

# **Sistema WEB de Venda e Divulgação de Produtos e Serviços**

**Gabriel Pedro Kreusch<sup>1</sup>, Hylson Vescovi Netto<sup>2</sup>**

<sup>1</sup>Fácil Informática Ltda. – Brasil

<sup>2</sup>Instituto Federal Catarinense - Campus Blumenau – Brasil

[gbr.kreusch@gmail.com](mailto:gbr.kreusch@gmail.com), [hylson.vescovi@ifc.edu.br](mailto:hylson.vescovi@ifc.edu.br)

## **1. Introdução**

Com a revolução dos computadores e o crescimento explosivo da Tecnologia da Informação (TI) fica, a cada segundo que passa, mais difícil de se encontrar alguém que não esteja conectado. Todos os dias terabytes e terabytes de informações são trafegados na rede no Brasil (SCHEIFER, 2020), e a conectividade é tão intensa que as gerações mais novas não veem um mundo sem internet. Mais comumente computadores, smartphones, televisores, e, mais recentemente, refrigeradores, aspiradores de pó autônomos e até máquinas de lavar vem se conectando à rede de algum modo para facilitar atividades rotineiras, ao ponto que, do seu trabalho, na empresa, com poucos cliques você pode ordenar que o seu aspirador de pó faça uma faxina na casa, ou que a cafeteira comece a passar seu café para estar pronto quando chegar em casa. São infinitas as aplicabilidades dessa tecnologia, e é óbvio que o mercado de trabalho não iria ficar de fora, diversas áreas exploraram esse meio e se reinventaram na última década.

O avanço da internet sobre o mercado de trabalho, nas mais variadas áreas, é iminente e inevitável, ao ponto de existirem profissões exclusivas do meio digital, como: criadores de conteúdos de entretenimento, seja no Youtube, Instagram ou Tik Tok, e afins; jogadores profissionais de jogos eletrônicos; serviços de criação e hospedagem de Websites; mercados de vendas de itens digitais e muito mais. Além disso, muitas oportunidades foram viabilizadas graças à rede mundial de computadores, hoje em dia é perfeitamente possível um brasileiro trabalhar para uma empresa norte-americana sem sair de seu país de origem. A recente pandemia da COVID-19, deixou isso ainda mais evidente, em uma situação em que era preciso preservar a vida dos funcionários ao mesmo tempo em que existe a necessidade de manter as atividades econômicas, observou-se que muitas das funções na empresa poderiam ser exercidas de forma remota pelo colaborador a partir da moradia do mesmo, mantendo a eficiência demonstrada no ambiente da empresa ao mesmo tempo em que são evidentes as economias em questão de recursos e espaço da parte do empregador. Uma das áreas que mais se reinventou com a expansão tecnológica foi a de vendas.

Desde o seu princípio as redes de computadores vem ampliando horizontes quando se trata de comunicação agilizada e de longa distância, características muito proveitosas quando o assunto é vender, seja um produto, ou um serviço. E, sem dúvidas, esse ramo se beneficiou muito disso, grandes, médias e, até, pequenas empresas de vendas tiveram qualquer tipo de barreira quebrada e se tornaram internacionais, com

plataforma própria ou através de plataformas de terceiros, como o Mercado Livre, a Aliexpress, o eBay, Wish, o próprio Facebook, e outras redes sociais, e muitas outras, construindo uma ponte direta entre o comprador e o comerciante. Esses são os principais conceitos que se pretende alcançar com esse trabalho, a disponibilidade, o alcance e a facilidade. Por meio de um website simples, intuitivo, customizável, expansível e unificado, pretendemos viabilizar uma alternativa para que os comércios instalem sua própria plataforma online de vendas, ao mesmo tempo em que tem o controle do seu negócio com estatísticas e dados operacionais, expandindo seus horizontes e aumentando a eficiência da gestão.

### **1.1. Tema/Problema**

Imagine, você, um pequeno comerciante no ramo de informática, vendendo equipamentos e peças de computadores e impressoras, e suprimentos, e prestando serviços de manutenção nos mesmos. Esse comerciante toca sozinho, com ajuda de sua esposa, o negócio da família. Esses dois indivíduos realizam todas as atividades envolvendo sua empresa em todos os níveis organizacionais, das vendas e prática dos serviços, até as atividades administrativas e estratégicas. Suas vendas são realizadas por telefone, redes sociais e plataformas de terceiros, os registros dessas operações se dá por meio de notas fiscais, geradas a partir de um sistema desktop, na máquina de trabalho da esposa, os serviços prestados e a venda de produtos semi novos e restaurados, também praticados por esses comerciantes, fica registrada apenas em conversas com o comprador ou na memória do casal.

Agora, pense o quão proveitoso seria um sistema unificado de venda e gerenciamento da atividade econômica dessa loja. Provavelmente ajudaria bastante, não? Esses pequenos comerciantes são nosso principal foco neste trabalho, e seus problemas são, também, nossos problemas. Desse modo, pretende-se atingir uma solução eficaz que tenha plena serventia ao nosso alvo, auxiliando no desenvolvimento e expansão do mesmo.

### **1.2. Objetivos Propostos/Solução dos Problemas**

O objetivo deste trabalho é desenvolver um sistema web que permita anunciar itens e serviços, bem como realizar o processo de venda, através de um módulo e-commerce; permitir o gerenciamento de estoque, entrada e saída dos produtos no estoque. A ideia futura para o sistema é abranger diversas áreas do negócio e permitir a fácil incorporação de novos módulos, como um sistema ERP (do inglês Enterprise Resource Planning, ou Planejamento de Recursos da Empresa em português), para que, posteriormente, seja obtida uma plataforma unificada de operação, gerenciamento e planejamento da empresa como um todo. Com esse projeto, espera-se obter um sistema que abranja toda a parte comercial, desde o controle e a quantificação dos estoques, até o processo de vendas online, tudo isso devidamente estruturado pensando na expansão para as demais áreas do negócio e do gerenciamento da empresa em si.

### **1.3. Escopo**

O sistema contará com um cadastro de produtos, realizado pelo administrador do sistema. Será possível, também, importar os produtos cadastrados no sistema de controle de estoque utilizado atualmente pelo cliente.

O sistema conterá um catálogo de produtos que irá dispor os itens cadastrados.

Será possível filtrar os itens do catálogo por categoria de produto.

Haverá uma página de contato contendo informações como, telefone, endereço no Google Maps e uma janela de e-mail rápida.

O sistema conterá um módulo de e-Commerce, onde o usuário poderá acessar o catálogo e adicionar ao carrinho os produtos desejados, para posteriormente fechar o pedido e efetuar o pagamento.

#### **1.4. Viabilidade do Projeto**

É pretendido, com este projeto, conceder uma plataforma unificada de venda e anúncio de produtos e serviços a pequenos e médios comerciantes, pretendendo oferecer uma plataforma unificada para venda e projeção do seu negócio.

O sistema busca ter uma interface simples e amigável, responsiva e adaptável a diversos dispositivos, visto que, por se tratar de uma aplicação *web*, se encontra disponível em qualquer plataforma.

#### **1.5. Método de Trabalho (arquitetura, ferramentas, tecnologias aplicadas)**

O sistema será desenvolvido utilizando a plataforma ASP.NET Core, em sua versão 3.1, da internacionalmente conhecida Microsoft, e, portanto, utilizará da linguagem de programação também desenvolvida e distribuída pela empresa, o C# (ou “C sharp”, como se lê) e será desenvolvido, por tanto, utilizando do paradigma de orientação a objetos. Para isso, utilizaremos a IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) Visual Studio 2019, igualmente criada e distribuída pela Microsoft, em sua versão *Community* (distribuída gratuitamente à comunidade).

O IDE é um programa de computador, geralmente utilizado para aumentar a produtividade dos desenvolvedores de software, bem como a qualidade desses produtos. Podem auxiliar, através de ferramentas e características, na redução de erros e na aplicação de técnicas como o RAD (Rapid Application Development) (DOS SANTOS, 2007).

Para a persistência das informações trafegadas neste sistema, como cadastro de itens e fluxos de vendas, utilizaremos o banco de dados Microsoft SQL Server. Este, por ser, também, desenvolvido e distribuído pela mesma empresa responsável pela plataforma e a linguagem de programação, é de fácil integração com essas ferramentas, sem mencionar que é um dos grandes nomes quando se fala em banco de dados relacional.

Ainda relacionado com a base de dados, ficará a cargo da biblioteca Entity Framework Core (ENTITYCORE, 2022), em sua versão 3.1.22, o gerenciamento do acesso ao banco e a automática criação das tabelas.

O Entity Framework Core trabalha partindo de modelos básicos de classes

seguindo o padrão POCO (sigla do inglês “Plain Old CRL Objects”), possuindo uma estrutura simples e independente de ligações com *frameworks* ORM, incluindo o Entity, permitindo que essas entidades sejam reutilizadas em outras possíveis bibliotecas de manipulação de informações provenientes da base de dados sem quaisquer problemas (GROFFE, 2014).

Para essa aplicação, adotamos o padrão de projetos MVC (*Model View Controller*, do inglês) que consiste na divisão e estruturação dos componentes de acordo com o seu papel para com o sistema. Neste sentido, a *View* se trata da parte visual da aplicação, o *Model* refere-se aos modelos de objetos, e, por fim, o *Controller* se trata da parte de acesso ao banco de dados.

As *views* do sistema foram construídas utilizando Razor Pages, tecnologia de integração de código fonte com a estruturação HTML da página *web*. Desenvolveremos as páginas de fórmula modularizada, criando componentes Blazor, e os integrando as *views* criando páginas altamente interativas, customizáveis e reaproveitáveis.

Obviamente que os três pilares de toda aplicação *web* não ficariam de fora desta, assim, contamos com as tecnologias HTML 5, Javascript e CSS. Contando com a ajuda do Bootstrap e JQuery para conceder beleza, praticidade e dinamismo ao projeto.

## 2. Trabalhos Correlatos Existentes

Este trabalho foi inspirado em sites de vendas e *marketplaces* como a Amazon, tentando reunir o melhor da experiência do usuário na utilização das interfaces e operações. Do mesmo modo como o site da Amazon (Figura 1), por exemplo, nesse projeto é possível consultar e comprar produtos e serviços, mas, diferente da Amazon, este sistema busca ir além.

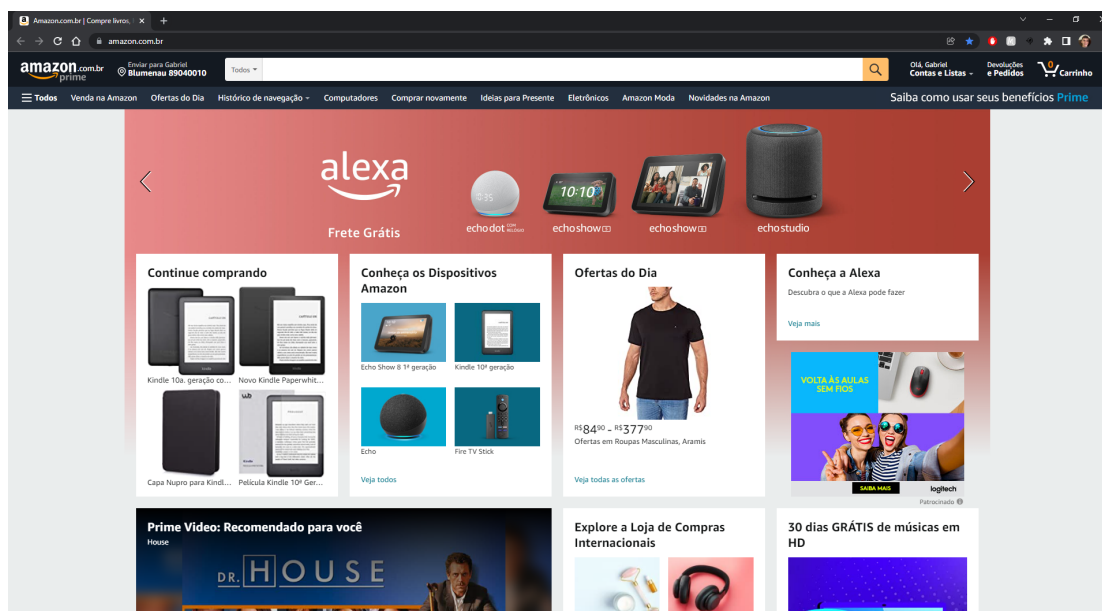


Figura 1 - Demonstração site da Amazon

É pretendido — e tudo que foi desenvolvido até aqui foi pensando nisso — uma expansão do sistema que permita o controle total do negócio, do anúncio e venda dos produtos até gráficos e relatórios de BI. Para isso o sistema foi preparado para ser modular e permitir fácil integração com novos componentes.

Contudo, diferente da Amazon (Figura 2), neste projeto pretende-se implementar uma interface mais limpa e minimalista (Figura 3), sem sobrecarga de componentes e controles, com o objetivo de permitir maior foco por parte dos usuários do sistema.

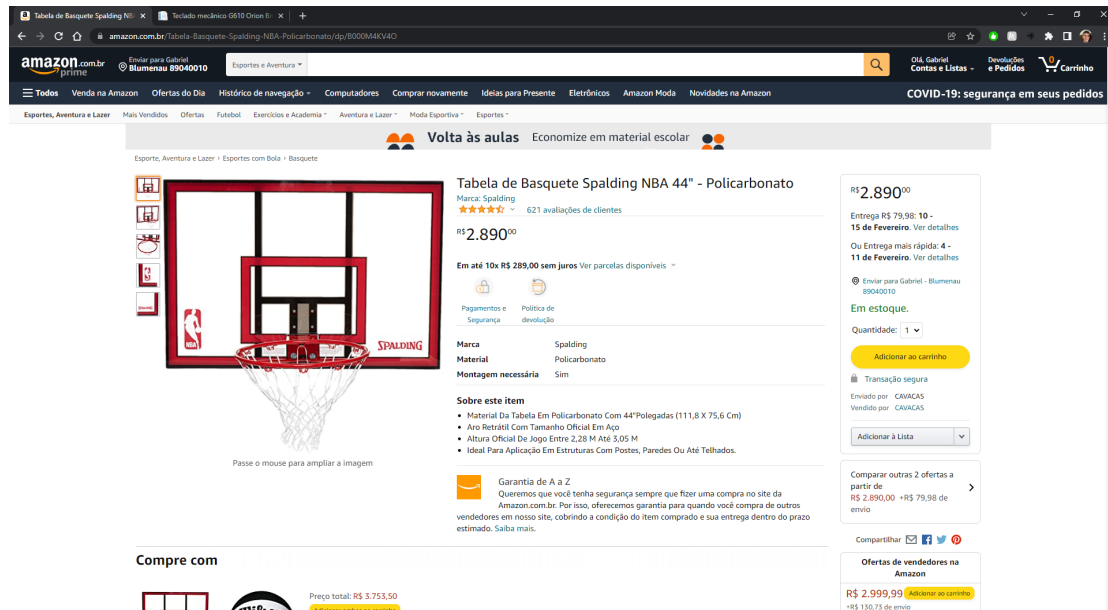


Figura 2 - Interface “carregada” da Amazon

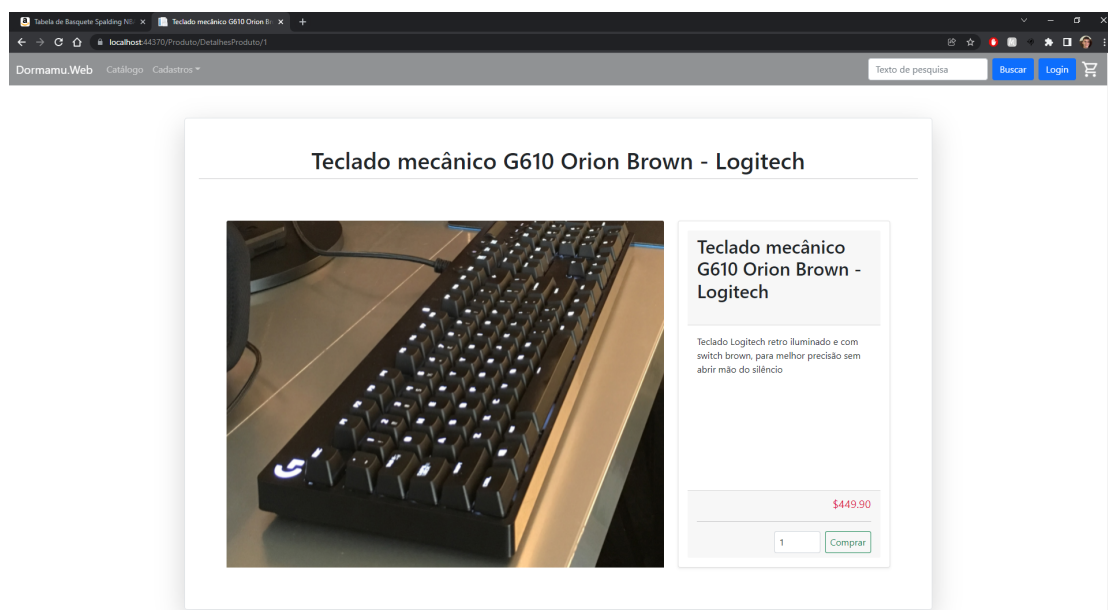


Figura 3 - Interface minimalista do sistema

### **3. Requisitos**

Foram levantados, junto ao cliente, e em análise ao sistema atualmente em uso por ele (como esse se trata de um sistema legado), os requisitos funcionais e não-funcionais que descrevem as características essenciais do projeto, conforme listados nas seções posteriores.

#### **3.1. Requisitos Funcionais**

Os requisitos funcionais descrevem, como o próprio nome sugere, as funcionalidades; as operações; as funções; rotinas; ou como for mais conveniente colocar; que devem ser desenvolvidas e adicionadas ao sistema; sua aplicabilidade como um software. Estas estão abaixo enumeradas:

**RF001** - Deve ser possível que Usuários se cadastrem no sistema, utilizando o e-mail e uma senha.

**RF002** - Deve ser possível ao Administrador do sistema cadastrar produtos, preenchendo os campos, marca, modelo, descrição, valor unitário, quantidade disponível, características técnicas, categoria, tipo. Os campos marca, categoria e tipo devem estar num campo de seleção e seus valores devem ser previamente cadastrados. Os campos modelo, descrição e características técnicas devem ser campos de texto. Também deve ser concedida ao Administrador as permissões de edição e exclusão dos registros de produtos.

**RF003** - Deve ser possível, ao Administrador, carregar fotos aos produtos para mostrar no catálogo e na página de visualização do produto. Isso pode ser feito tanto na criação, quanto na edição de um registro de produto.

**RF004** - Deve ser possível para o Administrador cadastrar os valores para os campos marca, categoria e tipo.

**RF005** - O sistema deve dispor os produtos cadastrados em um catálogo dividido por categorias.

**RF006** - Deve ser possível que Administrador e Usuário possam filtrar o catálogo de produtos com base nos campos da interface de produto, sendo eles: marca, modelo, valor unitário e tipo.

**RF007** - No catálogo, produtos indisponíveis devem ter essa informação destacada e devem ser alocados no final da lista.

**RF008** - Deve ser possível, para produtos indisponíveis, que o Usuário entre em contato e seja avisado quando o estoque for restabelecido.

**RF009** - Deve ser possível que Administrador e Usuário possa ordenar a busca por preço, mais vendidos e promoções.

**RF010** - O sistema deve possuir um módulo de e-Commerce implementando a API do PagSeguro.

**RF011** - No caso de o módulo de e-Commerce estar ativado, na página do produto deve-se mostrar o valor e a quantidade disponível e será assim como as opções do

sistema de “carrinho”.

**RF012** - No caso de o módulo de e-Commerce estar desativado, na página do produto deve-se mostrar apenas um formulário para envio de e-mail rápido.

**RF013** - O e-Commerce deve funcionar no sistema de “carrinho de compras”, onde o Usuário adicionará ao “carrinho” os produtos que deseja adquirir e suas respectivas quantidades, para, posteriormente, fechar o pedido.

**RF014** - Permitir que usuários do sistema busquem produtos utilizando textos. Buscando nos produtos os que apresentam as características do texto informado.

### **3.2. Requisitos Não-Funcionais**

Já esses requisitos, não tão intuitivamente como os requisitos do tópico anterior, descrevem como o sistema deve ser desenvolvido, com quais ferramentas, para qual tipo de servidor, padrões de projeto, etc. Segue a listagem:

**RNF001** - O sistema deve ser desenvolvido em C# ASP.NET.

**RNF002** - A parte cliente deve ser desenvolvida em HTML5 e Javascript, com auxílio das bibliotecas Bootstrap e JQuery.

**RNF003** - O sistema deve utilizar o banco de dados SQL Server.

**RNF004** - O sistema deverá contar com uma interface desenvolvida no padrão regulamentado pela Google, o Material Design Interface.

### **4. Diagramas UML**

Para esse projeto optamos por desenvolver um diagrama de caso de uso para compreender melhor as funções executadas por cada autor envolvido no sistema, já que deve haver um controle de visibilidade e de ações para os diferentes usuários do sistema; o Administrador, que terá acesso geral as funcionalidades e módulos do sistema, como controle de estoque e cadastro de anúncios; e o Usuário, que terá acesso apenas o módulo de compra e venda de produtos. Veja a seguir o diagrama de caso de uso que descreve esses relacionamentos (Figura 4) — tendo suas descrições anexas no [Apêndice A](#).

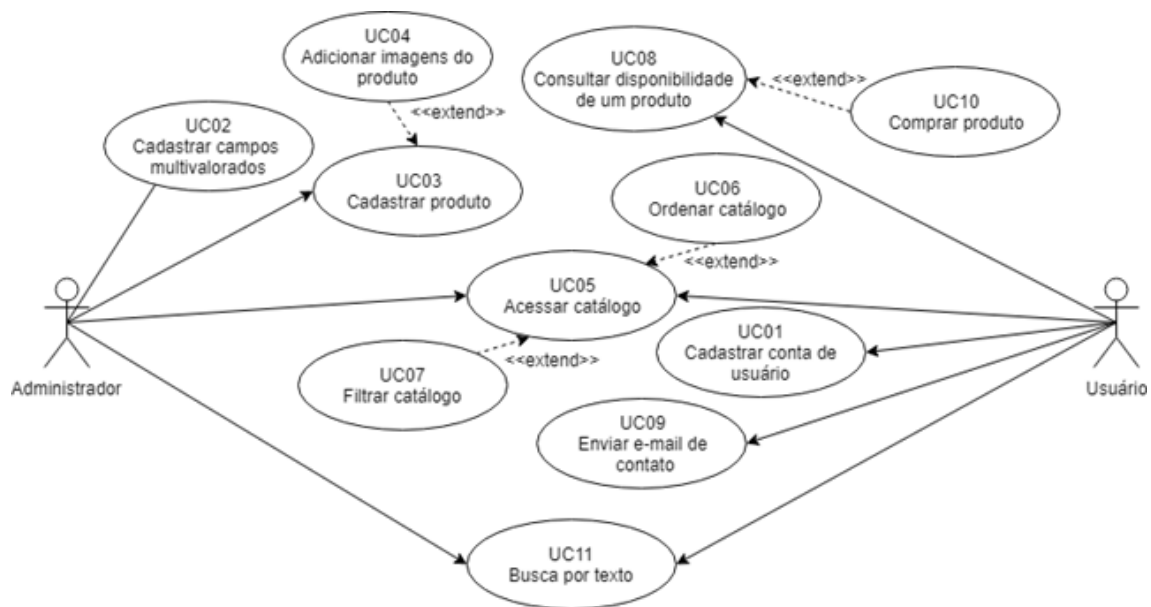


Figura 4 - Diagrama de caso de uso

Para demonstração dos relacionamentos entre os modelos do sistema, desenvolveu-se o diagrama de classes a seguir (Figura 5).

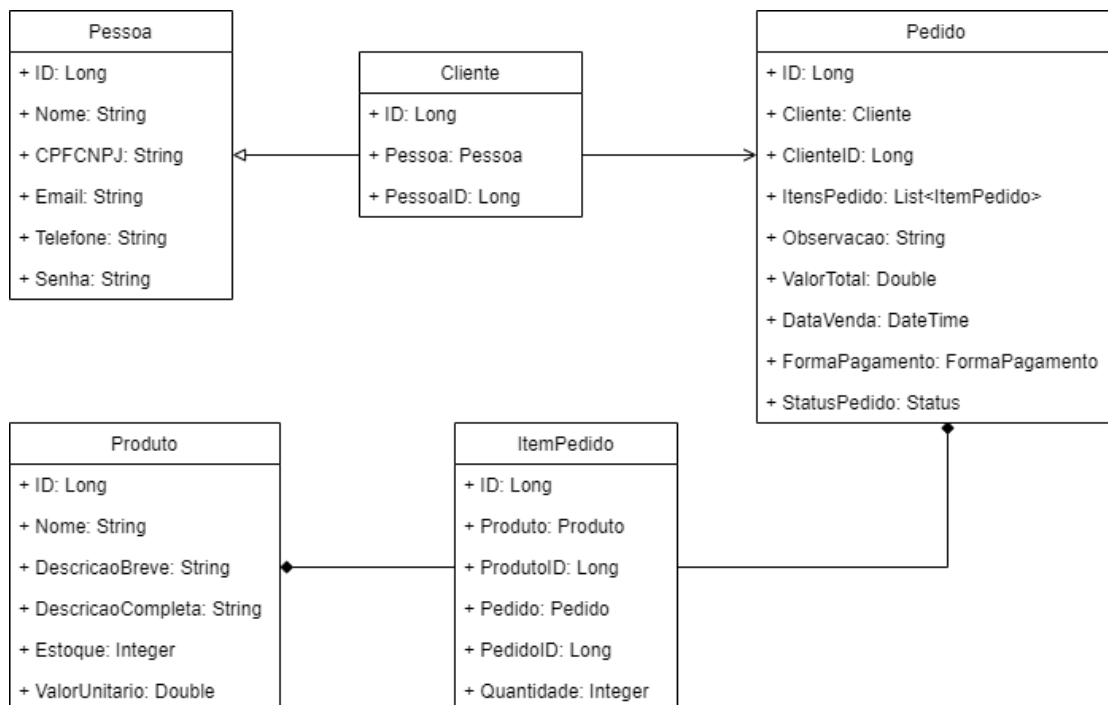


Figura 5 - Diagrama de classes



## 5. Modelagem de Dados

Conforme introduzido no tópico [Método de Trabalho](#), a modelagem do banco de dados ficou a cargo da biblioteca Entity Framework Core. Isso é possível através da abordagem Code-First adotada pela biblioteca, que consiste na modelagem do banco partindo de modelos simples de classes criadas na aplicação, ao invés de projetar o banco de dados primeiro (CODEFIRST, 2020). A figura a seguir (Figura 6) demonstra esse fluxo.

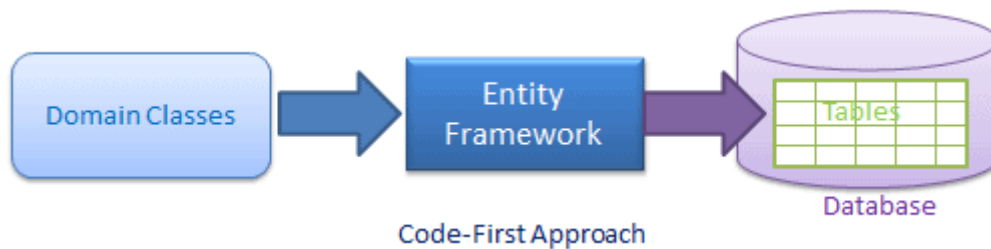


Figura 6 - Abordagem Code-First.

Através da utilização dessa ferramenta obteve-se o seguinte modelo de banco de dados (Figura 7).

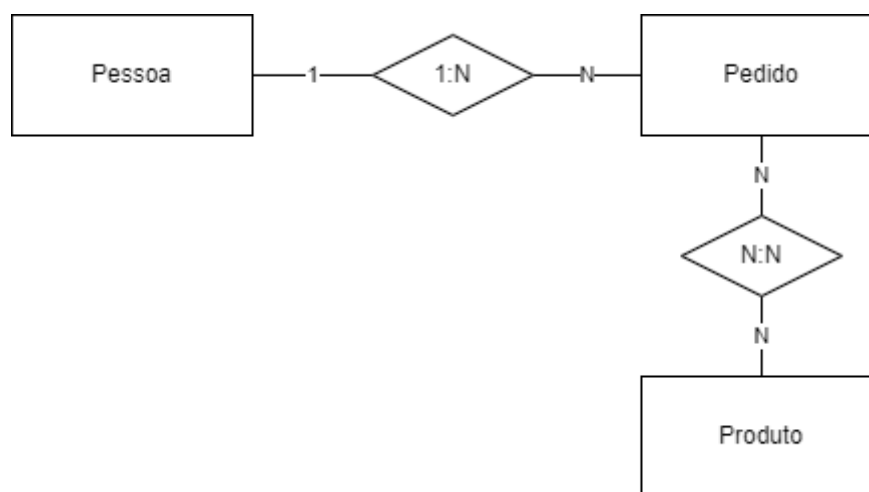


Figura 7 - Diagrama de Entidade e Relacionamento

## 6. Projeto da Interface

A interface do sistema foi trabalhada utilizando a biblioteca Bootstrap, de forma integrada as páginas Razor, utilizando de HTML e CSS para embelezamento das mesmas, também foram usados os componentes Blazor para permitir a modularização dos mesmos, mirando sempre em um *design* arrojado e moderno, responsivo e fluido, que fique bonito independente da plataforma do qual está sendo acessado. Abaixo é possível ver uma demonstração da tela principal do sistema (Figura 8).

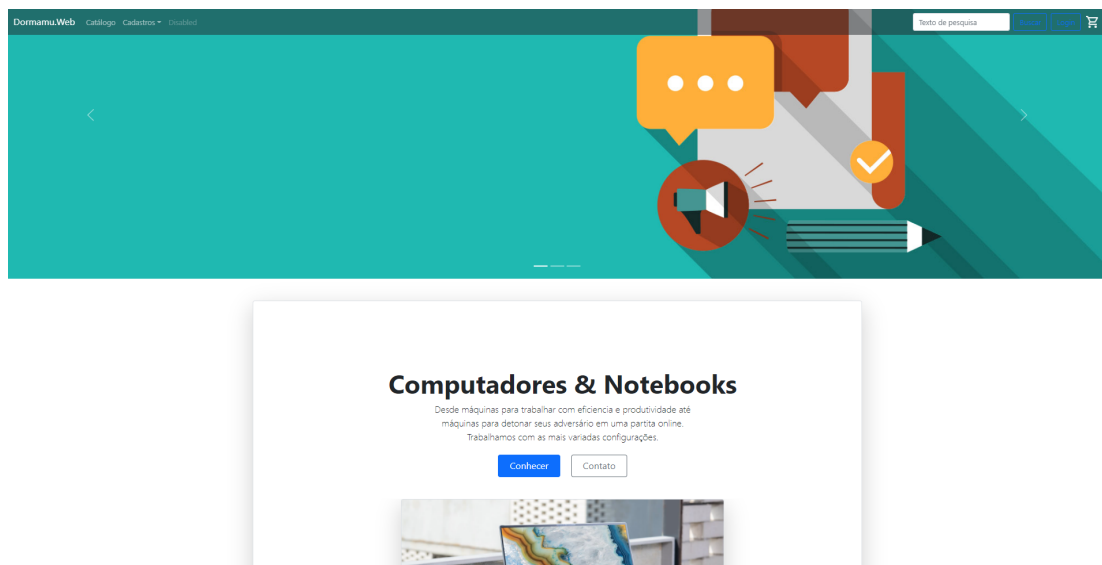


Figura 8 - Página 'Home'

Adotou-se uma estruturação minimalista, focando cada página em seu principal objetivo, sem muitos componentes apontando para outras funções.

Utilizamos muito dos conceitos desenvolvidos pela Google a partir do Android 5.0 Lollipop, o Material Design Interface (MATERIALDSIGN, 2014). O Material Design Interface, se trata de uma série de padrões de estruturação e estilo de sistemas e foi criado pensando em aprimorar a experiência com o usuário, agindo diretamente em suas percepções, a fim de passar a ideia de que os controles do sistema são objetos do mundo real. No nosso sistema, por exemplo, utilizamos de sombras e bordas para passar a ideia de que alguns elementos estão em um nível maior de elevação em relação a tela do dispositivo, fazendo-os parecer como pequenos blocos de anotações, ou mesmo folhetos deixados em cima do dispositivo. Essas características podem ser vistas na imagem a seguir (Figura 9).

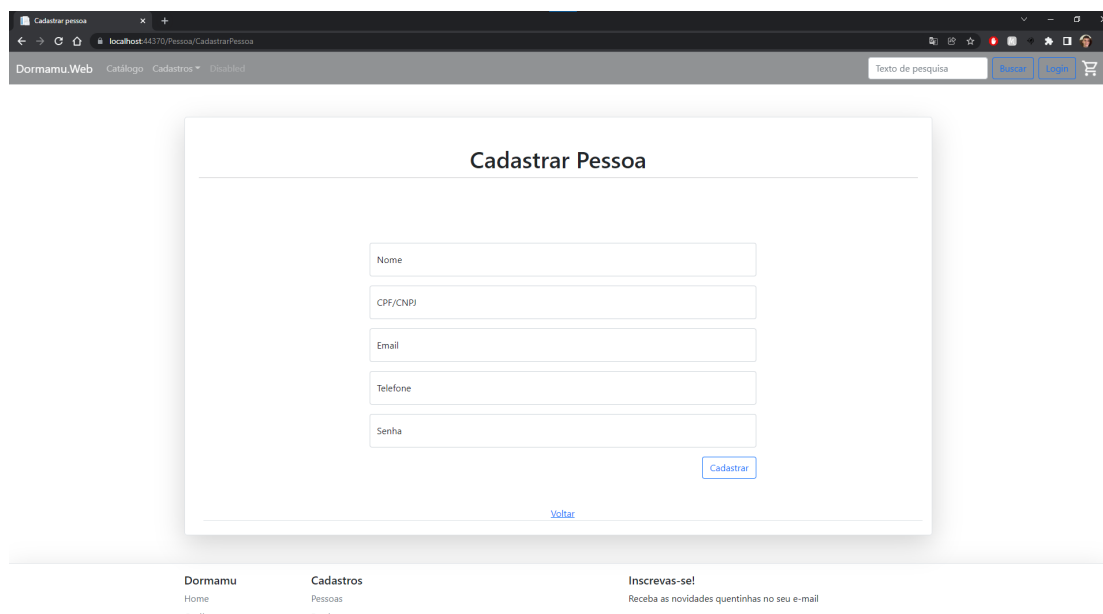


Figura 9 - Página de cadastro de pessoas

Este padrão foi aplicado igualmente em todas as páginas do sistema visando dar foco à função principal da rota acessada.

## 7. Implementação

O desenvolvimento desse projeto foi recheado de altos e baixos, ao ponto de que, cada vez que decaímos, tomamos impulso para ir cada vez mais alto, isso se mostra no fato de, inicialmente, o projeto ter começado com uma aplicação web utilizando .Net Framework 4.7.2 (MVCFRAMEWORK, 2018) para posteriormente avançarmos para um projeto .Net Core (MVCCORE, 2019), na sua versão 3.1.22, devido à simplicidade dos primeiros protótipos e a dificuldade de integrarmos componentes mais sofisticados que contribuíram em muito para construção do sistema, como foi o caso dos componentes Blazor, que se mostravam impossíveis à medida que avançávamos com os protótipos iniciais de modo que uma nova ‘solução’ (como o Visual Studio denomina o projeto) completamente alinhada ao .Net Core foi iniciada, aproveitando apenas a modelagem da antiga. A vasta gama de funcionalidades que se é possível integrar ao Core se mostrou muito vantajosa para a evolução do projeto e gerou demasiada empolgação e uma sede por buscar avançar sempre mais.

A aplicação *web* com .Net Core MVC em seu estado inicial é bem organizada, apenas a mantivemos deste modo no decorrer das implementações, e o Visual Studio 2019 foi um grande facilitador no processo de produção deste sistema, rapidamente se criava um modelo, o seu controlador e as *views*, com poucos cliques e algumas linhas de código (CRUDMVC, 2019). Este também viabilizou o fácil gerenciamento de pacotes com o NuGet (gerenciador de bibliotecas do Visual Studio), onde podemos encontrar uma vasta seleção de bibliotecas sem sequer sair da IDE. Por meio do Visual Studio ainda foi possível visualizar e manusear o banco de dados de modo muito prático. O ambiente e todo o ecossistema criado pela Microsoft teve grande impacto no bom andamento do projeto.

No projeto .Net Core MVC com Entity Framework temos os modelos como classes simples, tendo essas apenas a designação dos atributos seguindo o padrão POCO, conforme mencionado anteriormente (na seção [Método de Trabalho](#)). A partir desses modelos as tabelas e suas associações foram geradas automaticamente na base de dados.

O responsável pela designação de quais tabelas seriam criadas é a nossa classe de Contexto. Herdando as funcionalidades da classe DbContext, a classe Contexto define por meio de DBSets — que nada mais é do que uma lista dos registros da tabela específica que são compartilhados entre as rotinas e o banco de dados — quais das classes deverão gerar uma entidade na base de dados. Na sequência, fica a cargo dos controladores o manuseio das informações.

Os controladores, por meio de um objeto da classe Contexto, acessam os DBSets e por meio deles realizam as operações na base de dados e devolvem os resultados às views. Eles funcionam como as rotas do sistema, recebem as requisições e encaminham ao seu destino.

Por fim, as *views* são desenvolvidas como páginas Razor, que são páginas HTML que acoplam rotinas desenvolvidas em C#. Nestas são processados os dados e realizadas as requisições aos controladores. O diagrama a seguir ilustra esse funcionamento (Figura 10).

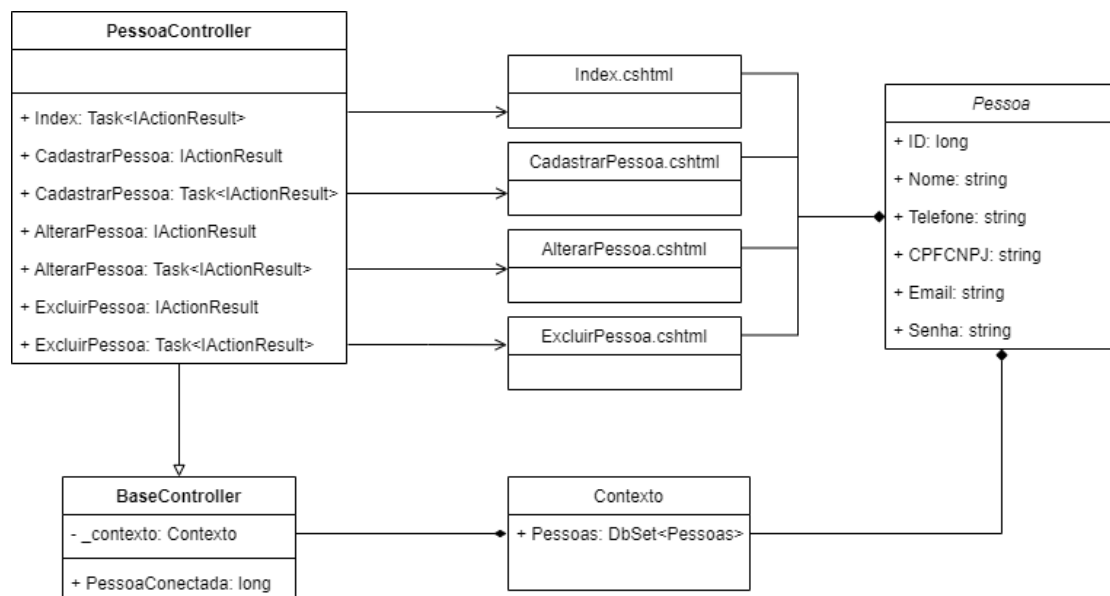


Figura 10 - Diagrama de Arquitetura

Toda a parte de servidor foi desenvolvida com o Asp .Net Core, que também marcou sua presença na parte cliente com as páginas Razor, que por sua vez foram construídas juntamente com os componentes Blazor e Bootstrap, garantindo a beleza e a praticidade do sistema. A biblioteca JQuery foi utilizada para enviar as informações ao servidor via Ajax e, integrada a biblioteca Sweet Alert 2, foi possível criar uma experiência fluida e amigável aos usuários do sistema.

Link do projeto: <https://github.com/gabrielkreusch/Dormamu.Web>

## 8. Resultados Obtidos

Concluiu-se que, por meio deste trabalho, obteve-se um sistema ágil, confiável, elegante, intuitivo, organizado, multi-plataforma, adaptável, sofisticado e receptivo a mudanças e atualizações. Um sistema *web* moderno com todas as vantagens de acessibilidade que um site pode ter. Modular e expansível, permitindo a integração de diversos módulos de trabalho permitindo um amplo gerenciamento do negócio, com controle de estoque e vendas, bem como divulgação do negócio, solucionando o problema proposto e entregando um produto vasto que atenda o quanto for possível as necessidades dos lojistas e empreendedores que o implantarem.

## 9. Considerações Finais e Trabalhos Futuros

Trabalhar com C#, Asp.Net Core, SQL Server e Entity Framework se mostrou muito prático e de fácil integração entre as tecnologias, sem dúvidas o ecossistema criado pela microsoft se mostrou determinante para o bom andamento deste projeto.

A partir desse ponto, pretendo trabalhar mais a questão de *business*, efetuando a implementação de *dashboards* e relatórios para permitir maior controle do negócio pelo cliente. Através de gráficos pretende-se dispor uma visualização do fluxo diário de transações executadas pelo sistema, e além dele, já que o propósito deste é, mais do que um site de vendas, ser uma plataforma online de gerenciamento de negócios.

Procurar-se-á aprimorar questões de segurança, validando sessões em páginas onde esse tratamento é necessário, evitando que dados possam ser manipulados e enviados através das rotas, e será efetuado criptografia de dados sensíveis na base — gerando *hashes* para senhas e afins — e adicionando criptografia assimétrica em transações de dados entre as páginas do sistema.

O layout das páginas *web* hão de ser revisados e ajustados conforme critérios do cliente.

## 10. Referências Bibliográficas

SCHEIFER, Verônica. Tráfego da internet brasileira na pandemia: quais foram os impactos?. Tecmundo, 2020. Disponível em: <<https://www.tecmundo.com.br/internet/154628-trafego-internet-brasileira-pandemia-impactos.htm>>. Acesso em: 29 de jan. 2022.

CRUDMVC. CRUD com ASP.NET CORE MVC 2. Produção: Programação Binária. Youtube, 2019. Disponível em: <[https://www.youtube.com/watch?v=7CVOXK8YUt4&t=323s&ab\\_channel=ProgramaçãoBinária](https://www.youtube.com/watch?v=7CVOXK8YUt4&t=323s&ab_channel=ProgramaçãoBinária)>. Acesso em: 29 jun. 2021.

MVCCORE. ENTITY Framework Core + Asp.NET Core Web API + SQL Server. Produção: ProgrAmada Mente. Youtube, 2019. Disponível em: <[https://www.youtube.com/watch?v=ipbSwv09dDU&t=3634s&ab\\_channel=ProgrAmadaMente](https://www.youtube.com/watch?v=ipbSwv09dDU&t=3634s&ab_channel=ProgrAmadaMente)>. Acesso em: 29 jun. 2021.

MVCFRAMEWORK. 59 - ASP.Net MVC - C# - TORNE-SE UM PROGRAMADOR. Direção: Danilo Aparecido. Produção: Danilo Aparecido - Torne-se um programador. Youtube, 2018. Disponível em: <[https://www.youtube.com/watch?v=UpOVuM8dXKQ&list=PLEdPHGYbHhlcxWx-LrVVYZ2RRdqltums&index=59&ab\\_channel=DaniloAparecido-Torne-seumprogramador](https://www.youtube.com/watch?v=UpOVuM8dXKQ&list=PLEdPHGYbHhlcxWx-LrVVYZ2RRdqltums&index=59&ab_channel=DaniloAparecido-Torne-seumprogramador)>. Acesso em: 29 jun. 2021.

DOS SANTOS, Alexandro Klein. Os IDE's (Ambientes de Desenvolvimento Integrado) como ferramentas de trabalho em informática. Universidade Federal de Santa Maria, 2017. Disponível em: <<http://www-usr.inf.ufsm.br/~alexks/elc1020/artigo-elc1020-alexks.pdf>>. Acesso em: 23 de jan. de 2022.

ENTITYCORE. Entity Framework Core. Microsoft, 2022. Disponível em: <<https://docs.microsoft.com/pt-br/ef/core/>>. Acesso em: 23 de jan. de 2022.

GROFFE, Renato Jose. Introdução ao Entity Framework Code First. Devmedia, 2014. Disponível em: <<https://www.devmedia.com.br/introducao-ao-entity-framework-code-first/30705>>. Acesso em: 23 de jan. de 2022.

CODEFIRST. What is Code-First?. Entity Framework Tutorial, 2020. Disponível em: <<https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>>. Acesso em: 23 de jan. de 2022.

MATERIALDSIGN. Material Design Interface. Material Design, 2014. Disponível em: <<https://material.io/design>>. Acesso em: 27 de jan. de 2022.

## 11. Apêndice A - Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos principais casos de uso descritos na seção de especificação deste trabalho. Nos Quadros de 1 até 11 são demonstrados todos os casos de uso definidos para o desenvolvimento do sistema.

### Quadro 1. Descrição do caso de uso cadastrar conta de usuário

**Caso de uso UC01** – Cadastrar conta de usuário.

**Ator:** Usuário.

**Objetivo:** Identificar o Usuário.

**Pré-condições:** Nenhuma.

**Pós-condições:** Usuário autenticado.

**Cenário Principal:**

1. Acessar o sistema;
2. Acessar o cadastro de usuários;
3. Preencher o formulário;
4. Validar informações;
5. Informar ao usuário que o cadastro foi realizado com êxito.

**Cenário Alternativo:**

1. Acessar o sistema;
2. Acessar o cadastro de usuários;
3. Preencher o formulário;
4. Validar informações;
5. Usuário já cadastrado com as informações fornecidas.

### Quadro 2. Descrição do caso de uso cadastrar campos multivalorados

**Caso de uso UC02** – Cadastrar campos multivalorados.

**Ator:** Administrador.

**Objetivo:** Facilitar o preenchimento dos campos dos produtos, cadastrando previamente valores que se repetem em diversos registros.

**Pré-condições:** Ter permissão de administrador.

**Cenário Principal:**

1. Acessar o sistema com permissão de administrador;
2. Acessar o cadastro de produtos;
3. Acessar a opção de “adicionar opção”, no campo que se deseja inserir um novo valor;
4. Cadastrar e salvar o novo valor.

### Quadro 3. Descrição do caso de uso cadastrar produto

**Caso de uso UC03** – Cadastrar produto.

**Ator:** Administrador.

**Objetivo:** Adicionar novo item ao catálogo de vendas. Pré-condições: Ter permissão de administrador.

**Cenário Principal:**

1. Acessar o sistema com permissão de administrador;
2. Acessar a opção de cadastro de novos produtos;
3. Preencher o formulário;
4. Salvar.

**Cenário Alternativo:**

1. Acessar o sistema com permissão de administrador;
2. Acessar um produto já cadastrado;
3. Realizar as alterações desejadas;
4. Salvar.

### Quadro 4. Descrição do caso de uso adicionar imagens ao produto

**Caso de uso UC04** – Adicionar imagens do produto.

**Ator:** Administrador.

**Objetivo:** Oferecer uma informação visual do produto ao Usuário.

**Pré-condições:** Ter permissão de administrador.

**Pós-condições:** As imagens cadastradas serão exibidas na página do produto e no catálogo.

**Cenário Principal:**

1. Acessar o sistema com permissão de administrador;
2. Acessar no cadastro de produtos, na criação ou edição de um produto, a opção de “adicionar imagens ao produto”;
3. Escolher as imagens desejadas;
4. Salvar.



#### **Quadro 5. Descrição do caso de uso acessar o catálogo**

**Caso de uso UC05** – Acessar catálogo.

**Ator:** Administrador e Usuário.

**Objetivo:** Mostrar os produtos disponíveis à venda.

**Pré-condições:** Nenhuma.

**Cenário Principal:**

1. Acessar o sistema;
2. Acessar uma categoria através do menu.
3. Cenário Alternativo:
4. Acessar o sistema;
5. Realizar uma busca na barra de pesquisa.

#### **Quadro 6. Descrição do caso de uso ordenar o catálogo de produtos**

**Caso de uso UC06** – Ordenar catálogo.

**Ator:** Administrador e Usuário.

**Objetivo:** Ordenar o catálogo de forma conveniente a facilitar a busca de um produto em específico.

**Pré-condições:** Nenhuma.

**Cenário Principal:**

1. Acessar o sistema na página do catálogo de produtos;
2. Alterar o valor do campo de seleção para a forma que se deseja ordenar os produtos.

#### **Quadro 7. Descrição do caso de uso filtrar os produtos do catálogo**

**Caso de uso UC07** – Filtrar catálogo.

**Ator:** Administrador e Usuário.

**Objetivo:** Mostrar apenas os produtos que atendem aos filtros realizados.

**Pré-condições:** Nenhuma.

**Pós-condições:** Visualizar somente os produtos que atendem aos filtros.

**Cenário Principal:**

1. Acessar o sistema na página do catálogo de produtos;
2. Acionar a opção “filtrar”.
3. Selecionar/Preencher os filtros desejados;
4. Clicar no botão “Filtrar”.

#### **Quadro 8. Descrição do caso de uso consultar disponibilidade de um produto**

**Caso de uso UC08** – Consultar a disponibilidade de um produto.

**Ator:** Usuário.

**Objetivo:** Verificar quando um determinado produto vai estar disponível.

**Pré-condições:** Possuir uma conta de Usuário.

**Pós-condições:** Possibilidade de realizar a compra.

**Cenário Principal:**

1. Acessar o sistema na página do catálogo de produtos;
2. Acionar a opção “filtrar”.
3. Selecionar/Preencher os filtros desejados;
4. Clicar no botão “Filtrar”.

#### **Quadro 9. Descrição do caso de uso enviar e-mail de contato**

**Caso de uso UC09** – Enviar e-mail de contato. Ator: Usuário.

**Ator:** Usuário.

**Objetivo:** Fornecer ao Usuário uma forma prática de contato com os administradores do sistema.

**Pré-condições:** Possuir uma conta de Usuário.

**Pós-condições:** Possibilidade de realizar a compra.

**Cenário Principal:**

1. Acessar o sistema na página de contatos;
2. Preencher os campos “Título” e “Mensagem”;
3. Clicar em “Enviar”.

**Cenário Alternativo:**

1. Acessar o sistema na página de contatos;
2. Preencher os campos “Título” e “Mensagem”;
3. Clicar em “Enviar”;
4. Sistema foi acessado sem autenticação de usuário;
5. Solicitar as informações de login.

#### **Quadro 10. Descrição do caso de uso comprar produto**

**Caso de uso UC10** – Comprar produtos.

**Ator:** Usuário.

**Objetivo:** Efetuar a compra de um produto.

**Pré-condições:** Possuir uma conta de Usuário.

**Cenário Principal:**

1. Acessar, com uma conta de usuário, o sistema no catálogo de produtos;
2. Acessar o(s) produtos que deseja comprar;
3. Adicionar ao carrinho o(s) produtos que deseja comprar;
4. Acessar o carrinho;
5. Clicar em “Finalizar compra”;
1. Informar endereço de entrega;
2. Selecionar forma de pagamento;
3. Direcionar para a página da API de e-Commerce para efetivar o pagamento;
4. Mostrar ao Usuário os dados da compra.

**Cenário Alternativo:**

1. Acessar, com uma conta de usuário, o sistema no catálogo de produtos;
2. Acessar o(s) produtos que deseja comprar;
3. Produto indisponível.

#### **Quadro 11. Descrição do caso de uso busca por texto**

**Caso de uso UC11** – Busca por texto.

**Ator:** Administrador e Usuário.

**Objetivo:** Encontrar um produto através de um texto que o caracterize.

**Pré-condições:** Nenhuma.

**Pós-condições:** Encontrar produtos caracterizados pelo texto de busca.

**Cenário Principal:**

1. Acessar o sistema e informar na barra de pesquisar o texto a ser procurado nos produtos disponíveis.