



**UNIVERSIDADE ESTADUAL PAULISTA**  
**“JÚLIO DE MESQUITA FILHO”**  
**Instituto de Ciência e Tecnologia**  
Câmpus de Sorocaba

**GABRIEL CHICOLI NUNES ROSA**

**SISTEMA DE ANÁLISE DE DADOS E FATURAMENTO**

Sorocaba

2024

GABRIEL CHICOLI NUNES ROSA

**SISTEMA DE ANÁLISE DE DADOS E FATURAMENTO**

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientadora: Prof<sup>ª</sup> Dr<sup>ª</sup> Luiza Amalia Pinto Cantão

Sorocaba

2024

R788s

Rosa, Gabriel Chicoli Nunes

Sistema de análise de dados e faturamento / Gabriel Chicoli Nunes

Rosa. -- Sorocaba, 2024

47 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba

Orientadora: Luiza Amalia Pinto Cantão

1. Análise de séries temporais - Processamento de dados. 2. Python (Linguagem de programação de computador). 3. Faturas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



UNIVERSIDADE ESTADUAL PAULISTA  
"JÚLIO DE MESQUITA FILHO"  
Instituto de Ciência e Tecnologia  
Câmpus de Sorocaba

## SISTEMA DE ANÁLISE DE DADOS E FATURAMENTO

GABRIEL CHICOLI NUNES ROSA

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO  
COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE  
**BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

Prof. Dr. Everson Martins  
Coordenador

### BANCA EXAMINADORA:

Prof<sup>ª</sup>. Dr<sup>ª</sup>. LUIZA AMÁLIA PINTO CANTÃO

UNESP – Campus de Sorocaba

Prof. Dr. IVANDO SEVERINO DINIZ

UNESP – Campus de Sorocaba

Prof. Dr. EVERSON MARTINS

UNESP – Campus de Sorocaba

14 de junho de 2024

## **AGRADECIMENTOS**

Agradeço à professora Luiza Amalia Pinto Cantão por todo o apoio durante a confecção deste trabalho.

Agradeço ao professor Renato Fernandes Cantão que esteve junto à professora Luiza, dando conselhos valiosos com todo seu conhecimento no aprendizado de máquinas.

Agradeço à minha família que me acompanhou durante toda a minha jornada de graduação.

Agradeço também meus amigos de república que me acompanharam durante vários momentos, tanto nos bons quanto nos ruins durante toda a minha vida acadêmica e me tornaram uma pessoa muito melhor durante todo esse tempo.

## RESUMO

Os sistemas de faturamento atuais costumam ser complexos e dependem de programadores especializados para implementar novas funcionalidades, além de serem muito caros. Este trabalho se propõe a abordar este problema, propondo um sistema que utiliza técnicas de programação em VBA e análise estatística com Python, especificamente utilizando modelos de redes neurais como o Long Short-Term Memory (LSTM). A finalidade do sistema é aprimorar o gerenciamento e a precisão no processo de faturamento em empresas que utilizam APIs para operações financeiras. Através de uma interface intuitiva desenvolvida em VBA, o sistema facilita o cadastro e a atualização de dados de clientes, enquanto a integração com a API gratuita de consulta ao CNPJ da receita federal assegura a precisão dessas informações. Paralelamente, a análise de dados realizada com Python permite a detecção de outliers e a previsão de comportamentos atípicos, contribuindo significativamente para a tomada de decisões estratégicas e prevenção de fraudes. Os resultados demonstram que o sistema não apenas atende às necessidades operacionais e estratégicas das empresas, mas também oferece robustez e confiabilidade, com potencial para ser adaptado e escalado para diferentes segmentos de mercado.

**Palavras-chave:** análise de dados; LSTM; Python; sistema de faturamento; VBA.

## ABSTRACT

Current billing systems tend to be complex and rely on specialized programmers to implement new features. This thesis proposes a solution to these challenges by developing a system that utilizes VBA programming techniques and statistical analysis with Python, specifically leveraging neural network models like Long Short-Term Memory (LSTM). The purpose of the system is to enhance management and accuracy in the billing processes of companies using APIs for financial operations. An intuitive VBA-developed interface facilitates the registration and updating of customer data, while integration with the free CNPJ query API from the federal revenue ensures the accuracy of this information. Concurrently, data analysis performed with Python enables the detection of outliers and the prediction of atypical behaviors, significantly contributing to strategic decision-making and fraud prevention. The results demonstrate that the system not only meets the operational and strategic needs of companies but also offers robustness and reliability, with the potential to be adapted and scaled for different market segments.

**Keywords:** billing system; data analysis; LSTM; Python; VBA.

## LISTA DE FIGURAS

|                                                                                                 |    |
|-------------------------------------------------------------------------------------------------|----|
| Figura 1 – Interface de cadastro de clientes .....                                              | 19 |
| Figura 2 – Planilha de cadastro de clientes, no formato de tabela.....                          | 19 |
| Figura 3 – Cadastro de itens .....                                                              | 20 |
| Figura 4 – Modalidades de precificação.....                                                     | 20 |
| Figura 5 – Cadastro de itens na modalidade “Escalonamento”.....                                 | 21 |
| Figura 6 – Cadastro do cliente no formulário.....                                               | 22 |
| Figura 7 – Contratos cadastrados .....                                                          | 22 |
| Figura 8 – Cadastro resgatado no <i>userforms</i> . ....                                        | 23 |
| Figura 9 – Fluxo da detecção de outliers. ....                                                  | 26 |
| Figura 10 – Fluxo de treino e teste para a validação temporal. ....                             | 30 |
| Figura 11 – Exemplo de cliente já cadastrado .....                                              | 35 |
| Figura 12 – Cadastro de item no formato de escalonamento. ....                                  | 36 |
| Figura 13 – Demonstrativo de exemplo. ....                                                      | 38 |
| Figura 14 – Demonstrativo em formato PDF .....                                                  | 39 |
| Figura 15 – Tela para identificação de outliers.....                                            | 40 |
| Figura 16 – Erro por iteração para o “ <i>IssuerId</i> ” 4 .....                                | 42 |
| Figura 17 – Volume predito x volume real para os primeiros itens do “ <i>IssuerId</i> ” 4 ..... | 43 |

## **LISTA DE TABELAS**

|                                                                 |    |
|-----------------------------------------------------------------|----|
| Tabela 1 – Formato da fonte de dados para o processamento ..... | 25 |
|-----------------------------------------------------------------|----|

## SUMÁRIO

|            |                                                                               |           |
|------------|-------------------------------------------------------------------------------|-----------|
| <b>1</b>   | <b>INTRODUÇÃO .....</b>                                                       | <b>11</b> |
| <b>2</b>   | <b>DESENVOLVIMENTO .....</b>                                                  | <b>13</b> |
| <b>2.1</b> | <b>Revisão da Literatura .....</b>                                            | <b>13</b> |
| 2.1.1      | VBA e o Potencial no Faturamento Corporativo.....                             | 13        |
| 2.1.2      | Python e a Análise de Dados no Faturamento .....                              | 16        |
| <b>2.2</b> | <b>Proposição .....</b>                                                       | <b>18</b> |
| 2.2.1      | Interface de cadastro de clientes, preços e modalidades de precificação ..... | 18        |
| 2.2.1.1    | <i>Cadastro de clientes .....</i>                                             | <i>19</i> |
| 2.2.1.2    | <i>Cadastro de itens.....</i>                                                 | <i>20</i> |
| 2.2.2      | Análise Estatística e Uso de LSTM (Long Short-Term Memory) .....              | 23        |
| <b>3</b>   | <b>METODOLOGIA .....</b>                                                      | <b>25</b> |
| <b>3.1</b> | <b>Análise e Modelagem Preditiva.....</b>                                     | <b>26</b> |
| <b>3.2</b> | <b>Avaliação de Desempenho.....</b>                                           | <b>27</b> |
| 3.2.1      | Erro Quadrático Médio (MSE).....                                              | 28        |
| 3.2.2      | Erro Absoluto Médio (MAE) .....                                               | 28        |
| 3.2.3      | Validação de séries temporais.....                                            | 29        |
| <b>3.3</b> | <b>Fórmulas e suas personalizações no Microsoft Excel®.....</b>               | <b>31</b> |
| <b>3.4</b> | <b>A utilização de userforms no VBA para cadastro de clientes.....</b>        | <b>32</b> |
| <b>3.5</b> | <b>O Power Query® no carregamento de dados para o Excel®.....</b>             | <b>33</b> |
| <b>4</b>   | <b>RESULTADOS E DISCUSSÃO.....</b>                                            | <b>34</b> |
| <b>4.1</b> | <b>Userforms para cadastro de clientes.....</b>                               | <b>35</b> |
| <b>4.2</b> | <b>Geração de demonstrativos em PDF .....</b>                                 | <b>38</b> |
| <b>4.3</b> | <b>Detecção de outliers na série temporal.....</b>                            | <b>39</b> |
| <b>5</b>   | <b>CONCLUSÃO .....</b>                                                        | <b>45</b> |
|            | <b>REFERÊNCIAS .....</b>                                                      | <b>46</b> |

## 1 INTRODUÇÃO

A síntese de tecnologia e finanças, conhecida como fintech, está resultando em mudanças significativas nos serviços bancários e apresentando novos desafios. As fintechs têm um enorme potencial para crescer e impactar a indústria bancária, combinando tecnologias emergentes como blockchain, IA e aprendizado de máquina (Varma *et al.*, 2022). No cenário atual do mercado financeiro existe uma exigência de ferramentas cada vez mais sofisticadas para o gerenciamento e análise de dados. Neste contexto, o presente trabalho propõe o desenvolvimento de um sistema inovador, focado na análise e tratamento de dados relacionados aos preços e volumetrias de consumo de APIs (*application programming interface*) voltadas para a bancarização de empresas de diversos ramos, permitindo que sejam realizadas transferências, PIX, emissão de boletos, entre outras atividades relacionadas a bancos dentro do aplicativo dessa empresa.

Através de uma interface intuitiva e versátil, o sistema facilitará o cadastro de preços em diferentes metodologias de precificação, refletindo a diversidade e complexidade do mercado. A integração entre as interfaces de cadastro e visualização dos dados promete não apenas uma experiência de usuário fluida, mas também uma análise mais eficiente e correlações precisas entre itens e preços.

A realidade de sistemas de faturamento complexos, que têm a versatilidade de suportar diferentes tipos de precificação é que, apesar de serem mais completos que a solução apresentada por este trabalho, podem ser muito lentos e dependem de empresas de consultoria que cobram caro pelo serviço. Ao utilizar o VBA (*Visual Basic for Applications*), a expectativa é de que a manutenção do sistema possa ser realizada por profissionais que possuam conhecimentos básicos na linguagem, já que existe uma extensa documentação disponível na internet.

O principal desafio nesta etapa foi a criação de um formulário para preenchimento dos dados de precificação de cada cliente, que foi realizado utilizando um *userforms* do VBA, que possibilita a abertura de caixas de diálogo para interação do usuário com o programa, tornando o preenchimento mais intuitivo, simplificando a entrada de dados e interação com planilhas, como ressalta Best Excel Tutorial (Creating [...], 2023).

Além disso, o formulário traz integrada uma consulta a uma API da Receita Federal<sup>1</sup> que traz os dados do cliente conforme o CNPJ digitado, o que facilita a inscrição de novos clientes no formato caso necessário.

---

<sup>1</sup> Disponível em: <https://receitaws.com.br/>. Acesso em: 26 maio 2024.

Já a impressão do demonstrativo para envio aos clientes é realizada no formato “.pdf” e é realizada pela própria interface do demonstrativo.

Por fim, o sistema conta com uma etapa de análise dos dados utilizando uma ferramenta de inteligência artificial desenvolvida em linguagem *python* que tem como objetivo a identificação de valores que eventualmente estejam fora da média esperada para um determinado item ou cliente, criando uma janela de oportunidades para a análise de tais dados.

A análise de dados é realizada utilizando uma ferramenta estatística, programada em linguagem *Python* que utiliza a metodologia do *Long Short-Term Memory* (LSTM). Esta escolha permite incorporar aprendizado de máquina na detecção de anomalias nos dados, especialmente útil para entender sequências temporais e padrões de venda, além de possíveis erros que possam acontecer na etapa de processamento dos dados (Filho, 2023). É comum que bases de dados enfrentem problemas ao processar transações corretamente devido a instabilidades no sistema, erros humanos no preenchimento de bases de dados, entre outros fatores. A capacidade do LSTM de modelar dependências de longo prazo nos dados é crucial para a tomada de decisões estratégicas e para garantir a integridade dos dados entregues aos clientes, já que ele é utilizado para classificações e previsões de séries temporais com intervalos de tempo indeterminados (Hochreiter; Schmidhuber, 1997).

Além disso, existem diversas aplicações da análise de dados de faturamento que podem ser utilizadas em áreas como vendas, relacionamento de clientes, desenvolvimento de produtos, gestão financeira, entre outras que não serão o foco deste trabalho, porém merecem uma menção.

## 2 DESENVOLVIMENTO

### 2.1 Revisão da Literatura

Em um ambiente de negócios cada vez mais conduzido por dados, a capacidade de extrair *insights* acionáveis através da análise estatística é vital.

*A inferência estatística é uma das partes da Estatística. Esta é a parte da metodologia da ciência que tem por objetivo a coleta, redução, análise e modelagem dos dados, a partir do que, finalmente, faz-se a inferência para uma população da qual os dados (a amostra) foram obtidos (Bussab; Morettin, 2010, p.18).*

Esta observação sublinha a importância de se ter sistemas robustos para gerenciar a coleta e processamento de dados. No contexto de empresas que vendem produtos de software, como APIs voltadas para aplicativos de carteira digital, um sistema de faturamento integrado torna-se não apenas uma ferramenta de gestão financeira, mas também uma fonte crucial de dados para análises e tomadas de decisão. A capacidade de gerenciar eficientemente as transações e, ao mesmo tempo, coletar informações relevantes é uma combinação poderosa que pode direcionar estratégias de negócios e identificar oportunidades de crescimento.

Além disso, é importante analisar também a importância de um sistema que evita falhas no contexto de uma empresa que vende softwares. Como cita Oliveira (2022), o valor percebido pelo consumidor é tomado como parâmetro balizador do preço a ser cobrado pelo produto/serviço. Isso significa que se um produto carrega consigo assertividade, ele também tem um valor agregado mais alto, o que é de extrema importância para um software de tamanha complexidade.

#### 2.1.1 VBA e o Potencial no Faturamento Corporativo

O VBA (*Visual Basic for Applications*) é uma linguagem de programação incorporada em muitos softwares da Microsoft, sendo amplamente utilizado em ambientes corporativos, principalmente devido à sua integração direta com aplicativos como o Microsoft Excel®. Essa característica tem tornado o VBA uma ferramenta poderosa para empresas que buscam soluções de automação e personalização rápidas para tarefas rotineiras. Como ressalta Korol (2003) “[...] *using Excel’s built in language, VBA [...] you can bring a lot of automating power to your spreadsheets, whether you are creating them for yourself or for others.*”, o que justifica o uso da linguagem para aplicações mais simples em planilhas.

Apesar de o VBA ser tradicionalmente considerado uma linguagem pouco adequada para sistemas robustos e abrangentes de faturamento, quando comparado a outras linguagens como

Python, ele possui certas vantagens. A vasta documentação disponível, juntamente com muitos profissionais familiarizados com a linguagem, facilita a implementação e manutenção de soluções baseadas em VBA. Além disso, a prevalência do VBA em ambientes corporativos, devido à sua integração com ferramentas de escritório, proporciona uma curva de aprendizado menor para os funcionários que já estão familiarizados com a interface do Microsoft Office®. Apesar disso, VBA é uma linguagem interpretada e não compilada, o que pode resultar em desempenho mais lento para tarefas que exigem processamento intensivo. A interface de desenvolvimento é considerada menos moderna e intuitiva em comparação com outras ferramentas de desenvolvimento de software mais recentes.

Dentro do escopo de um sistema de faturamento, o VBA pode ser utilizado para criar automações que facilitem a entrada de dados, geração de relatórios e até mesmo a integração com outras ferramentas ou sistemas. E aqui, os *userforms*, que são uma ferramenta de desenvolvimento de formulários do Excel desempenham um papel fundamental.

*Com o userforms você exibe informações e permite que o usuário insira informações. Os controles InputBox e MsgBox são maneiras simples de fazer isso. Você pode usar os controles de userform no editor do VB para criar formulários mais complexos. (Jelen; Syrstad, 2008, p.126)*

No trecho citado de Jelen e Syrstad (2008, p.126), a sigla "VB" refere-se ao *Visual Basic*, que é uma linguagem de programação orientada a eventos e um ambiente de desenvolvimento integrado da Microsoft. No contexto de *userforms*, VB é normalmente associado ao *Visual Basic for Applications* (VBA), uma versão do *Visual Basic* incorporada em programas da Microsoft como Excel, Word e Access.

Esses formulários podem ser personalizados para se adaptar especificamente às necessidades do faturamento, garantindo uma interface intuitiva e eficaz para os cadastros de informações que serão necessários na construção do software para os usuários finais.

Em suma, embora o VBA possa não ser a primeira escolha para um sistema de faturamento do zero, sua adaptabilidade, familiaridade e capacidade de personalização o tornam uma escolha prática, especialmente quando considerando ambientes corporativos onde o Microsoft Office® é uma ferramenta padrão.

É importante ressaltar também a necessidade da automatização de processos em empresas financeiras atualmente, que se tornam cada vez mais eficientes em processos repetitivos através de métodos de automação personalizados como o deste trabalho.

*The ability for professional services firms to manage and recover office-related expenses, as well as to provide accurate, efficient, and automated expense management tracking, has become increasingly crucial as firms strive to maintain profitability while legitimately recouping overhead costs. (Davis, 2003, p.13)*

### 2.1.2 Python e a Análise de Dados no Faturamento

O Python emergiu como uma das linguagens de programação mais versáteis e acessíveis para análise de dados, graças ao seu ecossistema extensivo e bem estabelecido de bibliotecas e ferramentas.

*Python's improved open-source libraries (such as pandas and scikit-learn) have made it a popular choice for data analysis tasks. Combined with Python's overall strength for general-purpose software engineering, it is an excellent option as a primary language for building data applications. (McKinney, 2017)*

A biblioteca pandas<sup>2</sup>, por exemplo, permite a manipulação e análise de dados de forma eficiente com suas estruturas de dados, *dataframes* e séries. Dentro deste contexto, é importante entender os conceitos de *dataframes* e "series", que são estruturas de dados fundamentais em linguagens de programação voltadas para análise de dados, como Python com a biblioteca Pandas. Um *dataframe* é uma estrutura de dados bidimensional, muito semelhante a uma tabela de dados, com linhas e colunas. Cada coluna pode conter dados de diferentes tipos (numéricos, *string*, booleanos, etc.), e é semelhante a uma planilha do Excel ou uma tabela de banco de dados. Já uma "serie" é uma estrutura de dados unidimensional e pode ser vista como uma única coluna de um *dataframe*. Cada series possui um índice, que é uma sequência de etiquetas, similar a uma lista (Machine Learning Plus, 2021).

Isso facilita a análise dos dados oriundos de sistemas de faturamento. Já a biblioteca *scikit-learn*<sup>3</sup> oferece uma ampla gama de ferramentas para aprendizado de máquina, permitindo desde regressões simples até modelos de predição mais complexos. Essas ferramentas podem ser úteis para analisar padrões de compra, prever tendências de vendas e identificar segmentos de clientes, entre outros.

Para empresas que comercializam produtos de software, o uso de Python na análise de dados do faturamento pode oferecer *insights* valiosos sobre o comportamento dos clientes, eficácia de campanhas de marketing, e potenciais oportunidades ou ameaças no mercado. A capacidade do Python de integrar-se com outras ferramentas, sistemas e bases de dados torna a linguagem ainda mais valiosa para a criação de um sistema de faturamento integrado, onde a coleta, processamento e análise dos dados são processos contínuos e interligados.

---

<sup>2</sup> Disponível em: <https://pandas.pydata.org/>. Acesso em: 26 maio 2024.

<sup>3</sup> Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 26 maio 2024.

Além disso, vale ressaltar que para a análise correta dos dados, é necessário tratá-los e para isso, existem vários métodos. Como ressaltam Morettin e Singer (2019, p.34) em geral, a primeira tarefa de uma análise estatística de um conjunto de dados consiste em resumi-los. As técnicas disponíveis para essa finalidade dependem do tipo de variáveis envolvidas.

Para a escolha do modelo de aprendizado de máquina ideal para o presente trabalho, Hodge e Austin (2004, p.18) em sua análise de métodos para detecção de outliers apontam: "*There is no single universally applicable or generic outlier detection approach.*". Apesar disso, existem diversos modelos que podem ser utilizados no contexto, tais como o Long Short-Term Memory (LSTM) e o Multilayer Perceptron (MLP), entre outros.

A implementação do modelo Long Short-Term Memory (LSTM) na detecção de outliers em *datasets* de tamanho moderado é particularmente vantajosa devido à sua habilidade em processar e aprender de sequências de dados com dependências de longo prazo. A capacidade do LSTM de capturar nuances temporais complexas e adaptar-se a padrões dinâmicos nos dados faz dele uma ferramenta poderosa em cenários onde métodos tradicionais menos flexíveis falham em identificar anomalias eficazmente.

Além disso, a implementação de LSTM em plataformas de programação como *Python* é relativamente direta, graças a bibliotecas de aprendizado de máquina como *TensorFlow*<sup>4</sup> e *PyTorch*<sup>5</sup>, que fornecem abstrações e ferramentas poderosas para construir e treinar redes neurais recorrentes com menos esforço de codificação. Estas bibliotecas facilitam o gerenciamento das complexidades associadas às dependências de longo prazo dos dados, essenciais para o sucesso do LSTM, permitindo que desenvolvedores e pesquisadores implementem soluções eficientes para problemas que requerem análise de sequências temporais.

No contexto de programação em Python para a implementação de redes neurais recorrentes como a LSTM as bibliotecas *TensorFlow* e *PyTorch* emergem como ferramentas cruciais. *TensorFlow*, desenvolvida pelo Google, é amplamente reconhecida por sua robustez e capacidade de lidar com operações em larga escala, sendo uma escolha popular em ambientes de produção. Por outro lado, *PyTorch*, criado pelo *Facebook's AI Research lab*, destaca-se pela sua interface intuitiva e flexibilidade, tornando-se uma favorita na comunidade de pesquisa para o desenvolvimento rápido e a experimentação em IA. Ambas as bibliotecas oferecem recursos

---

<sup>4</sup> Disponível em: <https://pypi.org/project/tensorflow/>. Acesso em: 26 maio 2024.

<sup>5</sup> Disponível em: <https://pytorch.org/>. Acesso em: 26 maio 2024.

avançados para a construção e o treinamento de redes neurais recorrentes, facilitando a implementação de modelos complexos e reduzindo a barreira técnica para a experimentação e o desenvolvimento no campo do aprendizado de máquina.

Essa facilidade de implementação torna o LSTM uma opção viável para pesquisadores e praticantes que buscam uma solução eficiente e acessível.

O LSTM pode enfrentar desafios em *datasets* extremamente grandes devido ao custo computacional associado à gestão de dependências de longo prazo e nem sempre é ideal para *datasets* muito pequenos devido ao risco de *overfitting*, que é uma situação comum em aprendizado de máquina, onde um modelo se ajusta excessivamente aos dados de treinamento, adaptando-se não somente às suas características, mas também aos ruídos e irregularidades do conjunto. Apesar disso, dentre os tipos de modelagem existentes, sua capacidade de modelar sequências temporais complexas o torna especialmente valioso em muitas aplicações práticas, inclusive na aplicação em que é utilizado no contexto deste trabalho.

## **2.2 Proposição**

No contexto do desenvolvimento de um sistema de análise de dados e faturamento, a seção de proposição deste documento visa detalhar a motivação e a estrutura propostas para a implementação do sistema. Esta seção é dividida em várias subseções, cada uma focando em um aspecto específico do sistema.

### **2.2.1 Interface de cadastro de clientes, preços e modalidades de precificação**

Na concepção do sistema de faturamento, a facilidade de entrada de dados é crucial. Para isso, foi desenvolvida uma interface de usuário utilizando userforms no VBA, uma ferramenta robusta que permite a criação de formulários personalizados dentro do ambiente do Excel.

### 2.2.1.1 Cadastro de clientes

A interface mostrada na Figura 1 demonstra a abordagem que foi utilizada para o cadastro de clientes de maneira eficiente e intuitiva.

**Figura 1** – Interface de cadastro de clientes

Fonte: Autoria própria

Ao inserir o CNPJ de um cliente no campo designado, a interface interage com a API da Receita Federal, uma ferramenta gratuita que fornece uma rica base de dados sobre empresas cadastradas. A consulta automática a essa API permite que, ao digitar o CNPJ, o nome da empresa e outros dados relevantes sejam preenchidos automaticamente nos campos correspondentes, otimizando o processo de cadastro e garantindo a precisão das informações.

Os dados obtidos, incluindo o nome, endereço e outros detalhes relevantes do cliente, são diretamente inseridos em uma planilha do Excel. Essa planilha possui uma estrutura conforme ilustra a Figura 2, onde cada cliente recebe um ID único. Esse ID é gerado automaticamente pelo sistema no momento do cadastro e é essencial para a apuração e análise de dados da fatura, que serão fundamentais para a criação dos demonstrativos de faturamento.

**Figura 2** – Planilha de cadastro de clientes, no formato de tabela

| id_cliente | cliente                                                | logradouro            | numero | complemento               | municipio      | bairro       | uf         | cep                    | email | cnpj               |
|------------|--------------------------------------------------------|-----------------------|--------|---------------------------|----------------|--------------|------------|------------------------|-------|--------------------|
| 3          | UNIVERSIDADE ESTADUAL PAULISTA JULIO DE MESQUITA FILHO | AV PROFESSOR MAI S/N  |        |                           | BOTUCATU       | UNESP CAM SP | 18.618-687 | diretoria@fmb.unesp.br |       | 48.031.918/0019-53 |
| 1          | DOCK SOLUCOES EM MEIOS DE PAGAMENTO S A                | AVENIDA TAMBORE 267   |        | CONJ 261 A SALA 1 BARUERI | ALPHAVILLE SP  |              | 06.460-000 | fiscal@dock.tech       |       | 08.744.817/0001-86 |
| 2          | COCA COLA INDUSTRIAS LTDA                              | PRAIA DE BOTAFOGO 374 |        | ANDAR 12 PARTE B          | RIO DE JANEIRO | BOTAFOGO RJ  | 22.250-040 |                        |       | 45.997.418/0001-53 |

Fonte: Autoria própria

### 2.2.1.2 Cadastro de itens

A implementação do cadastro de itens é realizada por meio do preenchimento direto em uma planilha de Excel, como ilustra a Figura 3. Esta abordagem foi adotada devido à simplicidade dos dados requeridos: código do item, nome e métrica. A necessidade de um *userforms* foi descartada nesse caso, simplificando o processo de entrada de dados.

**Figura 3** – Cadastro de itens

| item_code | item_name            | Métrica                   |
|-----------|----------------------|---------------------------|
| 1         | PIX                  | R\$ / Transação           |
| 2         | PIX Consultas DICT   | R\$ / Consulta            |
| 3         | Gestão de tesouraria | R\$ / Mês                 |
| 4         | Contas Ativas        | R\$ / Conta Ativa         |
| 5         | Cartões Ativos       | R\$ / Cartão desbloqueado |

Fonte: Autoria própria

Além do cadastro básico, é essencial associar cada item a uma modalidade de faturamento específica, que pode variar conforme as negociações com cada cliente.

As modalidades disponíveis, indicadas na Figura 4, são “Padrão”, “Escalonamento” e “Mensalidade”. A seleção da modalidade de faturamento é feita pelo usuário com base no que foi negociado com o cliente, permitindo a flexibilidade necessária para acomodar diferentes estruturas de preços e acordos comerciais.

**Figura 4** – Modalidades de precificação

Cadastro de Clientes

CNPJ: 00.280.273/0001-37 Razão Social: SAMSUNG ELETRONICA DA

Item: PIX Item\_code: 1

Modalidade:   
 Padrão  
 Escalonamento  
 Mensalidade

id\_Cliente: 4

Preço: Adicionar Remover ↩

| Item | Preço | Modalidade | Volumetria De | Volumetria Até |
|------|-------|------------|---------------|----------------|
|      |       |            |               |                |
|      |       |            |               |                |
|      |       |            |               |                |
|      |       |            |               |                |
|      |       |            |               |                |
|      |       |            |               |                |

Cadastrar

Fonte: Autoria própria

A modalidade “Padrão” é mais simples e depende somente do cadastro de um preço que é multiplicado ao volume consumido. Já a “Mensalidade” ignora a volumetria do arquivo e considera um preço fixo mensal para todos os meses.

Já a modalidade “Escalonamento” permite que o usuário cadastre uma faixa de preços de acordo com a volumetria utilizada, fazendo com que o preço diminua ou aumente para diferentes faixas.

Para o cadastro de vários preços e volumetrias, foi necessária a utilização de outro formulário de preenchimento separado, que possui validações referentes ao cadastro desse tipo de preço, conforme demonstrado na Figura 5.

**Figura 5** – Cadastro de itens na modalidade “Escalonamento”

The image shows a software interface for item registration. A modal window titled 'Escalonamento' is open over a 'Cadastro de Clientes' window. The 'Escalonamento' modal has the following elements:

- Fields for 'Início da Faixa' (containing '150001') and 'Final da faixa' (empty).
- A 'Preço' field (empty).
- 'Adicionar' and 'Remover' buttons.
- A table with the following data:
 

| Volumetria De | Volumetria Até | Preço    |
|---------------|----------------|----------|
| 1             | 5000           | R\$ 0,90 |
| 5001          | 15000          | R\$ 0,80 |
| 15001         | 50000          | R\$ 0,70 |
| 50001         | 150000         | R\$ 0,60 |
- 'Cadastrar' buttons at the bottom right of the modal and the background window.

The background 'Cadastro de Clientes' window has fields for 'CNPJ', 'Item', 'Modalidade', and 'Preço', along with a list of 'Item' entries.

Fonte: Autoria própria

Assim, ao clicar no botão “Cadastrar” da janela de escalonamento, o sistema leva os dados já tratados para o formulário de cadastro de clientes, considerando o limite do sistema para o final da última faixa, conforme a Figura 6.

**Figura 6 – Cadastro do cliente no formulário**

| Item                 | Preço        | Modalidade    | Volumetria De | Volumetria Até |
|----------------------|--------------|---------------|---------------|----------------|
| PIX                  | R\$ 0,90     | Escalonamento | 1             | 5000           |
| PIX                  | R\$ 0,80     | Escalonamento | 5001          | 15000          |
| PIX                  | R\$ 0,70     | Escalonamento | 15001         | 50000          |
| PIX                  | R\$ 0,60     | Escalonamento | 50001         | 9999999999     |
| PIX Consultas DICT   | R\$ 0,50     | Padrão        | 1             | 9999999999     |
| Gestão de tesouraria | R\$ 5.000,00 | Mensalidade   | 1             | 1              |

Fonte: Autoria própria

Por fim, clicando no botão “Cadastrar” da janela de cadastro de clientes, as informações serão incluídas na planilha sequencialmente conforme o que foi registrado no formulário. A tabela com o cadastro dos contratos está disposta na Figura 7.

**Figura 7 – Contratos cadastrados**

| id_cliente | cliente                | item_code | item_name           | fee           | modalidade    | faixa de | faixa ate     |
|------------|------------------------|-----------|---------------------|---------------|---------------|----------|---------------|
| 3          | UNIVERSIDADE ESTADUAL  | 3         | Gestão de tesourari | R\$ 2,00      | Padrão        | 1        | 9.999.999.999 |
| 3          | UNIVERSIDADE ESTADUAL  | 5         | Cartões Ativos      | R\$ 5.000,00  | Mensalidade   | 1        | 1             |
| 1          | DOCK SOLUCOES EM MEIOS | 1         | PIX                 | R\$ 1,00      | Padrão        | 1        | 9.999.999.999 |
| 2          | COCA COLA INDUSTRIAS   | 1         | PIX                 | R\$ 1,00      | Escalonamento | 1        | 5.000         |
| 2          | COCA COLA INDUSTRIAS   | 1         | PIX                 | R\$ 0,90      | Escalonamento | 5.001    | 10.000        |
| 2          | COCA COLA INDUSTRIAS   | 1         | PIX                 | R\$ 0,80      | Escalonamento | 10.001   | 50.000        |
| 2          | COCA COLA INDUSTRIAS   | 1         | PIX                 | R\$ 0,70      | Escalonamento | 50.001   | 9.999.999.999 |
| 2          | COCA COLA INDUSTRIAS   | 2         | PIX Consultas DICT  | R\$ 0,50      | Escalonamento | 1        | 1.000         |
| 2          | COCA COLA INDUSTRIAS   | 2         | PIX Consultas DICT  | R\$ 0,25      | Escalonamento | 1.001    | 9.999.999.999 |
| 2          | COCA COLA INDUSTRIAS   | 3         | Gestão de tesourari | R\$ 10.000,00 | Mensalidade   | 1        | 1             |
| 2          | COCA COLA INDUSTRIAS   | 4         | Contas Ativas       | R\$ 5.000,00  | Mensalidade   | 1        | 1             |
| 4          | SAMSUNG ELETRONICA DA  | 1         | PIX                 | R\$ 0,90      | Escalonamento | 1        | 5.000         |
| 4          | SAMSUNG ELETRONICA DA  | 1         | PIX                 | R\$ 0,80      | Escalonamento | 5.001    | 15.000        |
| 4          | SAMSUNG ELETRONICA DA  | 1         | PIX                 | R\$ 0,70      | Escalonamento | 15.001   | 50.000        |
| 4          | SAMSUNG ELETRONICA DA  | 1         | PIX                 | R\$ 0,60      | Escalonamento | 50.001   | 9.999.999.999 |
| 4          | SAMSUNG ELETRONICA DA  | 2         | PIX Consultas DICT  | R\$ 0,50      | Padrão        | 1        | 9.999.999.999 |
| 4          | SAMSUNG ELETRONICA DA  | 3         | Gestão de tesourari | R\$ 5.000,00  | Mensalidade   | 1        | 1             |

Fonte: Autoria própria

Dessa forma, o cadastro de cada um dos clientes está pronto para ser utilizado para a confecção do demonstrativo de faturamento, sendo que, caso seja necessária qualquer alteração no contrato ou atualização cadastral de clientes, basta digitar novamente o CNPJ do cliente na

tela de cadastro e as informações serão novamente dispostas no formulário conforme a Figura 8.

**Figura 8** – Cadastro resgatado no *userforms*.

| Item                 | Preço      | Modalidade    | Volumetria De | Volumetria Até |
|----------------------|------------|---------------|---------------|----------------|
| PIX                  | R\$ 0,90   | Escalonamento | 1             | 5000           |
| PIX                  | R\$ 0,80   | Escalonamento | 5001          | 15000          |
| PIX                  | R\$ 0,70   | Escalonamento | 15001         | 50000          |
| PIX                  | R\$ 0,60   | Escalonamento | 50001         | 9999999999     |
| PIX Consultas DICT   | R\$ 0,50   | Padrão        | 1             | 9999999999     |
| Gestão de tesouraria | R\$ 5.0... | Mensalidade   | 1             | 1              |

Fonte: Autoria própria

Além disso, também foram implementados um botão de remover itens, caso seja necessário remover algum item por consequência de alguma eventual negociação ou erro de cadastro, além de um botão de desfazer que pode ser utilizado dentro da mesma iteração de cadastro para desfazer qualquer ação realizada no cadastro, seja ela de adição ou remoção de itens.

### 2.2.2 Análise Estatística e Uso de LSTM (*Long Short-Term Memory*)

O *Long Short-Term Memory* (LSTM), uma classe de redes neurais recorrentes, é um meio eficaz de detectar outliers nos dados de faturamento. A capacidade do LSTM de aprender e modelar dependências temporais complexas o torna uma ferramenta valiosa na identificação de padrões anômalos que podem indicar erros de entrada de dados ou variações significativas nas tendências de faturamento. O sistema implementa algoritmos de aprendizado de máquina que treinam o LSTM com dados históricos para reconhecer o que constitui um valor típico. Quando um novo dado é introduzido, o LSTM avalia sua conformidade com o modelo aprendido, sinalizando aqueles que são estatisticamente significativos como potenciais outliers. Esta detecção precoce permite intervenções rápidas, corrigindo erros antes que eles afetem os relatórios financeiros ou análises preditivas.

Além disso, ao analisar variações de grande escala, a organização pode discernir se são devidas a mudanças de mercado ou a circunstâncias atípicas que requerem atenção especializada. Para essa implementação, foi utilizada a biblioteca *TensorFlow* da linguagem *Python*, que permite a aplicação da metodologia de detecção de erros através de uma programação relativamente simples quando comparada a outros métodos de detecção de erros, mantendo a eficiência dos métodos aplicados através de bibliotecas completas e de código aberto que permitem a globalidade da aplicação de tais métodos.

Dessa forma, é gerado um arquivo com todos os dados relevantes de faturamento pelo próprio sistema *Excel*, que deve ser fornecido ao código através do carregamento de arquivos no formato '.csv' da biblioteca *pandas*. Os dados são lidos e são criadas métricas para identificação do item e cliente que eventualmente for previsto como um *outlier* dentro do *dataset* proposto.

O treinamento do modelo é feito com dados dos meses anteriores, o que garante uma maior performance do modelo por diversos motivos. Primeiramente, dados antigos são ricos em padrões e tendências que o modelo pode aprender a identificar e replicar. Eles fornecem uma linha de base de comportamento normal contra a qual novas informações podem ser comparadas. Isso é especialmente importante na detecção de outliers, pois o modelo deve discernir entre variações normais e anormalidades genuínas. Além disso, o uso de dados históricos ajuda a calibrar o modelo para a realidade operacional específica da empresa, levando em conta sazonalidades, ciclos econômicos e a evolução dos hábitos dos clientes ao longo do tempo. Essa familiaridade permite que o LSTM faça previsões mais precisas e forneça análises mais profundas e acionáveis.

### 3 METODOLOGIA

Este estudo adota uma abordagem híbrida, integrando ferramentas tradicionais de análise de dados e processamento de planilhas com técnicas avançadas de aprendizado de máquina. O fluxo de trabalho foi estruturado para explorar as capacidades do VBA e Excel no manuseio e preparação preliminar dos dados, além do armazenamento dos dados dos clientes e da confecção dos demonstrativos. Além disso, utilizou-se o Power Query para a transformação e limpeza dos dados, e o Python, especificamente as bibliotecas *TensorFlow* e *Pandas*, para a análise preditiva avançada usando redes neurais LSTM.

Os dados foram inicialmente coletados em formato CSV e armazenados localmente. Utilizando o Power Query®, uma ferramenta integrada ao Excel®, os dados foram inspecionados para identificar quaisquer inconsistências óbvias, como entradas vazias ou formatos de dados inconsistentes. Além disso, houve a remoção de duplicatas, o tratamento de valores ausentes e a agregação de dados quando necessário.

Como fonte dos dados, foi utilizada uma base de dados presente no site *Kaggle*<sup>6</sup>. A base original possui poucas descrições sobre as características dos dados, porém após análise, constatou-se que são dados referentes à venda de produtos de varejo entre 2018 e 2019 de uma loja que possui sede em diversos países do mundo. A base foi alterada de forma a condizer com o problema proposto neste documento. Assume-se que para uma aplicação real, os dados já estarão corretamente formatados e prontos para uso, de forma que a extração destes será realizada por um time separado daquele que apura o faturamento.

A fonte em questão apresenta o formato da Tabela 1 que mostra um exemplo de como os dados devem ser recebidos para que o tratamento possa ser realizado.

**Tabela 1** – Formato da fonte de dados para o processamento

| TransactionTime | ItemCode | IssuerId | Volume | process_date |
|-----------------|----------|----------|--------|--------------|
| 12/02/2018      | 7        | 4        | 3684   | 12/02/2023   |
| 12/02/2018      | 10       | 4        | 4887   | 12/02/2023   |
| 12/02/2018      | 2        | 4        | 2427   | 12/02/2023   |
| 12/02/2018      | 9        | 4        | 2787   | 12/02/2023   |
| 12/02/2018      | 5        | 4        | 4674   | 12/02/2023   |
| 12/02/2018      | 8        | 4        | 5721   | 12/02/2023   |

Fonte: Autoria própria

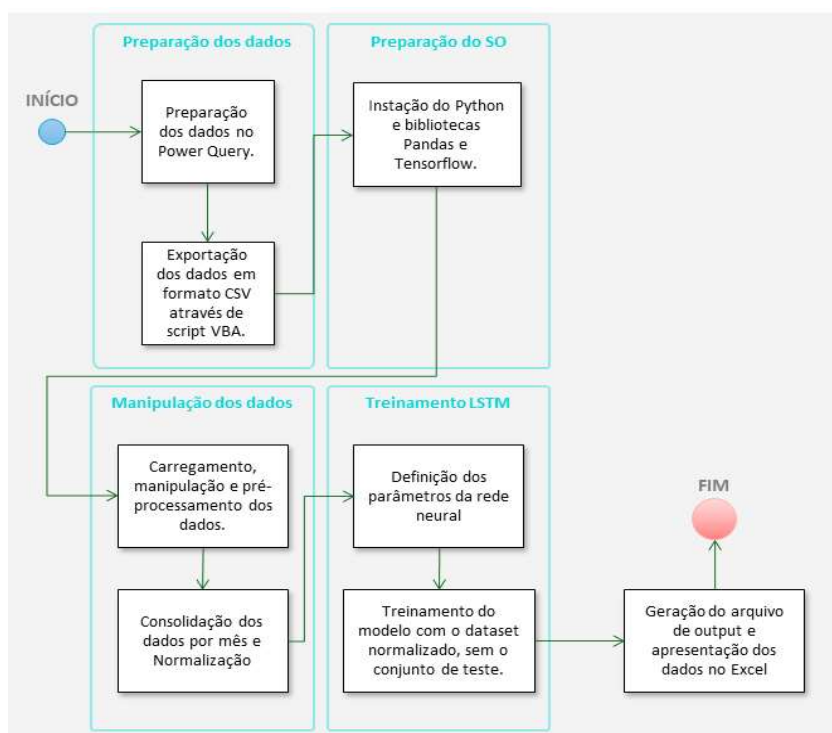
<sup>6</sup> Disponível em: <https://www.kaggle.com/datasets/vipin20/transaction-data>. Acesso em: 26 maio 2024.

### 3.1 Análise e Modelagem Preditiva

Após a preparação dos dados, eles foram exportados do Excel em um formato compatível para análise avançada. O ambiente de desenvolvimento *Python*, com a utilização das bibliotecas *Pandas* e *TensorFlow*, foi configurado para a implementação do modelo de rede neural *LSTM*.

A biblioteca *Pandas* foi utilizada para o carregamento, manipulação e pré-processamento dos dados. Isso incluiu a normalização adicional dos dados, a transformação das séries temporais em um formato adequado para o treinamento do modelo *LSTM*, e a divisão dos dados em conjuntos de treinamento e teste. A Figura 9 ilustra o fluxo realizado pelo programa desde o recebimento dos dados até a geração do arquivo contendo os outliers.

**Figura 9** – Fluxo da detecção de outliers.



Fonte: Autoria própria

A biblioteca *TensorFlow* foi empregada para construir o modelo de rede neural *LSTM*. O modelo foi configurado com parâmetros específicos, incluindo 2 camadas *LSTM*, que permitem ao modelo capturar dependências temporais mais complexas nos dados sequenciais (Greff *et al.*, 2017). A primeira camada *LSTM* extrai características básicas, enquanto a segunda camada refina essas características, melhorando a capacidade do modelo de aprender padrões mais sofisticados.

Além disso, foram utilizadas 50 unidades LSTM em cada camada. O número de unidades (neurônios) em cada camada LSTM determina a capacidade de armazenamento de memória da camada. 50 unidades é um ponto de partida comum que equilibra a capacidade do modelo de aprender padrões complexos e o custo computacional, já que, como afirma Greff *et al.* (2017), o tamanho da camada oculta no desempenho da rede LSTM é importante, destacando que redes maiores tendem a apresentar melhor desempenho, embora com retornos decrescentes. Além disso, observa-se que o tempo de treinamento requerido aumenta conforme o tamanho da rede, considerando tanto o tempo de computação necessário para cada época quanto a velocidade de convergência.

A função de ativação ReLU (*Rectified Linear Unit*), conforme descrito por Liang (2021), é amplamente utilizada em redes neurais profundas, calculando a saída de cada nó com base nas entradas individuais e nos pesos, sendo uma função não saturada e eficiente em termos de tempo de treinamento comparado a outras funções saturadas como a sigmoide. Além disso, a ReLU ajuda a resolver o problema de gradiente de desaparecimento e é simples na sua implementação, aumentando a flexibilidade e robustez das redes neurais.

A taxa de aprendizado utiliza o otimizador 'adam' que ajusta automaticamente durante o treinamento, tornando o processo mais eficiente e estável. Isso facilita a obtenção de bons resultados sem a necessidade de ajustes manuais frequentes.

A estrutura do modelo foi otimizada através de experimentação iterativa, com o objetivo de minimizar o erro de predição.

### 3.2 Avaliação de Desempenho

Para assegurar a precisão e a generalização do modelo de predição de volumes utilizando a arquitetura de redes neurais LSTM, adotou-se a implementação de métricas de erro padrão e a técnica de *time series split*. As métricas escolhidas foram o erro quadrático médio (MSE) e o erro absoluto médio (MAE), que fornecem *insights* sobre a precisão das predições do modelo em relação aos valores reais (scikit-learn, 2024), e são extensamente utilizadas na bibliografia. As informações desta seção foram baseadas na documentação oficial da biblioteca *scikit-learn*, que foi utilizada para os cálculos das métricas apresentadas.

A técnica de validação *time series split* foi empregada para avaliar a robustez do modelo, mitigando o risco de *overfitting* e garantindo que o desempenho avaliado seja representativo para diferentes conjuntos de dados.

### 3.2.1 Erro Quadrático Médio (MSE)

O RMSE (*Root Mean Squared Error*) é uma métrica que quantifica a raiz quadrada da média dos quadrados das diferenças entre os valores preditos pelo modelo e os valores reais observados. Especificamente, cada diferença entre a predição e o valor real é elevada ao quadrado. Isso assegura que erros maiores tenham um impacto desproporcionalmente maior no valor final do RMSE, tornando-o uma métrica valiosa quando é crucial evitar grandes desvios das predições.

O RMSE é calculado como:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

onde  $y_i$  representa o valor real e  $\hat{y}_i$  o valor predito. Um valor de RMSE mais baixo indica um modelo com predições mais precisas e menos erros.

A biblioteca scikit-learn do Python oferece uma implementação direta dessa métrica através da função `mean_squared_error`, que pode ser adaptada para calcular o RMSE aplicando-se a raiz quadrada ao resultado. Essa função é crucial para quantificar a precisão de modelos em tarefas de regressão, onde a precisão das predições e a penalização de erros significativos são prioritárias.

O RMSE é frequentemente preferido em situações em que é fundamental que o modelo minimize grandes erros, como em aplicações financeiras ou de planejamento, onde os custos dos erros de predição podem ser muito elevados. Além disso, um valor mais baixo de RMSE indica um modelo que está realizando predições mais próximas aos valores reais, refletindo um desempenho geral melhor e mais confiável. Portanto, essa métrica fornece um indicativo claro do erro médio, além de destacar a importância de evitar discrepâncias significativas nas predições, ajudando a ajustar modelos para alcançar maior precisão e confiabilidade.

### 3.2.2 Erro Absoluto Médio (MAE)

O MAE (*Mean Absolute Error*) é uma métrica utilizada para avaliar a performance de modelos de predição, proporcionando uma compreensão clara e direta da média das magnitudes absolutas das diferenças entre os valores preditos e os valores reais. A principal vantagem do MAE sobre métricas como o RMSE é que, ao não elevar ao quadrado as diferenças, ele atribui

uma penalidade linear para os erros. Isso significa que todas as diferenças, grandes ou pequenas, são tratadas de forma uniforme, tornando o MAE particularmente útil em situações em que os outliers não devem receber uma penalidade excessiva que poderia distorcer a avaliação geral do modelo.

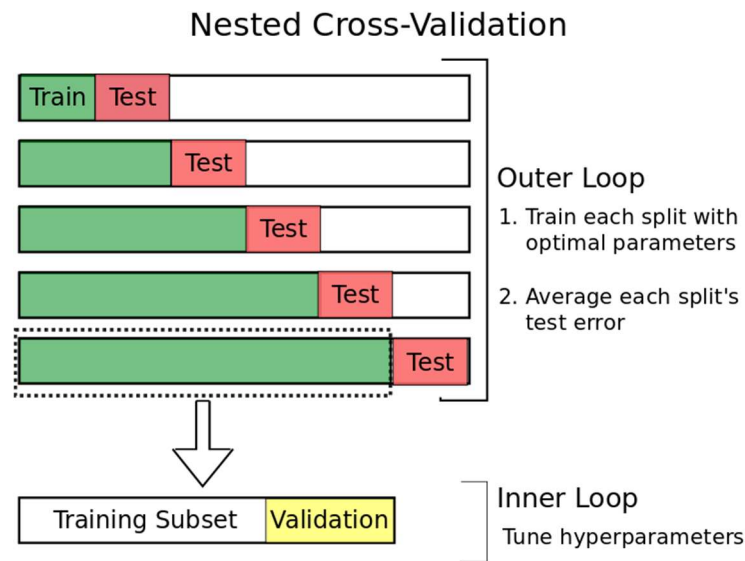
A biblioteca *scikit-learn* do *Python* implementa o MAE através da função *mean\_absolute\_error*. Esta função é amplamente utilizada para testar a precisão de modelos em problemas de regressão. A utilização do MAE na *scikit-learn* é bastante direta, exigindo apenas os valores reais e os valores preditos como entrada. O MAE é expresso pela fórmula:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

onde  $y_i$  é o valor real e  $\hat{y}_i$  o valor predito. A métrica MAE é frequentemente usada quando se deseja ter uma interpretação direta dos erros de predição. Por exemplo, em um contexto de predição de demanda de vendas, um MAE de 10 indica que, em média, as predições do modelo desviam-se 10 unidades (na mesma escala da variável de interesse) dos valores reais. Isso auxilia os analistas e gestores a entenderem a precisão das predições de forma clara, permitindo que tomem decisões com base na qualidade das predições fornecidas pelo modelo.

### 3.2.3 Validação de séries temporais

Na metodologia empregada no estudo, a validação de séries temporais foi realizada utilizando a técnica de *time series split* disponível na biblioteca *scikit-learn* do *Python*. Esta técnica é distinta da validação cruzada tradicional, pois não realiza uma divisão aleatória dos dados. Em vez disso, ela segmenta o conjunto de dados em N partes, onde cada partição subsequente de treinamento inclui os dados das partições anteriores, expandindo-se progressivamente. O conjunto de teste, por sua vez, é sempre uma partição que segue imediatamente o conjunto de treinamento mais recente. Isso garante que o treinamento e a avaliação dos modelos respeitem a sequência natural dos eventos, uma consideração crucial para dados temporais onde a proximidade temporal e a tendência são informativas. O funcionamento da validação pode ser encontrado na Figura 10.

**Figura 10** – Fluxo de treino e teste para a validação temporal.

Fonte: *Working [...]* (2020).

Para o contexto específico deste trabalho, a técnica foi adaptada para lidar com múltiplas séries temporais, cada uma correspondente a um "*IssuerId*" e um "*ItemCode*" específicos ao longo de um ano inteiro, conforme o contexto dos dados indicado na Tabela 1. Cada série temporal é tratada como uma unidade distinta dentro do modelo, refletindo variações específicas e tendências inerentes a cada combinação de identificador de emissão e código de item. Isso implica que, para cada par "*IssuerId*" e "*ItemCode*", um modelo separado é treinado e validado, proporcionando uma análise detalhada e ajustada às características específicas de cada série.

A estrutura de dados e a metodologia de validação são configuradas de tal modo que cada trecho temporal subsequente no qual o modelo é testado e serve como uma validação rigorosa do aprendizado do modelo nos dados anteriores. Isso é particularmente importante em cenários comerciais ou financeiros, onde as previsões precisam ser não apenas precisas, mas também consistentes ao longo do tempo.

Além disso, a abordagem adotada permite avaliar a performance dos modelos sob condições de deslocamento temporal, onde tendências e padrões podem evoluir. A análise dos resultados de cada *fold* de teste ajuda a identificar não apenas o desempenho médio do modelo ao longo do tempo, mas também quaisquer pontos de falha ou alterações significativas nas tendências dos dados. Essa informação é crítica para ajustar o modelo antes de sua aplicação em ambientes operacionais reais, garantindo que ele seja robusto e adaptável a mudanças nas condições subjacentes do mercado ou do ambiente.

Portanto, a aplicação meticulosa da validação de séries temporais em dados específicos de cada "IssuerId" e "ItemCode" por um período de um ano oferece uma avaliação abrangente e uma base sólida para confiar na capacidade preditiva e na adaptabilidade dos modelos desenvolvidos.

Essas metodologias são fundamentais para assegurar que o modelo de predição de volumes performa bem em média, e é confiável e eficaz em uma variedade de condições, minimizando o risco de decisões baseadas em previsões imprecisas.

### 3.3 Fórmulas e suas personalizações no Microsoft Excel®

O uso de fórmulas do Excel® desempenha um papel crítico na manipulação e análise de dados, permitindo cálculos automatizados e eficientes que são essenciais para a validação e interpretação dos resultados. O Excel®, como uma ferramenta versátil de planilha, oferece uma ampla gama de fórmulas padrão que facilitam desde operações matemáticas básicas até funções complexas de estatística e análise financeira.

As fórmulas padrão no Excel® são empregadas para executar cálculos automáticos que incluem somas (SUM), médias (AVERAGE), e operações condicionais (IF). Estas fórmulas são utilizadas para processar rapidamente grandes volumes de dados, permitindo que decisões baseadas em dados sejam tomadas de forma eficaz e eficiente. Por exemplo, a função XLOOKUP é frequentemente usada para buscar dados em uma tabela ou um intervalo por linha de forma automática.

Além disso, o Excel permite a criação de fórmulas personalizadas através do uso de funções definidas pelo usuário (UDFs - *User Defined Functions*) que são escritas em VBA (*Visual Basic for Applications*). Essas funções são especialmente úteis para realizar cálculos que vão além das capacidades das fórmulas pré-definidas do Excel®, permitindo aos usuários adaptarem as funções matemáticas e lógicas às necessidades específicas do seu trabalho.

No âmbito deste estudo, fórmulas personalizadas são utilizadas para abordar cenários específicos que não podem ser resolvidos eficientemente com as fórmulas padrão. Por exemplo, uma fórmula personalizada pode ser desenvolvida para calcular indicadores específicos que requerem a aplicação de regras de negócio complexas ou para a análise de séries temporais com ajustes específicos que não são diretamente suportados pelas funções padrão do Excel.

A metodologia para implementar essas fórmulas envolve a programação em VBA, onde o código é desenvolvido para realizar uma função específica, depois é integrado à planilha como uma função que pode ser chamada assim como qualquer outra função padrão do Excel. Isso permite uma flexibilidade significativa, pois é possível definir exatamente como os dados serão processados, adaptando o cálculo às peculiaridades dos dados ou às exigências específicas da análise.

### 3.4 A utilização de *userforms* no VBA para cadastro de clientes

A implementação de um *userforms* no VBA (*Visual Basic for Applications*) é uma metodologia chave para o cadastro de clientes em uma tabela Excel, facilitando a entrada de dados e a interação com o usuário de maneira estruturada e eficiente. Este *userforms* serve como uma interface gráfica que permite aos usuários inserirem informações relevantes sobre clientes de forma organizada e sem erros, assegurando que os dados sejam coletados e registrados corretamente.

O *userforms* foi projetado com campos específicos para a entrada de dados, como o CNPJ e os itens referentes aos itens do faturamento. Cada campo é validado para garantir a integridade dos dados inseridos, utilizando-se de validações como a verificação do formato e da completude dos campos obrigatórios.

Além disso, uma característica importante deste *userforms* é a integração com a API da Receita Federal para a consulta do CNPJ das empresas. Ao inserir o CNPJ no formulário, o sistema automaticamente consulta a API e preenche os campos do formulário com os dados públicos disponíveis, como nome da empresa, endereço e status do CNPJ.

A integração da API da Receita Federal é realizada por meio de chamadas HTTP diretamente do VBA. O código VBA é configurado para enviar o número do CNPJ à API e receber um conjunto de dados em formato JSON, que é então parseado e distribuído nos campos apropriados do *userforms*. Esta automação não só simplifica o processo de entrada de dados como também garante que as informações estejam sempre atualizadas e sejam consistentes com os registros oficiais.

A aplicação de um *userforms* no VBA, combinada com a utilização da API da Receita Federal, é uma metodologia eficaz para o gerenciamento de dados de clientes em um ambiente corporativo. Esta abordagem não apenas melhora a eficiência do processo de cadastro de clientes, mas também aumenta a precisão dos dados coletados. O uso de tecnologia API assegura

que o sistema esteja sempre sincronizado com as bases de dados governamentais, proporcionando confiabilidade nas informações cadastrais.

### **3.5 O Power Query® no carregamento de dados para o Excel®**

Power Query® é uma ferramenta poderosa para importação e transformação de dados dentro do ambiente Microsoft Excel® e Power BI®. Ela permite aos usuários extraírem dados de diversas fontes, limpar esses dados e reformatá-los de maneira que sejam mais adequados para análise ou relatórios. As funcionalidades do Power Query® incluem filtragem de dados, alteração de tipos de dados, mesclagem e agregação de tabelas, além da possibilidade de automatizar o processo de atualização dos dados com refrescamento periódico.

No contexto do trabalho atual, o Power Query® é utilizado especificamente para o carregamento de dados que são recebidos em um formato específico, preparando-os para utilização no sistema. Essa ferramenta simplifica a integração de dados ao permitir que usuários configurem uma vez o processo de importação e transformação, o qual pode ser reaplicado automaticamente sempre que os dados forem atualizados ou recebidos novamente. Isso economiza tempo e assegura que o processamento dos dados seja consistente e confiável, facilitando a manutenção e a gestão de dados dentro do sistema. O uso do Power Query, portanto, representa um componente crítico na estrutura do trabalho de análise de dados, maximizando a eficiência e a eficácia na manipulação de informações complexas.

## 4 RESULTADOS E DISCUSSÃO

Considerando o cenário descrito e o objetivo do presente trabalho de desenvolver um sistema para o gerenciamento e análise de dados financeiros no contexto de bancarização de empresas, esta sessão de resultados visa apresentar a eficácia e os benefícios das soluções propostas. O foco é demonstrar como as diferentes funcionalidades do sistema contribuem para uma análise precisa e eficiente dos dados de preços e volumetrias de consumo de APIs, além de facilitar o cadastro e a precificação em diversas metodologias.

Primeiramente, serão apresentados os resultados obtidos com o uso do *userforms* no VBA para o cadastro de preços. Esta funcionalidade provou ser crucial para reduzir erros de entrada de dados e agilizar o processo de inclusão de novos clientes, beneficiando-se significativamente da integração com a API da Receita Federal. Tal integração permite um preenchimento automático e mais preciso dos dados cadastrais, melhorando a qualidade das informações armazenadas no sistema.

Além disso, a funcionalidade de geração de relatórios em formato PDF a partir da interface do sistema é discutida, destacando como essa capacidade de automatizar a emissão de demonstrativos aumenta a eficiência operacional e proporciona uma entrega de dados clara e profissional aos clientes.

Por fim, é analisado o impacto da ferramenta de inteligência artificial desenvolvida em Python na identificação de valores atípicos, detalhando como essa análise avançada contribui para uma gestão mais estratégica de preços e identificação de oportunidades. Através de exemplos específicos e análises de dados, demonstraremos como a ferramenta IA (Inteligência Artificial) conseguiu destacar discrepâncias e fornecer dados valiosos que podem influenciar decisões de negócios.

O sistema operacional instalado no computador utilizado para executar o sistema é o Microsoft® Windows 10 Pro, versão 10.0.19045. A fabricante é a LENOVO®, modelo 81FE, e o tipo de sistema é baseado em x64. A configuração inclui um processador Intel® Core™ i5-8250U CPU @ 1.60GHz, com 4 núcleos e 8 threads. A BIOS instalada é a versão LENOVO 8TCN61WW, datada de 19/05/2021. A memória RAM tem 8 GB (8192MB RAM). A placa de vídeo utilizada é a Intel® UHD Graphics 620 que possui 4178 MB de memória total aproximadamente, sendo 128 MB de memória de vídeo (VRAM) e 4050 MB de memória compartilhada.

Ao final desta sessão, espera-se fornecer uma base sólida de conhecimento que possa ser utilizada para futuras investigações, além de contribuir para a literatura existente na área de modelagem e análise de dados.

#### 4.1 *Userforms* para cadastro de clientes

O *userforms* VBA para Cadastro de Preços, parte integrante do sistema desenvolvido, exibe uma operacionalidade intuitiva e dinâmica adaptável ao status do CNPJ consultado. Inicialmente, a janela está com todos os campos bloqueados, exceto o do CNPJ para que não seja possível o preenchimento de dados sem que estejam relacionados a um cliente específico. Quando um CNPJ é inserido, o sistema verifica automaticamente se este já possui um cadastro associado. Em caso afirmativo, o cadastro existente é imediatamente exibido, conforme ilustrado na Figura 11, onde itens como "PIX", "PIX Consultas DICT" e "Contas Ativas" são listados juntamente com suas respectivas precificações e volumes. Isso permite uma revisão e atualização rápida dos dados, garantindo que as informações estejam sempre corretas e atualizadas. Além disso, sempre que um mesmo cliente for cadastrado mais de uma vez, é exibida uma mensagem avisando que o cadastro anterior é apagado e substituído pelo novo, atualizando as informações do CNPJ e precificação dos itens do cliente.

**Figura 11** – Exemplo de cliente já cadastrado

The screenshot shows a window titled 'Cadastro de Clientes' with a close button (X) in the top right corner. The window contains several input fields and buttons:

- CNPJ:** 00.038.166/0001-05
- Razão Social:** BANCO CENTRAL DO
- Item:** A dropdown menu.
- Item\_code:** A dropdown menu.
- Modalidade:** A dropdown menu.
- id\_Cliente:** 5
- Preço:** An input field.
- Buttons:** 'Adicionar', 'Remover', and 'Desfazer' (with a circular arrow icon).

Below the input fields is a table with the following data:

| Item               | Preço    | Modalidade | Volumetria De | Volumetria Até |
|--------------------|----------|------------|---------------|----------------|
| PIX                | R\$ 0,25 | Padrão     | 1             | 9999999999     |
| PIX Consultas DICT | R\$ 0,50 | Padrão     | 1             | 9999999999     |
| Contas Ativas      | R\$ 0,35 | Padrão     | 1             | 9999999999     |
|                    |          |            |               |                |
|                    |          |            |               |                |
|                    |          |            |               |                |

At the bottom right of the window is a 'Cadastrar' button.

Fonte: Autoria própria

Para novos CNPJ, o formulário se apresenta inicialmente em branco, convidando ao preenchimento e cadastro de novos itens e precificações. Este processo é simplificado pela

interface clara e direta do *userforms*, facilitando a entrada de dados e promovendo um fluxo de trabalho eficiente.

É importante destacar que o armazenamento dos dados é realizado em uma tabela que fica em uma planilha oculta dedicada para tal que posteriormente é utilizada na criação do demonstrativo.

Além disso, é importante destacar que foi desenvolvida uma forma de precificação que calcula o valor do item e multiplica pelo volume apenas. Este tipo de precificação se demonstra especialmente importante porque a cada faixa de volumetria, o preço diminui, incentivando assim o crescimento da operação dos clientes que, em vários contextos, tem custos adicionais menores para quem fornece o serviço. Esta modalidade de precificação foi chamada de “escalonamento”.

Quando a modalidade escalonamento é selecionada, conforme evidenciado pela Figura 12, o sistema revela uma nova janela onde os preços podem ser definidos em faixas de volumetria.

**Figura 12** – Cadastro de item no formato de escalonamento.

The figure consists of two overlapping screenshots of a web application interface. The left screenshot shows the 'Cadastro de Clientes' form. It has fields for 'CNPJ' (00.038.166/0001-05), 'Razão Social' (BANCO CENTRAL DO), 'Item' (Requisição de API), 'Modalidade' (Escalonamento), and 'Preço'. Below these fields is a table with columns: Item, Preço, Modalidade, Volumetria De, and Volumetria Até. The right screenshot shows the 'Escalonamento' sub-form. It has fields for 'Início da Faixa' (50001), 'Final da faixa', and 'Preço'. Below these fields is a table with columns: Volumetria De, Volumetria Até, and Preço, showing a single entry with Volumetria De 1, Volumetria Até 50000, and Preço R\$ 0,30.

Fonte: Autoria própria

Neste contexto, é possível considerar que o início de cada faixa é sempre dado pelo volume final da faixa anterior acrescentado de uma unidade. Dessa forma, o campo início da faixa é fixo e determinado pelo próprio código quando se inicia o cadastro de uma nova faixa. Além disso, o valor final da última faixa é sempre considerado como o valor máximo de volumetria suportado pelo sistema, de forma que, caso o usuário cadastre apenas uma faixa, não deve haver diferença de comportamento entre a modalidade comum e a modalidade escalonamento.

Além disso, foi construída também a modalidade de mensalidade, na qual o valor de volumetria é desconsiderado e é aplicado um valor fixo mensalmente.

## 4.2 Geração de demonstrativos em PDF

A funcionalidade de geração de demonstrativos em PDF do sistema é um dos pilares do trabalho apresentado, e depende de alguns outros procedimentos realizados anteriormente. Como exemplo, foi utilizado um demonstrativo com dados fictícios para a Universidade Estadual Paulista Júlio de Mesquita Filho, referente a 01-2024, para fins de ilustração da capacidade do sistema de geração do arquivo PDF.

Inicialmente, deve-se cadastrar o cliente conforme descrito no passo anterior. Em seguida, os dados são mostrados na tela e o usuário deve selecionar o período de referência e o cliente para o qual o demonstrativo é gerado. Em seguida, o usuário analisa o demonstrativo a ser gerado, seleciona o caminho em que o demonstrativo será salvo e aperta o botão “Salvar Demonstrativo”. A Figura 13 ilustra o procedimento a ser realizado.

**Figura 13** – Demonstrativo de exemplo.

Fonte: Autoria própria

O layout do relatório é organizado de forma a destacar as métricas mais importantes, como volume de transações e a taxa de serviço aplicada "Fee", que são vitais para o entendimento dos custos associados a cada serviço utilizado pelo cliente. Por exemplo, o demonstrativo exibe várias faixas de volume para transações PIX, onde cada intervalo possui um preço por transação diferenciado. Além disso, existe a possibilidade de adição de algumas outras funcionalidades, como anuidades, valores fixos por faixa de transação, entre outros que não foram incluídos a este trabalho por uma questão de simplicidade e por se tratar de uma prova de conceito.

Dessa forma, para que os dados sejam mostrados de forma clara e de fácil entendimento, cada item é apresentado com seu volume correspondente e o valor total calculado, culminando em um total geral que resume o custo dos serviços prestados. A formatação e o design do PDF garantem que todos os dados sejam de fácil leitura e análise, com uma estrutura que permite

aos clientes compreenderem rapidamente os detalhes financeiros de suas operações. O resultado pode ser visualizado na Figura 14.

**Figura 14** – Demonstrativo em formato PDF



**Demonstrativo 01-2024**  
Universidade Estadual Paulista Julio De  
Mesquita Filho

| Item                 | Métrica                   | Volume      | Fee          | Valor Total       |
|----------------------|---------------------------|-------------|--------------|-------------------|
| Gestão de tesouraria | R\$ / Mês                 | 84.354      | R\$ 2,00     | R\$ 168.708,00    |
| Cartões Ativos       | R\$ / Cartão desbloqueado | 1           | R\$ 5.000,00 | R\$ 5.000,00      |
|                      |                           | <b>Soma</b> | <b>R\$</b>   | <b>173.708,00</b> |

Fonte: Autoria própria

Conforme demonstrado, existe espaço para personalização do logo da empresa que irá utilizar o sistema. Todo o layout também pode ser alterado facilmente por uma pessoa que tenha conhecimentos sobre a ferramenta Excel®, que nos dias de hoje é amplamente utilizada mundialmente.

O documento finaliza com a soma total de todos os serviços, fornecendo uma visão geral dos custos totais para o cliente, assim como uma visão para o próprio usuário do valor da nota fiscal que deve ser emitida para cobrança dos serviços prestados.

### 4.3 Detecção de outliers na série temporal

O uso de Redes Neurais Recorrentes *Long Short-Term Memory* (LSTM) emergiu como uma técnica eficaz na previsão de eventos atípicos, ou outliers, em séries temporais financeiras. Este estudo aplicou um modelo LSTM para identificar discrepâncias em um conjunto de dados de transações bancárias, usando como base o histórico do faturamento de cada um dos clientes ao longo de um ano completo para realizar a predição de volumetrias futuras.

Os resultados obtidos a partir do modelo LSTM indicam sua capacidade de detectar variações significativas no volume de transações, as quais podem ser indicativas de

comportamentos anômalos ou fraudes. A Figura 15 exemplifica a interface utilizada para a apresentação dos outliers identificados.

**Figura 15** – Tela para identificação de outliers

|    | IssuerId | ItemCode | process_date | Volume | cliente          | item_name          |
|----|----------|----------|--------------|--------|------------------|--------------------|
| 38 | 5        | 1        | 01/02/2023   | 51198  | BANCO CENTRAL DO | PIX                |
| 39 | 5        | 1        | 01/03/2023   | 27030  | BANCO CENTRAL DO | PIX                |
| 40 | 5        | 1        | 01/04/2023   | 28317  | BANCO CENTRAL DO | PIX                |
| 41 | 5        | 1        | 01/05/2023   | 25356  | BANCO CENTRAL DO | PIX                |
| 42 | 5        | 1        | 01/06/2023   | 27936  | BANCO CENTRAL DO | PIX                |
| 43 | 5        | 1        | 01/07/2023   | 27450  | BANCO CENTRAL DO | PIX                |
| 44 | 5        | 1        | 01/08/2023   | 33426  | BANCO CENTRAL DO | PIX                |
| 45 | 5        | 1        | 01/09/2023   | 48777  | BANCO CENTRAL DO | PIX                |
| 46 | 5        | 1        | 01/10/2023   | 35913  | BANCO CENTRAL DO | PIX                |
| 47 | 5        | 1        | 01/11/2023   | 36951  | BANCO CENTRAL DO | PIX                |
| 48 | 5        | 1        | 01/12/2023   | 63618  | BANCO CENTRAL DO | PIX                |
| 49 | 5        | 1        | 01/01/2024   | 105495 | BANCO CENTRAL DO | PIX                |
| 50 | 5        | 2        | 01/02/2023   | 45258  | BANCO CENTRAL DO | PIX Consultas DICT |
| 51 | 5        | 2        | 01/03/2023   | 27645  | BANCO CENTRAL DO | PIX Consultas DICT |
| 52 | 5        | 2        | 01/04/2023   | 25905  | BANCO CENTRAL DO | PIX Consultas DICT |
| 53 | 5        | 2        | 01/05/2023   | 32511  | BANCO CENTRAL DO | PIX Consultas DICT |
| 54 | 5        | 2        | 01/06/2023   | 24315  | BANCO CENTRAL DO | PIX Consultas DICT |
| 55 | 5        | 2        | 01/07/2023   | 28038  | BANCO CENTRAL DO | PIX Consultas DICT |
| 56 | 5        | 2        | 01/08/2023   | 26394  | BANCO CENTRAL DO | PIX Consultas DICT |
| 57 | 5        | 2        | 01/09/2023   | 43614  | BANCO CENTRAL DO | PIX Consultas DICT |
| 58 | 5        | 2        | 01/10/2023   | 44580  | BANCO CENTRAL DO | PIX Consultas DICT |
| 59 | 5        | 2        | 01/11/2023   | 35517  | BANCO CENTRAL DO | PIX Consultas DICT |
| 60 | 5        | 2        | 01/12/2023   | 48768  | BANCO CENTRAL DO | PIX Consultas DICT |
| 61 | 5        | 2        | 01/01/2024   | 107652 | BANCO CENTRAL DO | PIX Consultas DICT |

Fonte: Autoria própria

Para os testes realizados no *dataset*, o parâmetro *threshold* utilizado foi de 0,5 desvio padrão. Isso significa que erros foram identificados apenas se a variação for maior que 0,5 desvio padrão dos dados normalizados. Este parâmetro deve ser ajustado conforme o desejo do usuário para se adequar ao modelo de negócio.

Por meio do botão "Detectar Outliers", um código *Python* é acionado, utilizando um arquivo de lote (.bat) para ativar um script que processa os dados e carrega os resultados. Apesar de o programa apresentar uma lentidão considerável no sistema em que foi testado, o desempenho melhora em sistemas que possuem placa de vídeo dedicada, principalmente as mais modernas que possuem módulos dedicados para o processamento de inteligência artificial.

O código identificou cinco *outliers* para o *dataset* utilizado, levando em consideração o mês de janeiro de 2024.

Para o "IssuerId" 5, o "ItemCode" 2 foi considerado como um outlier pela análise. Os volumes para este item apresentaram variações moderadas ao longo de 2023, com volumes entre 24.315 e 48.768. O volume de janeiro de 2024 foi de 107.652, o que é mais que o dobro

do volume máximo registrado no mês de dezembro de 2023. Este aumento sugere um outlier significativo, considerando o padrão relativamente estável de volumes ao longo de 2023.

Na etapa de implementação do modelo LSTM, uma análise aprofundada do conjunto de dados foi fundamental para estabelecer uma base sólida para a predição de outliers. A utilização das métricas *Mean Absolute Error* (MAE) e *Root Mean Squared Error* (RMSE) revelou uma performance promissora do modelo, com valores médios de erro indicando uma habilidade satisfatória do modelo em capturar e prever discrepâncias nos dados.

Para avaliar o conjunto de dados inicial, foram feitas algumas medições gerais mais simplificadas, que indicam algumas características do modelo e tornam a análise da predição de erros mais assertiva. Neste caso, observou-se que o volume médio de transações por mês é de 44.015,99, com uma variação substancial, conforme indicado por um desvio padrão de 21.896,35. Isso sugere uma distribuição ampla nos volumes de transações, com alguns valores extremamente altos que distorcem a média.

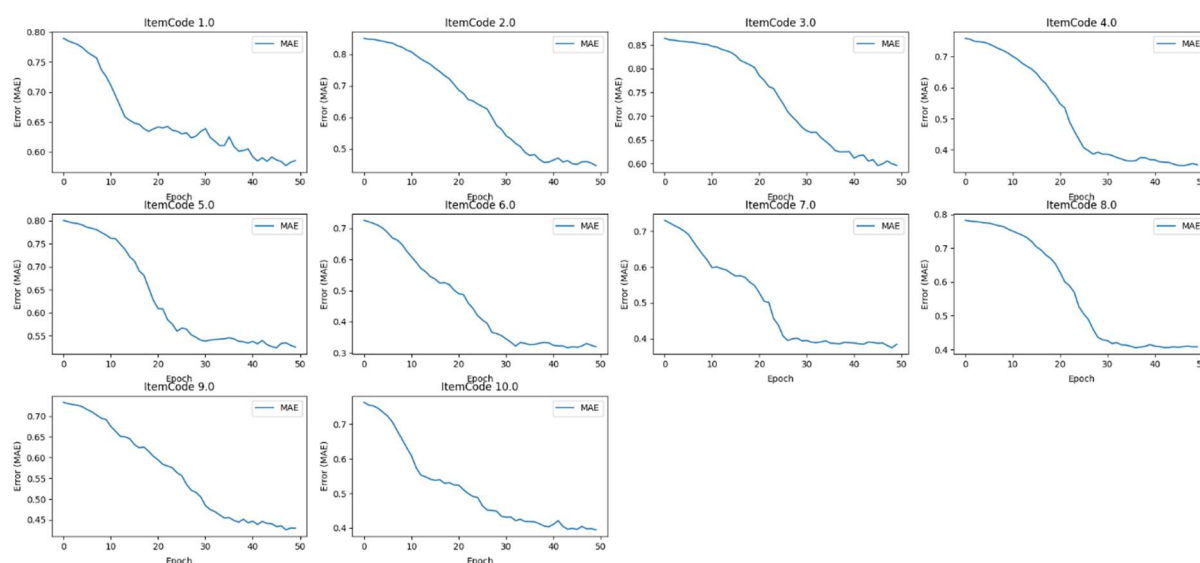
A análise revelou também uma instância de volume mínimo negativo de -24.309, o que pode ser indicativo de operações de estorno ou reversões, um comportamento também relevante para ser captado por modelos preditivos na identificação de padrões atípicos. Ademais, uma tendência de aumento no volume total mensal foi evidenciada no conjunto de dados ao longo dos meses no geral. Tais considerações enriquecem a discussão sobre o desempenho do modelo LSTM.

Para avaliar o desempenho do modelo de predição proposto, utilizou-se a biblioteca *scikit-learn* para calcular o RMSE e o MAE. O RMSE registrado foi de aproximadamente 11.500, enquanto o MAE foi de cerca de 8.350, variando alguns pontos para baixo ou para cima dependendo da *seed* aleatória que é atribuída para cada treinamento pela biblioteca. Esses valores indicam a média dos erros na predição de transações que variam menos de um desvio padrão no caso do erro absoluto. Considerando que o principal objetivo deste trabalho é a predição de outliers, e não a predição exata dos dados em geral, um erro que a princípio é aparentemente alto sugere apenas que o limiar para a identificação de outliers na série temporal analisada deve ser superior a meio desvio padrão. Com base nessa análise, adotou-se um *threshold* de 0,5 ou meio desvio padrão para a detecção de *outliers* na série temporal por ser o valor que mais se mostrou adequado para demonstração no modelo de dados.

Para ajudar na avaliação do modelo, a biblioteca *tensorflow* possui um parâmetro chamado *history* que grava os valores de erro encontrados a cada iteração de treinamento do

modelo até não encontrar mais melhoras significativas. A quantidade de iterações ou *epochs*, como são chamadas foi definida como 50, porém o modelo possui um parâmetro de parada chamado *patience* que interrompe as iterações quando não há mais melhoras significativas após 10 *epochs* de treinamento. A Figura 16 apresenta graficamente a melhora do erro com relação às iterações de treinamento para o “*IssuerId*” 4.

**Figura 16** – Erro por iteração para o “*IssuerId*” 4

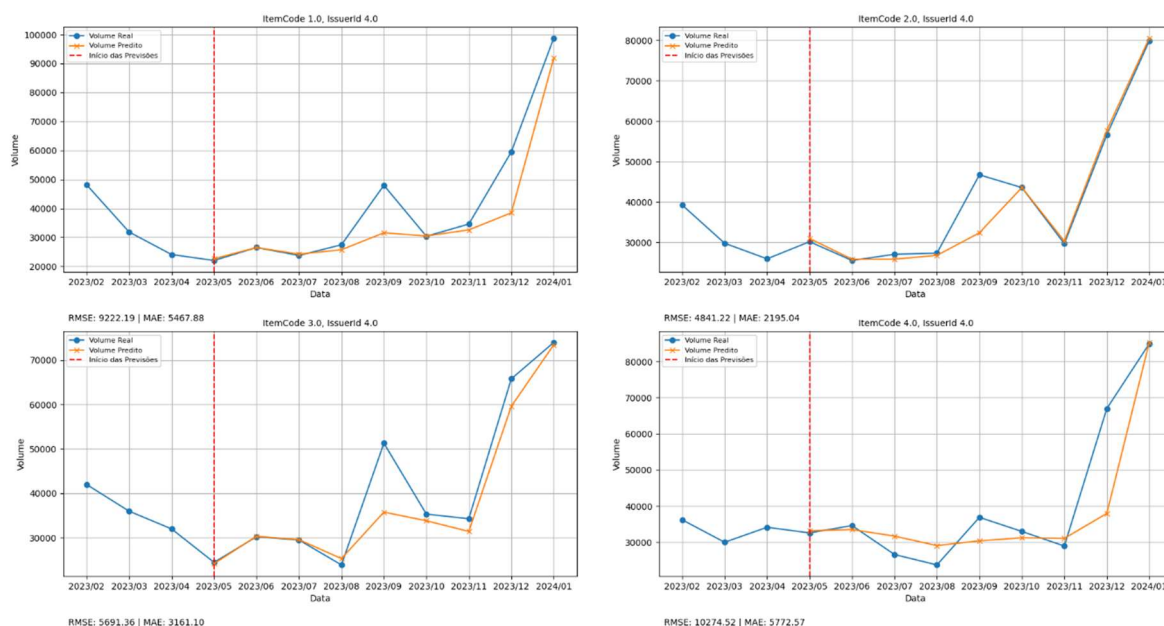


Fonte: Autoria própria

Pode-se observar, então, que o modelo apresenta melhoras significativas ao longo de 50 *epochs* para os casos normais. Para o “*ItemCode*” 1, foi observada uma grande variação no erro ao final da predição, indicando um possível *outlier*. Nesse caso, o modelo foi capaz de prever corretamente o item como um *outlier*, já que os volumes para este item apresentaram variações moderadas ao longo de 2023, com volumes entre 22.056 e 59.517. O volume de janeiro de 2024 foi de 98.808, o que é quase o dobro do volume máximo, que foi registrado no mês de dezembro de 2023.

Outra métrica que pode ser utilizada para a avaliação do modelo é a utilização de uma janela deslizante que aumenta o volume de dados do treinamento conforme sua disponibilidade e faz predições ao longo da série temporal para cada um dos pontos. A Figura 17 representa a comparação dessas predições com o volume real observado para os itens de 1 a 4 do “*IssuerId*” 4.

**Figura 17** – Volume predito x volume real para os primeiros itens do “*IssuerId*” 4



Fonte: Autoria própria

A figura gerada utilizando um script *Python* mostra quatro gráficos, cada um representando uma combinação de “*ItemCode*” e “*IssuerId*”, comparando os volumes reais em azul com os volumes preditos em laranja ao longo do tempo. Cada gráfico tem uma linha tracejada vermelha que indica o ponto de início das previsões. Abaixo de cada gráfico estão os valores de RMSE e MAE, que quantificam a precisão das previsões.

No primeiro gráfico, pode-se observar a variação do volume ao longo do tempo. A linha azul representa o volume real e a linha laranja mostra o volume predito pelo modelo LSTM. O erro para esta combinação é RMSE: 9222.19 e MAE: 5467.88, indicando a diferença média entre os valores reais e preditos. Como o volume predito para janeiro/2024 apresentou uma diferença maior que meio desvio padrão com relação ao volume real, o valor foi identificado como um outlier.

O segundo gráfico segue o mesmo formato. Aqui, o modelo também tenta seguir a tendência dos dados reais. A linha laranja dos volumes preditos segue de perto a linha azul dos volumes reais, especialmente antes e depois do ponto de início das previsões. Os erros calculados para esta combinação são RMSE: 8431.22 e MAE: 2195.04, sugerindo uma predição relativamente mais precisa em comparação com o primeiro gráfico.

No terceiro gráfico, a comparação entre os volumes reais e preditos é apresentada novamente. Este gráfico mostra uma tendência semelhante, com o modelo LSTM acompanhando os volumes reais ao longo do tempo. Os valores de RMSE e MAE são 5691.36 e 3161.10, respectivamente, indicando um desempenho de predição um pouco melhor em relação aos gráficos anteriores.

O quarto e último gráfico, exibe a quarta combinação de “*ItemCode*” e “*IssuerId*”. Neste gráfico, os volumes reais e preditos também são comparados, com a linha laranja das predições seguindo a linha azul dos dados reais. Os erros RMSE e MAE para esta combinação são 10274.52 e 5772.57, respectivamente, mostrando que as predições têm um erro maior em comparação com os outros gráficos.

Em resumo, os quatro gráficos ilustram como o modelo LSTM foi capaz de prever os volumes ao longo do tempo para diferentes combinações de “*ItemCode*” e “*IssuerId*”. A linha tracejada vermelha destaca o início das predições, e as métricas de erro RMSE e MAE fornecem uma medida da precisão dessas predições. Embora o modelo siga bem a tendência geral dos dados reais, a precisão varia entre as diferentes combinações, refletida nos diferentes valores de RMSE e MAE.

No contexto prático, as predições geradas pelo modelo LSTM são vitais para a detecção de fraudes e comportamentos anormais em transações financeiras. Oscilações acentuadas no volume de transações, bem como transações reversas, são aspectos onde o modelo deve provar ser eficaz, potencialmente protegendo contra riscos financeiros e operacionais.

Portanto, a integração do modelo LSTM com as métricas de avaliação e a análise temporal evidencia uma ferramenta robusta na identificação de comportamentos atípicos nas transações, servindo de referência para futuras melhorias e aplicações práticas do sistema de faturamento.

A discussão sobre os resultados enfoca não apenas a precisão do modelo LSTM em detectar outliers, mas também sua relevância prática. Os outliers identificados foram analisados para determinar se representavam verdadeiras anomalias ou variações benignas do comportamento esperado. Observou-se que o modelo foi capaz de discernir com sucesso entre variações normais de volume, que podem ser causadas por flutuações de mercado, e variações atípicas que sugerem a necessidade de uma investigação mais aprofundada.

## 5 CONCLUSÃO

O presente trabalho desenvolveu um sistema de análise de dados e faturamento, destacando-se pela integração de ferramentas como VBA e Python para otimizar processos em ambientes corporativos. Este sistema é notavelmente relevante para empresas que buscam uma solução robusta e adaptável para a gestão de preços e volumetrias de consumo de APIs, facilitando operações financeiras como transferências PIX e emissões de boletos.

Primeiramente, o uso de VBA para o desenvolvimento de *userforms* personalizados permitiu um cadastro mais intuitivo e seguro de dados de clientes, minimizando erros de entrada e aumentando a eficiência operacional. A integração do sistema com APIs, como a da Receita Federal, garantiu a precisão e a atualização constante dos dados cadastrais, que são cruciais para a confiabilidade dos processos de faturamento.

A análise estatística e a utilização de tecnologias de aprendizado de máquina, especificamente o modelo LSTM, foram fundamentais para a detecção de outliers e para a predição de comportamentos anômalos, que poderiam comprometer a integridade dos dados financeiros. Este aspecto é particularmente valioso para prevenir fraudes e otimizar a tomada de decisões estratégicas baseadas em dados confiáveis e precisos.

Ademais, a capacidade do sistema de gerar relatórios detalhados em formato PDF proporciona uma ferramenta adicional para comunicação e transparência com os clientes, fortalecendo relações comerciais e promovendo um ambiente de negócios mais integrado e informado.

Por fim, a aplicação prática deste sistema em um ambiente corporativo demonstra sua capacidade de adaptar-se a diversos requisitos de negócios, tornando-o uma solução escalável e eficaz para empresas de diferentes tamanhos e segmentos. A implementação bem-sucedida deste sistema é um testemunho do potencial de combinar ferramentas tradicionais de programação e técnicas avançadas de análise de dados para melhorar significativamente a eficiência e a eficácia dos processos de faturamento empresarial.

## REFERÊNCIAS

BUSSAB, P. A.; MORETTIN, J. M. **Estatística Básica**. 6ª ed. São Paulo: Saraiva, 2010.

CREATING UserForms in Excel. [S. l.]: Best Excel Tutorial, 28 jul. 2023. Disponível em: <https://best-excel-tutorial.com/creating-userforms-in-excel/>. Acesso em: 25 maio 2024.

DAVIS, L. M. **For Law Firms, Better Technology to Do a Tough Job Better**. Of Counsel, v. 22, n. 2, p. 13, 2003. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=9072584&lang=pt-br&site=ehost-live>. Acesso em: 3 fev. 2024.

FILHO, M. **Como Prever Séries Temporais Com LSTM em Python**. Mario Filho | Machine Learning, 2023. Disponível em: <https://mariofilho.com/como-prever-series-temporais-com-lstm-em-python>. Acesso em: 25 maio 2024.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. **LSTM: A Search Space Odyssey**. IEEE Transactions on Neural Networks and Learning Systems, v. 28, n. 10, p. 2222, 2017.

HODGE, V., AUSTIN, J. **A Survey of Outlier Detection Methodologies**. Kluwer Academic Publishers, v. 22, p. 85–126, 2004.

HOCHREITER, S.; SCHMIDHUBER, J. **Long short-term memory**. Neural computation, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

JELLEN, B.; SYRSTAD, T. **VBA e macros para Microsoft Office Excel 2007**. 1ª Edição E-book. São Paulo: Pearson, 2008. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 17 set. 2023.

LIANG, X.; XU, J. **Biased ReLU neural networks**. Neurocomputing, v. 423, p. 71-79, 2021.

MACHINE LEARNING PLUS. **Pandas Series**. Machine Learning Plus. 2021. Disponível em: <https://www.machinelearningplus.com/python/pandas-series>. Acesso em: 25 maio 2024.

McKINNEY, W. **Python for data analysis: Data wrangling with Pandas, NumPy, and IPython**. 2<sup>nd</sup> ed. California: O'Reilly Media, Inc., 2017.

MORETTIN, P. A.; SINGER, J. M. **Introdução à Ciência de Dados: Fundamentos e Aplicações**. Versão parcial preliminar. São Paulo: Departamento de Estatística Universidade de São Paulo, 2019.

OLIVEIRA, A. **Guia Prático da Contabilidade Gerencial**. 1ª Edição. Rio de Janeiro: Freitas Bastos, 2022. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 28 ago. 2023.

SCIKIT-LEARN. **Metrics and scoring: quantifying the quality of predictions**. 2024. Disponível em: [https://scikit-learn.org/stable/modules/model\\_evaluation.html#regression-metrics](https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics). Acesso em: 24 out. 2023.

KOROL, J. **Learn Microsoft Excel 2002 VBA Programming with XML and ASP**. Pap/Cdr Edition. Texas: Wordware Pub., 2003.

VARMA, P.; NIJER, S.; SOOD, K.; GRIMA, S.; RUPEIKA-APOGA, R. **Thematic Analysis of Financial Technology (Fintech) Influence on the Banking Industry**. Risks, v. 10, a. 186, 2022. Disponível em: <https://doi.org/10.3390/risks10100186> Acesso em: 25 mai. 2024

WORKING with Time Series data: splitting the dataset and putting the model into production. [S. l.]: Stack Exchange, 23 jul. 2020. Disponível em: <https://stats.stackexchange.com/questions/453386/working-with-time-series-data-splitting-the-dataset-and-putting-the-model-into>. Acesso em: 25 maio 2024.