

INTEGRAÇÃO DE SISTEMAS ERP: OTIMIZAÇÃO DE API PARA EFICIÊNCIA E SEGURANÇA FINANCEIRA NA CONCILIAÇÃO DE CARTÕES DE CRÉDITO E DÉBITO

Aécio Flávio Cruz da Silva, Ana Cláudia Souza Vidal Negreiros.

Resumo: A integração entre sistemas de gestão empresarial (ERP) tem se mostrado, de forma geral, essencial para otimizar processos operacionais e garantir segurança financeira nas transações comerciais. Neste estudo, foi desenvolvida uma Interface de Programação de Aplicações (API) que automatiza a conciliação financeira de pagamentos por cartão de crédito e débito em uma empresa varejistas/atacadista. A solução proposta permite a transferência automática de dados entre o ERP e a plataforma conciliadora, reduzindo erros manuais e melhorando a precisão dos registros financeiros. A API foi implementada na linguagem Python, em que ela utiliza requisições de Protocolo Transferência de Hipertexto (HTTP) para exportação e importação de dados, garantindo compatibilidade com diferentes sistemas. Os resultados demonstram que a automação melhora a eficiência operacional, reduz inconsistências financeiras e fortalece a gestão empresarial. Dessa forma, a integração automatizada via API se apresenta como uma estratégia eficaz para aprimorar a segurança e a confiabilidade na conciliação de transações eletrônicas.

Palavras-chave: *Integração de sistemas; ERP; Conciliação financeira; API; Automação de processos.*

1. INTRODUÇÃO

A integração entre sistemas de gestão empresarial (*Enterprise Resource Planning - ERP*) tem se tornado uma necessidade estratégica para empresas que buscam otimizar processos internos e aumentar sua eficiência operacional e reduzir riscos financeiros [1]. O ERP atua como uma plataforma centralizada para a gestão de diferentes áreas dentro de uma empresa, como finanças, logística, vendas e recursos humanos, o que contribui para a tomada de decisões mais rápidas e eficazes, além de proporcionar maior controle sobre os fluxos financeiros e faturamento [2]. No setor varejista, especialmente no segmento de cosméticos, a dinâmica de vendas pode envolver uma variedade de métodos de pagamento, com destaque para o uso de cartões de crédito e débito, que representam uma parcela significativa do faturamento [3].

Neste cenário, a conciliação financeira se torna um processo essencial para garantir que todas as transações realizadas com cartões de crédito e débito sejam corretamente registradas nos sistemas financeiros da empresa, garantindo que os valores processados sejam alinhados com os montantes efetuados pelos pagamentos pelas operadoras de cartões [4]. Contudo, quando o processo é realizado manualmente, surgem desafios significativos, no qual a necessidade de comparar extratos bancários e registros de vendas sem automação torna o processo demorado e propenso a erros, como a falha na reconciliação de valores e inconsistências de dados [5]. Além disso, a ausência de um sistema integrado de monitoramento dificulta a identificação de irregularidades, como cobranças indevidas e discrepâncias nos valores recebidos, expondo a empresa a riscos financeiros [6]. Dessa forma, a implementação de soluções automatizadas de conciliação financeira integradas ao ERP se apresenta como uma estratégia fundamental para garantir maior eficiência, segurança e transparência na gestão dos pagamentos eletrônicos [7].

Este estudo tem como objetivo principal desenvolver uma interface de programação de aplicação (*Application Programming Interface - API*) que possibilite a integração entre o sistema ERP de uma empresa varejista e o sistema ERP conciliador de cartões de crédito e débito. A proposta de API será responsável por exportar automaticamente os dados de vendas do ERP e importar as informações de pagamentos processados pela conciliadora de cartões, eliminando a necessidade de conferências manual, fato que gera significativamente a incidência de erros. Além de otimizar a gestão operacional, a solução abordará aspectos críticos como a segurança das transações, a conformidade de taxas cobradas e a supervisão dos registros financeiros, garantindo maior precisão e agilidade no processo de conciliação. Um dos problemas mais recorrentes nesse contexto é a divergência entre os valores esperados e os valores efetivamente creditados pelas administradoras de cartões, podendo resultar em perdas financeiras significativas ao longo do tempo [8]. Diante desse cenário, esse estudo busca responder a seguinte questão central: "Como a integração via API entre ERP e sistema terceirizado pode melhorar a eficiência e a segurança na conciliação de cartões de crédito e débito em uma empresa atacadista/varejista?". A resposta a essa pergunta poderá contribuir para a formulação de estratégias inovadoras que aprimorem a gestão financeira, reduzam os riscos operacionais e fortaleçam a competitividade das empresas no mercado varejista.

2. REFERENCIAL TEÓRICO

2.1. Enterprise Resource Planning - ERP

O ERP, ou Sistema de Gestão Integrada, pode ser definido como uma solução empresarial abrangente, desenvolvida para integrar todos os departamentos e funções de uma organização em um único sistema de informatizado [9]. Esse sistema surgiu a partir da evolução do Planejamento de Requisitos de Materiais (*Material Requirements Planning* - MRP) e do Planejamento de Recursos de Manufatura (*Manufacturing Resource Planning* - MRP II), principalmente voltados para o controle de produção e estoque [10].

Com o avanço tecnológico, os ERPs expandiram suas funcionalidades, incorporando módulos que abrangem áreas como marketing e vendas, gestão de produtos, compras, distribuição e varejo, gestão de instalações industriais, qualidade, recursos humanos, finanças, contabilidade e serviços de informação [11]. Desde os primeiros sistemas de processamento de dados e computadores mainframe no final da década de 1950 e início dos anos 1960, os ERPs evoluíram de ferramentas isoladas para soluções completas que centralizam todas as operações empresariais. Durante a década de 1990, a popularização dos ERPs foi impulsionada pelo avanço dos softwares comerciais flexíveis, permitindo que SAP e Oracle desenvolvessem soluções modulares adaptáveis a diferentes setores e portes empresariais [12].

Atualmente, a complexidade crescente dos processos empresariais exige que os ERPs sejam capazes de se comunicar com plataformas externas para garantir operações mais ágeis e eficientes [13]. A integração entre um ERP e sistemas externos permite a automação da troca de informações em tempo real, reduzindo a necessidade de intervenções manuais e minimizando erros operacionais [14]. Nos últimos anos, os ERPs vêm incorporando novas tecnologias como inteligência artificial (IA) e machine learning (ML), permitindo análises preditivas e automatizações de processos repetitivos. Esses avanços aumentam a eficiência operacional possibilitando melhores adaptações as mudanças do mercado [15].

2.2. Integração de Sistemas

Pode-se definir integração de sistemas como o processo de conectar diferentes sistemas, aplicações ou softwares que possam compartilhar dados e funcionalidades de maneira eficiente e automatizada [16]. A funcionalidade é permitir que sistemas independentes trabalhem de forma coordenada, eliminando barreiras que impedem o compartilhamento de dados, melhorando a eficiência operacional [17].

Um exemplo relevante dessa integração ocorre na conciliação financeira de vendas com cartão de crédito e débito, onde as transações registradas no ERP podem ser automaticamente comparadas e validadas com os dados processados por uma conciliadora de cartões [18]. Esse processo garante maiores resultados nos registros contábeis e transparência na gestão financeira [19].

A automação desse processo evita inconsistências financeiras, reduz a necessidade de conferências manuais e proporciona maior controle sobre o fluxo de caixa da empresa. Com a integração desses sistemas, as organizações conseguem melhorar a tomada de decisões estratégicas, uma vez que os dados financeiros ficam centralizados e acessíveis em tempo real [20]. Empresas que operam como varejistas e atacadistas, podem se beneficiar significativamente dessa automação, reduzindo riscos e aumentando a precisão dos dados analisados [21].

A segurança na comunicação entre sistemas ERP e sistema conciliador é um fator crítico para a integridade e confiabilidade das operações financeiras e empresariais [22]. A falta de protocolos adequados pode expor dados seguros a riscos como fraudes financeiras, interceptação de informações e ataques cibernéticos, entre as principais ameaças estão os ataques *man-in-the-middle* (MITM), nos quais hackers interceptam dados trocados entre sistemas, e a visibilidade de acesso indevido, que pode comprometer informações sigilosas [23]. Para mitigar esses riscos, é essencial a implementação de protocolos de criptografia ponta a ponta, como TLS (*Transport Layer Security*), além de mecanismos de autenticação robusta, como OAuth 2.0 autenticação e multifator (MFA) [24].

Um dos desafios enfrentado pelas empresas ao integrar sistemas ERP com plataformas externas é a necessidade de compatibilidade com sistemas legados, que podem utilizar tecnologias obsoletas e dificultar a integração, além disso, o alto custo inicial de implementação pode ser uma dificuldade para médias e pequenas empresas, exigindo integração gradual [25].

2.3. Interface de Programação (API)

A API (*Application Programming Interface*) que em português significa Interface de Programação de Aplicações, desempenha um papel fundamental nesse processo, pois permite a comunicação estruturada entre diferentes sistemas, softwares ou dispositivos, funcionando como uma ponte que possibilita a troca de informação de forma estruturada e padronizada, garantindo que aplicações distintas possam interagir de maneira eficiente e segura [26].

Uma API é um conjunto de regras e protocolos que estabelece a forma como diferentes softwares devem interagir, facilitando a troca de informações de maneira automatizada e segura [27]. Essa tecnologia possibilita a integração de funcionalidades sem a necessidade de intervenção manual, otimizando processos e aumentando a

eficiência operacional [28].

Entre as inúmeras APIs disponíveis no mercado, uma das mais utilizada é a API do Paypal®, que permite integrar diversas funcionalidades de pagamentos em sistemas ERP, aplicativos móveis e sites de vendas, facilitando transações entre usuário e comerciantes. Essa API se destaca pela robustez e ampla gama de funcionalidades prontas para uso, permitindo maior controle sobre o tratamento e a segurança dos dados [8].

A automação da conciliação financeira desempenha um papel fundamental no cumprimento das normas contábeis e fiscais, garantindo que todas as transações sejam registradas de forma precisa e transparente [29]. Regulamentações como as Normas Internacionais de Relatório Financeiro (*International Financial Reporting Standards* - IFRS) e as normas do Conselho Federal de Contabilidade (CFC) exigem que as empresas tenham um controle rigoroso sobre suas operações financeiras, assegurando que receitas e despesas sejam devidamente conciliadas e declaradas [30,31]. Dessa forma, a automação melhora a rastreabilidade das transações e fortalece a governança financeira das empresas [6].

As APIs desempenham um papel essencial na transformação digital das empresas, permitindo a escalabilidade dos negócios ao integrar sistemas de terceiros, como e-commerce, sistemas de pagamento online e análise de dados, facilitando o processo de inovação contínua e possibilitando que as empresas adotem novas tecnologias sem a necessidade de ter altos custos com reestruturação no setor de TI [32].

2.4. Linguagem de Programação Python

A linguagem de programação Python destaca-se como uma linguagem de alto nível, multiparadigma, que prioriza a produtividade e clareza no código, baseado na sua codificação limpa e intuitiva, aliado a tipagem dinâmica, reduz a complexidade no desenvolvimento de soluções integradas como interfaces entre sistemas ERP e plataformas externas [33]. Essas características tornam a linguagem especialmente valiosa em projetos que demandam rápido desenvolvimento e adaptação contínua, comuns em ambientes empresariais onde os requisitos mudam frequentemente, além disso, a linguagem oferece suporte nativo a paradigmas de programação como orientação a objetos, procedural e funcional, permitindo a criação de softwares reutilizáveis e escaláveis, essenciais para conectar diferentes sistemas com segurança e eficácia [34]. Seu ecossistema robusto de bibliotecas e frameworks são um dos pilares da eficácia e eficiência do Python em integrações complexas, ferramentas como Requests, para comunicação HTTP e SQLAlchemy, para conexão com banco de dados, simplificam a extração, transformação e o envio e recebimento de informações entre sistemas [35].

Nesse sentido, com uma comunidade ativa e a vasta documentação disponível tornam o Python uma escolha segura para implementações críticas [33]. Em ambientes ERP, onde a escalabilidade e a manutenibilidade são prioritárias, essa rede de suporte assegura que desenvolvedores possam construir interfaces robustas, com ciclos de atualização ágeis e alinhados às demandas dinâmicas dos negócios [36].

2.5. Processo de Conciliação Financeira

A conciliação financeira desempenha um papel fundamental na gestão empresarial, garantindo a integridade e a precisão dos dados financeiros [37]. Ao comparar transações registradas em sistemas distintos, como extratos bancários, transações de cartões e registros internos de fluxo de caixa, esse processo identifica divergências, erros de lançamento, possíveis fraudes e evita perdas financeiras [38]. Entre os tipos de conciliações financeiras, destacam-se a conciliação de cartões, a conciliação de pagamentos, a conciliação de recebíveis e a conciliação bancária. A conciliação de cartões tem como objetivo verificar se as informações que as operadoras de cartões disponibilizam sobre as vendas de uma empresa contêm inconsistências, falhas operacionais, fraudes nos valores pagos, prazos acordados ou descontos cobrados [39].

Dentro desse tipo de conciliação, há etapas específicas a serem validadas: a conciliação das vendas, a conciliação do recebimento e a conciliação bancária. A conciliação financeira automatizada contribui para a saúde financeira da empresa ao eliminar atrasos na identificação de problemas. Erros não detectados, como pagamentos duplicados ou lançamentos omitidos, podem gerar prejuízos significativos [37]. Ao utilizar Python para automatizar a integração dessa operação, as empresas não apenas aumentam a eficiência operacional, mas também fortalecem a gestão financeira [40]. Isso é particularmente relevante em ambientes empresariais, onde a precisão dos dados e a capacidade de gerar comprovantes confiáveis são exigências estratégicas para a sustentabilidade do negócio [41].

3. METODOLOGIA

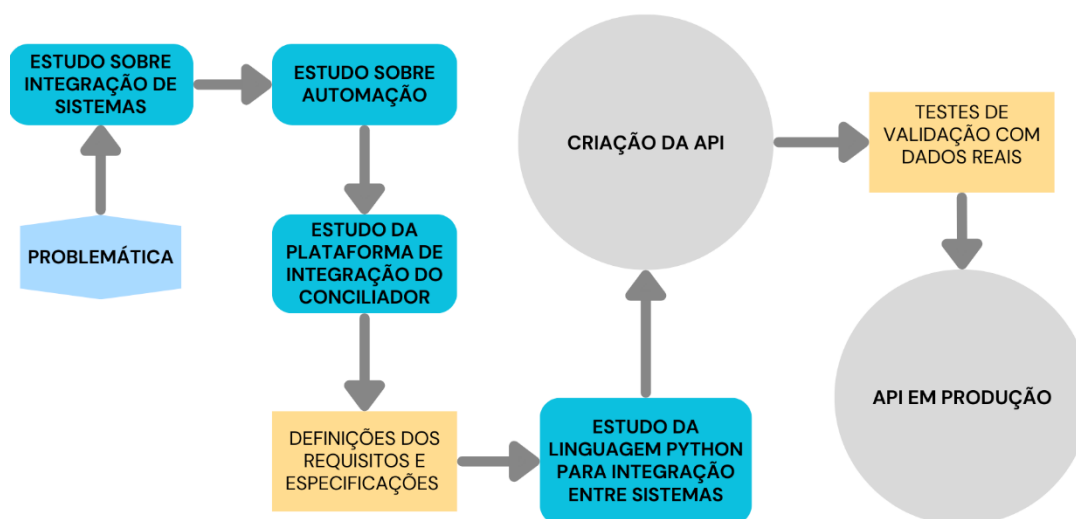
Esta pesquisa caracteriza-se como um estudo de caso, pois analisou um problema específico dentro de uma empresa do ramo varejista/atacadista e propôs uma solução para otimizar a conciliação de vendas com cartão de crédito e débito entre o sistema ERP e uma plataforma conciliadora. De acordo com Zou [42], um estudo de caso é uma investigação empírica que explora um fenômeno contemporâneo dentro de seu contexto real, especialmente quando as fronteiras entre o fenômeno e o contexto não são claramente definidas. O setor varejista é responsável por comercializar produtos diretamente ao consumidor final, abrangendo lojas físicas e online. Já o setor atacadista

atua na venda de grandes volumes para empresas, como varejistas e de serviços, garantindo eficiência na distribuição e redução de custos. A pesquisa apresenta uma abordagem predominantemente aplicada, pois busca desenvolver uma solução para um problema prático. Além disso, caracteriza-se como um estudo de natureza quantitativa e qualitativa; no qual, abordagem quantitativa é evidenciada na análise dos registros transacionais e na mensuração da eficiência da API desenvolvida, enquanto a abordagem qualitativa é utilizada na compreensão das dificuldades operacionais enfrentadas pela empresa antes da implementação da solução.

O estudo foi realizado em uma empresa do setor varejista/atacadista que enfrentava desafios na conciliação de vendas feitas via cartão de crédito e débito. A empresa possui um sistema ERP para gerenciar suas operações, mas enfrentava dificuldades na comunicação e integração dos dados com a plataforma conciliadora de cartões. Essa falta de padronização nos formatos dos dados entre os sistemas resulta em erros operacionais e retrabalho manual, aumentando custos e reduzindo a eficiência do processo financeiro. A pesquisa surgiu da observação dessas dificuldades, motivando o desenvolvimento de uma API para automatizar a integração entre os sistemas, garantindo maior precisão e agilidade no processamento dos dados financeiros. Com isso foi seguido as seguintes etapas: revisão da literatura sobre integração de sistemas, automação de processos e APIs para conciliação financeira. Foram analisadas pesquisas sobre boas práticas no desenvolvimento de APIs e métodos de integração entre sistemas distintos. Além disso, estudou-se a documentação da plataforma conciliadora e do sistema ERP da empresa para compreender os requisitos técnicos e os desafios de integração. Com base na revisão da literatura e na análise das necessidades da empresa, definiu-se que a melhor solução seria o desenvolvimento de uma API para atuar como intermediária entre o ERP e a plataforma conciliadora. Para isso, foram levantados os requisitos técnicos e especificações da API, incluindo os formatos de dados aceitos, métodos de autenticação e protocolos de comunicação. A figura 1 traz uma representação gráfica das etapas do estudo para modelagem da API.

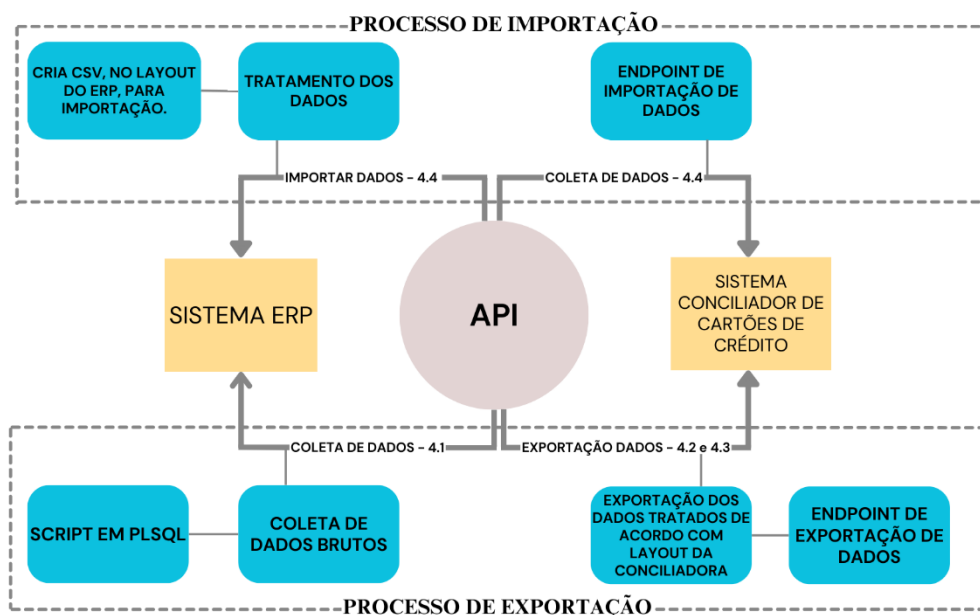
A API foi desenvolvida utilizando a linguagem de programação Python, com a biblioteca *requests* para facilitar a transmissão de dados pela rede. O método HTTP POST foi adotado para a comunicação entre os sistemas, permitindo a exportação e importação de dados de forma estruturada. A API recebe dados do ERP, os organiza conforme o layout exigido pela conciliadora e os transmite para processamento. As respostas da API são processadas no formato JSON, garantindo compatibilidade com os sistemas envolvidos. Após o desenvolvimento, foi testada com dados reais para validar sua funcionalidade e eficiência na comunicação entre os sistemas. A exportação dos dados do ERP foi feita por meio de consultas SQL, garantindo que as transações fossem extraídas corretamente. A importação dos dados foi ajustada conforme o layout do ERP, assegurando que os registros financeiros fossem conciliados de forma correta. Foram analisados os impactos da implementação, considerando a redução de erros operacionais e o ganho de tempo no processo de conciliação financeira. Dessa forma, a metodologia adotada garantiu uma abordagem estruturada para a identificação do problema, a definição da solução e a validação dos resultados, proporcionando uma contribuição significativa para a empresa analisada. A Figura 2 traz uma representação gráfica do funcionamento da API para a integração de sistemas.

Figura 1: Fluxograma das etapas do estudo.



Fonte: Autoria Própria (2024)

Figura 2: Fluxograma de funcionamento da API para integração dos sistemas.



Fonte: Autoria Própria (2024)

4. RESULTADOS E DISCUSSÃO

4.1 Coleta de Dados

O primeiro procedimento que é feito pela API é a extração dos dados diretamente no banco de dados, onde é armazenado todas as informações geradas pela empresa, incluindo os registros das transações de vendas em cartão de crédito e débito. Esse procedimento é feito através de um *script* criado em SQL, que é uma extensão da linguagem padrão. A busca dessas informações é baseada no layout exigido pela conciliadora que dentre todas as informações de layout, as obrigatórias são: número da venda, data da venda, valor da venda, número de parcelas, código da filial, bandeira do cartão, formato de captura da venda, número da parcela, valor bruto da parcela. O Quadro 1 sintetiza essas informações.

Quadro 1. Objetos Obrigatórios exigidos pela *WebService* da conciliadora

<i>Campo</i>	<i>Descrição</i>
<i>lotevenda.nrVendas</i>	Feito validação comparando o tamanho da lista de vendas
<i>lotevenda.totalVendas</i>	Feito validação comparando o valor somado da lista de vendas
<i>loteVenda.vendas</i>	Lista de Vendas
<i>venda.data</i>	Data da Venda
<i>venda.valor</i>	Valor da Venda
<i>venda.nrParcelas</i>	Número de parcelas da venda. Se maior que 1, campo passa a ser obrigatório.
<i>venda.parcelas</i>	Obrigatório se a <i>venda.nrParcelas</i> > 1
<i>venda.formaCaptura</i>	Forma de captura da venda (TEF ou POS)
<i>parcela.numero</i>	Número da parcela
<i>parcela.valor</i>	Valor Bruto da Parcela

Fonte: Autoria Própria (2025)

4.2 Tratamento dos Dados

Após a coleta, o código começar a tratá-los, o mesmo cria um lote e encapsula todas as informações. A partir disso, o layout fornecido pelo sistema conciliador assume sua devida importância. O layout é basicamente um modelo que define como os dados precisam ser organizados e tratados para envio ou recebimento do lote de vendas. No lote da venda, serão enviadas todas as informações referentes a venda realizada pelo cliente, os campos mostrados no Quadro 2 apresenta o layout para tratamento dos formatos dos campos do lote da venda propriamente dita, que serão enviados no campo denominado *loteVenda.vendas* dentro do lote de venda.

No Quadro 3 é apresentado o layout para a venda e no Quadro 4, é apresentado o layout para informações das parcelas de cada venda, se houver. Nesse quadro, o símbolo * significa que são informação não são obrigatórios,

mas o layout recomenda que sejam inseridos no lote da venda pois essas informações são os campos chave para a validação correta no processo de conciliação da venda.

Quadro 2: Atributos Lote Venda Cliente exigidos pela *WebService* da conciliadora

Campo	Formato	Descrição
<i>nsuLote</i>	Número	Número único do lote de vendas
<i>codigoCliente</i>	Caractere	Código externo que identifica o cliente
<i>nrVendas</i>	Número	Número de vendas no lote
<i>totalVendas</i>	Número	Valor total de vendas deste lote
<i>Vendas</i>	Lista	Lista de Vendas

Fonte: Autoria Própria (2025)

Quadro 3: Atributos das Vendas exigidas pela *WebService* da conciliadora

Campo	Formato	Descrição
<i>nsuOrigem*</i>	Caractere	Campo chave da venda para o cliente
<i>nsuAdministradora*</i>	Número	NSU da venda na Administradora
<i>nrAutorizacao*</i>	Caractere	Número da autorização da venda
<i>redeAdquirente</i>	Número	Número da rede adquirente (responsável pela liquidação financeira da transação)
<i>data</i>	Data/Hora	Data e Hora da venda
<i>nrCartao</i>	Caractere	Número do cartão
<i>loja</i>	Caractere	Nome da Loja
<i>codigoEstabelecimento</i>	Número	Código do estabelecimento que identifica a loja na adquirente
<i>valor</i>	Número	Valor da venda
<i>nrParcelas</i>	Número	Número de parcelas da venda
<i>bandeira</i>	Número	Número da bandeira
<i>modalidade</i>	Número	1 – Débito, 2 – Crédito
<i>banco</i>	Número	Banco onde será creditado o valor para o cliente
<i>agencia</i>	Número	Agência onde será creditado o valor para o cliente
<i>conta</i>	Número	Conta onde será creditado o valor para o cliente
<i>codigoLoja</i>	Número	Código da loja no Conciliador
<i>formaCaptura</i>	Caractere	Forma de captura (T – TEF, P – POS)
<i>parcelas</i>	Lista	Lista de Parcelas da venda

Fonte: Autoria Própria (2025)

Quadro 4: Atributos das Parcelas exigidos pela *WebService* da conciliadora

Campo	Formato	Descrição
<i>nsuOrigem</i>	Caractere	Campo chave da parcela para o cliente
<i>dataCredito</i>	Data/Hora	Data de prevista de crédito
<i>status</i>	Número	Código do status da parcela
<i>numero</i>	Número	Número da parcela
<i>dataPagamento</i>	Data/Hora	Data que o crédito foi realizado
<i>dataCancelamento</i>	Data/Hora	Data que o cancelamento foi realizado
<i>valorTaxaAdministracao</i>	Número	Valor da taxa de administração
<i>percTaxaAdministracao</i>	Número	Percentual da taxa de administração
<i>valorTaxaAntecipacao</i>	Número	Valor da taxa de antecipação
<i>percTaxaAntecipacao</i>	Número	Percentual da taxa de antecipação
<i>valorAjuste</i>	Número	Valor de ajuste
<i>valorPago</i>	Número	Valor pago
<i>Valor</i>	Número	Valor da parcela
<i>chaveParcela</i>	Número	Campo que identifica unicamente uma parcela para o conciliador
<i>percTaxaAntecipacaoNominal</i>	Número	Percentual da taxa de antecipação nominal

Fonte: Autoria Própria (2025)

4.3 Envio dos Dados Através do *Endpoint*

Após a formatação dos dados, o código fez a montagem do *endpoint* para realizar o envio das informações. Um *endpoint* é um ponto de acesso específico em uma API que permite a comunicação entre sistemas. Ele funciona como uma URL (*Uniform Resource Locator*) que em português significa Localizador Uniforme de Recursos, que define onde e como um cliente pode solicitar ou enviar dados a um servidor. Cada *endpoint* está associado a uma função ou recurso específico, como buscar informações, enviar dados ou executar operações. O lote criado é basicamente um objeto literal, ou seja, um conjunto de dados estruturados em pares *chave-valor*, dados estruturados em pares chave-valor consistem em uma organização de informações onde cada dado é associado a uma chave única, que o identifica, e um valor correspondente, sendo amplamente utilizado em arquivos de configuração, APIs e bancos de dados devido à sua simplicidade e eficiência na recuperação e armazenamento de dados.

No lote temos as seguintes chaves: *nrVendas*, *totalVendas* e *vendas*, que significam, respectivamente, o número de vendas dentro da lista, o somatório dos valores das vendas e a lista de vendas que é representado pelos colchetes. Esse lote depois de preenchido é colocado dentro do *endpoint* chamado *enviarLoteVendaCliente*. Esse *endpoint* além do lote de vendas espera mais dois parâmetros que são, a versão da API do conciliador e o token de acesso do usuário. Essas informações são armazenadas em variáveis de ambiente configuradas antes da execução da aplicação. Com o *endpoint* pronto a API realiza o envio através do protocolo HTTP obtivo através da biblioteca *requests* da linguagem Python. Assim que o arquivo chega à API da conciliadora, é feito um processo de validação da integridade do arquivo. Primeiro é validado o token do usuário, para garantir que ele está cadastrado junto à conciliadora. Em seguida, é verificado se o número de vendas enviado no lote confere com o total de vendas processadas.

Por fim, o valor total é comparado com o valor total das vendas enviadas dentro do *endpoint*. Essa validação é crucial para garantir a precisão e a confiabilidade dos dados. Caso ocorra algum problema durante a validação, o arquivo não é processado, e uma mensagem de retorno é enviada com código que indica o tipo de erro. Os códigos de retorno são compostos por um formato específico: “*TIPO (id do erro)*”. O TIPO pode ser “MSG” (mensagem de sucesso), “ERR” (Erro Inesperado) ou “NGC” (Regra de negócio não atendida). Por exemplo, o código *MSG_001* indica que o lote foi recebido com sucesso, enquanto *ERR_001* indica um problema na validação do token. Esses códigos são retornados na mesma requisição HTTP para facilitar a interpretação e o tratamento dos erros pelo desenvolvedor.

Além disso, a API foi projetada para lidar com possíveis falhas de comunicação ou problemas durante o envio do arquivo. Se a requisição falhar, devido a qualquer problema, o código registra o erro em um arquivo de log. Esse log contém detalhes do lote para conferência e conta também com o código do tipo de erro e a mensagem retornada pela requisição. Como exemplo, a Figura 3 apresenta o registro do log de erro gerado pela aplicação. Essa funcionalidade é essencial para monitorar o funcionamento da aplicação e garantir que eventuais erros sejam identificados e corrigidos rapidamente.

Figura 3: Arquivo de log com retorno do *endpoint enviarLoteVendaCliente*

```
16/01/2025, 17:18:32:
{
  "dataLote": "15/01/2025",
  "NumeroVendas": 849,
  "valorTotalVendas": "58990.26",
  "codigoRequisicao": 114624469,
  "codigoRetorno": "MSG_001",
  "detalheRetorno": "Resquisicao atendida com sucesso. Número de vendas inseridas: 849",
  "excluido": "Lote Enviado com sucesso."
}
```

Fonte: Autoria Própria (2025)

Outro ponto importante é a flexibilidade da aplicação. A API foi desenvolvida para adaptar-se a diferentes necessidades. Por exemplo, se a conciliadora alterar o layout do arquivo ou adicionar novos campos, basta ajustar o código responsável pela geração do corpo do arquivo, sem a necessidade de refazer toda a aplicação. Essa flexibilidade é um dos pontos fortes do projeto pois permite que a aplicação seja facilmente atualizada ou expandida.

4.4 Importação de Dados no Sistema ERP

O processo inicia com a leitura das configurações necessárias para a requisição na API da conciliadora. Essas informações estão armazenadas no arquivo *config*, que possui extensão INI. Dentro desse arquivo, há uma chave *[data]*, onde ficam os dados essenciais para a requisição. A Figura 4 apresenta esses dados.

Figura 4: Arquivo com parâmetros para requisição de dados no webservice

```
[data]
dataPagamentoInicial =
dataPagamentoFinal =
codigoLoja =
caminhoArq =
codigoFiliais = {"2":10383,"3":10386,"4":10384,"5":10385}
```

Fonte: Autoria Própria (2025)

O usuário responsável por essa operação pode configurar as informações conforme sua necessidade. No entanto, é obrigatório definir no arquivo *config.ini* os códigos das filiais da empresa, registrados dentro da plataforma conciliadora. Esses códigos podem ser obtidos facilmente com o suporte da conciliadora. Após receber os dados, o usuário deve associar cada filial ao código correspondente fornecido pelo suporte.

Após a configuração dos códigos das filiais, as demais configurações são opcionais. A API considera a data de pagamento inicial e a data de pagamento final como dia anterior à requisição, ou seja, sempre buscará informações referentes ao dia anterior. Além disso, existem outras duas configurações disponíveis: *codigoLoja* - permite que o usuário solicite informações de uma loja específica, caso necessário; *caminhoArq* - Define o local onde a API armazenará o arquivo CSV com as informações que serão importadas pelo sistema ERP do cliente. Caso o arquivo *config.ini* não esteja previamente criado, a API identifica e cria o arquivo com valores padrão. A função *lerConfiguracoes* é responsável por essa tarefa, garantindo que a aplicação tenha todos os dados necessários para realizar a requisição à API.

Além das informações para aquisição dos dados a serem importados, o *endpoint* também espera dados para validação do usuário, versão referente a API da conciliadora e o endereço do serviço web da plataforma (URL). Todas essas informações são obrigatórias e necessárias para validação da coleta de dados na API da conciliadora. Aqui podemos destacar a validação do usuário através do token de acesso. Esse token é fornecido pela conciliadora para que o usuário possa realizar as requisições via serviço web. De posse do token, sugere-se a criação de uma variável de ambiente para que possa armazenar o token de forma segura. Por fim, na posse desses dados a aplicação realiza a requisição através da biblioteca *requests* com o método HTTP do tipo *POST* através do *endpoint consultaVendaAdm* à API. A resposta da API é então processada em formato JSON, sendo armazenada em uma estrutura de dados previamente definida para organização e manipulação posterior.

A estratégia adotada para a importação dos dados leva em consideração a limitação imposta pela API da conciliadora quanto ao número máximo de registros por requisição. Conforme especificado pela documentação do serviço, cada consulta retorna, no máximo, 100 registros, sem possibilidade de ajuste desse valor. Diante desse cenário, ao receber a resposta da requisição, o código verifica o número total de páginas a serem consultadas. Caso haja um volume superior a 100 registros, é necessário entrar em contato com o setor de desenvolvimento do conciliador para analisar a possibilidade de alteração no protocolo de consulta.

Após a coleta integral dos dados, foi implementado um processo de geração e salvamento de arquivos em formato CSV – *Comma-Separated Values*, projetado especificamente para integração com o sistema ERP da empresa. O layout desse arquivo foi estruturado com base no formato já existente no ERP, seguindo a padronização adotada para importação de arquivos de retorno financeiro, como ocorre com os boletos bancários. Além disso, a estrutura do CSV aproveita o layout de uma empresa parceira do sistema, garantindo confiabilidade e minimizando a necessidade de ajustes manuais. Para assegurar a correta importação de dados pelo ERP, todas as formatações e ordenações necessárias foram aplicadas durante a construção do arquivo CSV. O código organiza as informações seguindo a sequência exigida pelo sistema, ajustando campos com valores monetários, datas e identificadores, garantindo que o arquivo seja processado sem erros. Esse layout é estruturado em três seções principais: Header(Início), Detalhes da venda e Trailer(Fim). O Header contém informações gerais, como data e hora da criação do arquivo, a data do período de importação das transações e o número sequencial do arquivo CSV, garantindo rastreabilidade dos dados.

A seção Detalhes da venda reúne informações sobre cada transação da venda, incluindo o identificador da transação, código do estabelecimento, data da venda, número do cartão de crédito, valores brutos e líquidos, detalhes de parcelamento e dados bancários. Além disso, registra informações específicas do sistema, como códigos de autorização, NSU – Número Sequencial Único, código que identifica transações. Por fim, o Trailer encerra o arquivo CSV, marcando o término da sequência de registros. A Figura 5 ilustra um exemplo do arquivo CSV gerado, demonstrando a estrutura e formatação dos dados conforme os requisitos do ERP.

Figura 5. Arquivo CSV para importação dos dados no ERP

credito-2024-11-25-2024-11-25.csv	
1	0;20241125;;20241125;20241125;V3.6;0;215;215
2	10;;3;20240125;;3001429;651675*****3965;93030;10;90332;;20241125;20241125;10;CRÉDITO;TEF;;748;2207;289230;2698;290;;;
3	10;;2;20240130;;4004224;548514*****5381;2990;10;2930;;20241125;20241125;10;CRÉDITO;TEF;;748;2207;289230;60;201;;;
4	10;;3;20240130;;1021662;606282*****5734;10000;10;9710;;20241125;20241125;10;CRÉDITO;TEF;;748;2207;289230;290;290;;;
5	10;;4;20240130;;3025778;523421*****0707;25900;10;25385;;20241125;20241125;10;CRÉDITO;TEF;;748;2207;289230;515;199;;;
6	10;;5;20240130;;2021651;525496*****7834;4490;10;4401;;20241125;20241125;10;CRÉDITO;TEF;;748;2207;289230;89;198;;;
7	10;;3;20240227;;2022669;544315*****7606;4690;10;4597;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;93;198;;;
8	10;;3;20240227;;2022686;545368*****9789;7990;10;7831;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;159;199;;;
9	10;;5;20240227;;1018349;528860*****9335;3869;10;3792;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;77;199;;;
10	10;;4;20240228;;3027903;230650*****1016;6490;10;6361;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;129;199;;;
11	10;;7;20240229;;1005506;425850*****7318;2990;10;2930;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;60;201;;;
12	10;;2;20240229;;4005991;410863*****6569;9876;9;9679;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;197;199;;;
13	10;;2;20240229;;4005955;545368*****7268;4690;10;4597;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;93;198;;;
14	10;;2;20240229;;3005002;459314*****8837;3490;10;3421;;20241125;20241125;9;CRÉDITO;TEF;;748;2207;289230;69;198;;;
15	10;;2;20240328;;1000001;516292*****3988;33500;10;32833;;20241125;20241125;8;CRÉDITO;TEF;;748;2207;289230;667;199;;;
16	10;;4;20240328;;3030181;530033*****0364;9500;10;9311;;20241125;20241125;8;CRÉDITO;TEF;;748;2207;289230;189;199;;;
17	10;;4;20240424;;2030638;650487*****2274;4350;10;4224;;20241125;20241125;7;CRÉDITO;TEF;;748;2207;289230;126;290;;;
18	10;;4;20240427;;3032063;409281*****2996;5690;10;5577;;20241125;20241125;7;CRÉDITO;TEF;;748;2207;289230;113;199;;;
19	10;;4;20240427;;3032150;406655*****6865;10990;10;10771;;20241125;20241125;7;CRÉDITO;TEF;;748;2207;289230;219;199;;;
20	10;;2;20240429;;4009047;548721*****4532;14680;10;14388;;20241125;20241125;7;CRÉDITO;TEF;;748;2207;289230;292;199;;;
21	10;;3;20240429;;694;545368*****2324;60000;10;58806;;20241125;20241125;7;CRÉDITO;POS;;748;2207;289230;1194;199;;;
22	10;;3;20240429;;696;545368*****2324;5000;10;4900;;20241125;20241125;7;CRÉDITO;POS;;748;2207;289230;100;200;;;
23	10;;4;20240429;;3032191;550209*****4732;3890;10;3813;;20241125;20241125;7;CRÉDITO;TEF;;748;2207;289230;77;198;;;
24	10;;4;20240429;;3032220;230888*****7964;13290;10;13026;;20241125;20241125;7;CRÉDITO;TEF;;748;2207;289230;264;199;;;

Fonte: Autoria Própria (2025)

5. CONCLUSÃO

Este estudo teve como objetivo desenvolver uma interface de programação de aplicação (API) para integrar o sistema ERP de uma empresa varejista/atacadista com o sistema conciliador de cartões de crédito e débito, visando automatizar a exportação de dados de vendas e a importação de informações de pagamentos processados pela conciliadora. A proposta da API buscou eliminar a necessidade de conferências manuais, reduzir a incidência de erros e garantir maior segurança, conformidade e precisão nos registros financeiros. Os resultados obtidos confirmam que a implementação da API atendeu plenamente aos objetivos propostos, demonstrando sua eficácia na otimização da gestão operacional e financeira.

A API desenvolvida proporcionou uma melhoria significativa no processo de baixa de títulos no contas a receber do ERP, além de reduzir drasticamente os erros cometidos pelos usuários do setor financeiro. Outro resultado relevante foi o controle mais preciso sobre as taxas cobradas pela administradora de cartões, permitindo uma análise detalhada dos custos envolvidos. Esses resultados evidenciam que a solução não apenas agilizou a conciliação financeira, mas também fortaleceu a segurança e a conformidade dos processos, confirmando as hipóteses iniciais do estudo.

Além dos benefícios operacionais e financeiros, a implementação da API trouxe impactos positivos para a competitividade da empresa. Ao proporcionar maior controle e confiabilidade nos processos, a solução permitiu que a organização se destacasse no mercado, oferecendo uma gestão mais transparente e eficiente. Essa melhoria na gestão interna reflete diretamente na experiência do cliente, fortalecendo a reputação da empresa e sua capacidade de adaptação às demandas do setor.

Para trabalhos futuros, recomenda-se a expansão da API para integrar outros sistemas financeiros e operacionais, ampliando ainda mais a automação e a eficiência dos processos. Além disso, sugere-se a incorporação de funcionalidades de análise preditiva e relatórios personalizados, que possam auxiliar na tomada de decisões estratégicas. Por fim, estudos adicionais podem ser conduzidos para avaliar o impacto da API em diferentes setores e escalas de negócios, validando sua aplicabilidade e benefícios em contextos variados. A adoção de soluções tecnológicas inovadoras, como a API desenvolvida neste estudo, representa um diferencial estratégico para empresas que buscam eficiência, segurança e sustentabilidade em suas operações.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] OGUNDIPE, Tolulope; EWIM, Somto Emmanuel; SAM-BULYA, Ngodoo Joy. Enhancing financial reporting and management efficiency through enterprise resource planning (ERP) systems: a theoretical review for large-scale energy operations. *International Journal of Management & Entrepreneurship Research*, v. 6, n. 10, p. 3415-3458, out. 2024. DOI: 10.51594/ijmer.v6i10.1655.
- [2] WULAN, Titis Sri; NOVIKA, Putri Wahyu; NURVIANTI, Elmi; PUTRA, Feby Arma. Impact of ERP System Implementation on Operational and Financial Efficiency in Manufacturing Industry. *Journal of Economic Education and Entrepreneurship Studies*, [S. l.], v. 5, n. 3, p. 491–501, 2024. DOI: 10.62794/je3s.v5i3.4328.
- [3] IRWAN, Fakhira; HAANURAT, A. Ifayani; JAYA, Asri. Analysis of Financial Performance in Cosmetics Industry Companies Listed on the Indonesian Stock Exchange. *International Journal of Economic Research and Financial Accounting*, [S. l.], v. 3, n. 1, 2024. DOI: 10.55227/ijerfa.v3i1.218.
- [4] DAMA, Hais; SELVI; HAIRUN, Sri Vany. Receivables turnover, cash turnover, and sales growth against liquidity in cosmetics sector companies listed on the Indonesia Stock Exchange. *Journal of Economics, Finance*

and Management Studies, v. 7, n. 7, p. 4095-4105, jul. 2024. DOI: 10.47191/jefms/v7-i7-27.

- [5] KARAMANI, Brunela. Automation of data validation processes in banking: a case study on validating Albanian identity numbers. *SEEU Review*, v. 19, n. 1, 2024. DOI: 10.2478/seeur-2024-0022.
- [6] ALAO, Olakunle Babatunde; DUDU, Oritsematosan Faith; ALONGE, Enoch O.; EZE, Chukwuka Emmanuel. Automation in financial reporting: a conceptual framework for efficiency and accuracy in U.S. corporations. *Global Journal of Advanced Research and Reviews*, v. 2, n. 2, p. 40-50, 2024. DOI: 10.58175/gjarr.2024.2.2.0057.
- [7] ADIREDDY, Santosh Nikhil Kumar. Idempotency and reconciliation in payment software. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, v. 12, n. 4, 2024. DOI: 10.22214/ijraset.2024.60774.
- [8] ADELEKE, Adams Gbolahan; SANYAOLU, Temitope Oluwafunmike; EFUNNIYI, Christianah Pelumi; AKWAWA, Lucy Anthony; AZUBUKO, Chidimma Francisca. API integration in FinTech: challenges and best practices. *Finance & Accounting Research Journal*, v. 6, n. 8, p. 1531-1554, ago. 2024. DOI: 10.51594/farj.v6i8.1506.
- [9] BARNAL, Laura-Eugenia-Lavinia. ERP Systems - Reliable Tools in Corporate Reporting of Organizations. *Audit Financiar*, v. XXII, n. 4(176), p. 783-790, 2024. DOI: 10.20869/AUDITE/2024/176/028.
- [10] SRI, M. Teja; SURESH, Nunna; VARALAKSHMI, T. Integration of ERP in Optimizing Business Process. *International Research Journal on Advanced Engineering and Management*, v. 2, n. 5, p. 1584-1587, maio 2024. DOI: 10.47392/IRJAEM.2024.0216.
- [11] SLACK, Nigel; BRANDON-JONES, Alistair; JOHNSTON, Robert. Administração da produção. 8. ed. Tradução de Daniel Vieira. São Paulo: Atlas, 2018. ISBN 978-85-97-01537-9.
- [12] SANTOS, E. R. A relevância dos sistemas Enterprise Resource Planning (ERP) para a análise de negócios em uma empresa ou organização. *Revista Tecnia*, [S. l.], v. 9, n. 1, p. 13, 2024. DOI: <<https://periodicos.ifg.edu.br/tecnia/article/view/950>>.
- [13] WEDHA, Bayu Yasa; HINDARTO, Djarot. Maximizing ERP Benefits with Enterprise Architecture: A Holistic Approach. *Journal of Computer Networks, Architecture and High Performance Computing*, [S. l.], v. 5, n. 2, p. 703-713, 2023. DOI: 10.47709/cnahpc.v5i2.2790.
- [14] ALHAYEK, Wiam Yahea; ODEH, Rasha A. Abu. Exploring the impact of ERP systems on business efficiency. *Benha Journal of Applied Sciences (BJAS)*, v. 9, n. 9, p. 175-177, 2024. DOI: 10.21608/bjas.2024.318158.1491.
- [15] POKALA, Poornachandar. The integration and impact of artificial intelligence in modern enterprise resource planning systems: a comprehensive review. *International Journal of Computer Engineering and Technology (IJCET)*, v. 15, n. 6, p. 79-88, nov.-dez. 2024. DOI: 10.5281/zenodo.14050064.
- [16] BAGGA, Jaspreet. Introduction to Integration: A Practical Guide to SAP Integration Suite: SAP's Cloud Middleware and Integration Solution. *Berkeley*, p. 1-32, 2023. DOI: 10.1007/978-1-4842-9337-9_1.
- [17] FEHRATBEGOVIĆ, Adnan; KOVAČEVIĆ, Selma. Integration of Software Systems in the Corporate Environment. *B&H Electrical Engineering*, v. 18, n. 2, 2024. DOI: 10.2478/bhee-2024-0010.
- [18] PRAYOGA, Muhamad Rico Aditya; SUHIRMAN. Integrated payment system to improve financial management using payment gateway. *Journal Inovtek Polbeng - Seri Informatika*, v. 9, n. 2, p. 727, 2024. DOI: 10.35314/yjtjrn52.
- [19] HARIANTO, Kevin Joshua; TARIGAN, Zeplin Jiwa Husada; SIAGIAN, Hotlan; BASANA, Sautma Ronni; JIE, Ferry. The effect of digital ERP implementation, supply chain integration and supply chain flexibility on business performance. *International Journal of Data and Network Science*, 2024. DOI: 10.5267/j.ijdns.2024.5.017.
- [20] EFUNTADE, Olubunmi Omotayo; EFUNTADE, Alani Olusegun. Application Programming Interface (API) and management of web-based accounting information system (AIS): security of transaction processing system, general ledger and financial reporting system. *Journal of Accounting and Financial Management*, v. 9, n. 6, p. 1-18, 2023. DOI: 10.56201/jafm.v9.no6.2023.pg1.18.
- [21] KIPILIMBA, Theobald Francis. Financial Reporting Revolution: How it Integration Drives Efficiency and Accuracy. *Archives of Current Research International*, [S. l.], v. 24, n. 6, p. 534-557, 2024. DOI: 10.9734/acri/2024/v24i6811.
- [22] ZHAO, Chenchen. API Common Security Threats and Security Protection Strategies. *Frontiers in Computing and Intelligent Systems*, [S. l.], v. 10, n. 2, p. 29-33, 2024. DOI: 10.54097/k5djs164.
- [23] OSUNDARE, Olajide Soji; IKE, Chidiebere Somadina; FAKEYEDE, Ololade Gilbert; IGE, Adebimpe Bolatito. Secure communication protocols for real-time interbank settlements. *Computer Science & IT Research Journal*, v. 4, n. 3, p. 436-457, 2023. DOI: 10.51594/csitj.v4i3.1498.
- [24] OLAOLUWA, Favour; POTTER, Kaledio; IoT security risk assessment and mitigation. *Preprints*, 2024. DOI: 10.20944/preprints202409.0468.v1.
- [25] FEHRATBEGOVIĆ, Adnan; KOVAČEVIĆ, Selma. Integration of software systems in the corporate environment. *B&H Electrical Engineering*, v. 18, n. 2, p. 44-49, 2024. DOI: 10.2478/bhee-2024-0010.
- [26] ISARAPU, Rohini; GOWDA, Sharanthchandra. APIs in cloud data services: a technical overview. *International Journal of Computer Engineering and Technology (IJCET)*, v. 15, n. 6, p. 1853-1864, 2024. DOI: 10.34218/IJCET_15_06_158.

- [27] MUDASSIR, Mohammed; MUSHTAQ, Mohammed. The role of APIs in modern software development. *World Journal of Advanced Engineering Technology and Sciences*, v. 13, n. 1, p. 1045–1047, 2024. DOI: 10.30574/wjaets.2024.13.1.0515.
- [28] PASUNURI, Kranthi Kumar. Digital Workflow Automation in Telecommunications: a systems integration approach to operational excellence. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, v. 11, n. 1, p. 20-28, jan./fev. 2025. ISSN 2456-3307. DOI: 10.32628/CSEIT2511120520.
- [29] UPPUTURI, Vamsi. Automation and Data Integrity in Regulatory Submissions: Innovations for Decentralized Clinical Trials. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, v. 11, n. 1, 2025. DOI: 10.32628/CSEIT251112105.
- [30] ASRI, Marselinus. The Effect of Accounting Earnings and Company Size on Abnormal Stock Returns in Food and Beverage Sub-Sector Manufacturing Companies. *Atestasi : Jurnal Ilmiah Akuntansi*, [S. l.], v. 7, n. 1, p. 687–703, 2024. DOI: 10.57178/atestasi.v7i1.864.
- [31] LAWALATA, Josina; SALLE, Ilham Z.; YULIANA, Leny. The Impact of International Financial Reporting Standards on Global Accounting Practices. *Advances in Applied Accounting Research*, [S. l.], v. 2, n. 2, p. 83–93, 2024. DOI: 10.60079/aaar.v2i2.262.
- [32] BAKARE, Oluwaseun Adeola; ACHUMIE, Godwin Ozoemenam; OKEKE, Njideka Ihuoma. Revolutionizing financial inclusion through strategic API integration and innovation. *Finance & Accounting Research Journal*, v. 6, n. 10, p. 1832-1860, out. 2024. DOI: 10.51594/farj.v6i10.1624.
- [33] SHARMA, Anush; CHOUDHARY, Ankit; HIMANSHU; CHAUDHARY, Aman. Exploring Python: a comprehensive guide for data science, machine learning, and IoT. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, v. 8, n. 10, out. 2024. ISSN 2582-3930. DOI: 10.55041/IJSREM37816.
- [34] PITRODA, Piyush D.; DONGA, Bhavika C.; DOMADIYA, Hasmukh B.; DOMADIYA, Dipti H. Beyond the basics: a detailed survey of advanced Python applications and innovations. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, v. 12, n. X, out. 2024. ISSN 2321-9653. DOI: 10.22214/ijraset.2024.64457.
- [35] SRIVASTAVA, Ankit. Use of Python in data science, data integration and data engineer. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, v. 8, n. 7, p. 1–X, July 2024. ISSN 2582-3930. DOI: 10.55041/IJSREM36294.
- [36] SANTOS, Mario Martinelli dos. Streamlining ERP Implementations: A Consultant's Guide to Success. *International Journal of Scientific Research and Management (IJSRM)*, v. 11, n. 01, p. 4516–4524, jan. 2023. ISSN 2321-3418. DOI: 10.18535/ijsrc/v11i01.em12.
- [37] MUNOZ, Justin; JALILI, Mahdi; TAFAKORI, Laleh. Automating Bank Reconciliation Through Graph Representation Learning and Downstream Record Linkage. *Royal Melbourne Institute of Technology (RMIT University)*, 2024. Disponível em: SSRN. DOI: 10.2139/ssrn.4698441.
- [38] DEVAVARAPU, Yaswanth; BEDADHALA, Raviteja Reddy; SHAIK, Shoaib Sohail; PENDELA, Charan Ram Kumar; ASHESH, K. Credit Card Fraud Detection Using Outlier Analysis and Detection. *International Conference on Intelligent Technologies (CONIT)*, Bangalore, India. IEEE, 2024. DOI: 10.1109/CONIT61985.2024.10626480.
- [39] CHEW, Peter. Unsupervised-learning financial reconciliation: a robust, accurate approach inspired by machine translation. *Proceedings of the First ACM International Conference on AI in Finance*, 2021, Association for Computing Machinery. DOI: 10.1145/3383455.3422517.
- [40] XIA, Shumin. Application of Python Automation Tool Combined with OCR Technology in "Financial Checkup" of Universities. *Accounting and Corporate Management*, v. 5, p. 130–136, 2023. DOI: 10.23977/acccm.2023.051220.
- [41] XU, Yitong. Um estudo sobre a aplicação de Python na análise financeira corporativa. *Frontiers in Business, Economics and Management*, [S. l.], v. 15, n. 2, p. 31–36, 2024. DOI: 10.54097/rpvn638.
- [42] ZOU, Patrick X. W.; XU, Xiaoxiao. Case study research: Methodology and Strategy. *John Wiley & Sons, Ltd*, 2023. p. 97-115. DOI: 10.1002/9781394190256.ch5.