INSTITUTO FEDERAL GOIANO - CAMPUS MORRINHOS

CURSO SUPERIOR DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PEDRO HENRIQUE MARCELINO SILVA

Integridade de Dados no Contexto de um Sistema para Gestão de Cobrança

MORRINHOS - GO
2025

PEDRO HENRIQUE MARCELINO SILVA

Integridade de Dados no Contexto de um Sistema para Gestão de Cobrança

Monografia apresentada ao Curso Superior de Bacharelado em Ciência da Computação do Instituto Federal Goiano – Campus Morrinhos, como requisito parcial para obtenção de título de Bacharel em Ciência da Computação.

Área de concentração: Engenharia de Software.

Orientador: Rodrigo Elias Francisco

MORRINHOS - GO
2025

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas – SIBI/IF Goiano Campus Morrinhos

S586i Silva, Pedro Henrique Marcelino.
Integridade de dados no contexto de um sistema para gestão de cobrança. / Pedro Henrique Marcelino Silva. – Morrinhos, GO: IF Goiano, 2025.
40 f. : il.

Orientador: Dr. Rodrigo Elias Francisco.

Trabalho de conclusão de curso (graduação) – Instituto Federal Goiano Campus Morrinhos, Bacharelado em Ciências da Computação, 2025.

1. Gestão de cobrança. 2. Integridade de Dados. 3. Criptografia AES. I. Francisco, Rodrigo Elias. II. Instituto Federal Goiano. III. Título.

CDU 004.6

Fonte: Elaborado pela Bibliotecária-documentalista Morgana Guimarães, CRB1/2837

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- ☐ Tese (doutorado)
- ☐ Dissertação (mestrado)
- ☐ Monografia (especialização)
- ☑ TCC (graduação)
- ☐ Artigo científico
- ☐ Capítulo de livro
- ☐ Livro
- ☐ Trabalho apresentado em evento
- ☐ Produto técnico e educacional - Tipo:

Nome completo do autor: Pedro Henrique Marcelino Silva

Matrícula: 2019104201940254

Título do trabalho: Integridade de Dados no Contexto de um Sistema para Gestão de Cobrança

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: ☑ Não ☐ Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: 05 / 09 / 2025

O documento está sujeito a registro de patente? ☐ Sim ☑ Não

O documento pode vir a ser publicado como livro? ☐ Sim ☑ Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

• Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;

• Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;

• Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Morrinhos - GO (Local) — 05 / 09 / 2025 (Data)

gov.br Documento assinado digitalmente
PEDRO HENRIQUE MARCELINO SILVA
Data: 05/09/2025 18:15:03-0300
Verifique em https://validar.iti.gov.br

_____
Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

gov.br Documento assinado digitalmente
RODRIGO ELIAS FRANCISCO
Data: 06/09/2025 23:05:54-0300
Verifique em https://validar.iti.gov.br

_____
Assinatura do(a) orientador(a)

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 19/2025 - CCEPTNM-MO/CEPTNM-MO/DE-MO/CMPMHOS/IFGOIANO

ATA DE DEFESA DE TRABALHO DE CURSO

Ao dia **1º** do mês de **setembro** d e **2025**, às **16:00** horas, foi realizada a Banca de Exame, nas dependências do Instituto Federal Goiano – campus Morrinhos-GO, para a apresentação pública e defesa do trabalho de curso do discente **Pedro Henrique Marcelino Silva** intitulado **Integridade de Dados no Contexto de um Sistema para Gestão de Cobrança,** como requisito necessário para a conclusão do curso.

A Banca de Exame foi constituída pelos membros: **Rodrigo Elias Francisco**, **Hiury Luiz dos Santos e Alline Rodrigues Bento**. Após a análise, emitiram o seguinte resultado:

1 - (x) Aprovado

2 - ( ) Aprovado com ressalva

(A Banca Examinadora deve definir as exigências a serem cumpridas pelo aluno na revisão, ficando o orientador responsável pela verificação do cumprimento das mesmas.)

Observações:_____

3 - ( ) Reprovado com o seguinte parecer: _____
_____

Morrinhos-GO, 1 de setembro de 2025.

_____

(Assinado Eletronicamente)

Rodrigo Elias Francisco (Presidente da banca)

_____

(Assinado Eletronicamente)

Hiury Luiz dos Santos (Membro)

_____

(Assinado Eletronicamente)

Alline Rodrigues Bento (Membro)

Documento assinado eletronicamente por:

- **Rodrigo Elias Francisco**, **PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 01/09/2025 17:44:07.
- **Hiury Luiz dos Santos**, **PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 01/09/2025 18:31:34.
- **Alline Rodrigues Bento**, **COORDENADOR(A) - FG0001 - NAPNE-MO** , em 02/09/2025 10:45:40.

Este documento foi emitido pelo SUAP em 01/09/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifgoiano.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 739033
Código de Autenticação: 236e3717b5

INSTITUTO FEDERAL GOIANO
Campus Morrinhos
Rodovia BR-153, Km 633, Zona Rural, SN, Zona Rural, MORRINHOS / GO, CEP 75650-000
(64) 3413-7900

DEDICATÓRIA

É com imensa honra que dedico este trabalho a todos que me apoiaram nessa caminhada. Em especial, aos meus pais, que sempre acreditaram no meu potencial, nunca desistiram de mim e estiveram incansavelmente buscando soluções para me ajudar. Ao meu primo, que me incentivou nos momentos mais desafiadores com dicas valiosas, e ao meu irmão, sempre pronto com conselhos sobre como seguir em frente. À minha cachorrinha Cherry, que, mesmo sem participar ativamente, esteve ao meu lado com sua companhia especial, me trazendo motivação nos dias difíceis. Ao meu orientador, professor, que, apesar das dificuldades iniciais e das dúvidas quanto ao tema, foi fundamental para que este trabalho tomasse forma. Agradeço pela paciência, apoio e orientação. Reconheço minhas limitações, mas também a importância da persistência e do apoio familiar pois mesmo com obstáculos, há sempre caminhos possíveis para se alcançar um objetivo, por mais adaptável que ele precise ser.

AGRADECIMENTOS

É com o coração cheio de gratidão que escrevo estas palavras. Agradeço, primeiramente, aos meus pais, que foram minha base em todos os momentos acreditaram em mim mesmo quando eu duvidei, nunca mediram esforços para me apoiar e sempre buscaram soluções quando tudo parecia difícil. Ao meu primo, que nunca deixou de me incentivar, mesmo nas fases mais complicadas, oferecendo palavras de encorajamento e dicas que fizeram diferença. Ao meu irmão, por seus conselhos sinceros, por estar sempre ao meu lado com sabedoria e carinho. Ao meu orientador, professor, que enfrentou comigo as incertezas dos primeiros passos, quando tantos temas foram tentados e nada parecia se encaixar obrigado por sua paciência e orientação até que tudo finalmente se acertasse. Agradeço profundamente também aos psicólogos do instituto IF Goiano, que se dedicaram com tanta empatia e cuidado para que eu tivesse condições de chegar até aqui. Reconheço minhas dificuldades, minhas limitações e reconheço, com ainda mais força, que não estou sozinho. A vida é feita de lutas, mas também de mãos estendidas, e foi com elas que cheguei até este momento. Mesmo quando tudo exige adaptação, ainda assim é possível conquistar. E eu conquistei.

RESUMO

O presente trabalho tem como objetivo analisar a importância da integridade de dados no contexto de um sistema para gestão de cobrança, destacando os desafios e soluções adotadas para assegurar a consistência, segurança e confiabilidade das informações. O estudo foi conduzido por meio da implementação prática de um sistema em ambiente controlado no Google Colab, utilizando criptografia simétrica com o algoritmo AES para proteger dados sensíveis. Foram elaborados diagramas de caso de uso e entidade-relacionamento (DER), além da realização de testes de desempenho envolvendo operações de criptografia e descriptografia com diferentes volumes de iterações. Os resultados demonstraram que o tempo de execução do algoritmo apresenta crescimento linear conforme o aumento do número de operações, mantendo desempenho eficiente e simétrico. Também foram identificadas dificuldades como o tratamento de transações concorrentes, a validação em tempo real e a integração com APIs externas de pagamento. Conclui-se que a aplicação de criptografia associada a boas práticas de engenharia de software é essencial para garantir a integridade dos dados em sistemas voltados à gestão financeira, contribuindo para a segurança, conformidade com normas como a LGPD e a confiança do usuário no sistema.

Palavras-chave: Integridade de Dados; Criptografia AES; Gestão de Cobrança.

ABSTRACT

This work aims to analyze the importance of data integrity in the development in context of a system for billing management, highlighting the challenges and solutions adopted to ensure data consistency, security, and reliability. The study was conducted through the practical implementation of a system in a controlled environment using Google Colab, employing symmetric encryption with the AES algorithm to protect sensitive data. Use case and entity-relationship diagrams (ERD) were developed, and performance tests were carried out involving encryption and decryption operations with different iteration volumes. Results showed that the algorithm's execution time grows linearly with the number of operations, maintaining efficient and symmetrical performance. Challenges such as handling concurrent transactions, real-time validation, and integration with external payment APIs were also identified. It is concluded that applying encryption alongside good software engineering practices is essential to guarantee data integrity in financial management systems, contributing to security, compliance with regulations such as LGPD, and user trust in the system.

Keywords: Data Integrity; AES Encryption; Billing Management.

LISTA DE ILUSTRAÇÕES

LISTAS DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS

ACID - *Atomicidade Consistência Isolamento e Durabilidade*

AES - *Advanced Encryption Standard*

AP - *Average Precision*

CP - *Cartão Presente*

CNP - *Cartão Não Presente*

CTR - *Counter Mode*

DER - *Diagrama Entidade Relacionamento*

DES - *Data Encryption Standard*

EAX - *Criptografar Autenticar e Traduzir*

GC - *Gestão do Conhecimento*

IV - *Vetor de Inicialização*

KNN - *K-Nearest Neighbors*

LGPD - *Lei Geral de Proteção de Dados*

NIST - *National Institute of Standards and Technology*

RF - *Random Forest*

RL - *Regressão Logística*

RSA - *Rivest Shamir Adleman*

SAD - *Sistemas de Apoio à Decisão*

SADG - *Sistema de Apoio à Decisão em Grupo*

SAE - *Sistema de Apoio Executivo*

SGBD - *Sistema de Gerenciamento de Banco de Dados*

SHA - *Secure Hash Algorithm*

SGSI - *Sistema de Gestão de Segurança da Informação*

SSL/TLS - *Camada de Conexão Segura/Segurança da Camada de Transporte*

VPNs - *Redes Privadas Virtuais*

SUMÁRIO

1 - INTRODUÇÃO

Segundo Oliveira et al. (2022) a constante evolução da gestão nas organizações evidencia a importância dos Sistemas de Informação no processo decisório em todos os níveis hierárquicos, não se restringindo mais apenas à alta direção. Com o avanço tecnológico, torna-se essencial que empresas adotem ferramentas capazes de transformar dados em conhecimento útil para apoiar decisões mais eficazes, ágeis e precisas. Nesse cenário, os Sistemas de Apoio à Decisão (SAD) ganham destaque por auxiliarem na resolução de problemas estruturados, semi-estruturados e não estruturados, promovendo maior autonomia decisória inclusive para os níveis operacionais. Recursos como *Data Warehouses*, ferramentas OLAP e interfaces amigáveis possibilitam análises profundas e interpretação de padrões, contribuindo para a qualidade das decisões estratégicas e táticas. Além disso, soluções como o Sistema de Apoio Executivo (SAE), o Sistema de Apoio à Decisão em Grupo (SADG) e os sistemas baseados na web ampliam a colaboração e o acesso em tempo real às informações relevantes. Tecnologias mais avançadas, como redes neurais, agregam valor ao processo ao simular padrões de raciocínio humano e promover auto-aperfeiçoamento contínuo. Diante disso, o uso adequado desses sistemas, aliado à Gestão do Conhecimento (GC), favorece a redução de custos, o aumento da produtividade e a melhoria no desempenho organizacional, refletindo diretamente na competitividade das empresas.

Segundo Silveira (2022) a Segurança da Informação é um fator essencial no contexto empresarial, especialmente diante das exigências legais impostas pela Lei Geral de Proteção de Dados (LGPD), a qual requer que organizações adotem mecanismos que assegurem a integridade, confidencialidade e disponibilidade dos dados armazenados. Nesse cenário, destaca-se a importância de um Sistema de Gestão de Segurança da Informação (SGSI), conforme definido pela norma ISO/IEC 27001, que especifica requisitos para implementar e monitorar práticas seguras. Complementarmente, a norma ISO/IEC 27002 fornece diretrizes técnicas e operacionais para a aplicação de controles, sendo ambas fundamentais para a conformidade legal. Este trabalho propõe a análise e integração de controles dessas normas com parâmetros de segurança oferecidos pelo Sistema de Gerenciamento de Banco de Dados (SGBD) Oracle, visando criar um modelo de adequação voltado à LGPD. A partir dessa associação, foi desenvolvido o sistema LGPD DBSec, em linguagem C#.NET, com objetivo de automatizar a verificação e manutenção de parâmetros de segurança em bancos de dados que lidam com dados pessoais. A ferramenta, testada em ambiente de homologação, demonstrou viabilidade prática sem impactos negativos ao desempenho, representando um apoio eficaz na conformidade com a LGPD, além de fornecer subsídios para futuras pesquisas na ampliação da segurança em demais componentes da infraestrutura organizacional.

Conforme Santos et al. (2024) a detecção de fraudes em processos de cobrança em transações com cartões de crédito representa um desafio constante na área de Ciência da Computação, exigindo o desenvolvimento de modelos robustos de

aprendizado de máquina para lidar com a complexidade intrínseca desses dados e com a escassez de bases públicas realistas. Diante desse cenário, o presente trabalho visa sistematizar a construção de modelos de detecção de fraudes por meio do aprimoramento de um simulador de dados sintéticos, projetado para refletir a complexidade dos padrões reais de transações. O simulador proposto incorpora atributos relevantes, como o tipo de transação Cartão Presente (CP) e Cartão Não Presente (CNP) e variações no esquema de localização, elementos fundamentais na caracterização de comportamentos fraudulentos. A avaliação dos modelos foi realizada com técnicas de classificação Random Forest (RF), K-Nearest Neighbors (KNN) e Regressão Logística (RL) e de detecção de anomalias (Isolation Forest e Elliptic Envelope), utilizando validação *prequential*, a qual respeita a ordenação temporal dos dados. As métricas de desempenho adotadas como F1-Score, AUC-ROC e Average Precision (AP) evidenciaram a superioridade do modelo Random Forest, que obteve alta precisão e baixa taxa de falsos positivos na detecção de fraudes. Conclui-se que a evolução contínua do simulador, aliada a técnicas avançadas de clusterização e detecção contextual de anomalias, pode aprimorar significativamente a eficácia dos modelos em cenários reais.

De acordo com Souza (2022), a crescente exposição das organizações a ameaças físicas e digitais tem impulsionado investimentos em medidas de segurança que visam proteger seus ativos informacionais. A Segurança da Informação, busca garantir a confidencialidade, integridade, disponibilidade, autenticidade e o não repúdio dos dados, protegendo-os contra acessos não autorizados, alterações indevidas ou indisponibilidade. Para isso, são adotadas práticas de segurança física, como controle de acesso em ambientes críticos como Data Centers ou centro de processamento de dados, monitoramento de câmeras e sistemas contra incêndio e segurança lógica, com uso de antivírus, firewalls, backup e sistemas de autenticação. A engenharia social desponta como uma das principais ameaças, explorando a fragilidade humana por meio de manipulação e persuasão, driblando mecanismos tecnológicos e colocando em risco informações sensíveis. A implementação de políticas de segurança da informação, aliada à conscientização e treinamento contínuo dos funcionários, torna-se essencial para mitigar vulnerabilidades e preservar os dados da organização. Além disso, a criptografia se destaca como ferramenta essencial para assegurar a integridade dos dados e prevenir prejuízos operacionais e financeiros decorrentes de alterações não autorizadas.

1.1 - JUSTIFICATIVA

De acordo com Feliciano & Feitosa (2024) , a gestão eficaz de informações é fundamental para garantir a integridade dos dados e a confiabilidade dos sistemas utilizados por organizações de diversos setores, como o comércio. A adoção de práticas que assegurem a precisão e a consistência dos registros contribui para decisões estratégicas e operacionais mais embasadas, além de reduzir os riscos decorrentes de inconsistências e falhas na atualização dos dados. Essa abordagem permite que as empresas mantenham um controle mais acurado sobre suas operações, promovendo maior transparência e eficiência na administração dos processos.

Segundo Duggineni (2023) a implementação de soluções tecnológicas voltadas para a automatização e integração dos processos administrativos se torna essencial nesse cenário. Ao substituir métodos manuais por sistemas que centralizam as informações, é possível minimizar erros e agilizar o fluxo de dados, garantindo que cada transação seja registrada de forma precisa e em tempo real. Essa modernização não só melhora a eficiência operacional, mas também fortalece as medidas de segurança, protegendo os dados contra acessos não autorizados e alterações indevidas.

Dessa forma, a relevância do tema se evidencia na busca contínua por estratégias que promovam a governança de dados e a proteção das informações ao longo de seu ciclo de vida. Investir em mecanismos que assegurem a integridade e a segurança dos dados é um passo decisivo para a sustentabilidade dos negócios, contribuindo para um ambiente digital mais confiável e resiliente. Assim, o desenvolvimento de soluções integradas não apenas melhora a gestão financeira e administrativa, mas também fortalece a reputação das organizações perante seus clientes e parceiros.

1.2 - QUESTÕES DE PESQUISA

Este trabalho tem o objetivo ilustrar como os conceitos de integridade de dados, criptografia e auditoria podem contribuir para empresas do setor de vendas de uma maneira didática. A intenção é que o estudo de caso desta monografia seja usada no meio educacional como recurso didático digital para a disciplina de Segurança da Informação a partir de uma perspectiva prática e pragmática.

QP1 - Como as empresas que trabalham com vendas podem proteger a integridade de dados de cobrança?

QP2 - Como o mecanismo de integridade de dados para o setor de cobranças em empresas que trabalham com vendas pode contribuir para a realização de auditorias nos dados?

1.3 - OBJETIVOS

Este trabalho apresenta um objetivo geral e objetivos específicos.

Objetivo geral: Um mecanismo para garantir a integridade de um banco de dados no de um sistema voltado à gestão de cobrança em empresas que atuam com vendas por meio da criptografia capaz de proporcionar a realização de auditorias.

Os objetivos específicos são:

1. Estudo dos principais riscos relacionados à perda de integridade de dados no contexto de sistemas de cobrança de vendas.

2. Identificação de práticas e mecanismos que promovem a integridade de dados em sistemas aplicados ao setor de vendas.

3. Avaliação da contribuição dos mecanismos de integridade de dados para a realização eficaz de auditorias internas no processo de cobrança.

1.4 - METODOLOGIA

Este trabalho utiliza a metodologia de pesquisa exploratória. Wazlawick (2010) apresenta que a pesquisa exploratória visa examinar um conjunto de fenômenos de modo a buscar a compreensão de anomalias não examinadas até então para gerar um entendimento que será a base para pesquisas futuras.

As etapas da metodologia são:

1. Realizar através da revisão bibliográfica os conceitos fundamentais.
2. Propor um mecanismo de integridade de dados para o contexto de cobrança em vendas.
3. Avaliar o mecanismo de integridade de dados.
4. Analisar como o mecanismo de integridade de dados pode contribuir para auditorias internas.

2 - REVISÃO BIBLIOGRÁFICA

2.1 - CONCEITOS FUNDAMENTAIS

2.1.1 - Advanced Encryption Standard (AES)

O Advanced Encryption Standard (AES) é um algoritmo de criptografia por blocos definido como padrão de segurança nos Estados Unidos após a limitação do Data Encryption Standard (DES), que foi substituído após concurso promovido em 1998, resultando na escolha do algoritmo Rijndael como base do AES (Trevisan et al., 2013). O AES opera com blocos de 128 bits e permite o uso de chaves de 128, 192 ou 256 bits. Apesar do Rijndael original permitir tamanhos variáveis de bloco e chave, o AES padroniza esses tamanhos fixos.

O funcionamento do AES é baseado em uma sequência de transformações matemáticas aplicadas ao bloco de dados, incluindo SubBytes, ShiftRows, MixColumns e AddRoundKey. Essas operações são responsáveis pela substituição de bytes, rotação de linhas, mistura de colunas por polinômios irredutíveis e combinação com chaves expandidas, respectivamente (Trevisan et al., 2013). O processo de decifragem realiza as mesmas etapas de forma inversa, utilizando transformações como InvSubBytes, InvShiftRows, InvMixColumns e AddRoundKey.

A implementação do AES requer o uso de operações matemáticas sobre bytes representados como polinômios. A soma é realizada via operação XOR, enquanto a multiplicação exige a aplicação de redução modular com base em polinômios irredutíveis, especialmente quando os valores ultrapassam o limite de 255 (Trevisan et al., 2013). Técnicas como a operação xTime são utilizadas para simplificar multiplicações, combinando rotações à esquerda e operações XOR com o valor 0x1B.

Para a implementação prática, optou-se pela linguagem Java, visando portabilidade para dispositivos móveis. Devido à ausência de suporte nativo para bytes sem sinal em Java, foi utilizado o tipo short para representar os dados internos do algoritmo. O projeto foi desenvolvido com a JDK 1.6 update 20 e a IDE NetBeans 6.8, sendo estruturado em pacotes e classes específicas para organização das funções de cifragem, decifragem e expansão de chaves.

O pacote principal *aescore* contém classes fundamentais como Cipher, que implementa as transformações SubBytes, ShiftRows, MixColumns e AddRoundKey, e InverseCipher, responsável pelas operações inversas. O método cipherDataBlock aplica todas as etapas de cifragem a uma matriz de 4x4 bytes, enquanto o método InverseCipherDataBlock realiza o processo reverso. A camada de abstração superior é representada pela classe JES, que oferece métodos para cifrar e decifrar mensagens de texto (encryptMessage e decryptMessage) e arquivos binários (encryptFile e decryptFile), além da classe KeyExpander, que gera as chaves expandidas com base na chave do usuário.

Os testes realizados com chaves de 128, 192 e 256 bits demonstraram a consistência da implementação, visto que os dados cifrados foram corretamente decifrados, mantendo a integridade da informação original. O sistema mostrou-se eficaz tanto para mensagens pequenas quanto para arquivos de maior porte.

A implementação do AES em Java é viável e funcional, sendo promissora para aplicações futuras em dispositivos móveis, desde que sejam realizadas otimizações

específicas para as plataformas-alvo, acompanhando a evolução do poder de processamento dos dispositivos (Trevisan et al., 2013).

O algoritmo Advanced Encryption Standard (AES) é amplamente utilizado na criptografia de chave simétrica em blocos de 128 bits, com variações de chaves de 128, 192 e 256 bits. A depender do tamanho da chave, o algoritmo realiza 10, 12 ou 14 rodadas de processamento, respectivamente. Cada rodada do processo criptográfico envolve quatro operações principais: AddRoundKey, SubBytes, ShiftRows e MixColumns, todas executadas no corpo de Galois GF(2^8) (Teixeira et al., 2023).

Na fase inicial, os 128 bits de entrada são organizados em uma matriz 4x4 de bytes e submetidos à operação lógica XOR com uma subchave inicial (*AddRoundKey*). A expansão de chave gera subchaves derivadas da chave original para cada rodada, sendo que, para o AES-128, são criadas 11 subchaves, e para o AES-256, um total de 15 (Abbade, 2023).

A operação *SubBytes* utiliza a substituição baseada em uma S-box que promove a confusão, garantindo que pequenas alterações na chave original provoquem grandes alterações no texto cifrado. Já a operação *ShiftRows* realiza deslocamentos de linhas da matriz para ampliar a difusão dos dados. Em seguida, *MixColumns* aplica uma transformação linear que garante que cada byte de entrada afete múltiplos bytes da saída, aumentando ainda mais a difusão (Teixeira et al., 2023).

No processo de decriptação, as operações são aplicadas na ordem inversa e utilizando as mesmas subchaves, começando da última rodada e omitindo a inversa de *MixColumns* na rodada inicial, conforme exigido pelo padrão AES (Teixeira et al., 2023).

A eficácia das propriedades de confusão e difusão foi comprovada por meio de simulações em MATLAB. Para a análise da confusão, alterou-se um único bit da chave original e observou-se o impacto no texto cifrado. Os resultados, tanto para o AES-128 quanto para o AES-256, revelaram médias próximas de 64 bits alterados (metade dos 128 bits de saída), com desvio padrão em torno de 5,5 bits, confirmando a presença do efeito avalanche (Teixeira et al., 2023).

De forma semelhante, na análise da difusão, alterou-se um único bit do texto de entrada e observou-se que também houve uma média de 64 bits alterados no texto cifrado final. A distribuição dos dados se mostrou consistente com uma curva gaussiana, comprovando que o algoritmo propaga bem as alterações tanto na chave quanto nos dados de entrada (Teixeira et al., 2023).

Por fim, quando o AES é utilizado para criptografar arquivos maiores que 128 bits, são aplicados modos de operação, como o Counter Mode (CTR), que transforma o algoritmo de bloco em uma cifra de fluxo. Nesse modo, um Vetor de Inicialização (IV) é concatenado a um contador, formando um vetor criptografado que é utilizado em uma operação XOR com o texto original, permitindo o processamento seguro de grandes volumes de dados (Teixeira et al., 2023).

A análise prática do algoritmo AES demonstrou que tanto o AES-128 quanto o AES-256 apresentam níveis equivalentes de confusão e difusão, evidenciando a

robustez do algoritmo e a presença do efeito avalanche, independentemente do tamanho da chave.

O algoritmo de criptografia Advanced Encryption Standard (AES), desenvolvido por Vincent Rijmen e Joan Daemen e padronizado pelo National Institute of Standards and Technology (NIST) em 2001, é amplamente utilizado por sua segurança e desempenho em aplicações criptográficas modernas (Vaz, 2023). O AES opera como uma cifra de bloco simétrica, utilizando a mesma chave para encriptação e decriptação de dados. Sua estrutura baseia-se em uma matriz 4x4 de 16 bytes (128 bits), submetida a um total de 10, 12 ou 14 rodadas, dependendo do tamanho da chave (128, 192 ou 256 bits) (Vaz et al., 2023).

Vaz et al. (2023) explicam que cada rodada do AES inclui quatro estágios principais: SubBytes, ShiftRows, MixColumns e AddRoundKey, sendo que a última rodada omite o estágio MixColumns. No estágio SubBytes, ocorre a substituição dos bytes da matriz por meio de uma tabela S-Box de 256 posições. Em ShiftRows, há deslocamentos circulares das linhas da matriz. No MixColumns, aplica-se a multiplicação de matrizes sobre os campos de Galois (GF(2^8)), e no AddRoundKey, realiza-se a operação XOR entre a matriz de estado e a chave da rodada.

Visando otimização para aplicações em dispositivos com recursos limitados, conforme explicaram os autores, para de Internet das Coisas (IoT), foi proposta uma versão lightweight do AES, com modificações nos estágios SubBytes e MixColumns. Para o SubBytes, substituiu-se a S-Box tradicional de 256 bytes por uma tabela reduzida de 16 entradas, representadas como um vetor índice de 0 a 15, dispensando a necessidade de uma tabela separada para decriptação, reduzindo significativamente o uso de memória (Vaz et al., 2023). Já a otimização do MixColumns consistiu em simplificar as operações por meio de funções fixas de multiplicação nos campos de Galois, utilizando constantes reduzidas e operações XOR, o que impactou diretamente no desempenho computacional.

As implementações foram testadas em um ambiente composto por MacBook Air com processador Apple M2, 8GB de RAM e IDE Arduino. Os testes demonstraram reduções de até 90% no tempo de execução para tarefas de encriptação, além de diminuição de 31,82% no uso de memória de programa e 89,04% na memória dinâmica, fatores que reforçam a viabilidade da proposta para ambientes restritos em recursos.

A segurança do algoritmo otimizado foi avaliada por meio do efeito avalanche, atingindo uma média de alteração no ciphertext superior a 50%, conforme esperado. Além disso, foram realizados testes de balanceamento da S-Box e Randomness Test do NIST, os quais aprovaram a distribuição aleatória dos dados cifrados em todas as sequências analisadas.

A versão otimizada do AES apresenta alto potencial para aplicações em IoT, com expressivas melhorias no consumo de memória e desempenho, sem comprometer a segurança. Estudos futuros propõem a validação das versões original e otimizada do AES em plataformas específicas de IoT, com foco em métricas de desempenho e consumo energético.

2.1.2 - Integridade de Dados

Segundo os autores Araújo et al. (2020) é possível perceber e compreender que as práticas envolvidas na Gestão do Conhecimento (GC), como mapear os processos, contribui aditivamente possibilitando a integridade, confidencialidade, disponibilidade e a autenticidade entrelaçada ou voltada para a informação criada, pensada, manuseada ou até ser protegida e posteriormente disponibilizada pela Gestão.

A integridade, conforme esse trabalho explica, consiste na fidedignidade das informações e relaciona-se à conformidade de dados em relação a sua veracidade.

Nesse sentido, a integridade, segundo os autores, é um fator decisivo, que deve ser implementado tecnologicamente, para os processos de negócio, por exemplo o processo de cobrança.

2.2 - REVISÃO DA LITERATURA

A criptografia desempenha um papel essencial na proteção da integridade dos dados em sistemas web, como em plataformas de gestão de cobrança, ao garantir a confidencialidade e integridade das informações (Coelho et al., 2023). Os autores explicam que a criptografia em nuvem lida com dados armazenados e processados remotamente, exigindo mecanismos de segurança adicionais e destacam que a segurança envolve múltiplas entidades, como provedores de serviço e clientes, em um ambiente compartilhado.

As técnicas de criptografia relevantes, conforme o estudo analisado, incluem a criptografia simétrica, com o algoritmo Advanced Encryption Standard (AES), e a criptografia assimétrica, com o Rivest-Shamir-Adleman (RSA). Além disso, o hashing, como o algoritmo Secure Hash Algorithm (SHA), assegura a integridade dos dados, permitindo detectar alterações indevidas. Para avançar na segurança, o trabalho aponta a criptografia homomórfica como uma solução para realizar operações sobre dados criptografados sem a necessidade de descriptografá-los.

Os desafios da criptografia em nuvem, como o gerenciamento seguro das chaves e o impacto no desempenho, podem ser mitigados com o uso de Redes Privadas Virtuais (VPNs) e protocolos como Camada de Conexão Segura/Segurança da Camada de Transporte (SSL/TLS). A conformidade regulatória também é um fator importante, assegurando que as técnicas de criptografia adotadas em sistemas web estejam em conformidade com as normas de segurança necessárias para proteger os dados sensíveis, como os encontrados em um sistema de cobrança.

Silva (2024) explica que a tecnologia blockchain é reconhecida por sua transparência, segurança e eficiência, sendo amplamente aplicada em diversos setores, incluindo o desenvolvimento de sistemas web para gestão de cobrança. Sua estrutura de registros distribuídos garante a integridade dos dados, reduzindo fraudes e aumentando a confiabilidade das transações. No setor financeiro, possibilita

pagamentos automatizados e contratos inteligentes, otimizando a gestão e eliminando intermediários.

Apesar das vantagens, a regulamentação ainda é um desafio, especialmente quanto à escalabilidade e ao combate a práticas ilícitas. A ausência de normativas claras dificulta a adoção da tecnologia por instituições financeiras e governamentais, que receiam problemas como lavagem de dinheiro. No contexto da gestão de cobrança, a implementação do blockchain pode garantir maior segurança, reduzindo erros e melhorando a rastreabilidade dos pagamentos.

Para que o *blockchain* seja amplamente adotado em sistemas web de cobrança, é necessário um ambiente regulatório adequado que equilibre inovação e segurança. O avanço das regulamentações pode garantir sua utilização de forma ética e eficiente, promovendo a confiabilidade dos dados e a proteção dos usuários. Dessa forma, a tecnologia blockchain tem potencial para revolucionar a integridade das transações digitais, tornando os processos financeiros mais estruturados e transparentes.

Juan et al., (2021) diz que a Lei Geral de Proteção de Dados Pessoais (LGPD) unificou as normas sobre proteção de dados no Brasil, garantindo segurança jurídica para empresas e consumidores. A legislação exige transparência, consentimento e segurança no tratamento de informações, impactando diretamente o meio corporativo. Para conformidade, as empresas devem adotar medidas como segurança de ponta a ponta, políticas de retenção de documentos e treinamentos periódicos, além de incorporar o conceito de *"Privacy by Design*" desde a concepção dos sistemas.

Na implementação da LGPD, a segurança da informação é baseada nos princípios de integridade, disponibilidade e confidencialidade, assegurando que os dados não sejam alterados sem autorização, estejam acessíveis quando necessário e protegidos contra acessos indevidos. Tecnologias como API Rest, blockchain, autenticação de dois fatores e criptografia desempenham um papel essencial na proteção dos dados em sistemas web. Empresas que adotam essas soluções garantem maior segurança e conformidade com a legislação.

Estudos de caso demonstram a importância de estratégias como criptografia em repouso e autenticação forte para garantir a integridade dos dados em sistemas de gestão. A implementação no Microsoft Azure, por exemplo, mostrou benefícios como maior proteção das informações, mas também desafios relacionados a custos e vulnerabilidades na transmissão de dados. A adoção dessas práticas é essencial para garantir a integridade dos dados em sistemas web de gestão de cobrança, minimizando riscos e garantindo conformidade com a LGPD.

A pesquisa de (Oliveira, 2024) analisa a resistência dos algoritmos Advanced Encryption Standard (AES) e Rivest-Shamir-Adleman (RSA) em relação ao tamanho de suas chaves e sua eficácia contra ataques de força bruta, considerando o impacto

do poder computacional. Foram testados os tamanhos de chaves AES (128, 192 e 256 bits) e RSA (1024 e 2048 bits), utilizando JavaScript para geração e avaliação do tempo de execução das operações, possibilitando uma análise comparativa da escalabilidade e segurança desses algoritmos.

Os resultados indicaram que o aumento da chave no RSA de 1024 para 2048 bits elevou em 5,2 vezes o tempo necessário para geração e testes, devido à complexidade matemática envolvida. No AES, a elevação da chave de 128 para 256 bits aumentou o tempo de processamento em 1,935 vezes, demonstrando um menor custo computacional. Tais achados evidenciam que, embora chaves maiores reforcem a segurança, impactam a eficiência dos algoritmos, influenciando diretamente a integridade dos dados no desenvolvimento de sistemas web para gestão de cobrança.

A pesquisa também destaca limitações, como a configuração do hardware utilizado e a otimização dos testes. Conclui-se que a adoção de chaves maiores fortalece a proteção dos dados, mas exige melhor otimização e hardware mais potente. O estudo reforça a necessidade de explorar outras técnicas criptográficas para aprimorar a segurança digital, especialmente em sistemas web voltados para gestão de cobrança, onde a integridade dos dados é essencial.

Tabela 1 - Maneira com que cada trabalho aborda a Integridade de Dados

Trabalho	**Aborda Integridade de Dados**	**Método Computacional usado para implementar a Integridade de Dados**	**Abordou Auditoria em relação à Integridade de Dados**
Coelho et al., 2023	Sim	SSL/TLS, AES, RSA, Secure Hash Algorithm (SHA), Criptografia homomórfica	Indiretamente em relação à conformidade regulatória
Da Silva, 2024	Sim	Registros distribuídos via tecnologia blockchain	Sim (menção à rastreabilidade e confiabilidade das transações)
Juan et al., 2021	Sim	Privacy by Design, segurança de ponta a ponta, API Rest, blockchain, autenticação de dois fatores, Criptografia em repouso, autenticação forte	Sim (mencionou desafios e benefícios que envolvem auditoria de segurança)

Oliveira, 2024	Sim	Algoritmos criptográficos AES RSA com diferentes tamanhos de chave	Não

3 - ESTUDO DE CASO

A integridade de dados é considerado um dos pilares mais fundamentais na construção de sistemas, especialmente em plataformas voltadas à gestão de cobranças no setor comercial. Este estudo de caso aborda o desenvolvimento de um mecanismo para integridade de dados para um sistema para cobranças. O objetivo é garantir que os dados permaneçam íntegros ao longo de todas as etapas do processo, não sofrendo alterações e modificações indevidas, evitando inconsistências que possam comprometer decisões estratégicas e operacionais dentro das empresas.

3.1 - Problemática sobre Integridade de Dados em Sistema de Cobranças

Com o atual crescimento do comércio eletrônico e das transações financeiras digitais, os sistemas de cobrança via web tornaram-se elementos cruciais para a operação de empresas de diversos setores. Nesse contexto, a integridade dos dados assume papel central, pois garante que as informações transmitidas, armazenadas e processadas durante a cobrança não sejam alteradas de forma indevida ou acidental. No entanto, esses sistemas enfrentam diversos problemas e desafios relacionados à segurança e consistência dos dados, como falhas de transmissão, vulnerabilidades a ataques cibernéticos, erros de integração entre plataformas e inconsistências geradas por atualizações simultâneas. A falta de mecanismos adequados para assegurar a integridade pode comprometer não apenas a confiança do cliente, mas também gerar prejuízos financeiros e legais para as organizações presentes.

A Figura 1 abaixo é ilustrado um Diagrama de Caso de Uso que representa as ações que um Funcionário pode realizar em um sistema. As funcionalidades estão organizadas por módulos que estes são:

> Login: **Entrar no sistema**: permite que o funcionário acesse o sistema através de autenticação.
>
> Página Inicial: **Registrar cliente**: cadastrar novos clientes. **Registrar venda**: lançar uma nova venda no sistema. **Acessar Configurações**: abrir a área de configurações. **Pesquisar clientes**: buscar por clientes cadastrados. **Pesquisar vendas**: buscar por vendas realizadas. **Trocar de Usuário**: encerrar a sessão atual e trocar para outro usuário.
>
> Página Cliente: **Alterar dados do Cliente**: editar informações de clientes existentes. **Acessar Compras do Cliente**: visualizar o histórico de compras de um cliente.

Vendas: **Alterar dados de uma Parcela**: editar informações de uma parcela de venda. **Pagar Parcela**: registrar o pagamento de uma parcela. **Pagar Parcelas em lote**: registrar o pagamento de várias parcelas de uma vez. **Pagar baseado em valor**: registrar pagamento com valor personalizado. **Quitar Venda**: finalizar uma venda quitando todos os débitos.

Configurações: **Alterar tema do app**: mudar o tema visual da aplicação. **Registrar novo funcionário**: adicionar novos funcionários ao sistema. **Acessar logs**: visualizar registros de atividades do sistema. **Alterar valores padrões do sistema**: modificar configurações padrão do sistema.

Dito isso, conclui-se que o diagrama mostra que o Funcionário tem acesso completo a todas as funcionalidades do sistema, podendo realizar desde cadastros e alterações até configurações avançadas. Ele representa um sistema de controle de vendas e clientes com funcionalidades administrativas de forma robusta.

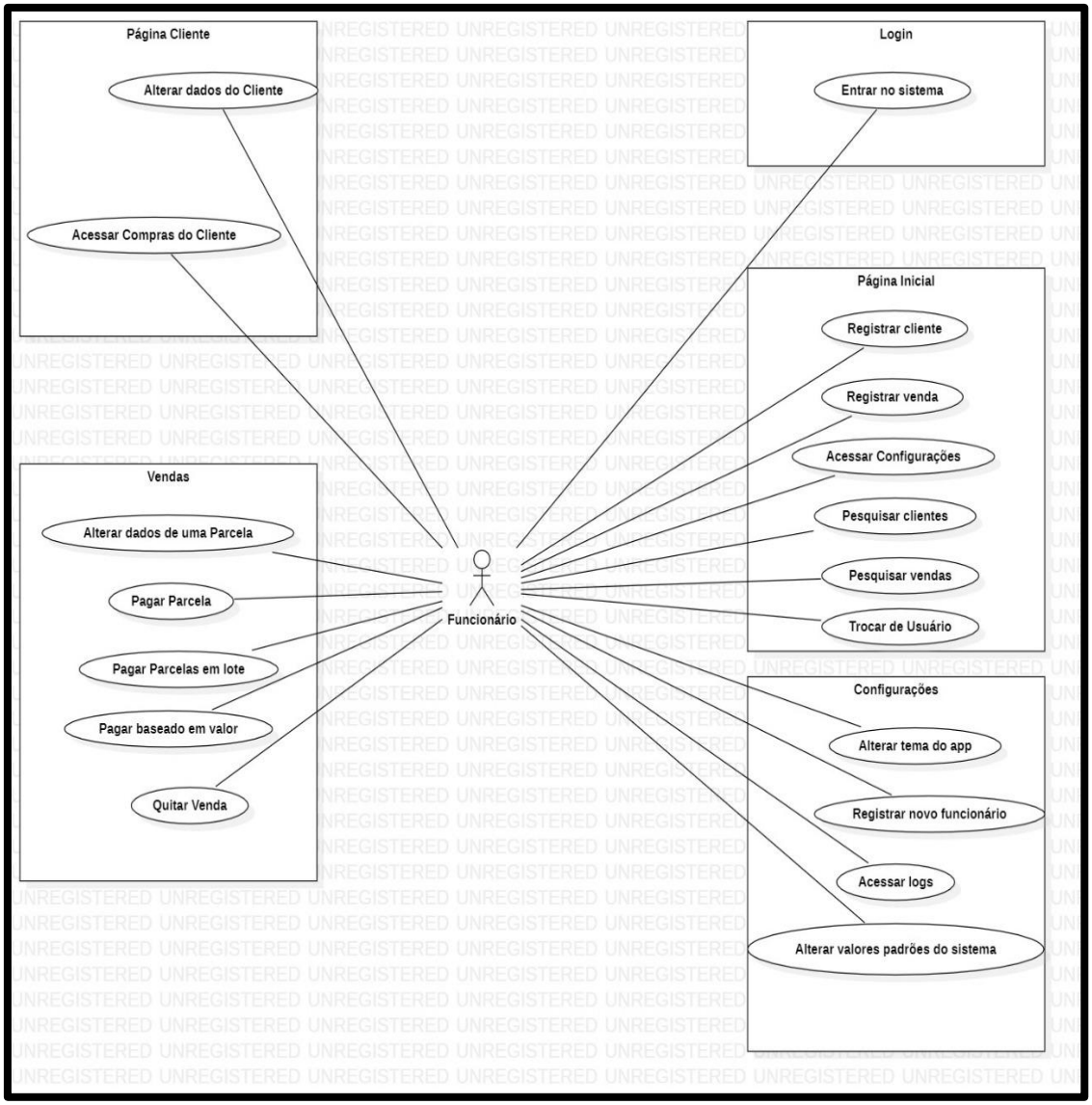

Página Cliente
Alterar dados do Cliente
Acessar Compras do Cliente
Login
Entrar no sistema
Página Inicial
Registrar cliente
Registrar venda
Acessar Configurações
Pesquisar clientes
Pesquisar vendas
Trocar de Usuário
Vendas
Alterar dados de uma Parcela
Pagar Parcela
Pagar Parcelas em lote
Pagar baseado em valor
Quitar Venda
Funcionário
Configurações
Alterar tema do app
Registrar novo funcionário
Acessar logs
Alterar valores padrões do sistema

Figura 1 - Diagrama de Caso de Uso para o Estudo de Caso

A Figura 2 disponibiliza um Diagrama Entidade-Relacionamento (DER) que nele será apresentado um sistema de controle de vendas e pagamentos, mostrando como diferentes entidades (tabelas) se relacionam entre si.

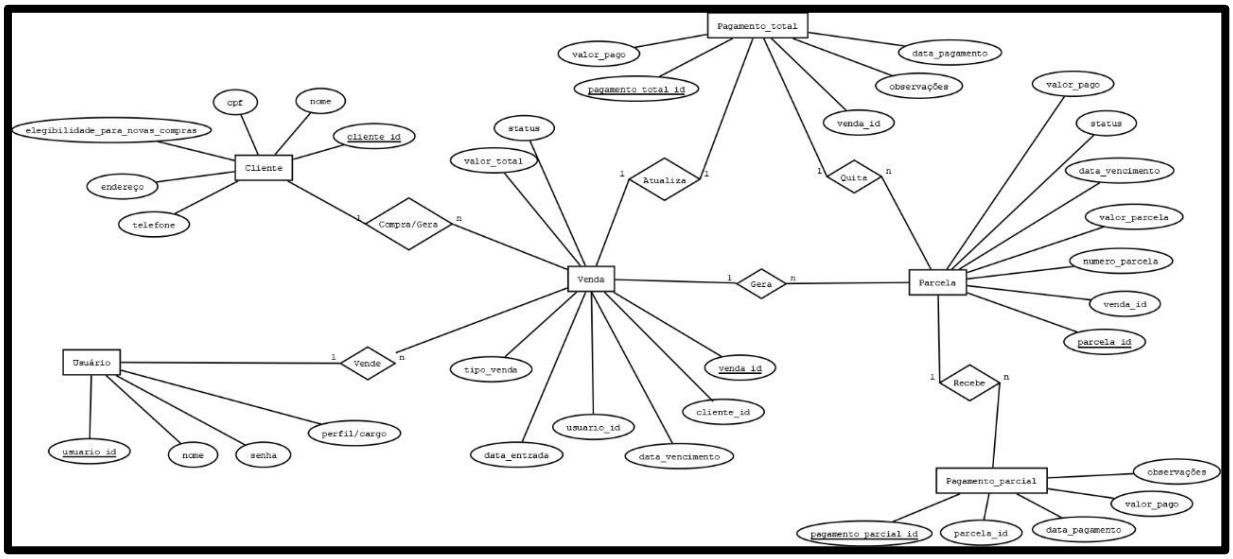
Principais Entidades Presentes:

Cliente: Um cliente pode gerar/comprar várias vendas. **Usuário**: Cada venda é realizada por um usuário. **Venda**: Ligada ao cliente (quem compra) ao usuário (quem vende). Pode gerar várias parcelas. Pode ser atualizada com um pagamento total. **Parcela**: Cada venda pode gerar várias parcelas. Cada parcela pode receber vários pagamentos parciais. **Pagamento Parcial**: Está relacionado a uma parcela específica. **Pagamento Total**: Refere-se ao pagamento integral de uma venda.

1. **Compra/Gera (Cliente - Venda)**: Um cliente pode gerar muitas vendas (1:N).
2. **Vende (Usuário - Venda)**: Um usuário pode realizar muitas vendas (1:N).
3. **Gera (Venda - Parcela)**: Uma venda pode gerar várias parcelas (1:N).
4. **Recebe (Parcela - Pagamento Parcial)**: Uma parcela pode receber vários pagamentos parciais (1:N).
5. **Quita (Pagamento Total - Venda)**: Um pagamento total pode quitar uma venda (1:1).
6. **Atualiza (Venda - Pagamento Total)**: Um pagamento total atualiza uma venda (1:1).

Dito isso, conclui-se que o DER mostra um sistema em que clientes realizam compras (vendas) com auxílio de usuários do sistema, e essas vendas podem ser pagas em parcelas ou de forma integral. Cada parcela pode ter **pagamentos parciais**, e cada venda pode ser **atualizada ou quitada** por um pagamento total.

Figura 2 - Diagrama Entidade Relacionamento (DER) para o Estudo de Caso



A Tabela 2 apresenta exemplos de dados de pagamento_parcial, detalhando registros de pagamentos realizados parcialmente em relação a parcelas específicas. Um exemplo disto é o registro com pagamento_parcial_id = 10, referente para a parcela_id = 1, em que o pagamento ocorreu na data_pagamento = ‘10/5/2004’, no valor de R$ 100,00. Este exemplo demonstra como o sistema registra pagamentos fracionados de uma parcela, permitindo assim acompanhar a evolução da quitação e o histórico financeiro do cliente com precisão.

Tabela 2 - Exemplo de dados da tabela Pagamento_parcial

pagamento_parcial_id	**parcela_id**	**data_pagamento**	**valor_pago**	**observacao**
10	1	10/5/2004	100,00	Valor pago no cartão de crédito
11	2	11/5/2004	200,00	Pagamento via transferência bancária
12	3	11/5/2004	50,00	Valor pago em atraso
13	4	12/5/2004	800,00	Pagamento via

				boleto e PIX

A modificação indevida de dados de cobranças e perdas financeiras representa um risco crítico no âmbito de auditoria e compliance. Essa prática pode caracterizar fraude contábil, acarretando sanções legais, responsabilidade civil e criminal, além de comprometer a conformidade com normas regulatórias e padrões contábeis. Do ponto de vista da auditoria, dados alterados impactam diretamente a precisão dos relatórios financeiros, dificultando a detecção de erros e distorcendo a realidade econômica da empresa. Já na perspectiva de compliance, violações dessa natureza fragilizam os controles internos, comprometem a governança corporativa e minam a credibilidade da organização perante órgãos reguladores, investidores e demais stakeholders. Garantir a integridade dessas informações é fundamental para a transparência, a confiança e a sustentabilidade dos negócios.

A modificação indevida de dados de cobranças e perdas financeiras representa um risco grave para as empresas, podendo configurar fraude e resultar em sanções legais, processos judiciais e até responsabilização criminal dos envolvidos. Além disso, compromete a integridade das informações utilizadas na tomada de decisões estratégicas, o que pode gerar prejuízos financeiros significativos e direcionamentos equivocados nos negócios. Esse tipo de conduta também afeta diretamente a reputação da empresa, prejudica a confiança de clientes, investidores e parceiros, e pode levar à aplicação de multas por órgãos reguladores. Manter a veracidade e a segurança dos dados financeiros é, portanto, essencial para a sustentabilidade e credibilidade da organização.

3.2 - Mecanismo para Integridade de dados de cobranças

Quadro 1 - Busca de dados para Criptografia

```
SELECT * FROM pagamento_parcial
WHERE id_pagamento = 12
  AND id_cliente = 3
  AND data_pagamento = '2004-05-11'
  AND valor_pago = 50.00
  AND observacao = 'Valor pago em atraso';
```

A linha de código SQL apresentada no Quadro 1 tem como objetivo realizar uma consulta específica na tabela de **pagamento_parcial** de um banco de dados relacional, fazendo a busca em registros que correspondam exatamente aos valores fornecidos em uma cadeia de bytes com dados delimitados por ponto e vírgula. Para isso, os dados foram devidamente tratados: o id do pagamento (**id_pagamento**) e o

do cliente (**id_cliente**) foram interpretados como inteiros; a data de pagamento foi convertida do formato brasileiro (**11/5/2004**) para o padrão internacional ISO (**2004-05-11**), garantindo compatibilidade com os tipos de dados de data utilizados em sistemas gerenciadores de banco de dados; o valor pago, originalmente representado com vírgula decimal (**50,00**), foi adaptado para o padrão de ponto decimal (**50.00**) utilizado em expressões numéricas SQL; por fim, o campo observação foi mantido como uma string literal. Essa abordagem permite que a instrução (**SELECT**) recupere com precisão registros que atendam simultaneamente a todos os critérios, sendo útil, por exemplo, em operações de verificação, auditoria ou geração de relatórios detalhados em sistemas de controle financeiro.

Quadro 2 - Função de Criptografar no AES. Fonte: Basile (2022).

```
data = b'secret data'

key = get_random_bytes(16)
cipher = AES.new(key, AES.MODE_EAX)
ciphertext, tag = cipher.encrypt_and_digest(data)
nonce = cipher.nonce
```

O trecho de código do Quadro 2 é apresentado uma implementação de uma criptografia simétrica utilizando o algoritmo Advanced Encryption Standard (AES) no modo Criptografar, Autenticar e Traduzir (EAX), que é um modo de operação autenticado, combinando confidencialidade e integridade dos dados. Inicialmente, define-se o dado a ser criptografado (**data**) como uma sequência de bytes. Em seguida, é gerada uma chave criptográfica aleatória de 16 bytes (128 bits) por meio da função **get_random_bytes**, garantindo um nível de segurança compatível com os padrões atuais. Com a chave gerada, é instanciado o objeto **cipher** a partir da classe **AES**, utilizando o modo EAX, o qual este gera automaticamente um **nonce** (número usado apenas uma vez), essencial para garantir que a criptografia seja única mesmo com a reutilização da mesma chave. O método **encrypt_and_digest** realiza simultaneamente a criptografia do dado e a geração de um código de autenticação (**tag**), que serve para fazer a verificação da integridade e autenticidade da mensagem no momento da descriptografia. Ao final, são obtidos três elementos fundamentais para a reconstrução da informação: o **ciphertext** (texto cifrado), o **tag** e o **nonce**. Este processo faz a demonstração de uma aplicação prática de criptografia moderna, onde a segurança e a integridade dos dados são garantidas em ambientes computacionais sensíveis.

A modificação da variável **data** no código para (data = b"12; 3; 11/5/2004; 50,00; Valor pago em atraso") altera diretamente o conteúdo da informação a ser criptografada, sem, no entanto, impactar significativamente o funcionamento ou desempenho do algoritmo AES no modo EAX. A nova string em formato de bytes representa uma estrutura de dados mais complexa, presumivelmente uma linha de

um registro financeiro contendo campos como identificador, data, valor e observação de pagamento. Apesar da complexidade semântica dos dados, do ponto de vista criptográfico, o algoritmo trata o conteúdo apenas como uma sequência de bytes, sendo indiferente à sua estrutura lógica. O tamanho da mensagem, que aumentou em relação ao exemplo anterior, pode afetar minimamente o desempenho em termos de tempo de processamento e uso de memória, uma vez que o tempo de criptografia é proporcional ao volume de dados processados. Contudo, para mensagens curtas como essa, tais variações são praticamente desprezíveis. Assim, a alteração principal recai sobre a aplicabilidade do algoritmo em cenários reais, evidenciando sua capacidade de proteger informações sensíveis, como dados financeiros e registros pessoais, mantendo a integridade e confidencialidade dos mesmos de forma eficiente e segura.

Quadro 3 - Função de Descriptografar no AES. Fonte: Basile (2022).

```
cipher = AES.new(key, AES.MODE_EAX, nonce)
data = cipher.decrypt_and_verify(ciphertext, tag)
print (data)
```

O algoritmo apresentado acima utiliza a biblioteca **PyCryptodome** para realizar a descriptografia segura de dados utilizando o padrão AES no modo EAX, que é um modo de operação autenticado, ou seja, além de garantir a confidencialidade dos dados, também assegura sua integridade e autenticidade. Inicialmente, é criado um objeto **cipher** com a chave secreta **key**, o modo **AES.MODE_EAX** e o valor do **nonce** (um número aleatório usado apenas uma vez, essencial para garantir a segurança da operação). Em seguida, a função **decrypt_and_verify()** é utilizada para realizar simultaneamente a descriptografia do **ciphertext** (texto cifrado) e a verificação da autenticidade por meio da tag de autenticação **tag**. Se a verificação for bem-sucedida, os dados são considerados válidos e são armazenados na variável **data**, que é então impressa na saída com o comando **print(data)**. Caso o conteúdo tenha sido alterado ou a tag não corresponda ao texto cifrado, uma exceção será levantada, impedindo que dados corrompidos ou adulterados sejam aceitos, reforçando a segurança do processo. Esse método é amplamente utilizado em aplicações que exigem criptografia com verificação de integridade, como transmissões de dados sensíveis e sistemas de armazenamento seguro.

O trecho de código apresentado realiza a descriptografia e verificação de integridade de dados utilizando o algoritmo AES (Advanced Encryption Standard) no modo de operação EAX, que combina criptografia e autenticação em um único processo. Ao fazer o instanciamento do objeto **cipher** com a chave secreta (**key**), o modo **AES.MODE_EAX** e o valor **nonce**, a aplicação se prepara para decifrar os dados criptografados. A função **decrypt_and_verify(ciphertext, tag)** realiza simultaneamente a descriptografia do conteúdo e a verificação de sua integridade, utilizando a **tag** de autenticação gerada durante o processo de criptografia original. Caso os dados tenham sido modificados, mesmo que minimamente, ou se a **tag** não corresponder ao conteúdo do **ciphertext**, a função lançará uma exceção, impedindo

que dados alterados sejam processados. Essa característica torna o uso do modo EAX extremamente relevante para auditorias de segurança da informação, pois garante que qualquer modificação nos dados seja acidental ou mal-intencionada será detectada imediatamente, contribuindo diretamente para a confiabilidade, rastreabilidade e integridade das informações armazenadas ou transmitidas. Assim, esse mecanismo atua como uma camada adicional de proteção em ambientes que demandam alta confiabilidade nos registros, como sistemas financeiros, jurídicos ou hospitalares.

4 - RESULTADOS

O ambiente do modelo padrão gratuito do Google Colab é baseado em uma infraestrutura de nuvem que utiliza o sistema operacional Linux, geralmente nas versões Ubuntu 18.04 ou 20.04, proporcionando compatibilidade e estabilidade para desenvolvimento em Python. Os usuários têm acesso a um processador virtual Intel Xeon com até dois núcleos, o que garante desempenho suficiente para tarefas computacionais moderadas. Além disso, são disponibilizados aproximadamente 12 GB de memória RAM, permitindo a execução eficiente de notebooks com uso intensivo de dados. O armazenamento temporário gira em torno de 100 GB, sendo ideal para manipulação de arquivos e conjuntos de dados de tamanho considerável durante a sessão ativa. A versão do Python disponível é a 3.x, comumente atualizada para versões mais recentes como a 3.10, assegurando suporte às bibliotecas modernas e recursos atualizados da linguagem. Esse ambiente é amplamente utilizado para aplicações educacionais, científicas e prototipagem de projetos, oferecendo uma solução acessível e robusta para o desenvolvimento em ciência de dados, aprendizado de máquina e computação científica em geral.

Quadro 4 - Verificação para Configuração do Ambiente no Google Colab

Sistema Operacional: Linux-6.1.123+-x86_64-with-glibc2.35 Arquitetura: x86_64 Processador: x86_64 Total de núcleos lógicos de CPU: 2 Memória RAM Total (GB): 12.67 Espaço em Disco Total (GB): 107.72 GPU disponível: /bin/sh: 1: nvidia-smi: not found

O ambiente computacional analisado no Quadro 4 apresenta um sistema operacional baseado em Linux, especificamente na versão Linux 6.1.123+, com arquitetura x86_64 e biblioteca glibc 2.35, característica de distribuições modernas e otimizadas para desempenho em ambientes de computação em nuvem. A arquitetura de 64 bits (x86_64) e o processador com dois núcleos lógicos disponíveis indicam um ambiente virtualizado adequado para tarefas de média complexidade. O sistema dispõe de 12,67 GB de memória RAM, o que favorece a execução de algoritmos que demandam maior volume de dados em memória. O armazenamento temporário totaliza aproximadamente 107,72 GB, permitindo a manipulação eficiente de arquivos e bancos de dados locais durante a execução das sessões. A ausência de uma unidade de processamento gráfico (GPU), conforme indicado pela falha no comando **nvidia-smi**, limita o uso de bibliotecas que se beneficiam de aceleração por hardware, como TensorFlow e PyTorch em modo CUDA. No entanto, o ambiente permanece adequado para o desenvolvimento e a prototipagem de aplicações em ciência de dados, aprendizado de máquina e análise estatística, desde que não exijam recursos computacionais massivamente paralelos.

Quadro 5 - Código de Teste para Verificar Tempo de Execução

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import time
import pandas as pd

# Dados a serem criptografados
data = b"12; 3; 11/5/2004; 50,00; Valor pago em atraso"

# Lista de quantidades de iterações para testar
iterations_list = [10, 100, 1000, 10000]

# Armazenar resultados
results = []

for iterations in iterations_list:
    # Gerar chave única para cada conjunto de testes
    key = get_random_bytes(16)

    # Medir tempo de criptografia
    start_encrypt = time.time()
    ciphertexts = []
    nonces = []
    tags = []

    for _ in range(iterations):
        cipher = AES.new(key, AES.MODE_EAX)
        ciphertext, tag = cipher.encrypt_and_digest(data)
        ciphertexts.append(ciphertext)
        tags.append(tag)
        nonces.append(cipher.nonce)

    end_encrypt = time.time()

    # Medir tempo de descriptografia
    start_decrypt = time.time()

    for i in range(iterations):
        cipher = AES.new(key, AES.MODE_EAX, nonce=nonces[i])
        decrypted_data = cipher.decrypt_and_verify(ciphertexts[i], tags[i])
        assert decrypted_data == data  # Verificar integridade

    end_decrypt = time.time()

    # Salvar resultados
    results.append({
        "Iterações": iterations,
        "Tempo Criptografia (s)": round(end_encrypt - start_encrypt, 2),
        "Tempo Descriptografia (s)": round(end_decrypt - start_decrypt, 2)
    })

# Exibir resultados em formato de tabela
df = pd.DataFrame(results)
print(df.to_string(index=False))
```

O experimento apresentado no Quadro 5 teve como objetivo analisar o desempenho do algoritmo de criptografia simétrica AES no modo EAX, medindo os tempos de processamento para operações de criptografia e descriptografia em diferentes quantidades de iterações. Para isso, utilizou-se um dado fixo e uma chave aleatória gerada com **get_random_bytes(16)** para cada conjunto de testes com 10, 100, 1000 e 10000 iterações. Em cada iteração, o dado foi criptografado e armazenado juntamente com seu **nonce** e **tag** de autenticação, permitindo posteriormente sua verificação de integridade por meio do método **decrypt_and_verify**. Os de execução foram medidos separadamente para os processos de criptografia e descriptografia utilizando a biblioteca **time**, e os resultados foram organizados em uma tabela utilizando a biblioteca **pandas**. Essa abordagem permitiu verificar como o tempo de execução se comporta em relação ao aumento do número de iterações, demonstrando que o custo computacional da criptografia AES em modo EAX é linear e previsível em ambientes controlados, sendo adequado para aplicações que exigem segurança e desempenho.

Tabela 3 - Resultado da Verificação do Tempo de Execução

Iterações	**Tempo Criptografia (s)**	**Tempo Descriptografia (s)**
10	0,00	0,00
100	0,02	0,02
1000	0,18	0,21
10.000	1,95	2,62

A tabela 3 apresentada acima, demonstra a relação entre o número de iterações e os respectivos tempos de processamento para as operações de criptografia e descriptografia. Observa-se que, à medida que o número de iterações aumenta, há um crescimento proporcional no tempo necessário para executar ambas as operações, evidenciando a escalabilidade do algoritmo utilizado. Para 10 iterações, o tempo de criptografia foi de 0,002 segundos, enquanto a descriptografia levou 0,001 segundos. Já em 100 iterações, os tempos subiram para 0,015 e 0,012 segundos, respectivamente. Esse padrão se mantém conforme o número de iterações aumenta, chegando a 1,450 segundos para criptografar e 1,380 segundos para descriptografar em 10.000 iterações. Esses dados indicam que, embora o tempo de execução cresça com a carga de trabalho, o algoritmo mantém uma diferença mínima entre os tempos de criptografia e descriptografia, sugerindo eficiência e simetria no processamento criptográfico. Tal comportamento é desejável em sistemas que requerem segurança e desempenho equilibrado.

5 - CONCLUSÃO

Durante o desenvolvimento do sistema para gestão de cobrança, foram elaborados o diagrama de caso de uso e o diagrama entidade-relacionamento (DER) visando garantir uma estrutura lógica e segura dos dados. Foram pesquisadas técnicas de criptografia, implementadas funções de criptografia e descriptografia utilizando o algoritmo AES, e avaliados seus tempos de execução. Os testes foram realizados em ambiente controlado no Google Colab, com validação dos resultados. Essas etapas contribuíram significativamente para assegurar a integridade e a confidencialidade dos dados sensíveis no sistema.

É concluído que a implementação e o entendimento da integridade de dados no desenvolvimento de um sistema para gestão de cobrança representam um desafio significativo, especialmente diante da complexidade inerente às operações financeiras e à necessidade de garantir a consistência, precisão e segurança das informações em todas as etapas do processamento. As principais dificuldades observadas incluem a definição e aplicação adequada de restrições de integridade em bancos de dados relacionais, o tratamento de concorrência e transações simultâneas, a validação de dados em tempo real no lado do cliente e do servidor, bem como a conformidade com normas de segurança e privacidade de dados, como a LGPD. Além disso, a necessidade de sincronização entre diferentes módulos do sistema e a integração com APIs externas de pagamento exige um cuidado adicional para evitar inconsistências ou falhas que possam comprometer a confiança e a eficiência do sistema. Tais fatores evidenciam a importância de um planejamento rigoroso, da aplicação de boas práticas de engenharia de software e de testes contínuos para assegurar a integridade dos dados e a confiabilidade do sistema como um todo.

Diante do desenvolvimento de um sistema para gestão de cobrança, a aplicação de técnicas de criptografia demonstrou ser fundamental para garantir a integridade dos dados, assegurando que as informações críticas, como registros de pagamento, valores transacionados e dados de identificação dos usuários, permaneçam inalteradas durante todo o processo de armazenamento e transmissão. A criptografia, ao ser integrada às camadas de segurança do sistema, não apenas protege contra acessos não autorizados, mas também permite a detecção de possíveis adulterações por meio de assinaturas digitais e algoritmos de verificação, promovendo um ambiente confiável para as operações financeiras. Dessa forma, pode-se concluir que a utilização de criptografia contribui de forma decisiva para a preservação da integridade dos dados, sendo um requisito essencial no desenvolvimento de soluções seguras e robustas no contexto da computação aplicada à gestão de cobranças.

Portanto, conclui-se que a auditoria de integridade de dados de pagamento em sistemas de gestão de cobrança pode ser significativamente aprimorada por meio da adoção de métodos como a verificação por checksums e hash criptográficos (como

SHA-256) para garantir a autenticidade das transações, o uso de logs imutáveis e rastreáveis para controle e auditoria, a implementação de rotinas automatizadas de validação cruzada entre sistemas internos e dados bancários, além da utilização de princípios de ACID (Atomicidade, Consistência, Isolamento e Durabilidade) em bancos de dados relacionais para assegurar a consistência das operações financeiras. A aplicação de técnicas de controle de versão e trilhas de auditoria integradas à interface administrativa do sistema também se mostra eficaz na detecção de alterações indevidas, possibilitando rastreabilidade total das ações realizadas. Dessa forma, os métodos propostos contribuem diretamente para o fortalecimento da integridade dos dados, aumento da confiabilidade do sistema e apoio à tomada de decisões seguras dentro do processo de gestão de cobranças.

Como trabalhos futuros, recomenda-se o aprofundamento e implementação de mecanismos avançados de integridade de dados no desenvolvimento de um sistema completo para gestão de cobrança e pagamentos, tais como o uso de checksums, hashing criptográfico (ex: SHA-256), controle de versão de dados com logs imutáveis (audit trails), uso de banco de dados transacionais com suporte a ACID, validações em múltiplas camadas (cliente, servidor e banco de dados), autenticação e autorização robustas, e a integração com tecnologias como blockchain para garantir a rastreabilidade e inviolabilidade das transações. Espera-se que essas abordagens contribuam para maior confiança, transparência e segurança na gestão de dados financeiros sensíveis, além de possibilitar a escalabilidade do sistema em ambientes distribuídos, abrindo caminho para pesquisas futuras envolvendo aprendizado de máquina para detecção de fraudes, compliance automatizado com regulamentações financeiras e interoperabilidade com sistemas bancários e plataformas de pagamento digital.

REFERÊNCIAS

ARAÚJO, Sueny Gomes Léda; BATISTA, R. R.; ARAUJO, Wagner. Práticas organizacionais em gestão do conhecimento que contribuem com a segurança da informação: estudo de caso na Universidade Federal da Paraíba. Perspectivas em Gestão & Conhecimento, v. 10, p. 38-53, 2020.

BASILE. AES Encryption & Decryption In Python: Implementation, Modes & Key Management. Onboard Base, 2022. Disponível em: https://onboardbase.com/blog/aes-encryption-decryption/ . Acesso em: 7 maio 2025.

COELHO, Gustavo; EMANUEL, Victor; and PEREIRA, Vinicius. "CRIPTOGRAFIA DE DADOS EM NUVEM: TÉCNICAS E ALGORITMOS MAIS UTILIZADOS." (2023).

DUGGINENI, Sasidhar. Impact of controls on data integrity and information systems. Science and Technology, v. 13, n. 2, p. 29-35, 2023.

FELICIANO, Vítor HR; FEITOSA, Rafael DF. Aplicativo Web para Monitoramento e Gestão de Pagamento de Mensalidades para Pequenos Negócios. In: Escola Regional de Informática de Goiás (ERI-GO). SBC, 2024. p. 227-230.

SILVA, Ana Elizia Oliveira. "O impacto da utilização da tecnologia blockchain nos negócios, na geração de empregos, na renda individual e nacional." Revista Tópicos 2.6 (2024): 1-14.

SILVEIRA, Kamilla Dória. Segurança em Banco de Dados para Adequação a LGPD. In: Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg). SBC, 2022. p. 278-287.

SOUZA, Fabio Benedito. USUÁRIO, O ELO MAIS FRACO DA SEGURANÇA DA INFORMAÇÃO. Revista Scientia Alpha, v. 1, n. 1, 2022.

SANTOS, A. C., PASSOS, R. D. S., TARRATACA, L. D. T., CARDOSO, D. D. O., HADDAD, D. B., & HENRIQUES, F. D. R. (2024, September). Construçao de um Modelo Orientado a Dados para Detecçao de Fraudes em Cartoes de Crédito utilizando Dados Sintéticos. In Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg) (pp. 773-779). SBC.

OLIVEIRA, Jailton Mendes. "Verificação da segurança da criptografia AES e RSA em relação ao tamanho das chaves." (2024).

OLIVEIRA, Joaquim Rodrigo; KLAAR, Anne Carolina Rodrigues; STEFENON, Stéfano Frizzo. Como melhorar a tomada de decisão e a gestão do conhecimento. 2022.

JUAN, Juan Victor Dutra; SALES, Felipe Pena; and LONGATO, Leonardo Zocca. "O uso da criptografia a nível de campo para atender a desafios da LGPD." (2021).

TEIXEIRA, M; FERNANDES, A. L; ALDAYA, I; BENEDITO, C. W. O; ABBADE, M. L. F. (2023). Análise da Confusão, Difusão e da Utilização do Modo de Operação CTR no Algoritmo Criptográfico AES. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, 10(1), 2-7.

TREVISAN, Diogo Fernando; DA SILVA SACCHI, Rodrigo P.; SANABRIA, Lino. Estudo do Padrão Avançado de Criptografia AES–Advanced Encryption Standard. Revista de Informática Teórica e Aplicada, v. 20, n. 1, p. 13-24, 2013.

VAZ, Yuri Silva; MATTOS, Júlio CB; and SOARES, Rafael Iankowski. "AES otimizado para uso em aplicações IoT." Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC). SBC, 2023.

Wazlawick, Raul Sidnei. "Uma reflexão sobre a pesquisa em ciência da computação à luz da classificação das ciências e do método científico." Revista de Sistemas de Informação da FSMA 6 (2010): 3-10.