

Universidade Paulista - UNIP

Henrique Franchini Zani

CIÊNCIA DE DADOS COMO ALIADA DO EMPREENDEDORISMO

Limeira

2023

Universidade Paulista - UNIP

Henrique Franchini Zani

CIÊNCIA DE DADOS COMO ALIADA DO EMPREENDEDORISMO

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da computação sob a orientação do professor Me. António Mateus Locci.

Limeira

2023

Henrique Franchini Zani

CIÊNCIA DE DADOS COMO ALIADA DO EMPREENDEDORISMO

Trabalho de conclusão de curso
apresentado à banca examinadora da
Faculdade UNIP, como requisito parcial à
obtenção do Bacharelado em ciência da
Computação sob a orientação do professor Me.
Antônio Mateus Locci.

Aprovada em XX de XXXXX de 201X.

BANCA EXAMINADORA

Prof. Dr. Nome completo

Prof. Me. Nome completo

Prof. Esp. Nome completo

DEDICATÓRIA

Dedico este trabalho aos meus amigos e familiares, que sempre me apoiaram em todas às minhas empreitadas que me incentivaram a buscar meus sonhos.

“Se as pessoas não acreditam que a matemática é simples, é apenas porque não percebem como a vida é complicada”.

(John von Neumann)

RESUMO

A presente obra foi feita em vista à crescente relevância da Ciência de Dados como aliada estratégica para empreendedores diante do vasto aumento de dados disponíveis. O seu objetivo consiste em desenvolver uma plataforma digital que utilize técnicas de *Web Scraping* para coletar produtos reais do *site* Amazon e que, em seguida, aplique o algoritmo Floresta Aleatória, a fim de gerar uma estimativa de vendas para cada produto selecionado, tendo como foco ajudar empreendedores a tomar decisões de maneira mais assertiva. Visando alcançar esse propósito, foram necessários os estudos das disciplinas de Ciência de Dados, Aprendizado de Máquina e o algoritmo Floresta Aleatória. O desenvolvimento da plataforma digital utilizou-se das linguagens de programação *Python* e *TypeScript*, através de ferramentas como *Angular* e *Firebase*. Baseou-se na arquitetura cliente-servidor para permitir uma comunicação mais organizada e modular, o que facilitou o desenvolvimento do protótipo. Os resultados obtidos são promissores, refletindo o êxito na implementação dos objetivos propostos. O *software* foi criado com sucesso, demonstrando sua viabilidade como uma ferramenta estratégica para empreendedores na era da Ciência de Dados. A técnica de *Web Scraping* foi eficientemente aplicada, permitindo a coleta sistemática de informações relevantes diretamente do *site* Amazon. Essa abordagem resultou na obtenção de dados adicionais que podem ser valiosos para investidores na tomada de decisões informadas. Além disso, o algoritmo Floresta Aleatória foi implementado bem-sucedidamente, gerando estimativas de vendas a todos os produtos. O projeto mostrou-se viável para criação de ferramentas de apoio, possibilitando aos empreendedores o uso dos dados para tomada de decisões.

Palavra-Chave: Ciência de Dados; Aprendizado de Máquina; Floresta Aleatória; Dados; *Web Scraping*;

ABSTRACT

Given the increasing relevance of Data Science as a strategic ally for entrepreneurs in the face of the vast increase in available data, the goal of this work is to develop a digital platform that employs Web Scraping techniques to collect real products from the Amazon website. Subsequently, the Random Forest algorithm is applied to generate a sales estimate for each selected product, with a focus on assisting entrepreneurs in making more informed decisions. To achieve this purpose, it was necessary to study the disciplines of Data Science, Machine Learning, and the Random Forest algorithm. For the development of the digital platform, programming languages such as Python and TypeScript were used, along with tools like Angular and Firebase. The platform is based on the client-server architecture to enable more organized and modular communication, facilitating the development of the prototype. The results obtained are promising, reflecting the success in implementing the proposed objectives. The digital platform was successfully created, demonstrating its viability as a strategic tool for entrepreneurs in the era of Data Science. The Web Scraping technique was efficiently applied, allowing the systematic collection of relevant information directly from the Amazon website. This approach resulted in obtaining additional data that can be valuable for entrepreneurs in making informed decisions. Furthermore, the Random Forest algorithm was successfully implemented, generating sales estimates for each selected product. The project has proven to be feasible for the creation of supportive tools, enabling entrepreneurs to use data for decision-making processes.

Key Words: *Data Science; Machine Learning; Random Forest; Data; Web Scraping;*

LISTA DE FIGURAS

Figura 1 - Ciência de Dados: disciplinas fundamentais.	18
Figura 2 - Relação entre conceitos de dados, informações.	19
Figura 3 - Os tipos de Aprendizado de Máquina.	23
Figura 4 - Árvores de decisões.	26
Figura 5 - Diagrama de usos.	37
Figura 6 - Arquitetura utilizada.	40
Figura 7 - Tela de <i>login</i>	43
Figura 8 - Tela de cadastro.	44
Figura 9 - Recuperação de senha.	44
Figura 10 - Tela de Início.	45
Figura 11 - Tela do Produto.	46
Figura 12 - Tela do Produto.	46
Figura 13 - Tela dos Produtos Salvos.	47
Figura 14 - Rota Search.	48
Figura 15 - Função para obter os produtos.	49
Figura 16 - Funções adicionais.	49
Figura 17 - Rota de pegar informações do produto.	50

LISTA DE QUADROS

Quadro 1 - Tipos de dados.	20
Quadro 2 - Serviços <i>Firebase</i>	32
Quadro 3 - Métodos de Requisição.	33
Quadro 4 - Códigos de status HTTP.	33
Quadro 5 - Requisitos Funcionais.	36
Quadro 6 - Requisitos não Funcionais.	37
Quadro 7- CSU 01: Pesquisar Produtos.	38
Quadro 8 - CSU 02: Visualizar Produtos.	39
Quadro 9 - CSU 03: Salvar Produtos.	39

LISTA DE ABREVIATURAS

AM - Aprendizado de Máquina

BaaS - *Back-End as a Service*

CD - Ciência de Dados

CSU - Casos de Uso Descritivos

DOD - Tomada de Decisão Orientada por Dados

GPL - *General Public License*

HTTP - *Hypertext Transfer Protocol*

JSON - *JavaScript Object Notation*

MVC - *Model-View-Controller*

OOB - *Out-of-Bag*

RF - Requisitos Funcionais

RNF - Requisitos não Funcionais

VSCode - *Visual Studio Code*

XGBoost - *Extreme Gradient Boosting*

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	14
1.1.1 Objetivo Geral	14
1.1.2 Objetivo específico.....	14
1.2 Justificativa.....	15
1.3 Metodologia.....	15
2 REVISÃO BIBLIOGRÁFICA	17
2.1 Ciência de dados.....	17
2.1.1 Mineração de Dados.....	19
2.1.2 Dados.....	19
2.1.3 Dados para Ciência de Dados	20
2.1.4 Tomada de decisão orientada em dados.....	21
2.2 Aprendizado de Máquina	23
2.2.1 Aprendizado Supervisionado	24
2.2.2 Regressão Linear.....	25
2.2.3 Árvores de Decisão	25
2.2.4 Floresta Aleatória	25
2.2.5 <i>Overfitting</i>	27
3 FERRAMENTAS E TÉCNICAS	29
3.1 Visual Studio Code (VSCode).....	29
3.2 Python	30
3.3 Framework Angular	30
3.4 Firebase	31
3.5 Arquitetura de software	32
3.6 Arquitetura Cliente-Servidor	32
4 DESCRIÇÃO GLOBAL DA PLATAFORMA DIGITAL	34
4.1 Interfaces.....	34
4.2 Interfaces do Sistema	34

4.3	Interface de Comunicação	35
4.4	Funções do Sistema	35
4.5	Restrições	35
5	MODELO PROPOSTO DA PLATAFORMA DIGITAL	36
5.1	Requisitos Funcionais	36
5.2	Requisitos Não Funcionais.....	37
5.3	Diagrama de Casos de Uso.....	37
5.4	Casos de Uso Descritivos (CSU).....	38
5.5	Modelo da Arquitetura.....	40
5.5.1	Servidor.....	41
5.5.2	Web Scraping	41
5.5.3	Cliente.....	42
5.5.4	Banco de dados.....	42
6	RESULTADOS.....	43
6.1	Protótipo Plataforma Digital.....	43
6.2	Códigos-Fonte.....	47
6.2.1	Python Server	Error! Bookmark not defined.
6.2.2	Algoritmo Web Scraping	48
6.2.2	Algoritmo Floresta Aleatória	50
7	CONSIDERAÇÕES FINAIS	52
7.1	Dificuldades encontradas	52
7.2	Trabalhos Futuros.....	53

1 INTRODUÇÃO

Nos últimos anos, presenciamos uma explosão de dados em praticamente todos os aspectos da nossa vida cotidiana. Com a popularização das redes sociais e plataformas on-line, a quantidade de dados gerados diariamente atingiu níveis exponenciais. Os consumidores passaram a deixar rastros digitais em todas as suas interações on-line. Cada compra, cada pesquisa, cada comentário é registrado e armazenado em bancos de dados gigantescos.

Diante do crescente acúmulo de informações, os consumidores têm à sua disposição uma ampla gama de opções de produtos ou serviços, o que torna essencial às empresas desenvolverem e implementarem estratégias para demonstrar suas vantagens em relação aos concorrentes, a fim de se destacarem em seu segmento de mercado e atrair novos clientes.

No entanto, essa explosão de dados trouxe consigo uma infinidade de oportunidades, mas também desafios significativos para o empreendedor. As empresas se viram diante do chamado "dilúvio de dados", em que a quantidade de informações disponíveis ultrapassava a capacidade humana de processá-las e de extrair *insights* relevantes. Foi nesse contexto que a Ciência de Dados emergiu como uma aliada para o empreendedorismo, sendo essencial para lidar com esse grande volume de estatísticas e transformá-las em conhecimento acionável.

Para aumentar as chances de sucesso em plataformas de mídia digital e social, as empresas devem identificar padrões insuspeitos utilizando técnicas de Aprendizado de Máquina. Assim, a indústria de vendas seria beneficiada por meio de ferramentas baseadas em áreas de pesquisa, como Ciências da Informação ou Ciências da Computação, que proporcionassem melhores, e mais direcionadas, informações, com o objetivo guiá-la em suas operações. A exemplo disso, podemos pensar numa ferramenta voltada à estimativa de vendas para determinado produto, que auxiliaria o mercado em negócios mais assertivos e seguros, implementando o algoritmo Floresta Aleatória com *Python*.

Até agora, as principais tarefas de Ciência de Dados incluíam melhorar a capacidade de armazenamento dos dados da empresa, realizar pesquisas de mercado e segmentação de consumidores, ou extrair informações importantes sobre os problemas da mesma. Tal ciência é um amplo ecossistema que engloba diferentes estratégias de identificação de padrões, modelos de análise, indicadores

de desempenho, variáveis estatísticas e habilidades técnicas ligadas a um grande conhecimento tecnológico. No entanto, a falta de recursos e expertise específica torna desafiador para muitas empresas explorar eficazmente as vastas quantidades de dados disponíveis. É nesse ponto que se evidencia a lacuna existente na capacidade das organizações de encontrar os produtos mais adequados para suas estratégias de negócios.

Com o objetivo de aumentar o uso da Ciência de Dados nas empresas, o presente estudo traz um projeto prático, por meio da implementação de uma plataforma digital que utiliza o algoritmo de Aprendizado de Máquina, chamado Floresta Aleatória, e que se baseia em dados reais do mercado, utilizando o *site* Amazon. Os aspectos mencionados serão estudados, explicados e analisados. A partir disso, empresas poderiam tomar decisões mais coerentes e precisas.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver uma plataforma digital com o propósito de auxiliar empreendedores na escolha estratégica de produtos para suas operações. A plataforma utilizará dados reais do *site* Amazon e o algoritmo Floresta Aleatória, para realizar estimativas de vendas semanal e mensal, capacitando, assim, os empreendedores a tomar decisões de maneira mais assertiva e potencializar seus negócios.

1.1.2 Objetivo Específico

Utilizar ferramentas e tecnologias como *Angular*, *Python*, *Firebase*, *Web Scraping* e o algoritmo Floresta Aleatória, para implementar a plataforma, tendo como foco atingir os seguintes objetivos:

- Configurar banco de dados e autenticação no *Firebase*;
- Criar servidor utilizando *Python*;
- Desenvolver algoritmo *Web Scraping* para capturar os dados do *site* Amazon;
- Desenvolver o algoritmo Floresta Aleatória com *Python*, a fim de gerar uma previsão de Estimativa de Vendas semanal e mensal;

- Limpar e formatar os dados para facilitar as análises;
- Criar o cliente para visualizar os dados utilizando *Angular*;
- Interpretar os dados para obter informações sobre os produtos;

1.2 Justificativa

O crescimento dos canais digitais e a crescente disponibilidade de dados transformaram o cenário digital. Para se manterem competitivas, as empresas precisam ser capazes de usar essas informações para tomar decisões e conduzir estratégias de mercado bem-sucedidas. A Ciência de Dados fornece uma poderosa caixa de ferramentas para análise e entendimento, permitindo que as empresas obtenham informações valiosas sobre comportamento, preferências e padrões do cliente.

Embora tal ciência tenha se tornado uma parte essencial do *marketing* moderno, ainda há muito a aprender sobre como ela pode ser usada para otimizar estratégias de vendas e gerar melhores resultados de negócios. Ao estudar a relação entre Ciência de Dados e mercado, podemos obter uma compreensão mais profunda do papel que ela desempenha no mesmo. Esta pesquisa pode ajudar as empresas a tomar decisões mais precisas, melhorando as vendas e aprimorando o envolvimento do cliente.

Além disso, os benefícios deste estudo vão além dos empreendimentos individuais. À medida que mais corporações começam a usar Ciência de Dados em suas estratégias, atinge-se uma melhor compreensão do campo, contribuindo para o desenvolvimento de melhores práticas e padrões da indústria. Isso pode levar a um uso mais eficiente e eficaz da tecnologia, beneficiando o setor como um todo.

1.3 Metodologia

A metodologia deste trabalho abrange várias etapas com o objetivo de alcançar a proposta estabelecida. Inicialmente, será conduzida uma revisão bibliográfica que abordará temas como Ciência de Dados, Aprendizado de Máquina e o algoritmo Floresta Aleatória.

Depois, será realizado um levantamento das tecnologias que serão utilizadas para o desenvolvimento do sistema e arquitetura do sistema proposto. Após isso,

será mostrado uma análise abrangente da plataforma apresentada, abordando seus aspectos relativos às interfaces, funcionalidades e restrições.

Em seguida, serão discutidos os conceitos cruciais que orientaram o desenvolvimento da plataforma, como requisitos funcionais, requisitos não funcionais, diagrama de uso e modelo de arquitetura.

Na fase final, será dedicado à apresentação dos resultados, onde a plataforma será demonstrada, códigos-fonte e implementações.

Por fim, serão discutidas as conclusões derivadas dos resultados obtidos, relacionando-os com os objetivos estabelecidos inicialmente. Dificuldades encontradas e áreas para trabalhos futuros também serão identificadas, contribuindo para o avanço contínuo da solução proposta.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, serão apresentados os fundamentos essenciais da Ciência de Dados e Aprendizado de Máquina, fornecendo uma base sólida que sustentará o desenvolvimento do sistema proposto.

2.1 Ciência de dados

Ciência de dados (CD) é essencialmente a disciplina dedicada à extração de informações, conhecimento e *insights* a partir de diversas fontes e grandes volumes de dados. Isso é alcançado por meio da coleta, processamento e análise de dados, utilizando tecnologias da informação. Esse processo pode ser automatizado através de métodos estatísticos, modelos matemáticos e ferramentas dessa tecnologia. (COELHO 2017).

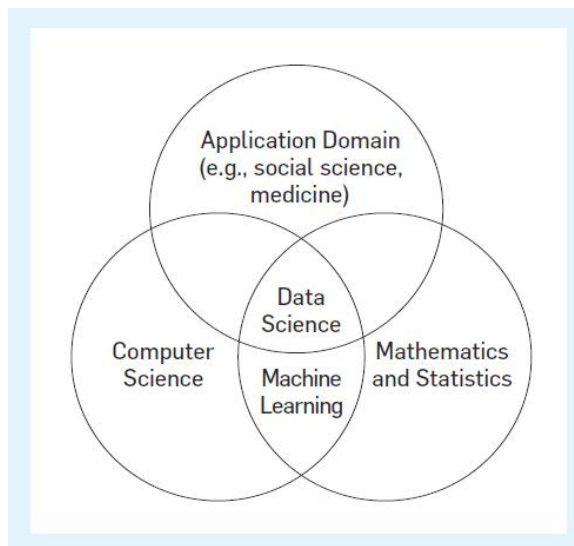
Conforme destacado por Coelho (2017), *Data Science*, ou Ciência de Dados, é uma área de estudo rigorosamente direcionada para dados e outras informações relacionadas às organizações e perspectivas sobre um determinado tema. Essencialmente, trata-se de uma ciência que se dedica à investigação das informações, abrangendo os estágios de recebimento, transformação e geração, culminando na análise estruturada desses dados.

Como também, de acordo com Provost e Fawcett (2016), a referida ciência visa primariamente melhorar a capacidade de uma organização tomar decisões sobre temas específicos, fato que sempre foi prioridade essencial para todas as empresas. Dessa forma, embora o pensamento de Alberto Boschetti e Luca Massaron (2016) dizer que tal disciplina representa um campo de conhecimento inovador, os seus elementos individuais já formam objetos de estudo e pesquisa ao longo de vários anos. Esses elementos abrangem tópicos como álgebra linear, modelagem estatística, visualização de dados, linguagem corporal, análise gráfica, aprendizado de máquina, inteligência empresarial, bem como o armazenamento e recuperação de dados. (MASSARON 2016, p.8)

A integração bem-sucedida na CD demanda a combinação de três disciplinas fundamentais: aptidão em ciência computação, matemática e estatísticas e conhecimento especializado, como evidenciado na Figura 1. De acordo com Vanderplas (2016, p. 11) *Data Science* é: “o conjunto de habilidades

interdisciplinares que estão se tornando cada vez mais importantes em muitas aplicações em toda a indústria e academia.”

Figura 1 - Ciência de Dados: disciplinas fundamentais.



Fonte: Adaptado (KOBY MIKE, 2022)

Novamente, segundo Provost e Fawcett (2016), a análise meticulosa dos dados visa extrair informações relevantes para a resolução de problemas e a geração de conhecimento. Os autores estabelecem uma analogia entre a CD e uma fábrica, onde a estrutura da primeira se assemelha a uma linha de produção industrial que requer recursos como mão-de-obra, matéria-prima e equipamentos para atingir seus objetivos. No contexto dessa ciência, os dados em si representam a matéria-prima, enquanto as modelagens matemáticas e estatísticas desempenham papéis análogos aos profissionais envolvidos, comparáveis, respectivamente, aos equipamentos e à mão-de-obra que moldam e transformam os mesmos. Esse processo visa a entrega de novas informações ou ideias, equiparadas ao produto gerado.

Conforme mencionado pelos autores acima, é imperativo aplicar técnicas e procedimentos da CD para oferecer soluções aos problemas reais através da participação ativa das organizações nesse domínio, a fim de alcançarem uma posição mais sólida na era da Indústria 4.0.

Para entender tais técnicas e procedimentos, se faz necessário conceituar os itens:

2.1.1 Mineração de Dados

A Mineração de Dados investiga dados em busca de padrões por meio de técnicas analíticas conduzidas por máquinas. Esse processo utiliza plenamente a computação na formulação de seus métodos de operação, conforme indicado por Vanderplas (2016).

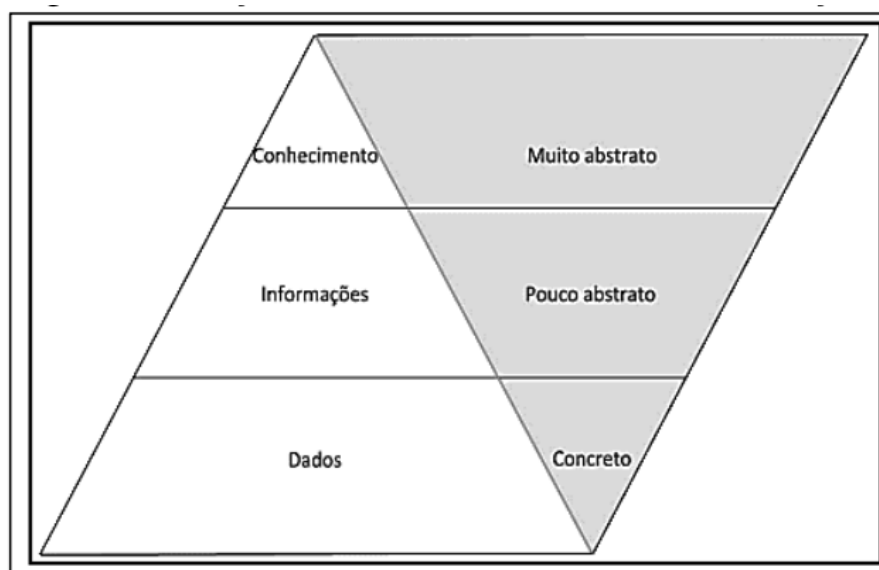
2.1.2 Dados

Dados são informações ou fatos pelos quais se deduzem outras informações adicionais. É possível conceituá-los como valores quantitativos que, quando combinados, permitem a mensuração. Esses valores podem ser obtidos por meio de observações documentadas, com a finalidade de obter informações posteriormente. De acordo com Setzer (1999, p.2), os dados podem ser armazenados e processados facilmente em qualquer ambiente computacional e tecnológico com essa finalidade específica.

Para alcançar um resultado relacionado ao valor do dado, são necessários diversos processos nos quais esses evoluem para informações e, posteriormente, para conhecimento útil, culminando na obtenção de *insights*. Conforme representado na Figura 2, ao utilizar um dado como informação, ele adquire uma maior abstração em relação ao seu significado e complexidade, elevando, assim, seu nível de abstração de informação para conhecimento.

A trajetória da transformação de dados em conhecimento é um processo complexo e multifacetado que envolve várias etapas. Inicialmente, os dados são coletados e documentados, fornecendo a base para análises posteriores. À medida que eles são processados, evoluem para informações, revelando padrões e tendências. Essas informações, por sua vez, constituem a matéria-prima para a construção do conhecimento, à medida que são interpretadas e contextualizadas.

Figura 2 - Relação entre conceitos de dados, informações.



Fonte: (THIERAUF, 1999, p. 23)

É possível afirmar que os computadores utilizam dados, enquanto os seres humanos lidam com informações. Esses princípios são cruciais para orientar o cientista de dados na tomada de decisões, visando gerar *insights* provenientes de análises. A eficácia dos resultados está diretamente relacionada ao nível de conhecimento que o profissional de *Data Science* possui sobre o objeto de estudo, destacando que um entendimento mais profundo contribui para a obtenção de resultados mais satisfatórios.

2.1.3 Dados para Ciência de Dados

Na Ciência de Dados, os *insights* são empregados como a fonte primordial de conhecimento. A partir do processamento das informações, podemos inferir que, quanto maior a escala de dados disponíveis, maior será a obtenção de resultados com informações precisas em relação ao objetivo proposto.

Há uma diferenciação no emprego de tipos de dados na CD, possibilitando a utilização de dados estruturados e não estruturados. O Quadro 1 delineia a distinção entre esses dois tipos conforme proposto por Elmasri e Navathe (2011).

Quadro 1 - Tipos de dados.

Tipo de Dados	Descrição
Dados estruturados	Esses dados são organizados em formato de

	tabelas, envolvendo linhas e colunas, tornando-se assim o tipo mais frequente e comum encontrado em diversas aplicações.
Dados semi-estruturados	Atuando como uma interface entre dados estruturados e não estruturados, esse tipo de dado não é completamente definido por um formato específico e é coletado de forma mais casual. Não segue um padrão estrutural definido para as informações e pode apresentar atributos em algumas entidades.
Dados não estruturados	Registra uma extensa quantidade de informações desejadas para armazenamento, sem se preocupar com a estrutura específica dos dados. É amplamente empregado para armazenar diversos tipos de dados, incluindo textos, vídeos, imagens, entre outros formatos.

Fonte: Adaptado de Elmasri; Navathe, p. 281 (2011)

2.1.4 Tomada de decisão orientada em dados

Tomada de Decisão Orientada por Dados (DOD) é a abordagem de fundamentar as decisões na análise de dados, em contraposição à confiança exclusiva na intuição. Por exemplo, um comerciante pode escolher anúncios com base em sua vasta experiência e intuição de que serão eficazes. No entanto, ele também pode decidir fundamentado na análise de dados sobre como os consumidores reagem a diferentes anúncios. Muitas vezes, a prática da DOD envolve uma combinação dessas abordagens, não sendo uma estratégia "tudo ou nada". Diversas empresas a adotam em graus variados, conforme explicado por Provost e Fawcett (2016).

Os benefícios da Tomada de Decisão Orientada por Dados têm sido inequivocamente comprovados. O economista Erik Brynjolfsson e seus colegas do MIT, juntamente com a Penns Wharton School, conduziram um estudo sobre como a DOD impacta o desempenho empresarial no uso de dados para decisões. Eles

demonstram que, estatisticamente, quanto mais uma empresa adota a abordagem orientada por dados, mais produtiva ela se torna, mesmo ao controlar uma ampla gama de possíveis fatores de confusão. As discrepâncias são significativas, com um aumento de 4%-6% na produtividade associado a um desvio padrão adicional na escala de DOD. Além disso, a DOD está correlacionada a um maior retorno sobre ativos, e a utilização dos mesmos, além de retorno sobre o patrimônio líquido e valor de mercado. Essa relação parece ser causal, conforme destacado por Provost e Fawcett (2016).

De acordo com Heinrichse (2013), a competição no atual mercado global demanda que as empresas possuam um nível de conhecimento superior ao que era necessário no passado. Além disso, para alcançar o sucesso, é essencial que as organizações tenham um entendimento mais aprofundado sobre seus clientes, mercados, tecnologias e processos, adquirindo essas informações antes de seus concorrentes.

Conforme destacado por Furlan e Filho (2005), possuir informações acessíveis é uma ferramenta poderosa para aqueles que precisam tomar decisões. Em consonância com esse princípio, as empresas iniciaram a prática de extrair dados de seus sistemas operacionais e armazená-los de forma separada dos dados operacionais.

A elaboração estratégica de qualquer empreendimento é sempre baseada nas informações disponíveis, o que significa que nenhuma estratégia pode superar a qualidade da informação da qual se origina (REZENDE, 2002).

De acordo com Patrocínio (2017), atualmente, observam-se dois cenários macro na aplicação de tomada de decisão orientada por dados, utilizando os princípios de CD: (1) descobertas derivadas de dados e (2) decisões repetitivas em larga escala. O primeiro cenário (1) está mais próximo do que é conhecido hoje como *Advanced Analytics*, em que as empresas adquirem novas informações simplesmente ao "analisar" os dados. Exemplos notáveis desse formato incluem casos no Walmart, como durante o furacão Frances e as associações de compra entre fraldas e cervejas, além da reorganização dos produtos em suas lojas físicas. Já o segundo cenário (2) pode ser exemplificado pelos sistemas de recomendação, nos quais a aplicação decide automaticamente quais produtos apresentar ao usuário.

2.2 Aprendizado de Máquina

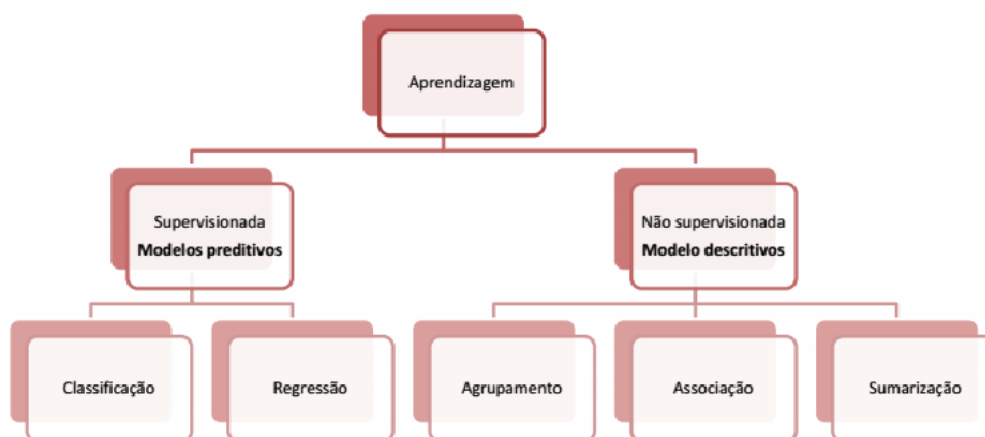
Nas últimas décadas, à medida que os desafios computacionais se tornaram mais complexos e a quantidade de dados gerados aumentou, ficou evidente a necessidade de ferramentas computacionais mais avançadas. Essas ferramentas deveriam ser mais autônomas, reduzindo, assim, a dependência da intervenção humana (FACELI, 2011). Para atender a essa demanda, é crucial que essas ferramentas possam, com base em experiências passadas (conjunto de treino), formular uma hipótese capaz de resolver o problema em questão. O processo pelo qual uma hipótese é derivada de informações anteriores é denominado Aprendizado de Máquina (AM) (FACELI, 2011).

No AM, os computadores são instruídos a aprender por meio de treinamento. Nesse processo, eles buscam derivar conclusões abrangentes a partir de um conjunto de exemplos específicos (FACELI, 2011). Dessa forma, os algoritmos de AM aprendem a deduzir uma função ou hipótese que seja capaz de solucionar um problema com base nos dados que representam instâncias desse problema. Esses dados compõem um conjunto conhecido como *dataset*, ou conjunto de dados em termos mais simples.

Cada elemento no conjunto de dados, conhecido por diferentes termos como objeto, padrão, exemplo ou registro, é composto por valores distintivos ou características que descrevem seus aspectos fundamentais, também referidos como campos ou variáveis. Em determinadas tarefas de aprendizado, um desses atributos é designado como atributo de saída (também chamado de atributo meta ou alvo), cujos valores podem ser estimados com base nos valores dos demais atributos (também denominados atributos preditivos) (FACELI, 2011).

Russell e Norvig (2009) categorizam os algoritmos de aprendizado de máquina em duas categorias: i) supervisionado, no qual os dados são etiquetados, ou seja, a máquina tem acesso à saída correta para cada entrada, e ii) não supervisionado, onde os dados de análise não possuem rótulos, ou seja, a máquina não tem conhecimento da saída correta para as entradas. Conforme representado na Figura 3

Figura 3 - Os tipos de Aprendizado de Máquina.



Fonte: Vilar (2017)

2.2.1 Aprendizado Supervisionado

Russell e Norvig (2009) explicam o aprendizado supervisionado da seguinte maneira: dado um conjunto $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subseteq X \times Y$ de treinamento com n pares de exemplos de entrada e saída, onde cada y_i foi gerado por uma função desconhecida $y_i = f(x_i)$, o objetivo é descobrir uma função h que se aproxime da função verdadeira f . Vale destacar que o domínio X e o contradomínio Y não precisam necessariamente ser valores numéricos.

A busca pela função de melhor desempenho na aprendizagem se estende além do conjunto conhecido, abrangendo também novos exemplos. A avaliação do desempenho da função h é realizada por meio de um conjunto de dados separado, conhecido como conjunto de testes, que difere do conjunto de treinamento. A função h , considerada uma hipótese, tem a responsabilidade de prever com precisão os valores de y_i para novos exemplos.

Quando a saída y faz parte de um conjunto discreto, a tarefa de aprendizado é denominada classificação, e busca prever o elemento que melhor representa y . Se, por outro lado, y pertencer a um conjunto contínuo, o desafio de aprendizado é chamado de regressão e procura estimar o valor \hat{y} mais próximo possível de y . Do ponto de vista técnico, a resolução de um problema de regressão envolve encontrar uma estimativa de y , uma vez que a probabilidade de obter exatamente o valor real de y tende a ser mínima (RUSSEL; NORVIG, 2009).

2.2.2 Regressão Linear

A regressão linear representa um dos métodos mais básicos de aprendizado de máquina supervisionado, especialmente quando o atributo alvo y é contínuo. Este modelo é frequentemente empregado em situações em que se busca criar uma função linear que minimize o erro quadrático, resultando em uma melhor adaptação a todos os pontos (exemplos do conjunto de treino). Para prever novos valores, o domínio, que pode consistir em uma ou várias variáveis, é submetido à função (RUSSEL; NORVIG, 2009).

2.2.3 Árvores de Decisão

Uma árvore de decisão é um modelo que opera com base em um vetor de valores de atributos como entrada, resultando em uma decisão única (um valor de saída) (RUSSEL; NORVIG, 2009). Tanto os valores de entrada quanto os de saída podem ser discretos ou contínuos. Quando lidam com valores de saída discretos, as árvores são denominadas árvores de classificação. Por outro lado, para prever valores de saída contínuos, é necessário recorrer a uma árvore de regressão (RUSSEL; NORVIG, 2009).

O processo de tomada de decisão de uma árvore de decisão se desenrola por meio da realização de uma sequência de testes. Cada nó na árvore corresponde a um teste do valor de um dos atributos de entrada, enquanto as ramificações desses nós são categorizadas com as possíveis respostas do teste (RUSSEL; NORVIG, 2009). Cada nó folha na árvore representa uma decisão que pode ser obtida através da função. A importância relativa de um atributo é maior quanto mais ele é utilizado para tomar decisões significativas durante o processo (RUSSEL; NORVIG, 2009).

2.2.4 Floresta Aleatória

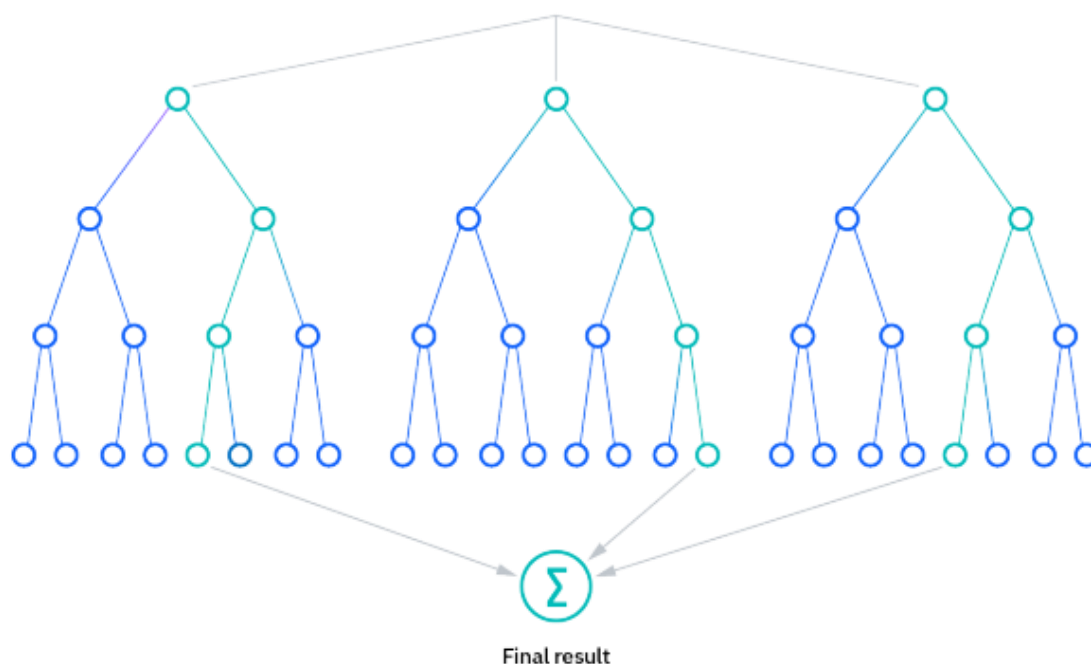
A floresta aleatória destaca-se como um dos algoritmos mais empregados em aprendizado supervisionado, devido à sua simplicidade e à sua aplicabilidade tanto em tarefas de classificação quanto de regressão. Essencialmente, trata-se de um conjunto (*ensemble*) de vários classificadores combinados para aprimorar os resultados ao realizar a predição de novos exemplos.

Este algoritmo realiza a separação de amostras de maneira aleatória a partir do conjunto de treino. Cada nova árvore é gerada utilizando uma amostra aleatória e um subconjunto de atributos também escolhidos aleatoriamente. Dentro desses atributos aleatórios, um é selecionado como o atributo de decisão mais representativo. Esse processo gera uma ampla diversidade de árvores, o que, em geral, resulta na criação de modelos mais robustos (OSHIRO, 2013). Uma das implementações mais eficientes da floresta aleatória é o *Extreme Gradient Boosting* (XGBoost).

Os algoritmos de floresta aleatória têm três hiper parâmetros principais que precisam ser configurados antes do treinamento. Isso inclui o tamanho do nó, o número de árvores e o número de características amostradas. A partir daí, o classificador de floresta aleatória pode ser utilizado para resolver problemas de regressão ou classificação.

O algoritmo de floresta aleatória é composto por uma coleção de árvores de decisão, e cada árvore no conjunto é constituída por uma amostra de dados retirada de um conjunto de treinamento com reposição, chamada de amostra *bootstrap*. Dessa amostra de treinamento, um terço é reservado como dados de teste, conhecidos como amostra *out-of-bag* (oob). Outra instância de aleatoriedade é então introduzida através do "*feature bagging*", adicionando mais diversidade ao conjunto de dados e reduzindo a correlação entre as árvores de decisão. Dependendo do tipo de problema, a determinação da predição variará. Para uma tarefa de regressão, as árvores de decisão individuais serão médias, e para uma tarefa de classificação, uma votação majoritária — ou seja, a variável categórica mais frequente — resultará na classe prevista. Finalmente, a amostra oob é então usada para validação cruzada, finalizando essa predição. Conforme representado na Figura 4.

Figura 4 - Árvores de decisões.



Fonte: IBM (2019)

2.2.5 Overfitting

Overfitting refere-se ao fenômeno de um ajuste excessivo aos dados. Em outras palavras, ocorre quando a hipótese se adapta de maneira muito precisa ao conjunto de dados utilizado durante o treinamento, mas revela-se ineficaz na predição de novos exemplos. Nesse cenário, também é descrito que a hipótese memorizou ou se especializou demasiadamente no conjunto de treinamento (FACELI, 2011).

Para evitar o *overfitting*, é essencial impedir que o modelo preditivo memorize excessivamente o conjunto de treinamento. Em modelos iterativos, a estratégia ideal envolve interromper o processo de aprendizado quando o erro associado ao conjunto de treinamento começa a exibir um decréscimo insignificante (FACELI, 2011), conhecido como *early stopping*. O *early stopping* é uma abordagem frequentemente eficaz para mitigar o super ajuste, consistindo na interrupção do aprendizado quando o erro (considerando a métrica de desempenho utilizada) deixa de diminuir, por exemplo, após uma sequência de n iterações.

Em certos algoritmos, é viável evitar o *overfitting* por meio do ajuste de hiper parâmetros, já que alguns desses parâmetros contribuem para melhorar a diversidade das árvores geradas pelo modelo. No entanto, é importante ressaltar

que o verdadeiro objetivo do ajuste de parâmetros é identificar uma configuração que otimize o desempenho do algoritmo empregado (FACELI, 2011). Uma das técnicas amplamente reconhecidas para a sintonia de hiper parâmetros é o *grid search*, que utiliza conjuntos predefinidos de valores para cada parâmetro, realizando uma análise combinatória para encontrar a melhor configuração possível.

3 FERRAMENTAS E TÉCNICAS

Este capítulo apresenta as ferramentas e tecnologias usadas para o desenvolvimento da plataforma.

3.1 Visual Studio Code (VSCode)

Desde o seu lançamento em 2015, o Visual Studio Code (VSCode), desenvolvido pela Microsoft, se estabeleceu como uma das ferramentas mais eficientes para o desenvolvimento de aplicativos no mercado. De acordo com a pesquisa *Stack Overflow Developer Survey* de 2023, o VSCode é o editor de código mais popular entre os desenvolvedores, sendo adotado por mais de 70% dos entrevistados (STACK OVERFLOW, 2022).

Entre suas principais vantagens, destaca-se a capacidade de suportar uma ampla variedade de linguagens de programação, desde as mais comuns até as menos conhecidas. O VSCode oferece alta configurabilidade, permitindo que os desenvolvedores personalizem a interface e incorporem funcionalidades conforme suas preferências. Sua popularidade é impulsionada pela integração com diversas ferramentas e extensões, abrangendo *linters*, *debuggers*, *plugins* para bancos de dados e *frameworks* específicos, proporcionando uma experiência de desenvolvimento otimizada. O VSCode está em constante evolução, mantendo-se poderoso e flexível graças a sua equipe de desenvolvimento ativa e à colaboração contínua da comunidade de usuários. Como *software* de código aberto, promove uma cultura colaborativa e o compartilhamento de ideias e soluções. Em resumo, o VSCode é um editor de código-fonte completo, flexível e altamente personalizável, com suporte a uma ampla variedade de linguagens e integração com diversas ferramentas e extensões (MICROSOFT, 2022).

O principal critério de escolha para esta ferramenta é o suporte a múltiplas linguagens, possibilitando o uso de uma única ferramenta para escrever códigos. Além disso, o VSCode é um *software* gratuito, e suas extensões simplificam a análise e correção de códigos nas linguagens utilizadas.

3.2 Python

Guido van Rossum criou a linguagem *Python* em 1991, com uma filosofia que valoriza o esforço do programador sobre o esforço computacional, dando prioridade à legibilidade do código por meio de uma sintaxe elegante, concisa e clara. *Python* é uma linguagem interpretada de alto nível que combina elementos de orientação a objetos, programação procedural e funcional. Possui tipagem dinâmica, tipagem forte e é compatível com várias plataformas. Uma característica única do *Python* em relação à sintaxe léxica, não presente na maioria das linguagens de programação, é a utilização de blocos de código delimitados por endentação, dispensando delimitadores do tipo *BEGIN*, *END* ou { e } (PYTHON, 2009).

Suas características incluem a capacidade de expressar operações complexas em um único comando por meio de tipos de dados de alto nível. Além disso, não é necessário realizar a declaração explícita de variáveis ou parâmetros formais.

O *Python* conta com uma licença livre em conformidade com a *General Public License* (GPL), o que significa que não impõe restrições quanto à utilização e venda. Essa característica, somada às várias outras mencionadas anteriormente, confere uma significativa atratividade, levando profissionais da área a estudarem e adotarem o *Python* como uma ferramenta essencial em seus trabalhos profissionais (PYTHON, 2009).

3.3 Framework Angular

"*Angular* é mais do que um simples *framework*; é uma abrangente plataforma de desenvolvimento projetada para criar *single-page apps* eficientes e sofisticadas." (ANGULAR, 2020).

Seshadri e Green (2014) afirmam que o *Angular* é uma biblioteca criada por engenheiros do Google, que optaram por desenvolvê-la com o objetivo de simplificar a manutenção de alguns de seus serviços *web*. Inicialmente, um código extenso de 18 mil linhas foi substancialmente reduzido para 1.5 mil linhas, representando uma significativa redução de aproximadamente 91% no volume total de código.

Esse progresso foi alcançado principalmente devido à considerável modularidade e reusabilidade incorporadas. A partir dessa concepção, os engenheiros direcionaram seus esforços para estabelecer um método simplificado

no desenvolvimento de aplicações *web*. O projeto pioneiro a empregar o *Angular* foi o Google Feedback, servindo como uma referência significativa para estudos sobre o funcionamento e a utilização de um *framework JavaScript* do ponto de vista dos desenvolvedores (SESHADRI; GREEN, 2014).

Em operação, o *Angular* organiza a aplicação de maneira semelhante ao padrão *Model-View-Controller* (MVC). Os dados são predominantemente apresentados de maneira direta por meio de estruturas JSON (*JavaScript Object Notation*), denominadas modelos. A interação do usuário é representada pela visão. As regras de negócios determinam quais partes do modelo serão ou não renderizadas. Essas regras são descritas nos controladores, que correspondem às lógicas aplicadas no funcionamento do sistema.

Podem-se destacar como vantagens dessa abordagem a clara separação entre as camadas da aplicação, resultando em um desenvolvimento mais usável e de fácil manutenção. A ausência de referências diretas das visões nos controladores proporciona independência, permitindo testes rápidos e simples sem a necessidade de instanciar um domínio.

3.4 Firebase

O *Firebase* é uma plataforma digital que simplifica o desenvolvimento de aplicações *web* e *mobile*, sendo categorizado como *Back-End as a Service* (BaaS). Criado por James Tamplin e Andrew Lee em 2011, o *Firebase* foi posteriormente adquirido pelo Google em 2014 (COODESH, 2021). O termo BaaS é definido por Batschinski (2016) como:

Um serviço de computação em nuvem que serve como *middleware*. O mesmo fornece aos desenvolvedores uma forma para conectar suas aplicações *mobile* e *web* a serviços na nuvem.

Conforme Andrade (2020), o BaaS é um tipo de serviço que oferece a infraestrutura e o *back-end* de uma aplicação de maneira simplificada, eliminando a necessidade de desenvolver essa solução manualmente. Algumas das funcionalidades mais comuns fornecidas pelo BaaS incluem autenticação, armazenamento, notificações, escalabilidade, entre outras. Dessa forma, os

desenvolvedores podem concentrar seus esforços exclusivamente na construção do *front-end*.

O *Firebase* oferece uma variedade de recursos, incluindo *realtime database*, *Google Analytics*, *Cloud Firestore*, *Authentication*, *Cloud Storage*, entre outros (FIREBASE, 2021). Neste trabalho, destacamos o uso do *Cloud Firestore*, *Authentication*. O *Cloud Firestore* é um banco de dados não relacional flexível e escalável, que sincroniza em tempo real com os aplicativos clientes e oferece suporte *offline* (FIREBASE, 2021). O *Authentication* é responsável por proporcionar uma autenticação segura e uma experiência de *login* aprimorada para o usuário final, permitindo *logins* por senhas, números de telefone, Google, Facebook, Twitter, GitHub, Apple, entre outros métodos (FIREBASE, 2021).

Para a realização deste trabalho, foram utilizados alguns serviços em *Firebase*, conforme indicado no Quadro 2.

Quadro 2 - Serviços *Firebase*.

Nome	Descrição
<i>Authentication</i>	Sistema de autenticação
<i>Firestore Database</i>	Banco de dados não relacional.

Fonte: Elaborado pelo autor.

3.5 Arquitetura do sistema

Este subcapítulo, apresenta a arquitetura que será utilizada na construção do protótipo do sistema.

3.6 Arquitetura Cliente-Servidor

A comunicação cliente-servidor é essencial para o funcionamento de diversos sistemas utilizados. Ela ocorre por meio do protocolo *Hypertext Transfer Protocol* (HTTP), em que o cliente indica o tipo de ação desejada e envia uma requisição ao servidor. Essa requisição é processada e, em seguida, recebe uma resposta (MENDES, 2021).

Conforme Caio Mendes (2021), os principais métodos de requisição que possibilitam ao cliente especificar o tipo de ação desejada estão descritos na Quadro 3.

Quadro 3 - Métodos de Requisição.

Métodos	Responsabilidade
<i>GET</i>	Retorna um dado do servidor.
<i>POST</i>	Alterar um dado no servidor.
<i>PUT</i>	Envia um dado para o servidor.
<i>DELETE</i>	Deleta um dado no servidor.

Fonte: Caio Mendes (2021).

Conforme Caio Mendes (2021), quando o servidor retorna uma resposta, ele o faz acompanhado de um status que define o resultado da requisição realizada. Os principais status estão descritos na Quadro 4.

Quadro 4 - Códigos de status HTTP.

Status	Tipo de resposta
1XX	Informação
2XX	Sucesso
3XX	Redirecionamento
4XX	Erro no cliente
5XX	Erro no servidor

Fonte: Caio Mendes (2021).

4 DESCRIÇÃO GLOBAL DA PLATAFORMA DIGITAL

Neste capítulo, é fornecida uma visão abrangente da plataforma, abordando suas interfaces, funcionalidades e limitações.

A criação da plataforma envolveu o uso de *Python* no desenvolvimento do servidor, enquanto o *framework Angular* foi empregado para a construção do cliente. A principal finalidade da plataforma abrange a pesquisa de produtos no *site* da Amazon, a capacidade de salvar itens e a visualização detalhada de produtos específicos para obter informações adicionais.

4.1 Interfaces

Esta seção oferece uma visão geral das interfaces da plataforma, abrangendo a interface do sistema e a interface de comunicação.

4.2 Interfaces do Sistema

Login: Esta opção possibilita que o usuário faça *login* utilizando as credenciais previamente estabelecidas, incluindo o nome de usuário e a senha registrados na tela de cadastro. Além disso, deve incorporar uma funcionalidade para a recuperação de senha por meio do envio de um e-mail, bem como um mecanismo para o cadastro de novos clientes.

Cadastro: Durante o processo de cadastro, o sistema solicitará informações essenciais, como nome de usuário, senha (com confirmação) e e-mail. Além disso, será incorporada a funcionalidade que permite ao usuário retornar facilmente para a tela de *login*.

Recuperação de Senha: Esta opção solicitará o e-mail cadastrado, possibilitando o envio de uma mensagem que permitirá ao usuário redefinir a senha por uma nova. Além disso, será incorporada a funcionalidade que permite ao usuário retornar facilmente para a tela de *login*.

Tela Inicial: Permite que o usuário visualize os resultados da pesquisa em uma tabela com base no *input* fornecido. Dois botões são fornecidos para aprimorar a

experiência do usuário: um para visualizar detalhes do produto e outro para salvar o produto.

Produto: Permite que o usuário visualize um produto específico que destaca informações específicas sobre o produto escolhido. Nessa interface, além dos detalhes básicos, como nome e preço, são apresentados *insights* adicionais e estimativa de vendas.

Meus Produtos: Permite que o usuário visualize os produtos em uma listagem organizada dos itens previamente salvos. Cada produto é apresentado de forma clara, exibindo detalhes relevantes como nome, preço e classificação.

4.3 Interface de Comunicação

O sistema vai usar o *Firebase* como banco de dados para realizar o armazenamento dos produtos salvos, e realizar a autenticação da plataforma.

4.4 Funções do Sistema

- Visualizar produtos do *site* Amazon;
- Visualizar estimativa de vendas de um produto;
- Salvar produtos;
- Visualizar produtos salvos;

4.5 Restrições

- 1- A utilização do aplicativo está condicionada à presença de uma conexão com a internet.
- 2- Se o usuário não estiver autenticado, será direcionado para a tela de *login*.
- 3- O *site* Amazon precisa estar funcionando.

5 MODELO PROPOSTO DA PLATAFORMA DIGITAL

Neste capítulo apresentam-se conceitos pivotais do desenvolvimento da plataforma digital, tal como requisitos funcionais (RF), requisitos não funcionais (RNF), diagrama de casos de uso, casos de uso descritivo, a arquitetura escolhida e tecnologias utilizadas.

5.1 Requisitos Funcionais

O quadro 5 apresenta os requisitos funcionais.

Quadro 5 - Requisitos Funcionais.

Id.	Descrição
RF 01	O usuário deve ter a capacidade de efetuar o <i>login</i> utilizando um nome de usuário e senha válidos.
RF 02	O usuário deve ser capaz de recuperar uma senha perdida.
RF 03	O usuário deve ter a capacidade de criar seu meio de acesso (<i>login</i>) fornecendo os seguintes dados devidamente validados: nome de usuário, senha e e-mail.
RF 04	O usuário deve ser capaz de visualizar a tela de <i>login</i> ao acessar a plataforma.
RF 05	O usuário deve ser capaz de pesquisar produtos de acordo com o <i>input</i> .
RF 06	O usuário deve ser capaz de visualizar resultados da pesquisa dos produtos.
RF 07	O usuário deve ser capaz de visualizar um produto específico para obter dados adicionais.
RF 08	O produto deve conter os seguintes dados: Nome, Preço, Estimativa de Vendas, Demanda, Potencial de Receita, Concorrência, Custos Gerais, Margem de Lucro.
RF 09	O usuário deve ser capaz de salvar um produto para visualização posterior ou referência.
RF 10	O usuário deve ser capaz de visualizar produtos salvos.
RF 11	O usuário deve ser capaz de efetuar o <i>log out</i> da plataforma.

Fonte: Elaborado pelo autor.

5.2 Requisitos Não Funcionais

O quadro 6 apresenta os requisitos não funcionais.

Quadro 6 - Requisitos não Funcionais.

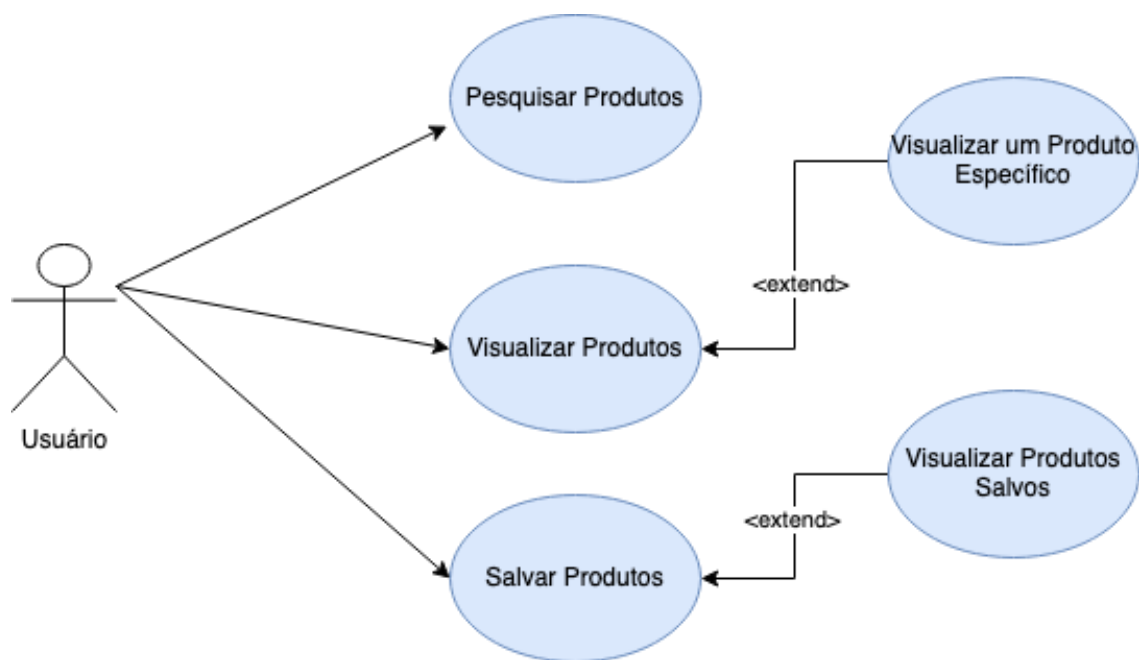
Id.	Descrição
RNF 01	As informações dos usuários são restritas e podem ser acessadas ou visualizadas apenas pelo próprio usuário.
RNF 02	Plataforma deve ser responsivo e intuitivo.
RNF 03	O servidor deve ser escrito na linguagem <i>Python</i> .
RNF 04	O cliente deve ser escrito na linguagem <i>Typescript</i> e usar o framework <i>Angular</i> .
RNF 05	O banco de dados deve ser o serviço <i>Firebase</i> .
RNF 06	O Algoritmo Floresta Aleatória deve ser implementado na linguagem <i>Python</i> .
RNF 07	<i>Web Scraping</i> deve ser implementado na linguagem <i>Python</i> .
RNF 08	O servidor deve ser capaz de suportar mais de um usuário simultaneamente.

Fonte: Elaborado pelo autor.

5.3 Diagrama de Casos de Uso

A figura 5 apresenta o diagrama de casos de uso do sistema.

Figura 5 - Diagrama de usos.



Fonte: Elaborado pelo autor.

5.4 Casos de Uso Descritivos (CSU)

Os quadros 7, 8 e 9 apresentam os casos de uso descritivos.

Quadro 7- CSU 01: Pesquisar Produtos.

Identificador: CSU 01
Nome: Pesquisar Produtos
Responsável: Henrique Franchini Zani
Requisitos Relacionados:
Descrição/Resumo: Permite realizar a pesquisa de produtos
Atores: Usuário
Pré-condições: Os atores devem estar conectados à internet e ter acessado o sistema.
Pós-condições: O sistema deve pegar dados do <i>site</i> Amazon.
Cenário Principal:
1- O usuário inseri o <i>input</i> da pesquisa.
2- O usuário clica no botão pesquisar.
3- O sistema carrega os produtos obtidos do <i>site</i> Amazon
4- Caso de uso encerrado.

Cenário de Exceção:
1- O sistema está fora de acesso.
2- O usuário não efetuou o <i>login</i> .

Fonte: Elaborado pelo autor.

Quadro 8 - CSU 02: Visualizar Produtos.

Identificador: CSU 02
Nome: Visualizar Produtos
Responsável: Henrique Franchini Zani
Requisitos Relacionados:
Descrição/Resumo: Permite visualizar o resultado da pesquisa dos produtos
Atores: Usuário
Pré-condições: Os atores devem estar conectados à internet e ter acessado o sistema.
Pós-condições: O sistema deve ter carregado os produtos do <i>site</i> Amazon.
Cenário Principal:
1- O sistema apresenta uma lista dos produtos com paginação.
2- O sistema mostra a mensagem “Dados obtidos com sucesso”
3- Caso de uso encerrado.
Cenários Alternativos:
1ª – Fluxo alternativo – Ator visualizar os dados do produto.
1- Usuário seleciona um produto específico.
2- O sistema apresenta dados adicionais do produto.
3- Caso de uso encerrado.
Cenário de Exceção:
1- O sistema está fora de acesso.
2- O usuário não efetuou o <i>login</i> .
3- O sistema mostra a mensagem “Erro ao tentar obter produtos”.

Fonte: Elaborado pelo autor.

Quadro 9 - CSU 03: Salvar Produtos.

Identificador: CSU 03
Nome: Salvar Produtos

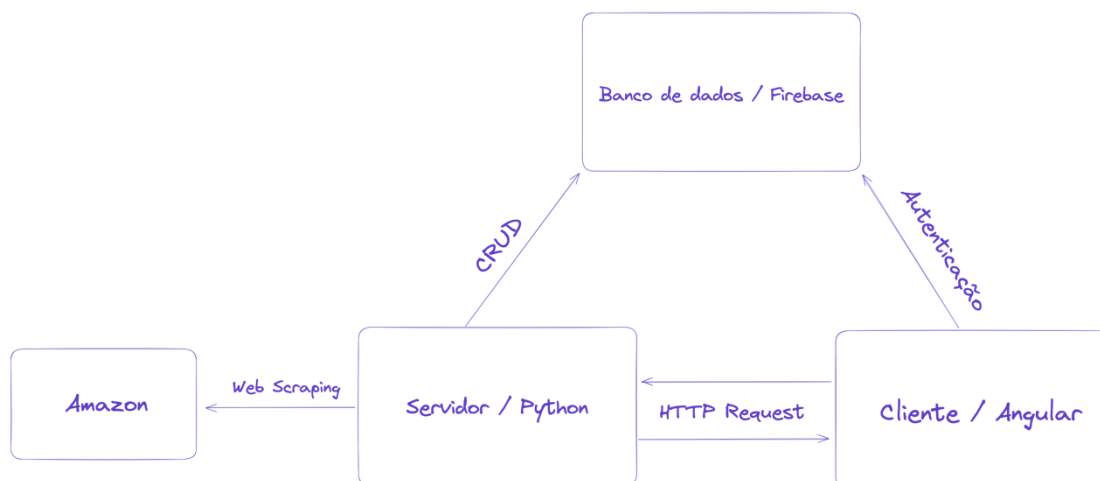
Responsável: Henrique Franchini Zani
Requisitos Relacionados:
Descrição/Resumo: Permite salvar um produto para visualização posterior ou referência.
Atores: Usuário
Pré-condições: Os atores devem estar conectados à internet e ter acessado o sistema.
Pós-condições: O sistema deve salvar o produto no banco de dados.
Cenário Principal:
1- O usuário clica no botão de salvar.
2- O sistema salva o produto no serviço Firebase.
3- Caso de uso encerrado.
Cenários Alternativos:
1ª – Fluxo alternativo – Ator visualizar os produtos salvos.
1- Usuário visualiza os produtos salvos.
2- Caso de uso encerrado.
Cenário de Exceção:
1- O sistema está fora de acesso.
2- O usuário não efetuou o <i>login</i> .
3- O sistema mostra a mensagem “Erro ao tentar obter produtos salvos”.

Fonte: Elaborado pelo autor.

5.5 Modelo da Arquitetura

A figura 6 mostra o modelo da arquitetura utilizada.

Figura 6 - Arquitetura utilizada.



Fonte: Elaborado pelo autor.

5.5.1 Servidor

O servidor, é a entidade que responde às requisições dos clientes, fornecendo os serviços ou recursos solicitados. Pode ser um computador dedicado ou um *software* específico em execução em um sistema. O servidor permanece em espera para receber requisições dos clientes, processa essas requisições e retorna as respostas correspondentes (FEMI, 2018).

O servidor, implementado em *Python*, desempenha um papel central na plataforma. Ele estabelece uma comunicação bidirecional essencial: por um lado, interage com o banco de dados *Firebase*, gerenciando operação de armazenamento de produtos; por outro lado, comunica-se com o cliente por meio do protocolo HTTP, facilitando a transferência de dados e comandos entre o servidor e a interface do usuário. Além disso, o servidor realiza *Web Scraping* no *site* da Amazon para coletar informações sobre produtos, e incorpora o algoritmo Floresta Aleatória que foi citado anteriormente para classificar esses produtos, fornecendo uma camada analítica robusta à plataforma.

5.5.2 Web Scraping

O *Web Scraping* é uma técnica de extração de dados da *web* por meio da automação de requisições HTTP para acessar e recuperar informações de páginas da internet. Essa prática envolve a análise estruturada do código-fonte das páginas,

permitindo a identificação e extração de dados específicos (VELOTIO TECHNOLOGIES, 2019).

5.5.3 Cliente

O cliente é a entidade que solicita serviços ou recursos a um servidor. Pode ser um *software*, como um navegador *web*, ou até mesmo um *hardware*, como um computador. O cliente inicia a comunicação, enviando requisições ao servidor para obter informações ou realizar operações específicas (MENDES, 2021).

A parte de visualização da plataforma é desenvolvida utilizando o *framework Angular*. O cliente é a interface que os usuários interagem, apresentando os resultados do algoritmo Floresta Aleatória fornecidos pelo servidor, bem como permitindo a navegação e interação com os produtos. Essa parte da arquitetura, com sua estrutura dinâmica e responsiva, garante uma experiência do usuário eficaz e intuitiva.

5.5.4 Banco de dados

Na arquitetura Cliente-Servidor, o banco de dados desempenha um papel central na gestão e persistência de dados essenciais para a aplicação. Ele atua como um repositório centralizado, armazenando informações cruciais como autenticação de usuários, dados de produtos e outras variáveis relevantes. Ao intermediar a comunicação entre o servidor e o cliente, o banco de dados garante a integridade e consistência dos dados, possibilitando que múltiplos clientes acessem e atualizem informações simultaneamente (MENDES, 2021).

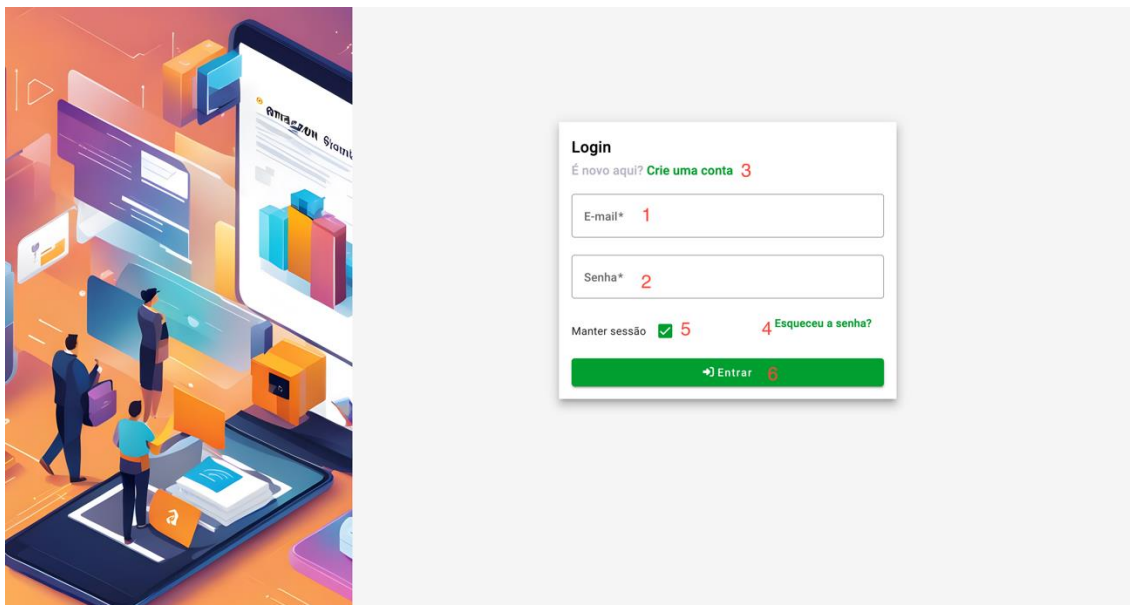
6 RESULTADOS

Neste capítulo, serão mostrados os resultados do objetivo geral proposto. Interfaces do sistema, códigos-fonte relevantes e implementações práticas que exemplificam a aplicação dos conceitos discutidos no referencial teórico.

6.1 Protótipo Plataforma Digital

A figuras 7, 8, 9, 10, 11, 12, 13 apresentam as telas da plataforma digital.

Figura 7 - Tela de *login*.



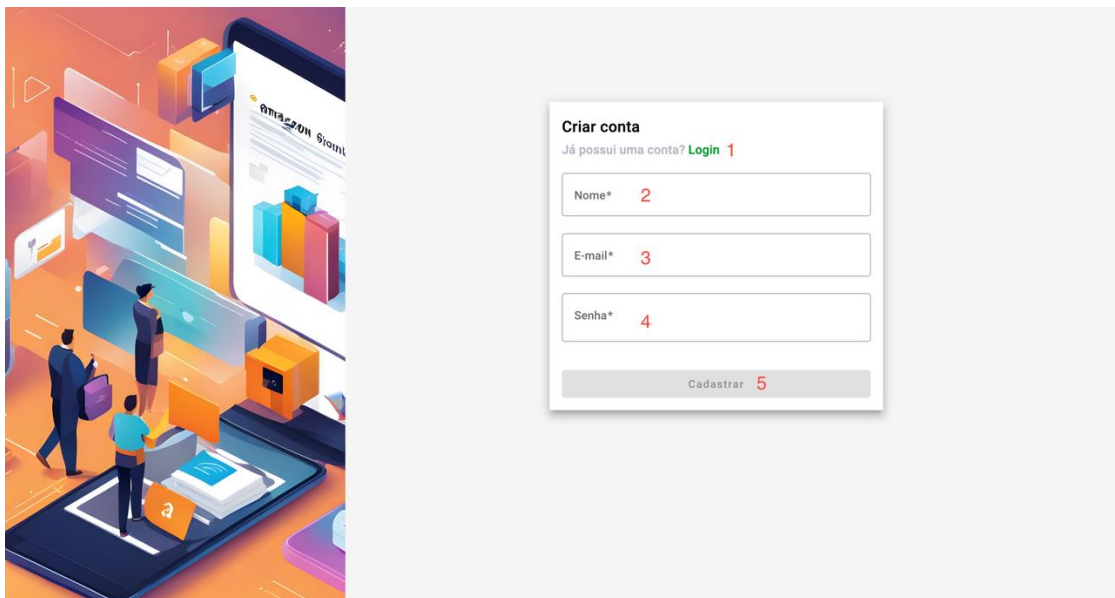
Fonte: Autoria própria.

A tela inicial é a primeira interface que surge quando o usuário acessa a plataforma, proporcionando a ele a opção de realizar o *login* no sistema, efetuar um novo registro ou recuperar sua conta.

- Pontos 1 e 2: Campos nos quais o usuário deve inserir suas credenciais previamente cadastradas.
- Ponto 3: Um botão que direciona o usuário para a tela de cadastro.
- Ponto 4: Um botão que direciona o usuário para a tela de recuperar senha.
- Ponto 5: Uma caixa de seleção para manter a sessão do usuário.

- Ponto 6: Um botão que o usuário deve pressionar após preencher os campos de nome de usuário e senha para realizar o *login* no aplicativo.

Figura 8 - Tela de cadastro.



Criar conta
Já possui uma conta? [Login](#) 1

Nome* 2

E-mail* 3

Senha* 4

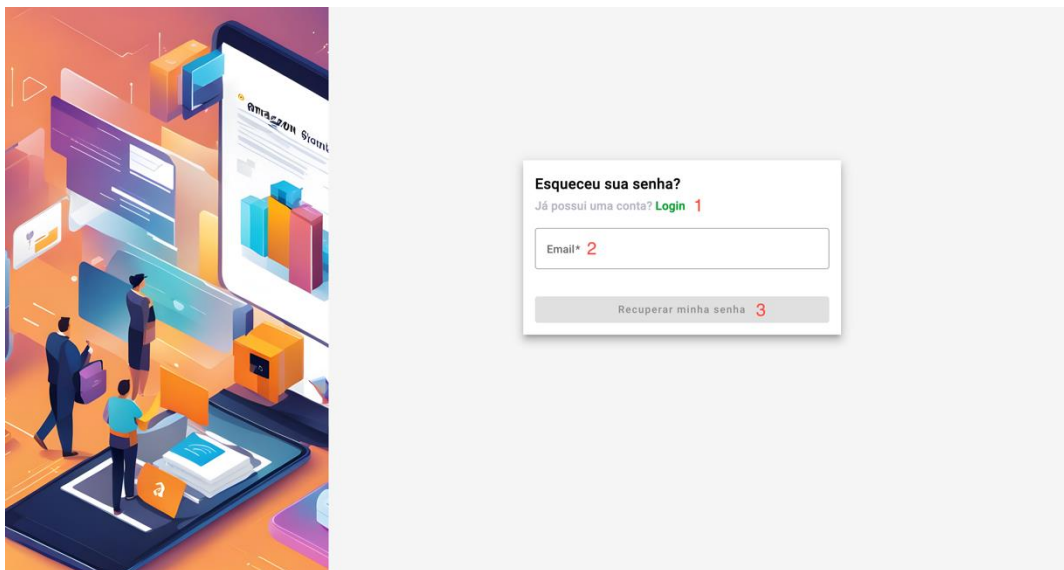
Cadastrar 5

Fonte: Autoria própria.

A tela de registro no sistema, acessível a partir da tela de *login* ao pressionar o botão "Cadastrar", é onde o usuário realiza seu cadastro.

- Ponto 1: Um botão que direciona o usuário para a tela de *login*.
- Pontos 2, 3, 4: Campos nos quais o usuário deve inserir seus dados correspondentes.
- Ponto 5: Após preencher os campos mencionados anteriormente, o usuário deve clicar neste local para registrar as informações.

Figura 9 - Recuperação de senha.



Fonte: Autoria própria.

A opção para recuperar a senha está disponível na tela de *login*, bastando pressionar o botão recuperar minha senha.

- Ponto 1: Um botão que direciona o usuário para a tela de *login*.
- Ponto 2: Um campo que o usuário precisa preencher para recuperar a senha.
- Ponto 3: Um botão que o usuário clica para recuperar a senha.

Figura 10 - Tela de Início.

Produto Certo

Início 1 Meus Produtos 2 H

Video Games

ps5

Pesquisar

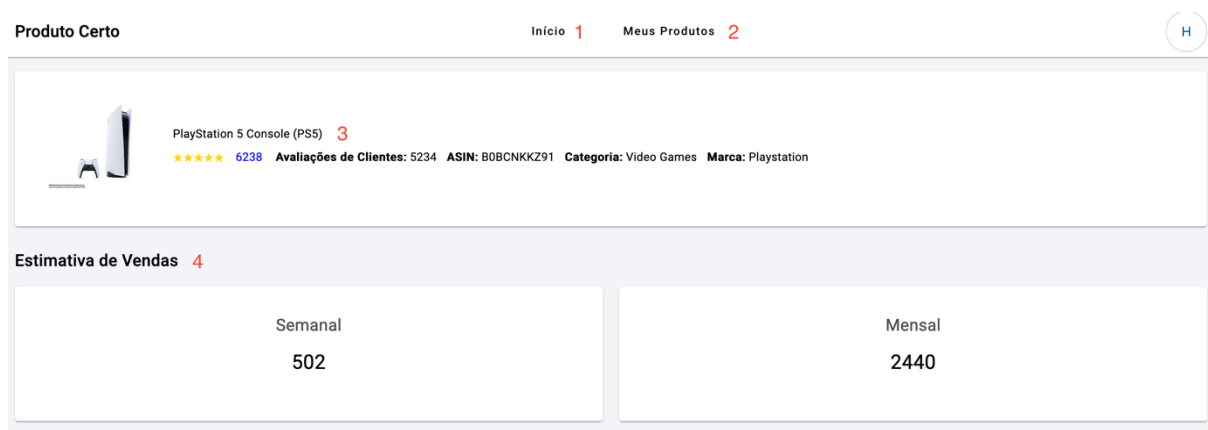
#	Título	Preço	Ações
1	PlayStation 5 Console - Marvel's Spider-Man 2 Bundle (slim) ★★★★★ 178	\$499.00	Visualizar
2	PlayStation®5 Digital Edition (slim) ★★★★★ 2	\$449.99	Visualizar
3	PlayStation®5 Console – Marvel's Spider-Man 2 Bundle ★★★★★ 102	\$499.00	Visualizar
4	PlayStation 5 Console CFI-1102A ★★★★★ 8088	0	Visualizar
5	Star Wars Jedi: Survivor Playstation 5 Video Game English EU Version Region Free ★★★★★ 1085	\$48.46	Visualizar
6	PlayStation DualSense Wireless Controller ★★★★★ 94746	\$69.00	Visualizar
7	Call of Duty Modern Warfare III - PS5 ★★★★☆ 156	\$59.99	Visualizar
8	Among Us: Ejected Edition - PlayStation 5 ★★★★★ 91	\$49.93	Visualizar
9	NowSkins Superhero Spider - Man PS5 Skin for Playstation 5, Premium 3M Vinyl Cover Skins Wraps Set for Playstation 5 Disc Edition and PSS Controller Stickers (PSS Disc Edition) ☆☆☆☆☆	\$25.99	Visualizar

Fonte: Autoria própria.

A tela onde o usuário pode pesquisar produtos, visualizar ou salvar um produto específico é acessada logo após realizar o *login*.

- Ponto 1: Um botão que direciona o usuário para a tela inicial.
- Ponto 2: Um botão que direciona o usuário para a tela dos produtos salvos.
- Pontos 3, 4: Campos nos quais o usuário deve inserir seus dados para realizar a pesquisa.
- Ponto 5: Um botão que realiza a busca dos produtos.
- Ponto 6: O usuário visualiza o resultado das pesquisas.
- Ponto 7: Um botão que visualiza um produto específico.
- Ponto 8: Um botão que salva um produto no banco de dados.

Figura 11 - Tela do Produto.



Fonte: Autoria própria.

Figura 12 - Tela do Produto.

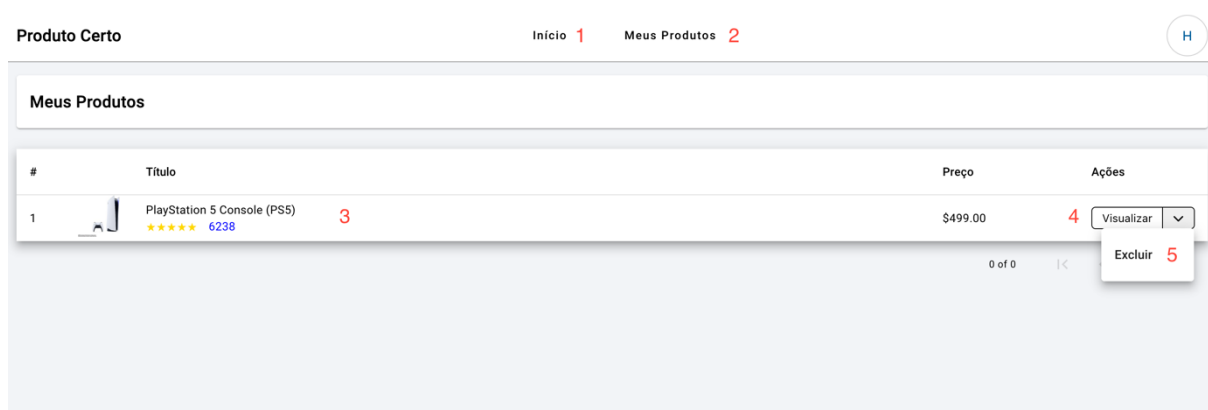


Fonte: Autoria própria.

A tela onde o usuário pode visualizar um produto específico, para obter insights extras e estimativa de vendas.

- Ponto 1: Um botão que direciona o usuário para a tela inicial.
- Ponto 2: Um botão que direciona o usuário para a tela dos produtos salvos.
- Ponto 3: O usuário visualiza detalhes do produto.
- Ponto 4: O usuário visualiza a estimativa de vendas que é obtido de acordo com o algoritmo Floresta Aleatória.
- Ponto 5: O usuário visualiza *insights* extras do produto de acordo com número de *reviews*, estimativa de vendas, número de estrelas, classificação dos mais vendidos e volumes mensais de vendas unitárias.

Figura 13 - Tela dos Produtos Salvos.



Fonte: Autoria própria.

A tela onde o usuário pode visualizar os produtos salvos.

- Ponto 1: Um botão que direciona o usuário para a tela inicial.
- Ponto 2: Um botão que direciona o usuário para a tela dos produtos salvos.
- Ponto 3: O usuário visualiza os produtos salvos.
- Ponto 4: Um botão que visualiza um produto específico.
- Ponto 5: Um botão para excluir um produto salvo do banco de dados.

6.2 Códigos-Fonte

6.2.2 Algoritmo Web Scraping

Figura 14 - Rota Search.

```
@app.route('/search', methods=['POST'])
def search():
    # Analisa os dados JSON da solicitação POST
    data = request.get_json()

    # Configura o registro para mensagens de informação
    logging.basicConfig(level=logging.INFO)

    # Verifica se 'search_input' está presente nos dados JSON
    if 'search_input' not in data:
        return jsonify({'error': 'Não existe nenhum valor para realizar a pesquisa'}), 400

    # Extrai a entrada de pesquisa dos dados JSON
    search_string = data['search_input']

    # Obtém o URL para a string de pesquisa
    url = get_url(search_string)

    # Verifica se 'url' está presente nos dados JSON
    if 'url' not in data:
        return jsonify({'error': 'URL inválido'}), 400

    # Inicializa uma lista vazia para armazenar todos os resultados
    all_results = []

    # Registra o URL para fins de depuração
    logging.info(url)

    # Obtém o conteúdo HTML do URL especificado
    html = get_page_html(url)

    # Extrai informações do produto do HTML
    products = get_products(html)

    # Extrai informações de avaliações do HTML
    reviews = get_reviews_from_html(html)

    # Adiciona informações do produto à lista de resultados
    all_results.append(products)

    # Processa cada avaliação e adiciona os resultados à lista
    for rev in reviews:
        data = orchestrate_data_gathering(rev)
        all_results.append(data)

    # Registra todos os resultados para fins de depuração
    logging.info(all_results)

    # Retorna os resultados como JSON
    return jsonify({'results': all_results})
```

Fonte: Autoria própria.

O algoritmo que realiza o Web Scraping no *site* Amazon.

Figura 15 - Função para obter os produtos.

```
def get_products(page_html: str) -> list:
    # Cria um objeto BeautifulSoup para analisar o HTML da página
    soup = BeautifulSoup(page_html, "html.parser")

    # Encontra todos os contêineres de produtos na página
    product_containers = soup.find_all("div", {"class": "s-result-item"})

    # Inicializa uma lista para armazenar as informações dos produtos
    products = []

    # Itera sobre cada contêiner de produto na página
    for product_container in product_containers:
        # Extrai informações específicas do contêiner do produto
        product_name = product_container.find("span", {"class": "a-text-normal"}).text.strip()
        product_price = product_container.find("span", {"class": "a-offscreen"}).text.strip()
        product_total_reviews = product_container.find("span", {"class": "a-size-base s-underline-
text"}).text.strip()
        product_total_stars = product_container.find("span", {"class": "a-icon-alt"}).text.strip()
        product_extra_info = product_container.find("span", {"class": "a-row a-size-base a-color-
secondary"}).text.strip()

        # Cria um dicionário com as informações do produto
        product_info = {
            'name': product_name,
            'price': product_price,
            'total_reviews': product_total_reviews,
            'total_stars': product_total_stars,
            'extra_info': product_extra_info
        }

        # Adiciona o dicionário à lista de produtos
        products.append(product_info)

    # Retorna a lista de produtos
    return products
```

Fonte: Autoria própria.

Figura 16 - Funções adicionais.

```

def get_url(search_input: str) -> str:
    url = 'https://www.amazon.com/s?k=' + search_input
    return url

def get_page_html(page_url: str) -> str:
    resp = requests.get(page_url, headers=headers)
    return resp.text

def get_reviews_from_html(page_html: str) -> BeautifulSoup:
    soup = BeautifulSoup(page_html, "lxml")
    reviews = soup.find_all("div", {"class": "a-section celwidget"})
    return reviews

def get_review_date(soup_object: BeautifulSoup):
    date_string = soup_object.find("span", {"class": "review-date"}).get_text()
    return date_string

def get_review_text(soup_object: BeautifulSoup) -> str:
    review_text = soup_object.find(
        "span", {"class": "a-size-base review-text review-text-content"}
    ).get_text()
    return review_text.strip()

def get_review_header(soup_object: BeautifulSoup) -> str:
    review_header = soup_object.find(
        "a",
        {
            "class": "a-size-base a-link-normal review-title a-color-base review-title-content a-text-
            bold"
        },
    ).get_text()
    return review_header.strip()

def get_number_stars(soup_object: BeautifulSoup) -> str:
    stars = soup_object.find("span", {"class": "a-icon-alt"}).get_text()
    return stars.strip()

def get_product_name(soup_object: BeautifulSoup) -> str:
    product = soup_object.find(
        "a", {"class": "a-size-mini a-link-normal a-color-secondary"}
    ).get_text()
    return product.strip()

def orchestrate_data_gathering(single_review: BeautifulSoup) -> dict:
    return {
        "review_text": get_review_text(single_review),
        "review_date": get_review_date(single_review),
        "review_title": get_review_header(single_review),
        "review_stars": get_number_stars(single_review),
        "review_flavor": get_product_name(single_review),
    }

```

Fonte: Autoria própria.

6.2.2 Algoritmo Floresta Aleatória

Figura 17 - Rota de pegar informações do produto.

```
@app.route('/product-insight', methods=['POST'])
def getProductInsights():

    # Obtém os dados da requisição
    data = request.get_json()

    # Carrega os dados históricos para o produto específico
    product_data = data['product_data']

    # Assume que 'Vendas' é a variável alvo
    target_variable = 'Sales'

    # Obtém produtos similares na mesma categoria
    similar_products_category = getSimilarProductsCategory(product_data['asin'],
    product_data['category'])

    # Combina os dados dos produtos similares com os dados do produto específico
    final_data = similar_products_category + [product_data]

    # Assume que 'Preço', 'Avaliações', 'Estrelas' e outras características relevantes estão em X
    X = final_data[['asin', 'sales_volume', 'bestseller_rank', 'price', 'reviews', 'stars',
    'extra_info', 'weekly_sales', 'monthly_sales']]

    # Assume que 'Vendas' é a variável alvo
    y = final_data[target_variable]

    # Cria e treina o modelo de Random Forest
    rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
    rf_model.fit(X, y)

    # Assume que novos_dados contém as características para a nova instância do produto
    results = getResults(rf_model)

    # Faz uma previsão para as vendas do novo produto
    predicted_sales = rf_model.predict(results)

    return jsonify({'results': predicted_sales})
```

Fonte: Autoria própria.

7 CONSIDERAÇÕES FINAIS

Quando foi iniciada a pesquisa, contestou-se o crescente desafio enfrentado pelas empresas na era da informação, marcada pelo exponencial aumento de dados. A justificativa para este estudo reside na necessidade premente de estratégias inovadoras para lidar com essa abundância de informações, especialmente no contexto empresarial. Ao optar por explorar a Ciência de Dados como uma aliada fundamental para empreendedores diante desse cenário, propôs-se uma solução prática e relevante para a gestão eficiente de dados comerciais, resgatando a importância intrínseca desse tema.

Ao atingir o objetivo geral estabelecido no início desta pesquisa, confirmamos que a plataforma digital desenvolvida corresponde às demandas identificadas, proporcionando uma solução para a análise de produtos. A integração bem-sucedida do *Web Scraping* no site Amazon e o algoritmo Floresta Aleatória para estimar vendas apoiam a abordagem escolhida, consolidando a Ciência de Dados como um recurso valioso para empreendedores na seleção e classificação de produtos.

O problema identificado no escopo deste estudo, relacionado à dificuldade na análise manual de grandes conjuntos de dados de produtos, foi enfrentado de frente. A utilização do algoritmo Floresta Aleatória emerge como uma solução, mitigando o desafio e proporcionando aos empreendedores uma ferramenta para a tomada de decisões. A contribuição prática dessa solução para a gestão de informações na era digital é inegável, oferecendo não apenas eficiência, mas também uma base sólida para a inovação e competitividade.

7.1 Dificuldades encontradas

Ocorreu dificuldades no decorrer deste trabalho. Uma delas foi, a falta de profundidade nos dados disponibilizados pela Amazon. A plataforma, por questões de privacidade e políticas de acesso, não oferece informações adicionais de vendas e apenas uma camada superficial. Isso implica que, embora tenha sido desenvolvida uma solução eficaz para a classificação de produtos, a ausência de detalhes mais profundos limita a amplitude das análises e insights que poderiam ser obtidos.

Um outro fator que ocasionou dificuldades foi na busca pelo algoritmo de classificação mais adequado para os dados obtidos neste projeto. A também falta de

profundidade dos dados, e diversidade dessas informações, tornaram desafiadora a identificação de uma abordagem que fosse capacitada na análise e categorização dos produtos provenientes da Amazon.

7.2 Trabalhos Futuros

Como sugestão de trabalhos futuros, propõe-se:

- Utilizar técnicas de aprendizado de máquina para prever a demanda de produtos em cadeias de suprimentos;
- Análise preditiva e *big data* para prever tendências de mercado;
- Implementar modelos de detecção de fraudes com base em aprendizado de máquina para fortalecer a segurança em transações on-line;

REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, A. P. de. **O que é Firebase?** 2020. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-firebase>>. Acesso em: 09 de set de 2023.

ANGULAR. **Introduction to the Angular Docs.** 2020. Disponível em <<https://angular.io/docs>>. Acesso em: 20 de set. de 2023.

BRYNJOLFSSON, Erik. **Os três pilares do futuro digital, segundo pesquisadores do MIT.** Disponível em 2017 <<https://cetax.com.br/data-science-ou-ciencia-de-dados/>> Acesso em 13 nov. 2023.

COELHO, Lucas. **Ciência de dados: o que é, conceito e definição.** Disponível em 2017 <<https://cetax.com.br/data-science-ou-ciencia-de-dados/>> acesso em 10 nov. 2023.

COODESH. **O que é Firebase?** 2021. Disponível em: <<https://coodesh.com/blog/dicionario/o-que-e-firebase/>>. Acesso em: 09 de set de 2023.

Faceli, A. C. Lorena, J. Gama, A. C. P. d. L. Carvalho, et al. **Inteligência artificial: Uma abordagem de aprendizado de máquina.** 2011. Acesso em 13 nov. 2023.

FAWCETT, Tom; PROVOST, Foster. **Data Science para negócios: O que você precisa saber sobre mineração de dados e pensamento analítico de dados.** Rio de Janeiro: Alta Books, 2016.

FEMI. **Understanding servers.** 2018. Disponível em: <<https://medium.com/@theoluwafemi/understanding-servers-cbc61f910b9b>>. Acesso em: 05 out 2023.

FIREBASE, D. **Documentação.** 2023. Disponível em: <<https://firebase.google.com/docs>>. Acesso em: 09 de set de 2023.

KNAFLIC, Cole N. **Storytelling com dados: Um guia sobre visualização de dados para profissionais de negócios.** Rio de Janeiro: Alta Books, 2019.

MENDES, Caio. **Arquitetura Cliente-Servidor**. 2021. Disponível em: <<https://medium.com/@caiomay.mendes/arquitetura-cliente-servidor-b2adeeb3632e>>. Acesso em: 03 maio 2023.

MICROSOFT. **Visual Studio Code Documentation**. 2022. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 26 set. 2023.

PYTHON. **Python Programming Language**. Disponível em <http://www.python.org/>. Acesso em 2 out. 2023.

Russell and P. Norvig. **Artificial Intelligence: A Modern Approach**. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009. ISBN 0136042597, 9780136042594.

SAURA, Jose R. **Using Data Sciences in Digital Marketing: Framework, methods, and performance metrics**. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2444569X20300329#bib013>>. Acesso em 21 mar. 2023.

SESHADRI, Shyam; GREEN, Brad. **Desenvolvendo com AngularJS**. 1ª ed. São Paulo: Novatec Editora, 2014

SETZER, Waldemar W. **Dado, Informação, Conhecimento e Competência, Revista de Ciência da Informação**, n.0, dezembro, 1999, artigo 1. Disponível em: <<https://www.ime.usp.br/~vwsetzer/datagrama.html>> Acesso em: 26 ago 2022.

STACK OVERFLOW. **2022 Developer Survey**. 2022. Disponível em: <<https://survey.stackoverflow.co/2022/>>. Acesso em: 26 set. 2023.

VANDERPLAS, Jake. **Python Data Science Handbook**, 1. ed. Sebastopol – Estados Unidos: Editora O'Reilly Media, 2016.

VBOSCHETTI, Alberto; MASSARON, Luca. **Python Data Science Essentials**, 2. ed. Birmingham – UK: Editora Packt, 2016.

Velotio Technologies. **Web Scraping: Introduction, Best Practices & Caveats**. 2018. Disponível em: <<https://medium.com/velotio-perspectives/web-scraping-introduction-best-practices-caveats-9cbf4acc8d0f>>. Acesso em: 06 out 2023.

WALLER, Matthew A. **Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management.**

Disponível em:<<https://onlinelibrary.wiley.com/doi/full/10.1111/jbl.12010>>. Acesso em 21 mar. 2023.