

Pedro Laurenti, Lucas Andrade e Allan Cândido

**PIM VIII**  
**Projeto Integrado Multidisciplinar VIII**

Brasil

Setembro de 2023

Pedro Laurenti, Lucas Andrade e Allan Cândido

**PIM VIII**  
**Projeto Integrado Multidisciplinar VIII**

Trabalho científico redigido colocando em prática as habilidades e conhecimento adquiridos no terceiro período do curso.

Universidade Paulista - UNIP  
Curso de Análise e Desenvolvimento de Sistemas

Orientador: Robson Batista Alves  
Coorientador: Tarcisio Peres

Brasil  
Setembro de 2023

Pedro Laurenti, Lucas Andrade e Allan Cândido

PIM VIII Projeto Integrado Multidisciplinar VIII - Brasil, Setembro de 2023 – 49p.

Orientador: Robson Batista Alves

Trabalho Científico – Universidade Paulista - UNIP

Curso de Análise e Desenvolvimento de Sistemas  
, Setembro de 2023.

1. Market Place. 2. Desenvolvimento. 3. Análise de Sistemas. I. Market. II. UNIP - Universidade Paulista. III. Faculdade de Análise e Desenvolvimento de Sistemas. IV. PIM VIII - Projeto Integrado Multidisciplinar.

Pedro Laurenti, Lucas Andrade e Allan Cândido

**PIM VIII**  
**Projeto Integrado Multidisciplinar VIII**

Trabalho científico redigido colocando em prática as habilidades e conhecimento adquiridos no terceiro período do curso.

Trabalho aprovado. Brasil, 24 de novembro de 2012:

---

**Robson Batista Alves**  
Orientador

---

**Professor**  
Convidado 1

---

**Professor**  
Convidado 2

Brasil  
Setembro de 2023

*Este projeto é dedicado a todos os desenvolvedores que já falharam várias vezes, mas  
nunca desistiram de suas paixões e ideias.*

*“Ser feliz ao realizar a jornada pode ser muito melhor do que chegar ao destino com sucesso.” - (Jordan Peterson)*

# Resumo

Este Projeto Integrado Multidisciplinar (PIM) propõe o desenvolvimento de um Sistema de Marketplace, abrangendo tanto aplicativos móveis quanto navegadores web. Unindo conhecimentos de Desenvolvimento de Software para Internet, Tópicos Especiais de Programação Orientada a Objetos e Programação Orientada a Objetos II, o projeto utiliza códigos ASPX para a interface em ASP.net, cria um protótipo Android em XML e implementa em **C#** o acesso ao Banco de Dados, incluindo classes como Cliente, Carrinho, Produto e Vendedor. A classe "CarrinhoRepository" gerencia carrinhos de compras. Este trabalho destaca não apenas a inovação no Sistema de Marketplace, mas também a aplicação prática dos conhecimentos adquiridos, resultando em uma solução tecnológica robusta alinhada às demandas do comércio digital contemporâneo.

**Palavras-chave:** Marketplace, Desenvolvimento de Software, Programação Orientada a Objetos, Aplicativos Móveis.

# Abstract

This Multidisciplinary Integrated Project (MIP) proposes the development of a Marketplace System, encompassing both mobile applications and web browsers. Integrating knowledge from Web Software Development, Special Topics in Object-Oriented Programming, and Object-Oriented Programming II, the project utilizes ASPX code for the interface in ASP.net, creates an Android prototype in XML, and implements database access in C#. This involves the creation of classes such as Client, Cart, Product, and Seller. The "CarritoRepository" class manages shopping carts. This work emphasizes not only innovation in the Marketplace System but also the practical application of acquired knowledge, resulting in a robust technological solution aligned with contemporary demands of the digital commerce landscape.

**Keywords:** Marketplace, Software Development, Object-Oriented Programming, Mobile Applications.



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Sobre o projeto</b>	<b>20</b>
1.1.1	Escopo do Projeto e PIM	20
<b>2</b>	<b>DESENVOLVIMENTO DE SOFTWARE PARA INTERNET</b>	<b>23</b>
<b>2.1</b>	<b>Fluxo de Usuários no Protótipo</b>	<b>23</b>
<b>2.2</b>	<b>Estruturação do Código ASPX</b>	<b>24</b>
<b>2.3</b>	<b>Visualização da Interface em Capturas de Tela</b>	<b>24</b>
2.3.0.1	Carrinho	24
2.3.0.2	Checkout	25
2.3.0.3	Login do Cliente	26
2.3.0.4	Layout Principal	27
2.3.0.5	Página de Produto	29
<b>3</b>	<b>TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS</b>	<b>31</b>
<b>3.1</b>	<b>Estruturação do Código XML</b>	<b>31</b>
<b>3.2</b>	<b>Exibição da Interface em Capturas de Tela</b>	<b>32</b>
3.2.1	Tela: carrinho.xml	32
3.2.2	Tela: checkout.xml	33
3.2.3	Tela: logincliente.xml	35
3.2.4	Tela: main_layout.xml	37
3.2.5	Tela: product.xml	39
<b>4</b>	<b>PROGRAMAÇÃO ORIENTADA A OBJETOS II</b>	<b>43</b>
4.0.1	Modelagem de Entidades e Relacionamentos no Banco de Dados	43
4.0.2	Estruturação do Código C# para Acesso às Classes e Entidades	45
4.0.3	Implementação da Classe Repositório 'CarrinhoRepository'	45
<b>5</b>	<b>CONCLUSÃO</b>	<b>47</b>
	<b>REFERÊNCIAS</b>	<b>49</b>

# Lista de ilustrações

Figura 1 – Ícone do grupo PIM (Desenvolvido pelos Autores). . . . .	20
Figura 2 – Logo da empresa-fantasia "Local Shopping"(Desenvolvido pelos Autores). . . . .	20
Figura 3 – Fluxo dos usuários "Comprador"e "Vendedor"(Desenvolvido pelos Autores). . . . .	23
Figura 4 – Tela de Carrinho (Desenvolvido pelos Autores). . . . .	24
Figura 5 – Tela de Checkout (Desenvolvido pelos Autores). . . . .	25
Figura 6 – Tela de Login (Desenvolvido pelos Autores). . . . .	26
Figura 7 – Tela de Início (Desenvolvido pelos Autores). . . . .	28
Figura 8 – Tela de Produto (Desenvolvido pelos Autores). . . . .	30
Figura 9 – IDE do Android Studio (Fonte: os Autores). . . . .	31
Figura 10 – Captura de Tela - Carrinho . . . . .	32
Figura 11 – Captura de Tela - Checkout . . . . .	34
Figura 12 – Captura de Tela - Login do Cliente . . . . .	35
Figura 13 – Captura de Tela - Layout Principal . . . . .	38
Figura 14 – Captura de Tela - Detalhes do Produto . . . . .	39

# Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
PIM	Projeto Integrado Multidisciplinar
IDE	Ambiente de desenvolvimento integrado
XML	eXtensible Markup Language
ASPX	Active Server Pages Extended



# 1 Introdução

A ascensão contínua da tecnologia tem remodelado significativamente o cenário comercial, impulsionando a transição para plataformas digitais que facilitam e agilizam as transações comerciais. Nesse contexto, o presente PIM (Projeto Integrado Multidisciplinar) propõe a concepção e implementação de um Sistema de Marketplace destinado à compra e venda de uma ampla variedade de produtos, acessível tanto por meio de aplicativos móveis quanto por navegadores web.

O foco principal deste projeto é a convergência de múltiplas disciplinas, destacando-se o Desenvolvimento de Software para Internet, Tópicos Especiais de Programação Orientada a Objetos e Programação Orientada a Objetos II. Para a disciplina de Desenvolvimento de Software para Internet, serão explorados os códigos ASPX para a criação da interface gráfica usando ASP.net, documentando o processo por meio de capturas de tela e a inclusão direta do código ASPX no trabalho.

No âmbito de Tópicos Especiais de Programação Orientada a Objetos, este TCC apresentará um protótipo de interface gráfica para dispositivos Android, elaborado por meio de XML. As capturas de tela e o código XML serão incorporados ao documento, proporcionando uma visão abrangente do desenvolvimento orientado a objetos voltado para interfaces mobile.

A disciplina de Programação Orientada a Objetos II será integralmente abordada com a implementação do código **C#** responsável pelo acesso ao Banco de Dados. Além disso, as classes de entidades essenciais para o sistema - Cliente, Carrinho, Produto e Vendedor - serão codificadas de maneira apropriada. A classe CarrinhoRepository, que desempenha um papel crucial na gestão dos carrinhos de compras, será desenvolvida e terá seus métodos implementados, assegurando a funcionalidade eficiente do sistema de Marketplace.

Este trabalho, assim, busca não apenas apresentar uma proposta de Sistema de Marketplace inovador, mas também evidenciar a aplicação prática dos conhecimentos adquiridos ao longo das disciplinas, culminando em uma solução tecnológica robusta e alinhada com as demandas contemporâneas do comércio digital.

Os Autores.

## 1.1 Sobre o projeto

### 1.1.1 Escopo do Projeto e PIM

Visando concretizar as disciplinas aprendidas neste bimestre, o corpo docente elaborou a proposta do PIM VIII. Este projeto representa a oportunidade de aplicar os conhecimentos adquiridos em Desenvolvimento de Software para Internet, Tópicos Especiais de Programação Orientada a Objetos e Programação Orientada a Objetos II.



Figura 1 – Ícone do grupo PIM (Desenvolvido pelos Autores).

A proposta do projeto abranje uma situação hipotética em que, como uma empresa terceirizada contratada para tal, ajudaremos na construção e desenvolvimento de certas partes de um Sistema Marketplace - empresa fantasia "Local Shopping- destinado à compra e venda de produtos diversos por meio de aplicativos móveis e plataformas web.



Figura 2 – Logo da empresa-fantasia "Local Shopping"(Desenvolvido pelos Autores).

---

Ao longo deste documento, exploraremos os detalhes da implementação, desde o fluxo do usuário dos protótipos até a estruturação do código em ASPX, XML e C#. Este trabalho não apenas demonstrará a aplicação dos conhecimentos teóricos, mas também evidenciará a capacidade de integrar esses conceitos em um projeto robusto e funcional, na prática.

O PIM VIII representa um desafio enriquecedor que permite a cada aluno contribuir para o desenvolvimento de uma solução tecnológica inovadora, alinhada com as demandas do mercado digital contemporâneo.

## 2 Desenvolvimento de Software para Internet

### 2.1 Fluxo de Usuários no Protótipo

A etapa inicial do desenvolvimento do Sistema de Marketplace envolve a análise detalhada do fluxo do usuário no protótipo. Nesta seção, serão explorados os caminhos que os usuários percorrerão ao interagir com a interface, destacando as diferentes etapas desde a busca de produtos até a conclusão da compra.

Com o propósito de proporcionar uma visão organizacional clara, elaboramos um diagrama representativo do fluxo do usuário, apresentado na Figura 3.

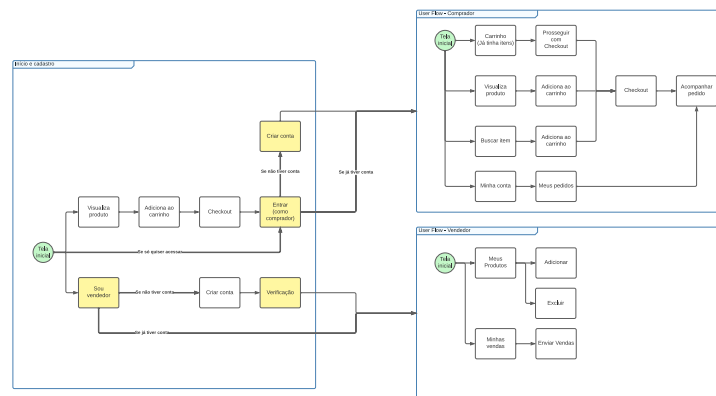


Figura 3 – Fluxo dos usuários "Comprador" e "Vendedor" (Desenvolvido pelos Autores).

Este diagrama oferece uma representação visual do percurso dos usuários, tanto "Comprador" quanto "Vendedor", dentro do protótipo. A análise detalhada desta representação é essencial para compreender as interações esperadas e identificar pontos cruciais de usabilidade. Dessa forma, estabelece-se uma base sólida para a próxima fase: a organização do código ASPX, onde traduziremos essa estrutura conceitual em implementação prática no Sistema de Marketplace.

O entendimento do fluxo do usuário é crucial para garantir uma experiência intuitiva e eficiente. Serão abordados aspectos como a navegação entre páginas, interações com elementos da interface, e feedbacks fornecidos ao usuário durante o processo de utilização.

Ao compreender o fluxo do usuário, estaremos preparados para a próxima etapa: a organização do código ASPX, onde transformaremos essa estrutura conceitual em uma implementação prática e funcional do Sistema de Marketplace.



## 2.2 Estruturação do Código ASPX

Na fase de desenvolvimento do Sistema de Marketplace, uma consideração crucial reside na estruturação eficaz do código ASPX, responsável por fornecer a interface gráfica através da plataforma ASP.net. A qualidade do código nessa etapa influencia diretamente a experiência do usuário e a eficiência operacional do sistema.

Nesta seção, exploraremos a organização do código ASPX, destacando princípios de design e as estratégias adotadas para garantir a coesão e a manutenibilidade do código. A compreensão detalhada dessa estrutura é fundamental para a transição suave da concepção do projeto para sua implementação prática.

## 2.3 Visualização da Interface em Capturas de Tela

### 2.3.0.1 Carrinho

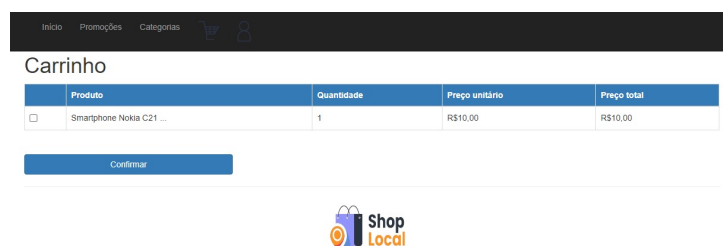


Figura 4 – Tela de Carrinho (Desenvolvido pelos Autores).

A tela do carrinho é o espaço virtual onde os usuários podem revisar e gerenciar os itens selecionados para compra. Nessa interface, eles podem adicionar, remover ou ajustar a quantidade de produtos no carrinho antes de prosseguir para o checkout.

```

1
2 <%@ Page Title="Carrinho" Language="C#" MasterPageFile="~/Site.Master"
   AutoEventWireup="true" CodeBehind="Carrinho.aspx.cs" Inherits="
   UNIPPIMVIII.Carrinho" %>
3
4 <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="
   server">
5     <h1>Carrinho</h1>
6     <table class="table table-bordered">
7         <thead>
8             <tr class="bg-primary">
9                 <th scope="col"></th>
10                <th scope="col">Produto</th>
11                <th scope="col">Quantidade</th>
12                <th scope="col">Preço unitário</th>
13                <th scope="col">Preço total</th>

```

```

14         </tr>
15     </thead>
16     <tbody>
17         <tr>
18             <th scope="row"><input type="checkbox" /></th>
19             <td>Smartphone Nokia C21 ...</td>
20             <td>1</td>
21             <td>R$10,00</td>
22             <td>R$10,00</td>
23         </tr>
24     </tbody>
25 </table>
26 <br />
27 <p>
28     <a runat="server" href="~/Checkout.aspx"><button type="button"
29     class="btn btn-primary" style="width:30%;height:auto;">Confirmar</
30 </asp:Content>

```

Listing 2.1 – Tela de Carrinho em ASPX

### 2.3.0.2 Checkout

The screenshot shows a web form titled "Checkout". It contains several input fields for user information: "Nome do remetente", "CEP", "Complemento", "Número", "Estado", and "Cidade". Each field is represented by a white rectangular box with a thin border. At the bottom of the form is a prominent blue button with the text "Confirmar" in white. The form is set against a light gray background with a subtle grid pattern.

Figura 5 – Tela de Checkout (Desenvolvido pelos Autores).

A tela de checkout é o ponto crucial onde os usuários finalizam suas compras. Aqui, eles fornecem informações de pagamento, endereço de entrega e revisam o resumo da compra antes de confirmar o pedido. Uma interface intuitiva é essencial para simplificar esse processo sensível.

```

1
2 <%@ Page Title="Checkout" Language="C#" MasterPageFile="~/Site.Master"
   AutoEventWireup="true" CodeBehind="Checkout.aspx.cs" Inherits="
   UNIPPIMVIII.Checkout" %>
3
4 <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="
   server">
5     <h1>Checkout</h1>
6
7     <p>Nome do remetente</p>
8     <input type="text" />
9     <p>CEP</p>
10    <input type="text" />
11    <p>Complemento</p>
12    <input type="text" />
13    <p>Número</p>
14    <input type="text" />
15    <p>Estado</p>
16    <input type="text" />
17    <p>Cidade</p>
18    <input type="text" />
19    <br />
20    <p>
21        <a runat="server" href="/">
22            <button type="button" class="btn btn-primary" style="width:
23            30%; height: auto;">Confirmar</button></a>
24 </asp:Content>

```

Listing 2.2 – Tela de Checkout em ASPX

### 2.3.0.3 Login do Cliente

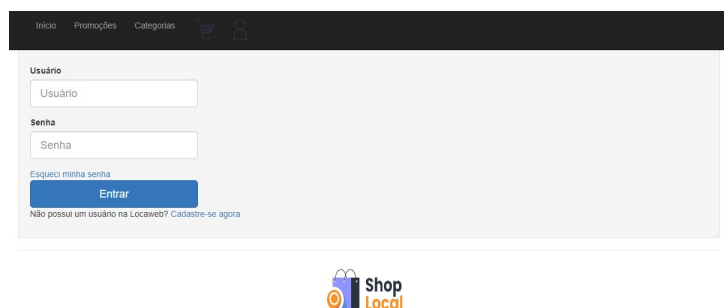


Figura 6 – Tela de Login (Desenvolvido pelos Autores).

A tela de login do cliente é a porta de entrada personalizada para usuários registrados. Aqui, eles inserem suas credenciais para acessar funcionalidades exclusivas, como histórico de compras, rastreamento de pedidos e configurações de conta.

```
1
2 <%@ Page Title="Login" Language="C#" MasterPageFile="~/Site.Master"
   AutoEventWireup="true" CodeBehind="LoginCliente.aspx.cs" Inherits="
   UNIPPIMVIII.LoginCliente" %>
3
4 <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="
   server">
5     <div class="box-parent-login">
6     <div class="well bg-white box-login">
7         <h1 class="ls-login-logo"></h1>
8         <form role="form">
9             <fieldset>
10
11                 <div class="form-group ls-login-user">
12                     <label for="userLogin">Usuário</label>
13                     <input class="form-control ls-login-bg-user input-lg" id="
14 userLogin" type="text" aria-label="Usuário" placeholder="Usuário">
15                 </div>
16
17                 <div class="form-group ls-login-password">
18                     <label for="userPassword">Senha</label>
19                     <input class="form-control ls-login-bg-password input-lg" id="
20 userPassword" type="password" aria-label="Senha" placeholder="Senha">
21                 </div>
22
23                 <a href="#" class="ls-login-forgot">Esqueci minha senha</a>
24
25                 <input type="submit" value="Entrar" class="btn btn-primary btn-
26 lg btn-block">
27                 <p class="txt-center ls-login-signup">Não possui um usuário na
28 Locaweb?
29                 <a href="#">Cadastre-se agora</a>
30                 </p>
31             </fieldset>
32         </form>
33     </div>
34 </div>
35 </asp:Content>
```

Listing 2.3 – Tela de Login em ASPX

#### 2.3.0.4 Layout Principal

O layout principal é a página inicial do Sistema de Marketplace, onde os usuários são recebidos com uma visão geral dos produtos em destaque, promoções e categorias. Essa tela desempenha um papel vital na navegação inicial e na exposição de produtos.

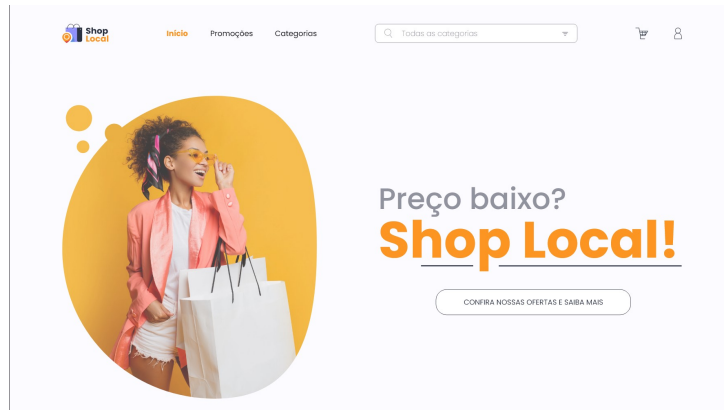


Figura 7 – Tela de Início (Desenvolvido pelos Autores).

```

1  <%@ Page Title="Pagina Inicial" Language="C#" MasterPageFile="~/Site.
2  Master" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits=
3  "UNIPPIMVIII._Default" %>
4  <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat
5  ="server">
6
7      <div class="row">
8          <div class="col-md-5" style="margin:auto; display:flex">
9              
10             </div>
11             <div class="col-md-7" style="height:311px;margin:auto;display:flex
12             ">
13                 
14             </div>
15         </div>
16         <br/>
17         <br/>
18         <br/>
19         <div class="row">
20             <h2>Roupas</h2>
21         </div>
22         <div class="row" >
23             <div class="col-md-3">
24                 <a runat="server" href="~/Produto.aspx"></a>
26             </div>
27             <div class="col-md-3">
28                 <a runat="server" href="~/Produto.aspx"></a>
30             </div>
31             <div class="col-md-3">

```

```

28     <a runat="server" href="~/Produto.aspx"></a>
30     </div>
31     <div class="col-md-3">
32         <a runat="server" href="~/Produto.aspx"></a>
34     </div>
35     <br />
36     <br />
37     <div class="row">
38         <h2>Celular</h2>
39     </div>
40     <div class="row" >
41         <div class="col-md-3">
42             <a runat="server" href="~/Produto.aspx"></a>
44         </div>
45         <div class="col-md-3">
46             <a runat="server" href="~/Produto.aspx"></a>
48         </div>
49         <div class="col-md-3">
50             <a runat="server" href="~/Produto.aspx"></a>
52         </div>
53     </div>
54 </asp:Content>
55

```

Listing 2.4 – Tela Principal em ASPX

### 2.3.0.5 Página de Produto

Ao acessar a página de um produto específico, os usuários encontram detalhes abrangentes sobre o item, incluindo descrição, preço, avaliações e opções de compra. Uma interface clara e informativa é crucial para influenciar as decisões de compra nessa fase crítica.

```

1
2 <%@ Page Title="Produto" Language="C#" MasterPageFile="~/Site.Master"
   AutoEventWireup="true" CodeBehind="Produto.aspx.cs" Inherits="
   UNIPPIMVIII.Produto" %>

```



Figura 8 – Tela de Produto (Desenvolvido pelos Autores).

```

3
4 <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat
  ="server">
5   <div class="row">
6     <div class="col-md-7">
7       <h1>Smartphone Nokia c21 Plus 4g 128GB Tela HD+ 6,5 Câmera Dupla
        13MP Android Bateria de 2 dias de duração +</h1>
8       <h2>R$100,00</h2>
9       <h3>Em até 12x sem juros</h3>
10      <p>
11        <a runat="server" href="Carrinho.aspx"><button type="button"
          class="btn btn-primary" style="margin:auto;width:70%;height:auto;">
            Adicionar ao Carrinho</button></a>
12      </p>
13      <br/>
14      <a runat="server" href="Checkout.aspx"><button type="button"
          class="btn btn-light" style="margin:auto;width:70%;height:auto;">
            Comprar</button></a>
15    </div>
16    <div class="col-md-5">
17      
18    </div>
19  </div>
20 </asp:Content>
21

```

Listing 2.5 – Tela de Produto em ASPX

## 3 Tópicos Especiais de Programação Orientada a Objetos

### 3.1 Estruturação do Código XML

Com base na análise do fluxo do usuário para sistemas mobile, procedemos à criação do protótipo de baixa fidelidade utilizando o Android Studio como ambiente de desenvolvimento integrado (IDE), Kotlin como linguagem de programação e XML como linguagem de marcação.

Para otimizar a organização e manutenção do código, optamos por uma abordagem que divide as telas em arquivos XML distintos, proporcionando modularidade e clareza estrutural. Ao todo, foram desenvolvidas cinco telas, cada uma correspondendo a uma etapa específica do fluxo do usuário "Comprador".

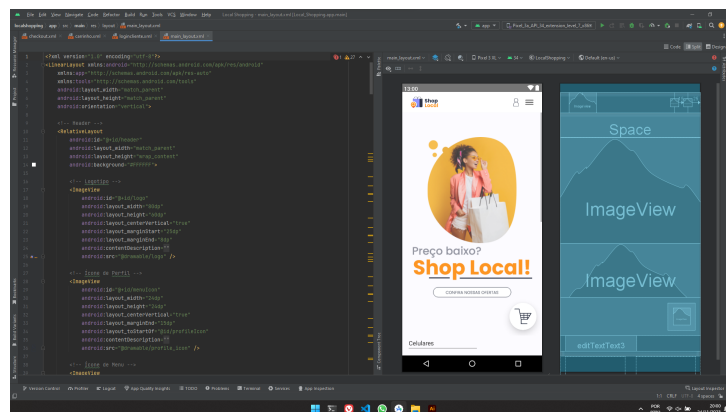


Figura 9 – IDE do Android Studio (Fonte: os Autores).

A estruturação modular permitiu uma implementação mais eficiente, facilitando a manutenção e aprimoramento de cada componente da interface. Cada tela foi cuidadosamente projetada e, posteriormente, vinculada aos seus respectivos arquivos em Kotlin, estabelecendo uma conexão fluida entre a lógica de programação e a apresentação visual.

Este enfoque na estruturação do código XML não apenas contribui para a coesão e clareza do código-fonte, mas também simplifica a implementação de futuras atualizações e a incorporação de novas funcionalidades ao Sistema de Marketplace para dispositivos móveis. Na próxima seção, exploraremos mais a fundo a integração desses elementos na programação Kotlin, detalhando como cada tela foi associada à lógica do aplicativo.



## 3.2 Exibição da Interface em Capturas de Tela

Nesta seção, apresentaremos capturas de tela que ilustram a interface do Sistema de Marketplace em sua fase de prototipagem para dispositivos móveis. Cada imagem corresponde a uma tela específica, proporcionando uma visão visual abrangente do design e da disposição dos elementos em diferentes etapas do processo.

### 3.2.1 Tela: carrinho.xml

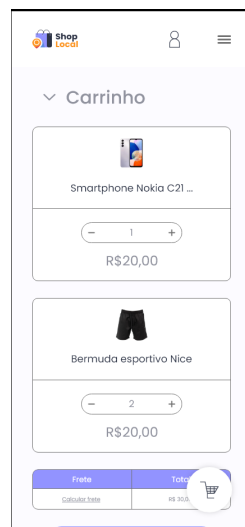


Figura 10 – Captura de Tela - Carrinho

A tela de carrinho exibe os produtos selecionados pelo usuário "Comprador", oferecendo uma visão consolidada das escolhas feitas durante a sessão de compras.

```

1
2 <?xml version="1.0" encoding="utf-8"?>
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical">
9
10    <!-- Header -->
11    <RelativeLayout
12        android:id="@+id/header"
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:background="#FFFFFF">
16
17        <!-- Logotipo -->
18        <ImageView

```

```
19         android:id="@+id/logo"
20         android:layout_width="80dp"
21         android:layout_height="60dp"
22         android:layout_centerVertical="true"
23         android:layout_marginStart="25dp"
24         android:layout_marginEnd="8dp"
25         android:contentDescription="@string/logo"
26         android:src="@drawable/logo" />
27
28     <!-- Ícone de Perfil -->
29     <ImageView
30         android:id="@+id/menuIcon"
31         android:layout_width="24dp"
32         android:layout_height="24dp"
33         android:layout_centerVertical="true"
34         android:layout_marginEnd="15dp"
35         android:layout_toStartOf="@id/profileIcon"
36         android:contentDescription="@string/menu"
37         android:src="@drawable/profile_icon" />
38
39     <!-- Ícone de Menu -->
40     <ImageView
41         android:id="@+id/profileIcon"
42         android:layout_width="24dp"
43         android:layout_height="24dp"
44         android:layout_alignParentEnd="true"
45         android:layout_centerVertical="true"
46         android:layout_marginEnd="25dp"
47         android:contentDescription="@string/test"
48         android:src="@drawable/menu_icon" />
49
50     </RelativeLayout>
51
52     <!-- Restante do código mandado em Anexo com esse documento -->
53
54 </LinearLayout>
```

Listing 3.1 – Exemplo de Código XML

### 3.2.2 Tela: checkout.xml

A tela de checkout apresenta o resumo da compra, permitindo ao usuário revisar os itens escolhidos, ajustar quantidades e proceder ao pagamento.

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <?xml version="1.0" encoding="utf-8"?>
```

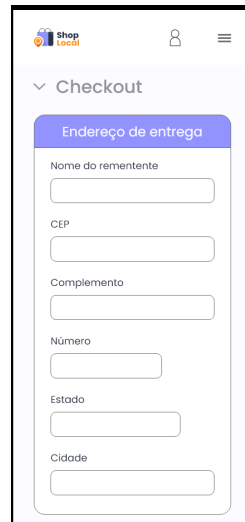


Figura 11 – Captura de Tela - Checkout

```

4  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
5
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:orientation="vertical">
9
10     <!-- Header -->
11     <RelativeLayout
12         android:id="@+id/header"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:background="#FFFFFF">
16
17         <!-- Logotipo -->
18         <ImageView
19             android:id="@+id/logo"
20             android:layout_width="80dp"
21             android:layout_height="60dp"
22             android:layout_centerVertical="true"
23             android:layout_marginStart="25dp"
24             android:layout_marginEnd="8dp"
25             android:contentDescription="@string/logo"
26             android:src="@drawable/logo" />
27
28         <!-- Ícone de Perfil -->
29         <ImageView
30             android:id="@+id/menuIcon"
31             android:layout_width="24dp"
32             android:layout_height="24dp"
33             android:layout_centerVertical="true"

```

```

34     android:layout_marginEnd="15dp"
35     android:layout_toStartOf="@id/profileIcon"
36     android:contentDescription="@string/menu"
37     android:src="@drawable/profile_icon" />
38
39     <!-- Ícone de Menu -->
40     <ImageView
41         android:id="@+id/profileIcon"
42         android:layout_width="24dp"
43         android:layout_height="24dp"
44         android:layout_alignParentEnd="true"
45         android:layout_centerVertical="true"
46         android:layout_marginEnd="25dp"
47         android:contentDescription="@string/test"
48         android:src="@drawable/menu_icon" />
49
50 </RelativeLayout>
51
52 <!-- Restante do código mandado em Anexo com esse documento -->
53
54 </LinearLayout>

```

Listing 3.2 – Código XML da tela de Checkout

### 3.2.3 Tela: logincliente.xml

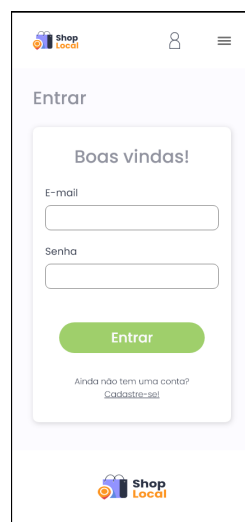


Figura 12 – Captura de Tela - Login do Cliente

A tela de login do cliente é o ponto de autenticação, onde o usuário pode acessar sua conta para personalizar a experiência de compra.

```

1
2 <?xml version="1.0" encoding="utf-8"?>

```

```
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
4   xmlns:app="http://schemas.android.com/apk/res-auto"
5   xmlns:tools="http://schemas.android.com/tools"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   android:orientation="vertical">
9
10  <!-- Header -->
11  <RelativeLayout
12      android:id="@+id/header"
13      android:layout_width="match_parent"
14      android:layout_height="wrap_content"
15      android:background="#FFFFFF">
16
17      <!-- Logotipo -->
18      <ImageView
19          android:id="@+id/logo"
20          android:layout_width="80dp"
21          android:layout_height="60dp"
22          android:layout_centerVertical="true"
23          android:layout_marginStart="25dp"
24          android:layout_marginEnd="8dp"
25          android:contentDescription="@string/logo"
26          android:src="@drawable/logo" />
27
28      <!-- Ícone de Perfil -->
29      <ImageView
30          android:id="@+id/menuIcon"
31          android:layout_width="24dp"
32          android:layout_height="24dp"
33          android:layout_centerVertical="true"
34          android:layout_marginEnd="15dp"
35          android:layout_toStartOf="@id/profileIcon"
36          android:contentDescription="@string/menu"
37          android:src="@drawable/profile_icon" />
38
39      <!-- Ícone de Menu -->
40      <ImageView
41          android:id="@+id/profileIcon"
42          android:layout_width="24dp"
43          android:layout_height="24dp"
44          android:layout_alignParentEnd="true"
45          android:layout_centerVertical="true"
46          android:layout_marginEnd="25dp"
47          android:contentDescription="@string/test"
48          android:src="@drawable/menu_icon" />
```

```
49
50     </RelativeLayout>
51
52     <ScrollView
53         android:layout_width="match_parent"
54         android:layout_height="match_parent"
55         android:background="#F1F0F3">
56
57         <LinearLayout
58             android:layout_width="match_parent"
59             android:layout_height="wrap_content"
60             android:orientation="vertical">
61
62             <TextView
63                 android:layout_width="wrap_content"
64                 android:layout_height="wrap_content"
65                 android:layout_margin="100px"
66                 android:padding="8dp"
67                 android:text="Entrar"
68                 android:textColor="#91939F"
69                 android:textSize="20sp"
70                 android:textStyle="bold" />
71
72
73         <!-- Restante do código mandado em Anexo com esse documento -->
74
75     </LinearLayout>
```

Listing 3.3 – Código da tela de Login

### 3.2.4 Tela: main\_layout.xml

A tela principal, ou `main_layout.xml`, é a interface inicial do aplicativo, apresentando opções de navegação e categorias de produtos.

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical">
9
10    <!-- Header -->
11    <RelativeLayout
12        android:id="@+id/header"
```

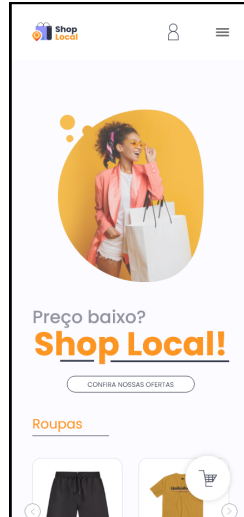


Figura 13 – Captura de Tela - Layout Principal

```

13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:background="#FFFFFF">
16
17     <!-- Logotipo -->
18     <ImageView
19         android:id="@+id/logo"
20         android:layout_width="80dp"
21         android:layout_height="60dp"
22         android:layout_centerVertical="true"
23         android:layout_marginStart="25dp"
24         android:layout_marginEnd="8dp"
25         android:contentDescription="@string/logo"
26         android:src="@drawable/logo" />
27
28     <!-- Ícone de Perfil -->
29     <ImageView
30         android:id="@+id/menuIcon"
31         android:layout_width="24dp"
32         android:layout_height="24dp"
33         android:layout_centerVertical="true"
34         android:layout_marginEnd="15dp"
35         android:layout_toStartOf="@id/profileIcon"
36         android:contentDescription="@string/menu"
37         android:src="@drawable/profile_icon" />
38
39     <!-- Ícone de Menu -->
40     <ImageView
41         android:id="@+id/profileIcon"
42         android:layout_width="24dp"
43         android:layout_height="24dp"

```

```

44         android:layout_alignParentEnd="true"
45         android:layout_centerVertical="true"
46         android:layout_marginEnd="25dp"
47         android:contentDescription="@string/test"
48         android:src="@drawable/menu_icon" />
49
50     </RelativeLayout>
51
52
53     <!-- Restante do código mandado em Anexo com esse documento -->
54
55 </LinearLayout>

```

Listing 3.4 – ECódigo Layout Principal do portal

### 3.2.5 Tela: product.xml

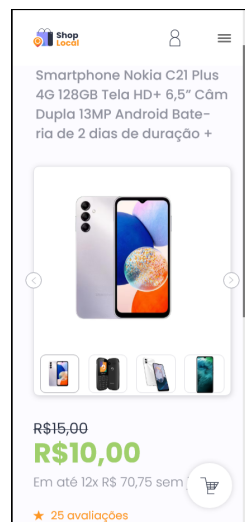


Figura 14 – Captura de Tela - Detalhes do Produto

A tela de detalhes do produto exibe informações específicas sobre um item selecionado, proporcionando ao usuário uma visão mais aprofundada antes da compra.

```

1
2 <?xml version="1.0" encoding="utf-8"?>
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical">
9
10 <!-- Header -->

```



```
11 <RelativeLayout
12     android:id="@+id/header"
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:background="#FFFFFF">
16
17     <!-- Logotipo -->
18     <ImageView
19         android:id="@+id/logo"
20         android:layout_width="80dp"
21         android:layout_height="60dp"
22         android:layout_centerVertical="true"
23         android:layout_marginStart="25dp"
24         android:layout_marginEnd="8dp"
25         android:contentDescription="@string/logo"
26         android:src="@drawable/logo" />
27
28     <!-- Ícone de Perfil -->
29     <ImageView
30         android:id="@+id/menuIcon"
31         android:layout_width="24dp"
32         android:layout_height="24dp"
33         android:layout_centerVertical="true"
34         android:layout_marginEnd="15dp"
35         android:layout_toStartOf="@id/profileIcon"
36         android:contentDescription="@string/menu"
37         android:src="@drawable/profile_icon" />
38
39     <!-- Ícone de Menu -->
40     <ImageView
41         android:id="@+id/profileIcon"
42         android:layout_width="24dp"
43         android:layout_height="24dp"
44         android:layout_alignParentEnd="true"
45         android:layout_centerVertical="true"
46         android:layout_marginEnd="25dp"
47         android:contentDescription="@string/test"
48         android:src="@drawable/menu_icon" />
49
50 </RelativeLayout>
51
52 <ScrollView
53     android:layout_width="match_parent"
54     android:layout_height="match_parent"
55     android:background="#FCFBFF">
56
57     <LinearLayout
```

```
58     android:layout_width="match_parent"
59     android:layout_height="wrap_content"
60     android:orientation="vertical">
61
62     <TextView
63         android:layout_margin="100px"
64         android:layout_width="wrap_content"
65         android:layout_height="wrap_content"
66         android:text="Smartphone Nokia C21 Plus 4G 128GB Tela HD+ 6,5
        Câmera Dupla 13MP Android Bateria de 2 dias de duração +"
67         android:textColor="#91939F"
68         android:textSize="20sp"
69         android:textStyle="bold"
70         android:padding="8dp"/>
71
72     <androidx.cardview.widget.CardView
73         android:layout_width="match_parent"
74         android:layout_height="wrap_content"
75         android:layout_margin="16dp"
76         android:cardBackgroundColor="#ffffff"
77         android:layout_gravity="center"
78         android:radius="16dp">
79
80     <!-- Restante do código mandado em Anexo com esse documento -->
81
82 </LinearLayout>
```

Listing 3.5 – Exemplo de Código XML

Essas capturas de tela oferecem uma representação visual do cuidadoso design da interface do usuário, facilitando a compreensão da experiência que o Sistema de Marketplace busca proporcionar aos usuários móveis.

## 4 Programação Orientada a Objetos II

O capítulo de Programação Orientada a Objetos II concentra-se em aspectos fundamentais para o desenvolvimento eficiente do Sistema de Marketplace proposto. Abordaremos a modelagem de entidades e relacionamentos no banco de dados, a estruturação do código em C# para acesso às classes e entidades, além da implementação crucial da classe repositório 'CarrinhoRepository'.

### 4.0.1 Modelagem de Entidades e Relacionamentos no Banco de Dados

Antes de adentrar a implementação em código, é imperativo estabelecer uma sólida modelagem de entidades e seus relacionamentos no banco de dados. Este passo é crucial para a integridade e eficiência do sistema. A definição clara das entidades, como Cliente, Carrinho, Produto e Vendedor, juntamente com seus relacionamentos, moldará a estrutura do banco de dados e, por consequência, a dinâmica do Sistema de Marketplace.

```
1  using System ;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  public class Cliente
8  {
9      public int Id { get; set; }
10     public string Nome { get; set; }
11     public string Cpf { get; set; }
12     public string Email { get; set; }
13     public string Senha { get; set; }
14     public string Endereco { get; set; }
15
16     public Carrinho Carrinho { get; set; }
17 }
18
19 public class Carrinho
20 {
21     public int Id { get; set; }
22     public DateTime DataPedido { get; set; }
23     public decimal ValorTotal { get; set; }
24     public string StatusPedido { get; set; }
25
26     public Cliente Cliente { get; set; }
27     public List<Produto> Produtos { get; set; }
```

```
28 }
29
30 public class Produto
31 {
32     public int Id { get; set; }
33     public string Descricao { get; set; }
34     public decimal Preco { get; set; }
35     public string Imagem { get; set; }
36     public string Status { get; set; }
37
38     public Vendedor Vendedor { get; set; }
39     public Categoria Categoria { get; set; }
40 }
41
42 public class Vendedor
43 {
44     public int Id { get; set; }
45     public string RazaoSocial { get; set; }
46     public string NomeFantasia { get; set; }
47     public string Cnpj { get; set; }
48     public string Email { get; set; }
49     public string Senha { get; set; }
50     public string Endereco { get; set; }
51     public decimal Comissao { get; set; }
52 }
```

Listing 4.1 – Entidades do código C#

Para representar a conexão das diversas Entidades com o Banco de Dados também desenvolvemos um código C# presente no seguinte trecho:

```
1 using Microsoft.EntityFrameworkCore;
2
3 public class LojaContext : DbContext
4 {
5     public DbSet<Cliente> Clientes { get; set; }
6     public DbSet<Carrinho> Carrinhos { get; set; }
7     public DbSet<Produto> Produtos { get; set; }
8     public DbSet<Vendedor> Vendedores { get; set; }
9     public DbSet<Categoria> Categorias { get; set; }
10
11     protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
12     {
13         optionsBuilder.UseSqlServer("Data Source=(localdb)\\MSSQLLocalDB;
Initial Catalog=LojaDB;Integrated Security=True");
14     }
```

15 }

## Listing 4.2 – Conexão com a Data Base

#### 4.0.2 Estruturação do Código C# para Acesso às Classes e Entidades

Com a modelagem de dados consolidada, avançaremos para a estruturação do código em C#. Este tópico abordará a forma como as classes e entidades definidas anteriormente serão acessadas e manipuladas no ambiente do Sistema de Marketplace. A clareza e eficácia dessa estruturação são cruciais para garantir a coerência entre a representação do modelo no código e no banco de dados.

#### 4.0.3 Implementação da Classe Repositório 'CarrinhoRepository'

A implementação da classe 'CarrinhoRepository' desempenha um papel central na gestão dos carrinhos de compras. Essa classe é responsável por operações como adicionar itens, remover produtos, calcular o total da compra e outras ações relacionadas ao carrinho do cliente. A eficiência e correção dessa implementação influenciam diretamente a experiência do usuário ao utilizar o Sistema de Marketplace.

No decorrer deste capítulo, cada tópico será explorado detalhadamente, proporcionando uma compreensão abrangente da Programação Orientada a Objetos II no contexto do desenvolvimento do Sistema de Marketplace. O objetivo é não apenas implementar funcionalidades, mas também assegurar a manutenibilidade e escalabilidade do sistema ao longo do tempo.

```
1 public class CarrinhoRepository
2 {
3     private readonly LojaContext _context;
4
5     public CarrinhoRepository(LojaContext context)
6     {
7         _context = context;
8     }
9
10    public async Task AdicionarCarrinhoAsync(Carrinho carrinho)
11    {
12        await _context.Carrinhos.AddAsync(carrinho);
13        await _context.SaveChangesAsync();
14    }
15
16    public async Task AtualizarCarrinhoAsync(Carrinho carrinho)
17    {
18        _context.Entry(carrinho).State = EntityState.Modified;
19        await _context.SaveChangesAsync();
20    }
21 }
```

```
20     }
21
22     public async Task ExcluirCarrinhoAsync(Carrinho carrinho)
23     {
24         _context.Carrinhos.Remove(carrinho);
25         await _context.SaveChangesAsync();
26     }
27
28     public async Task<Carrinho> ObterPorIdAsync(int id)
29     {
30         return await _context.Carrinhos.FindAsync(id);
31     }
32
33     public async Task<List<Carrinho>> ObterTodosAsync()
34     {
35         return await _context.Carrinhos.ToListAsync();
36     }
37 }
```

Listing 4.3 – DataBase

## 5 Conclusão

O desenvolvimento acelerado da tecnologia continua a transformar de maneira expressiva o panorama comercial, guiando a transição para plataformas digitais que simplificam e agilizam as transações comerciais. Em resposta a esse cenário dinâmico, o Projeto Integrado Multidisciplinar (PIM) propôs e delineou a concepção e implementação de um Sistema de Marketplace, oferecendo uma abordagem abrangente para a compra e venda de uma extensa gama de produtos, acessível tanto por aplicativos móveis quanto por navegadores web.

O cerne deste projeto reside na convergência de disciplinas cruciais, com destaque para Desenvolvimento de Software para Internet, Tópicos Especiais de Programação Orientada a Objetos e Programação Orientada a Objetos II. No âmbito do Desenvolvimento de Software para Internet, a exploração dos códigos ASPX para a criação da interface gráfica usando ASP.net foi detalhada, com documentação por meio de capturas de tela e inclusão direta do código ASPX neste trabalho.

Na esfera dos Tópicos Especiais de Programação Orientada a Objetos, este trabalho apresentou um protótipo de interface gráfica para dispositivos Android, desenvolvido em XML. As capturas de tela e o código XML integrados ao documento oferecem uma visão holística do desenvolvimento orientado a objetos voltado para interfaces móveis.

A Programação Orientada a Objetos II foi abordada de maneira abrangente, com a implementação do código C# responsável pelo acesso ao Banco de Dados. As classes de entidades fundamentais para o sistema - Cliente, Carrinho, Produto e Vendedor - foram codificadas de maneira apropriada. A classe CarrinhoRepository, crucial na gestão dos carrinhos de compras, foi desenvolvida com seus métodos implementados, assegurando a eficiência funcional do Sistema de Marketplace.

Este trabalho, além de apresentar uma proposta inovadora de Sistema de Marketplace, destaca a aplicação prática dos conhecimentos adquiridos ao longo das disciplinas. A solução tecnológica resultante é robusta e alinhada com as exigências contemporâneas do comércio digital. Ao integrar teoria e prática, este projeto não apenas representa uma conquista acadêmica, mas também ressalta a relevância e a aplicabilidade dos conceitos explorados.

Os autores, ao concluírem este projeto, reafirmam o compromisso com a excelência acadêmica e com a busca contínua por soluções inovadoras no campo da tecnologia. Este Sistema de Marketplace não é apenas o resultado de esforços acadêmicos, mas uma contribuição tangível para a evolução do cenário comercial digital.





# Referências

ROBSON CAMARGO. PMBOK: O que é e qual sua importância no gerenciamento de projetos? Disponível em: <<https://robsoncamargo.com.br/blog/PMBOK>> . Acesso em 2023. Nenhuma citação no texto.

TOTVS. O que é PMBOK e como ele auxilia no gerenciamento de projetos? Disponível em: <<https://www.totvs.com/blog/negocios/pmbok/>>. Acesso em 2023. Nenhuma citação no texto.

AEVO. PMBOK: Guia completo para entender tudo sobre a metodologia! Disponível em: <<https://blog.aevo.com.br/pmbok-guia-completo/>> Acesso em 2023. Nenhuma citação no texto.

SA E FARIAS. GABRIEL: Fluxo de usuário (User Flow): O que é? Como fazer? Disponível em: <<https://medium.com/7bits/fluxo-de-usuario-user-flow-o-que-como-fazer>> Acesso em 2023. Nenhuma citação no texto.