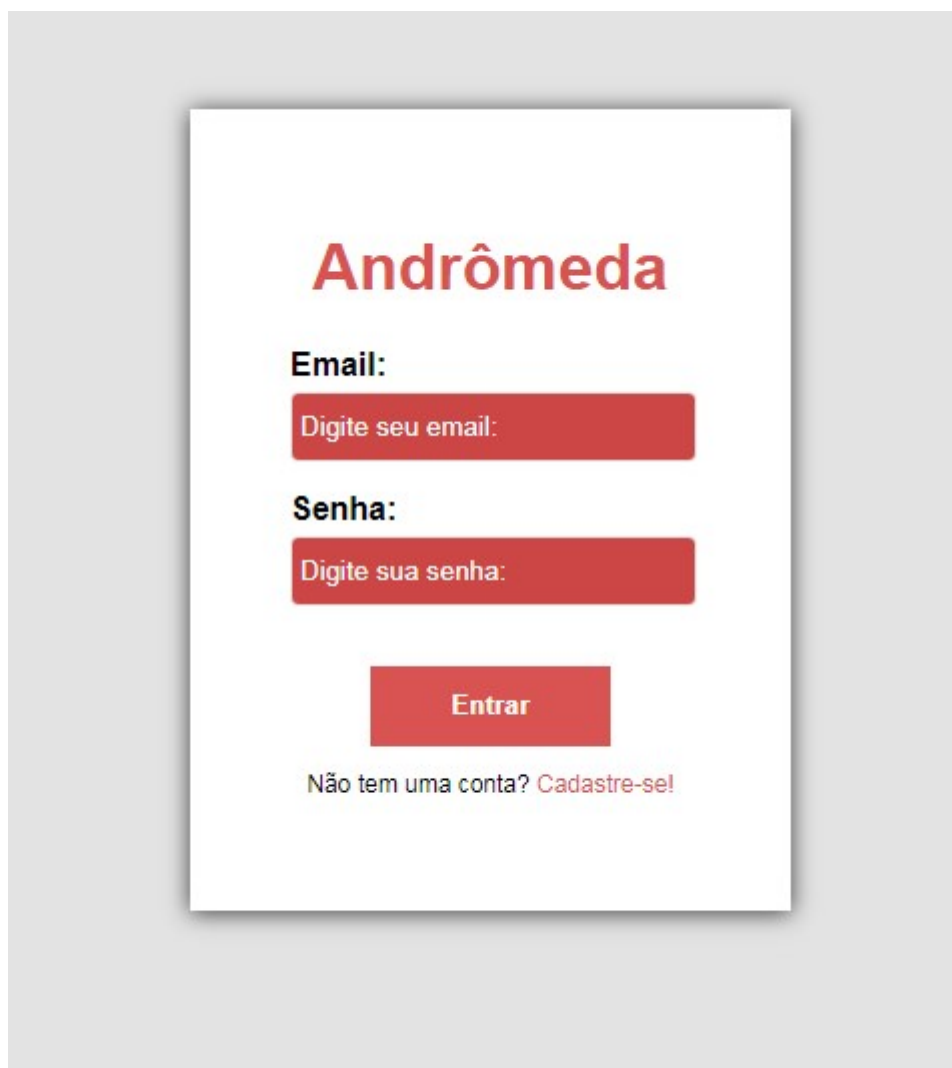
**Entrar**

Não tem uma conta? [Cadastre-se!](#)

**Fonte: Elaborado pelo Autor.**

Se algum dos campos não for devidamente preenchido aparecerá uma mensagem de erro e também os campos ficaram no tom vermelho como demonstra a Figura 8.

**Figura 8: Tela de login com campos não preenchidos.**



A imagem mostra a interface de login do sistema Andrômeda. No topo, o nome "Andrômeda" é exibido em uma fonte vermelha. Abaixo, há dois campos de entrada: "Email:" e "Senha:". Cada campo possui um botão vermelho com o texto "Digite seu email:" e "Digite sua senha:" respectivamente. Abaixo dos campos, há um botão vermelho com o texto "Entrar". Na base da tela, há o texto "Não tem uma conta? [Cadastre-se!](#)".

**Fonte: Elaborado pelo Autor.**

O usuário é capaz de adicionar produtos ao carrinho e visualizar o carrinho clicando na imagem do mesmo como mostra a figura 9, ele também é capaz de remover os produtos do carrinho ou alterar a sua quantidade.

**Figura 9 – Carrinho.**



**Fonte: Elaborado pelo Autor.**

Se o usuário tiver permissão do tipo administrador, onde ficava o login irá aparecer uma frase “Área do administrador” como demonstra a Figura 10, se clicado o administrador será enviado a área de administração

**Figura 10 - Administrador.**



**Fonte: Elaborado pelo Autor.**

Na área de administração o administrador pode gerenciar produtos, marcas, categorias, subcategorias, reserva e relatório como mostra a Figura 10.

**Figura 11 - Tela do administrador.**



Seja bem-vindo a página de edição do administrador, somente os administradores tem acesso a essa página, acima você poderá ver que há uma lista de ações que você poderá realizar para gerenciar o seu sistema. Qualquer problema contatar o seu desenvolvedor.

**Fonte: Elaborado pelo Autor.**

## 6 CONSIDERAÇÕES FINAIS

Este projeto teve como objetivo facilitar e agilizar o processo de reserva da loja Andrômeda inf, usando de tecnologias para agilizar a reserva de produtos e assim diminuir a quantidade de espera na loja tanto para compras quanto para visualização de produtos.

O sistema facilita a busca de produtos que são ofertados na loja e também a reserva de produtos, assim diminuindo o tempo de espera e agilizando o atendimento.

Na continuidade deste projeto entende-se como possibilidades de trabalhos futuros: (i) A possibilidade de se fazer compras *onlines*; (ii) Fazer o uso de api's para facilitar o cadastro e login de usuários; (iii) Criar um *chat online* para comunicação direta do usuário e o administrador.

## 7 REFERÊNCIAS

KABUM. Disponível em: [Https://www.kabum.com.br/](https://www.kabum.com.br/). Acesso em: 28/05/2019

ROCKETZ. Disponível em: [Https://rocketz.com.br/](https://rocketz.com.br/). Acesso em 28/05/2019

AMAZON. Disponível em: [Https://www.amazon.com.br/](https://www.amazon.com.br/). Acesso em 01/10/2019.

EIS, DIEGO; FERREIRA, ELCIO. HTML5 e CSS3 com farinha e pimenta. Lulu. com, 2012.

PRESSMAN, ROGER S. Engenharia de Software: uma abordagem profissional. 7. ed. São Paulo: Pearson Makron Books, 2011.

BEZERRA, EDUARDO. Princípios de Análise e Projeto de Sistemas com UML. Rio de Janeiro: Elsevier, 2007.

## 8 APÊNDICE A – Códigos do banco de dados.

```
-- phpMyAdmin SQL Dump
-- version 4.8.5
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: 20-Dez-2019 às 03:16
-- Versão do servidor: 10.1.38-MariaDB
-- versão do PHP: 7.3.3
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `andromeda`
--
```

```
-----
```

```
--
```

```
-- Estrutura da tabela `categoria`
```

```
--
```

```
CREATE TABLE `categoria` (  
  `codigo_categoria` int(11) NOT NULL,  
  `nome_categoria` varchar(80) NOT NULL,  
  `status_categoria` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--
```

```
-- Estrutura da tabela `compra`
```

```
--
```

```
CREATE TABLE `compra` (  
  `codigo_compra` int(11) NOT NULL,  
  `data_compra` date NOT NULL,  
  `forma_de_pagamento` varchar(100) DEFAULT NULL,  
  `cpf_usuario_compra` char(11) NOT NULL,  
  `status_compra` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--
```

```
-- Estrutura da tabela `itens_compra`
```

```
--
```

```
CREATE TABLE `itens_compra` (  
  `codigo_itens_compra` int(11) NOT NULL,  
  `descricao_itens` varchar(100) NOT NULL,
```

```
`valor_pago_itens` int(11) NOT NULL,  
`quantidade_itens` int(11) NOT NULL,  
`codigo_produtos` int(11) NOT NULL,  
`codigo_compra_itens` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-----

--

-- Estrutura da tabela `login`

--

```
CREATE TABLE `login` (  
  `codigo_login` int(11) NOT NULL,  
  `email_login` varchar(80) NOT NULL,  
  `senha_login` varchar(150) NOT NULL,  
  `codigo_tipo_usuario` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-----

--

-- Estrutura da tabela `marca\_produto`

--

```
CREATE TABLE `marca_produto` (  
  `codigo_marca_produto` int(11) NOT NULL,  
  `nome_marca_produto` varchar(80) NOT NULL,  
  `status_marca` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-----



--

-- Estrutura da tabela `produtos`

--

```
CREATE TABLE `produtos` (  
  `codigo_produto` int(11) NOT NULL,  
  `link_imagem` varchar(1000) NOT NULL,  
  `nome_produto` varchar(150) NOT NULL,  
  `descricao_produto` varchar(250) NOT NULL,  
  `valor_unitario` float NOT NULL,  
  `quantidade_produto` int(11) NOT NULL,  
  `codigo_subcategoria_produto` int(11) NOT NULL,  
  `codigo_marca_produto_produto` int(11) NOT NULL,  
  `status_produto` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-----

--

-- Estrutura da tabela `subcategoria`

--

```
CREATE TABLE `subcategoria` (  
  `codigo_subcategoria` int(11) NOT NULL,  
  `nome_subcategoria` varchar(100) NOT NULL,  
  `status_subcategoria` varchar(50) NOT NULL,  
  `codigo_sub_categoria` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-----

--

```
-- Estrutura da tabela `tipo_usuario`  
--  
  
CREATE TABLE `tipo_usuario` (  
  `tipo_usuario` int(4) NOT NULL,  
  `nome_tipo_usuario` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--  
-- Estrutura da tabela `usuario`  
--  
  
CREATE TABLE `usuario` (  
  `cpf_usuario` char(11) NOT NULL,  
  `nome_usuario` varchar(80) NOT NULL,  
  `data_nascimento` date NOT NULL,  
  `sexo_usuario` varchar(50) NOT NULL,  
  `endereco_usuario` varchar(100) NOT NULL,  
  `bairro_usuario` varchar(80) NOT NULL,  
  `telefone_usuario` varchar(20) NOT NULL,  
  `cep` int(11) NOT NULL,  
  `estado` varchar(50) NOT NULL,  
  `cidade` varchar(100) NOT NULL,  
  `codigo_usuario_login` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Indexes for dumped tables  
--
```

```
--
```

```
--
```

```

-- Indexes for table `categoria`
--
ALTER TABLE `categoria`
  ADD PRIMARY KEY (`codigo_categoria`);

--
-- Indexes for table `compra`
--
ALTER TABLE `compra`
  ADD PRIMARY KEY (`codigo_compra`),
  ADD KEY `cpf_usuario` (`cpf_usuario_compra`);

--
-- Indexes for table `itens_compra`
--
ALTER TABLE `itens_compra`
  ADD PRIMARY KEY (`codigo_itens_compra`),
  ADD KEY `codigo_compra` (`codigo_compra_itens`),
  ADD KEY `codigo_produtos` (`codigo_produtos`);

--
-- Indexes for table `login`
--
ALTER TABLE `login`
  ADD PRIMARY KEY (`codigo_login`),
  ADD KEY `codigo_tipo_usuario` (`codigo_tipo_usuario`);

--
-- Indexes for table `marca_produto`
--
ALTER TABLE `marca_produto`
  ADD PRIMARY KEY (`codigo_marca_produto`);

```

```
--  
-- Indexes for table `produtos`  
--  
ALTER TABLE `produtos`  
  ADD PRIMARY KEY (`codigo_produto`),  
  ADD KEY `codigo_marca_produto` (`codigo_marca_produto_produto`),  
  ADD KEY `codigo_subcategoria` (`codigo_subcategoria_produto`);
```

```
--  
-- Indexes for table `subcategoria`  
--  
ALTER TABLE `subcategoria`  
  ADD PRIMARY KEY (`codigo_subcategoria`),  
  ADD KEY `subcategoria_ibfk_1` (`codigo_sub_categoria`);
```

```
--  
-- Indexes for table `tipo_usuario`  
--  
ALTER TABLE `tipo_usuario`  
  ADD PRIMARY KEY (`tipo_usuario`);
```

```
--  
-- Indexes for table `usuario`  
--  
ALTER TABLE `usuario`  
  ADD PRIMARY KEY (`cpf_usuario`),  
  ADD KEY `codigo_usuario_login` (`codigo_usuario_login`);
```

```
--  
-- AUTO_INCREMENT for dumped tables  
--
```

```
--
```

```
--
```

```

-- AUTO_INCREMENT for table `categoria`
--
ALTER TABLE `categoria`
  MODIFY `codigo_categoria` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=20;

--
-- AUTO_INCREMENT for table `compra`
--
ALTER TABLE `compra`
  MODIFY `codigo_compra` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=44;

--
-- AUTO_INCREMENT for table `itens_compra`
--
ALTER TABLE `itens_compra`
  MODIFY `codigo_itens_compra` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=52;

--
-- AUTO_INCREMENT for table `login`
--
ALTER TABLE `login`
  MODIFY `codigo_login` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=35;

--
-- AUTO_INCREMENT for table `marca_produto`
--
ALTER TABLE `marca_produto`
  MODIFY `codigo_marca_produto` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=5;

```

```

--
-- AUTO_INCREMENT for table `produtos`
--
ALTER TABLE `produtos`
  MODIFY `codigo_produto` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=20;

--
-- AUTO_INCREMENT for table `subcategoria`
--
ALTER TABLE `subcategoria`
  MODIFY `codigo_subcategoria` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=27;

--
-- Constraints for dumped tables
--

--
-- Limitadores para a tabela `compra`
--
ALTER TABLE `compra`
  ADD CONSTRAINT `compra_ibfk_1` FOREIGN KEY (`cpf_usuario_compra`)
  REFERENCES `usuario` (`cpf_usuario`);

--
-- Limitadores para a tabela `itens_compra`
--
ALTER TABLE `itens_compra`
  ADD CONSTRAINT `itens_compra_ibfk_2` FOREIGN KEY
  (`codigo_compra_itens`) REFERENCES `compra` (`codigo_compra`),

```

```
ADD CONSTRAINT `itens_compra_ibfk_3` FOREIGN KEY (`codigo_produtos`)
REFERENCES `produtos` (`codigo_produto`);
```

```
--
```

```
-- Limitadores para a tabela `login`
```

```
--
```

```
ALTER TABLE `login`
```

```
ADD CONSTRAINT `login_ibfk_1` FOREIGN KEY (`codigo_tipo_usuario`)
REFERENCES `tipo_usuario` (`tipo_usuario`);
```

```
--
```

```
-- Limitadores para a tabela `produtos`
```

```
--
```

```
ALTER TABLE `produtos`
```

```
ADD CONSTRAINT `produtos_ibfk_1` FOREIGN KEY
(`codigo_marca_produto_produto`) REFERENCES `marca_produto`
(`codigo_marca_produto`),
```

```
ADD CONSTRAINT `produtos_ibfk_2` FOREIGN KEY
(`codigo_subcategoria_produto`) REFERENCES `subcategoria`
(`codigo_subcategoria`);
```

```
--
```

```
-- Limitadores para a tabela `subcategoria`
```

```
--
```

```
ALTER TABLE `subcategoria`
```

```
ADD CONSTRAINT `subcategoria_ibfk_1` FOREIGN KEY
(`codigo_sub_categoria`) REFERENCES `categoria` (`codigo_categoria`);
```

```
--
```

```
-- Limitadores para a tabela `usuario`
```

```
--
```

```
ALTER TABLE `usuario`
```

```
ADD CONSTRAINT `usuario_ibfk_2` FOREIGN KEY (`codigo_usuario_login`)
REFERENCES `login` (`codigo_login`);
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```