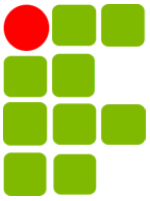


INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – IFPB
CAMPUS CAMPINA GRANDE
COORDENAÇÃO DO CURSO SUPERIOR DE ENGENHARIA DE COMPUTAÇÃO

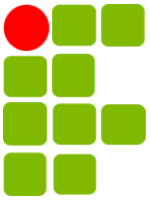
AVALIAÇÃO DA SEGURANÇA DE ACESSO À NOTA FISCAL DE CONSUMIDOR ELETRÔNICA (NFC-e): UM ESTUDO DE CASO EM EMPRESAS NO MUNICÍPIO DE CAMPINA GRANDE - PB

ESTEVÃO HOLANDA LIMA JUNIOR

JONATHAN SATURNINO PONTES DE SOUTO

Orientador: Elmano Ramalho Cavalcanti

Campina Grande, 2025
®Estevão Holanda Lima Júnior
®Jonathan Saturnino Pontes de Souto



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – IFPB
CAMPUS CAMPINA GRANDE
COORDENAÇÃO DO CURSO SUPERIOR DE ENGENHARIA DE COMPUTAÇÃO

AVALIAÇÃO DA SEGURANÇA DE ACESSO À NOTA FISCAL DE CONSUMIDOR ELETRÔNICA (NFC-e): UM ESTUDO DE CASO EM EMPRESAS NO MUNICÍPIO DE CAMPINA GRANDE - PB

ESTEVÃO HOLANDA LIMA JUNIOR

JONATHAN SATURNINO PONTES DE SOUTO

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação, do Instituto Federal da Paraíba – Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Elmano Ramalho Cavalcanti

Campina Grande, 2025

Ficha Catalográfica

Catalogação na fonte:
Ficha catalográfica elaborada por Gustavo César Nogueira da Costa - CRB 15/479

L732a Lima Júnior, Estevão Holanda
Avaliação da segurança de acesso à Nota Fiscal de Consumidor Eletrônica (NFC-e): um estudo de caso em empresas no município de Campina Grande - PB / Estevão Holanda Lima Júnior, Jonathan Saturnino Pontes de Souto. - Campina Grande, 2025.
65 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Engenharia de Computação) - Instituto Federal da Paraíba, 2025.
Orientador: Prof. Elmano Ramalho Cavalcanti.

1. Engenharia de computação 2. Nota Fiscal Eletrônica 3. Segurança da informação 4. Proteção de dados-Análise de vulnerabilidades I. Cavalcanti, Elmano Ramalho II. Título.

CDU 336.2

AVALIAÇÃO DA SEGURANÇA DE ACESSO À NOTA FISCAL DE CONSUMIDOR ELETRÔNICA (NFC-e): UM ESTUDO DE CASO EM EMPRESAS NO MUNICÍPIO DE CAMPINA GRANDE - PB

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação, do Instituto Federal da Paraíba – Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de Bacharel em Engenharia de Computação.

Conceito:_____ Data:___/___/____

Elmano Ramalho Cavalcanti
Orientador

Membro da Banca

Membro da Banca

Campina Grande, Paraíba, Brasil
Janeiro/2025

Dedicatória (Estevão)

Dedico este trabalho a toda minha família, em especial aos meus pais Estevão Cacimiro de Lima e Erileide Cunha de Holanda Lima, que acompanharam de perto a minha jornada e se fizeram presentes em todos os momentos.

Dedicatória (Jonathan)

Dedico esta pesquisa a minha família, que me apoiaram desde o começo nessa jornada e me fizeram perceber o valor do sacrifício, da fé e da resiliência, em especial meu pai João Tavares de Souto, minha mãe Naíza Izabel Soares de Pontes, meu irmão João Henrique S. Pontes de Souto, minha tia Maria Tavares de Souto, minha avó materna Maria Eletice Soares de Pontes *(in memoriam)* e meu avô paterno Severino Saturnino de Souto *(in memoriam)*, como também a minha namorada Rafaela Barbosa Santos, que me apoiou nos momentos difíceis e esteve presente em todos os momentos que precisei.

Agradecimentos (Estevão)

Gostaria de agradecer primeiro a Deus, antes de tudo, pelo dom da vida e por sempre garantir que minhas viagens fossem seguras, sei que ele é o principal responsável por todas as coisas, todos os eventos e sei que sem ele nada disso seria possível, hoje eu entendo qual o meu lugar no mundo e consigo ver o que ele preparou.

Também gostaria de agradecer à minha família, meu pai (Estevão Cacimiro) e a minha mãe (Erileide), sempre foram a base de tudo na minha vida, praticamente no fim da graduação eles se fizeram presentes, me apoiando no que eu precisasse sem deixar que eu desistisse, eu não tenho palavras para expressar minha gratidão a essas duas pessoas que Deus escolheu para serem meus pais, que me educaram a partir dos princípios deles.

Agradeço a minha namorada (Ana Clara) que sempre se dispôs a me ouvir, ajudar e incentivar no que podia durante todo esse percurso, ela me ajudou a tornar o processo mais fácil e mesmo quando não entendia nada, me apoiou, em todas as decisões e/ou quando eu queria tentar algo novo, esteve presente em quase todo processo, sempre me incentivando a chegar no meu objetivo final.

Não poderia deixar de expressar minha gratidão ao meu cunhado Renato Eduardo Marques por todo apoio e ajuda durante esta jornada, sem ele tenho certeza que seria muito mais difícil.

Ainda, expresso minha gratidão aos professores deste curso, cada um deixou sua marca e apesar de algumas divergências foi uma honra ter vocês como mestres, que antes qualquer conteúdo expressavam seu amor pela tecnologia, isso fez com que eu me apaixonasse ainda mais por ela, desde o início as palavras de vocês me fascinaram e hoje posso dizer com toda segurança que não me imagino fazendo outra coisa.

Também agradeço ao meu orientador Elmano, que aceitou nos orientar mesmo quando eu achava que estava tudo perdido, o senhor mostrou novos caminhos e deu direcionamentos que foram essenciais para concluir este trabalho, sei que não fui um orientando fácil, mas muito obrigado pela paciência.

Gostaria de expressar minha gratidão ao meu colega e amigo Jonathan por ter enfrentado este desafio comigo.

“Por último, mas não menos importante, eu quero me agradecer por acreditar em mim, eu quero me agradecer por fazer todo esse trabalho duro, eu quero me agradecer por não ter dias de folga, eu quero me agradecer por nunca desistir.”

Agradecimentos (Jonathan)

Quero agradecer primeiramente a Deus, por todas as suas providências em minha vida, por cada vez que senti sua presença nos meus melhores e piores dias me mostrando que sempre estava comigo, pela minha saúde e todos os livramentos onde o senhor colocou a mão e me tirou de toda e qualquer agonia.

Ao meu pai, que me moldou a ser a pessoa que sou hoje, por me mostrar que o trabalho árduo, a honestidade e a palavra firme é o que edifica o homem, agradeço por cada eu te amo dito ao seu modo, e a minha mãe, meu maior exemplo de fé, resiliência, superação e minha maior inspiração acadêmica, gratidão ao divino por você hoje estar podendo me ver realizar este sonho.

A todos os meus tios e tias, em especial minha tia Maria Tavares de Souto, que sempre me acolheu de braços abertos desde a infância, me ensinando, encorajando, apoiando e abençoando cada decisão minha.

A minha namorada Rafaela Barbosa Santos, que me acompanhou desde quase toda minha carreira acadêmica, sempre me ajudando em todo este árduo processo, me incentivando e me ensinando sempre a melhorar como aluno e pessoa. Gratidão por tudo que tem feito por mim até aqui, você é ímpar na minha vida e me faz ser melhor.

Ao meu irmão mais novo João Henrique, meu fiel escudeiro, que sempre me mostrou seu lado mais puro de amor fraternal, obrigado por confiar em mim sempre que precisei. Essa vitória também é sua!

Aos meus amigos, que influenciaram diretamente ou indiretamente na minha jornada, tenho todos como minha segunda família, obrigado por todo apoio, risadas, reflexões, vivências e superações, em especial a João Gabriel, Rennan, Raniel, Luana, Raphael, Guilherme, João Marcos, José Mateus e Eufrásio Ramalho, com certeza guardarei todos vocês comigo, do ensino médio ao final da vida.

Ao nosso orientador, Elmano Ramalho Cavalcanti, por ter nos apoiado nesta jornada e acreditar no nosso potencial para a realização deste trabalho, sempre nos mostrando, auxiliando e incentivando no que fosse necessário, meus sinceros agradecimentos.

Ao corpo docente do curso Engenharia de Computação, por todo conhecimento e experiências, além de acender cada dia mais a chama do amor à tecnologia e todas as suas nuances. Minha gratidão e respeito.

Por fim, agradeço ao meu amigo e colega Estevão, que enfrentou esta jornada comigo.

Epígrafe

“A tecnologia é a arte de transformar a ciência em soluções para o cotidiano.”

(Autor desconhecido).

Porque sou eu que conheço os planos que tenho para vocês', diz o Senhor, 'planos de fazê-los prosperar e não de causar dano, planos de dar a vocês esperança e um futuro.
(Jeremias 29:11)

SUMÁRIO

1. Introdução 19
2. Objetivos 21
2.1. Objetivo geral 21
2.2. Objetivos específicos 21
3. Fundamentação Teórica 22
3.1. Nota Fiscal eletrônica (NF-e) 22
3.2. Nota Fiscal de Consumidor Eletrônica (NFC-e) 23
3.3. Princípios de Funcionamento e Segurança da NF-e/NFC-e 24
3.3.1. Chave de Acesso 25
3.4. Geração de Números Pseudo-aleatórios 26
3.5. Situação das Empresas de Campina Grande em 2017 28
4. Metodologia 32
5. Levantamento e Análise dos Dados 34
6. Ferramenta para detecção do nível de segurança da NF-e/NFC-e 44
6.1. Tecnologias Utilizadas 44
6.2. Implementação do Backend 45
6.3. Endpoints da API 45
6.4. Funcionalidades de Análise 46
6.5. Implementação do Frontend 47
6.6. Comunicação com o Backend 48
6.7. Formatos de Dados 48
6.8. Resultado da Análise do Serviço 49
7. Análise técnica de sistemas de NFC-e de código aberto 52
8. Trabalhos Relacionados 58
9. Considerações Finais 60
Referências 62

LISTA DE SIGLAS E ABREVIATURA

AAMM - ano e mês de emissão

API - Interface de Programação de Aplicações

cDV - dígito verificador

cNF - Código da nota fiscal

CPF - Cadastro de Pessoa Física

cUF - Código da Unidade Federativa

DANFE - Documento Auxiliar da Nota Fiscal Eletrônica

GNU - GNU's not Unix

IA - Inteligência Artificial

LCG - Linear Congruential Generator

LFSRS - Linear Feedback Shift Registers

LGPD – Lei Geral de Proteção de Dados Pessoais

mod - modelo do documento

NFC-e - Nota Fiscal do Consumidor Eletrônica

NF-e - Nota Fiscal Eletrônica

NFS-e - Nota Fiscal de Serviços Eletrônica

PB - Paraíba

PRNG - Gerador de Números Pseudo Aleatórios

procEmi - Processo de emissão

QR Code - Quick Response Code

RE - Regular Expressions

SEFAZ - Secretaria da Fazenda

SER - série do documento, contendo três dígitos

SPED - Sistema Público de Escrituração Digital

tpEmis - forma de emissão, contendo um dígito

Web - World Wide Web

LISTA DE FIGURAS

Figura 1- Trecho do leiaute do DANFE da NFC-e..... 24
Figura 2: Estrutura do Projeto.....45
Figura 3: Classe NFCeAnalyzer..... 45
Figura 4: Endpoint de Análise (POST).....46
Figura 5: Função de integração com API de análise de chaves.....48
Figura 6: Tela da ferramenta de análise..... 50
Figura 7 - Função-método geraCodigoRandomico do projeto nfe.....52
Figura 8 - Histórico de emissão de notas fiscais do restaurante de renome em Campina Grande.....53
Figura 9 - Função-método completarComZerosAEsquerda do projeto Java_NFe 54
Figura 10 - Construtor da classe ChaveUtil do projeto Java_NFe.....55
Figura 11 - Utilização da função para gerar a chave de acesso do projeto DFe.NET..... 55
Figura 12 - Criação da função ObterChave do projeto DFe.NET..... 56
Figura 13 - Criação da função GetRandom do projeto DFe.NET..... 56
Figura 14 - Método random do projeto sped-nfe..... 57

LISTA DE QUADROS

Quadro 1- Levantamento sobre as vulnerabilidades na geração do código numérico da nota fiscal usado por alguns sistemas em 2017..... 30
Quadro 2 – Levantamento do total de empresas agrupadas por setor nos anos de 2017 e 2024..... 34
Quadro 3 - Total de empresas por porte em 2024..... 38

LISTA DE GRÁFICOS

Gráfico 1 - Níveis de vulnerabilidade das empresas em 2024..... 38
Gráfico 2 - Distribuição dos métodos não seguros de geração do cNF.....39
Gráfico 3 - Distribuição de vulnerabilidade por porte da empresa.....39
Gráfico 4 - Comparação de empresas seguras na emissão da NFC-e por setor nos anos de 2017 e 2024..... 41
Gráfico 5 - Comparação de empresas vulneráveis na emissão da NFC-e por setor nos anos de 2017 e 2024..... 42

AVALIAÇÃO DA SEGURANÇA DE ACESSO À NOTA FISCAL DE CONSUMIDOR ELETRÔNICA (NFC-e): UM ESTUDO DE CASO EM EMPRESAS NO MUNICÍPIO DE CAMPINA GRANDE - PB

Estevão Holanda Lima Junior

Jonathan Saturnino Pontes de Souto

[1]Instituto Federal de Ciência e Tecnologia– IFPB - Campus Campina Grande (IFPB) CEP 58432-300– Campina Grande - PB– Brazil

Resumo

Este trabalho revela que dados de, potencialmente, milhões de cidadãos e empresas do Brasil estão passíveis de serem acessados por qualquer pessoa. Nosso objetivo inicial foi analisar se os sistemas utilizados por empresas atendem à norma de segurança na geração da chave de acesso da Nota Fiscal de Consumidor Eletrônica (NFC-e). O levantamento detalhou o tipo de código numérico da nota fiscal (cNF) usado por mais de 200 empresas de Campina Grande - PB, comparando os resultados com dados coletados em 2017 pelo professor Elmano Cavalcanti. Foi desenvolvido um serviço web que, a partir de chaves de acesso, infere o algoritmo e informa o nível de vulnerabilidade dos dados contidos na NFC-e. Como uma contribuição adicional, também foram avaliados os métodos de geração do cNF em sistemas populares de código aberto de emissão de NF-e/NFC-e disponíveis na plataforma GitHub. A pesquisa concluiu que 52% das empresas ainda adotam práticas vulneráveis na emissão de notas fiscais por meio dos softwares utilizados. O webservice desenvolvido pode ser utilizado por qualquer empresa que queira descobrir se encontra-se vulnerável. Recomenda-se fiscalização rigorosa, conscientização e correção de sistemas que não garantam a aleatoriedade do cNF, destacando a importância do uso adequado de geradores pseudo-aleatórios para fortalecer a segurança.

Palavras chave: Geração de Chaves, Pseudorandomicidade, NF-e, Cibersegurança Fiscal.

Abstract

This study reveals that data from potentially millions of Brazilian citizens and companies can be accessed by anyone. Our initial objective was to analyze whether the systems used by companies comply with security standards in generating the access key for the Electronic Consumer Invoice (NFC-e). The study detailed the type of numerical code of the invoice (cNF) used by more than 200 companies in Campina Grande - PB, comparing the results with data collected in 2017 by Professor Elmano Cavalcanti.A web service was developed that, based on access keys, infers the algorithm and informs the level of vulnerability of the data contained in the NFC-e. As an additional contribution, the cNF generation methods in popular open-source NF-e/NFC-e issuance systems available on the GitHub platform were also evaluated.The research concluded that 52% of companies still adopt vulnerable practices when issuing invoices through the software they use. The developed web service can be used by any company that wants to determine if it is vulnerable. Strict oversight, awareness, and correction of systems that do not ensure the randomness of the cNF are recommended, highlighting the importance of properly using pseudo-random generators to enhance security.

Keywords: Key Generation, Pseudorandomity, NF-e, Fiscal Cybersecurity.

1. Introdução

Atualmente, mais de 60 milhões de notas fiscais de consumidor eletrônica (NFC-e) são emitidas diariamente no Brasil[1]. Esses documentos contêm dados sobre a empresa, os produtos que foram vendidos, a forma de pagamento e, em alguns casos, sobre o consumidor que os adquiriu, podendo incluir seu nome, CPF, endereço e número de telefone. Como também, é possível encontrar dados da lista de produtos que foram vendidos, com suas descrições, quantidades e preços unitários e totais, bem como dos descontos e também informações sobre as alíquotas dos impostos.

A princípio, o acesso a esses dados deveria ser permitido apenas para a empresa que vendeu o produto (emissor), o destinatário/comprador (em geral, uma empresa se for uma NF-e e uma pessoa física se for NFC-e) e, eventualmente, para o contador da empresa, sempre com o uso do certificado digital. Entretanto, para fins de praticidade para os cidadãos, o acesso é permitido via portal web para qualquer pessoa que possua a chave de acesso da nota fiscal. Utilizando o celular, por exemplo, o consumidor pode simplesmente apontar a câmera para o código QR impresso no DANFE (Documento Auxiliar da Nota Fiscal Eletrônica) e ser encaminhado para o portal da Secretaria da Fazenda (SEFAZ) da unidade federativa para visualização dos dados da nota. Este meio agiliza a consulta das informações, permitindo com que os consumidores possam verificar a autenticidade das notas e se o cumprimento das obrigações fiscais está ocorrendo por parte dos vendedores. Como também, essa funcionalidade é um passo importante na digitalização, contribuindo à redução do uso de papel e promovendo práticas mais sustentáveis aos comércios brasileiros.

Conforme é descrito na Seção 3.3.1, a chave de acesso, que contém 44 dígitos, deveria sempre possuir uma parte gerada aleatoriamente. Entretanto, nada impede que o sistema utilizado pela empresa não atenda a esse requisito de segurança[2]. De fato, de acordo com dados não publicados levantados em

[1] https://nfce.encat.org/estatisticas/

[2] Na realidade, com o passar do tempo, os sistemas das secretarias da fazenda passaram a fazer algumas poucas verificações simplificadas, rejeitando notas cujo código da nota fiscal (cNF) tenha valores como uma sequência única do mesmo número (ex: 11111111) ou sequências crescentes como 12345678. Também passou a rejeitar os casos em que o cNF é igual ao número da nota fiscal (nNF):

2017 pelo professor Elmano Ramalho Cavalcanti, mais de 70% das empresas do varejo no município de Campina Grande (PB) estavam vulneráveis a vazamento de dados (empresariais e pessoais) por utilizar sistemas que geravam chaves não aleatórias. Torna-se, evidentemente, imprescindível conhecer a situação atual desse cenário.

Os objetivos deste trabalho são apresentados na Seção 2. Em seguida, na Seção 3, apresentamos o referencial teórico, explicando o que é e como funciona a nota fiscal eletrônica, bem como explicando o conceito da chave de acesso e de geradores pseudo-aleatórios. Finalizamos a Seção 3 com os dados levantados em 2017 (ano de início da operação da NFC-e na Paraíba) sobre a situação da vulnerabilidade de empresas em Campina Grande. Os materiais e métodos utilizados nesta pesquisa estão descritos na Seção 4. As seções seguintes, 5 a 7, procuram trazer o cumprimento dos objetivos específicos. Em sequência, apresentamos alguns trabalhos relacionados (Seção 8), finalizando o trabalho com as considerações finais na Seção 9.

---

https://www.projetoacbr.com.br/forum/topic/79251-c%C3%B3digo-num%C3%A9rico-em-formato-inv%C3%A1lido/

2. Objetivos

2.1. Objetivo geral

O objetivo principal deste projeto é investigar se os sistemas utilizados por empresas emissoras de Nota Fiscal de Consumidor Eletrônica (NFC-e) estão gerando o código numérico da nota fiscal (cNF) de acordo com os padrões técnicos nacionais propostos pelo órgãos tributários.

2.2. Objetivos específicos

- OE1: Realizar um levantamento detalhado sobre o tipo de código numérico da nota fiscal (doravante denominado cNF) utilizado por empresas do varejo no município de Campina Grande - PB, comparando os resultados obtidos com os dados coletados pelo professor Elmano Cavalcanti em 2017;
- OE2: Desenvolver um serviço web que, a partir de uma ou mais chaves de acesso de uma mesma empresa, possa inferir o algoritmo utilizado para a geração do cNF, informando o nível de vulnerabilidade de acesso indevido aos dados das NFC-e da empresa;
- OE3: Avaliar o método de geração do cNF utilizado pelos sistemas de emissão de NF-e/NFC-e de código aberto mais populares disponibilizados na plataforma GitHub.

Dessa forma, este estudo tem como objetivo não apenas identificar e entender as vulnerabilidades do sistema, mas também criar recomendações práticas que ajudem a melhorar sua segurança e eficácia.

Os resultados do objetivo específico 1 são apresentados na Seção 5. A Seção 6 abordará o OE2, enquanto que o OE3 será discutido na Seção 7.

3. Fundamentação Teórica

A seção a seguir contempla os principais tópicos necessários para a compreensão do problema alvo da pesquisa: a Nota Fiscal eletrônica (NF-e), Nota Fiscal de Consumidor eletrônica (NFC-e), a chave de acesso utilizada para consulta da nota na SEFAZ e o conceito de gerador de números pseudo-aleatório (PRNG).

3.1. Nota Fiscal eletrônica (NF-e)

Conforme relatado por Rodrigues et al. (2016), os registros contábeis começaram a ser feitos manualmente, por meio do preenchimento de livros específicos. Com o tempo, essa prática foi sendo substituída pelo uso de máquinas, que surgiram em resposta às necessidades dos procedimentos contábeis. Hoje, a digitalização transformou a forma como a escrituração é realizada, permitindo que seja feita de maneira eletrônica. Essa mudança não apenas aumentou a eficiência do trabalho, mas também melhorou a qualidade dos serviços prestados.

Em 2004, no intuito de modernizar e informatizar o processo de emissão de notas fiscais, seguindo as tendências mundiais de governo eletrônico (MANOHARAN et al., 2021), o projeto Nota Fiscal eletrônica (NF-e) foi concebido e coordenado pelo ENCAT (Encontro Nacional dos Coordenadores e Administradores Tributários Estaduais), sendo desenvolvido em parceria com a Receita Federal do Brasil.

É importante compreender que o surgimento do projeto da NF-e está associado a um projeto maior, que culminou com o desenvolvimento do Sistema Público de Escrituração Digital (SPED), o qual foi projetado para melhorar a relação entre o fisco e o contribuinte, visando a modernização e simplificação das obrigações fiscais no Brasil (BONFIM et al., 2012). Outros benefícios do projeto são, de acordo com Oliveira et al. (2021) (OLIVEIRA et al., 2021), a redução dos custos de uso e armazenamento de papel e o aperfeiçoamento no combate à evasão e à elusão fiscal.

A NF-e é um documento fiscal digital (cujo formato é na extensão XML) que substitui as tradicionais notas fiscais em papel. Ela é gerada e transmitida eletronicamente, via serviços web (web services), pelas empresas para a Secretaria da Fazenda (SEFAZ), permitindo o registro e a fiscalização das operações de forma mais eficiente e segura (BRASIL, 2024c).

Por motivos semelhantes, outros documentos fiscais eletrônicos foram desenvolvidos pelo poder público, a exemplo da nota fiscal de serviços eletrônica (NFS-e) e da nota fiscal de consumidor eletrônica (NFC-e) (CUNHA, 2022). Esta última é bastante semelhante à NF-e, sendo utilizada na venda de produtos para consumidor final (NFC-e).

A implantação do projeto NF-e teve início em 2006. Dados oficiais de janeiro de 2025 indicam que já foram emitidas mais de 47 bilhões de notas fiscais, por um número que ultrapassa 2 milhões de empresas emissoras (BRASIL, 2024a).

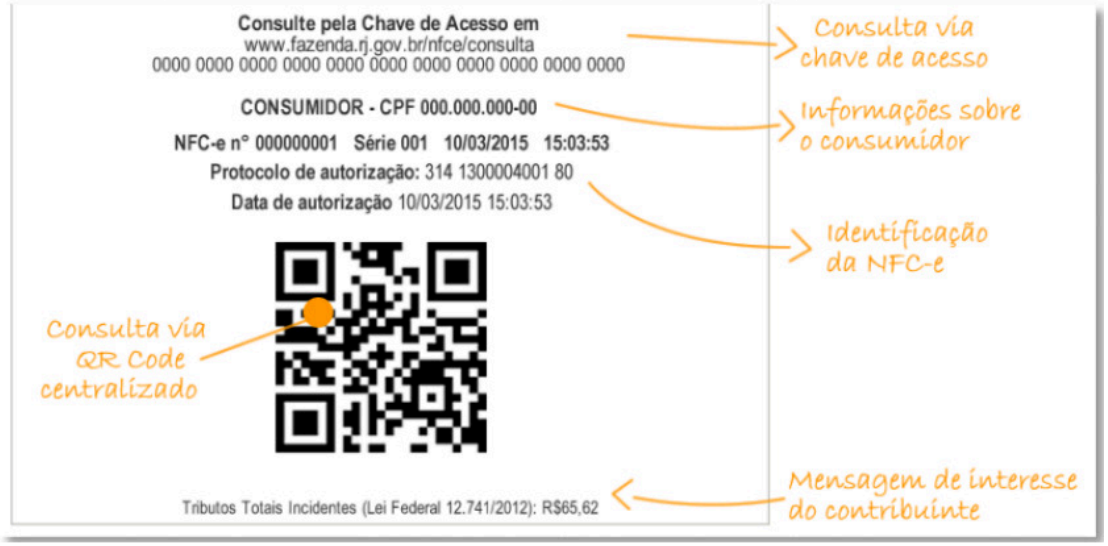
3.2. Nota Fiscal de Consumidor Eletrônica (NFC-e)

Dando sequência ao sucesso do projeto da NF-e, a NFC-e foi idealizada para substituir os documentos fiscais em papel utilizados no varejo (i.e., o cupom fiscal e a nota fiscal modelo 2) (GONZAGA, 2017). Em termos geográficos, a abrangência da NF-e é nacional, enquanto que a da NFC-e é estadual, devendo ser consultada exclusivamente no portal da SEFAZ da respectiva unidade federativa. Por conseguinte, diferentemente da NF-e, a adoção da NFC-e ocorreu de forma independente entre os estados, cada um estabelecendo seu próprio cronograma de obrigatoriedade (CAVALHEIRO, 2022).

Atualmente, a NFC-e está presente em todas as unidades da federação. No momento da escrita deste documento, já haviam sido cadastrados quase 1,9 milhões de emissores de NFC-e, os quais geram aproximadamente 70 milhões de documentos diariamente, totalizando quase 135 bilhões desde o início da operação, em 2013 (ENCAT, 2024a).

Por ser, em grande parte, destinada à pessoa física, a NFC-e tem algumas particularidades. No ato da compra, o consumidor recebe o Documento Auxiliar da Nota Fiscal Eletrônica (DANFE)(BRASIL, 2024b), conforme ilustrado na Figura 1, o qual contém, além dos dados dos produtos adquiridos, indicações para alguns campos importantes como a chave de acesso, a numeração da nota fiscal (nNF) e o CPF do consumidor. Uma regra importante que varia entre os estados é a que exige a inserção do CPF na nota para compras acima de determinado limite. Como exemplo, na Paraíba esse limiar é de R$500, na Bahia, R$1000, e de apenas R$200 no Ceará (TECH, 2023).

Figura 1- Trecho do leiaute do DANFE da NFC-e



Fonte: website da Receita Federal.

3.3. Princípios de Funcionamento e Segurança da NF-e/NFC-e

Logo após o pagamento por um produto em um estabelecimento comercial, o sistema da empresa deverá criar o arquivo XML (que representa a NFC-e), assinalá-lo com o certificado digital do emissor (garantindo autenticidade) e transmiti-lo via *web service* à SEFAZ do estado em que a

empresa está cadastrada como contribuinte (COLOMBO; NETO; BARROS, 2022).

De acordo com a versão mais recente do Manual de Orientação ao Contribuinte (ENCAT, 2024b), ao receber o arquivo xml, o sistema da SEFAZ verifica se o mesmo atende a uma série de regras de validação (ENCAT, 2023). Caso aprovado, a SEFAZ autorizará o uso da nota fiscal, armazenando-a em seu banco de dados e disponibilizando o documento (xml) para consulta pela Internet, pelas partes envolvidas (emitente e destinatário) e aos terceiros legitimamente interessados. O acesso à NF-e é feito tanto no site da SEFAZ do estado do emissor como no portal nacional da NF-e, mantido pela Receita Federal, enquanto que o acesso à NFC-e é feito exclusivamente na SEFAZ estadual.

A implementação da NFC-e tem como objetivo principal reduzir os custos com equipamentos e simplificar as obrigações acessórias dos contribuintes, além de permitir que o Fisco acompanhe as transações comerciais em tempo real. Para isso, os contribuintes precisam se cadastrar na SEFAZ. O processo de implantação é acessível, pois não requer grandes investimentos; geralmente, é suficiente ter uma impressora comum, fazer algumas adaptações no *software* já utilizado e garantir o acesso à internet. Além disso, é fundamental escolher uma solução de sistema adequada para a NFC-e, considerando os *softwares* que serão usados para a geração, envio, autorização, impressão e arquivamento das notas fiscais. Essa escolha deve estar alinhada com as legislações nacional e estadual (SEFAZ/RN, 2016; SEFAZ/MG, 2018).

3.3.1. Chave de Acesso

Um dos componentes cruciais da nota fiscal é a chave de acesso, que é composta por 44 dígitos que referenciam metadados do documento fiscal, de acordo com a sequência a seguir (ENCAT, 2024b):

- cUF: indica o código da UF do emitente (e.g., PB é 25);
- AAMM: ano e mês de emissão;
- CNPJ do emitente: 14 dígitos;

- mod: modelo do documento (55 para NF-e e 65 para NFC-e);
- ser: série do documento, contendo três dígitos;
- nNF: número do documento, contendo nove dígitos (geralmente sequencial);
- tpEmis: forma de emissão, contendo um dígito;
- cNF: código numérico, contendo oito dígitos (deve ser aleatório);
- cDV: dígito verificador (módulo 11), contendo um dígito.

O dígito verificador (DV) da chave de acesso da NF-e/NFC-e é baseado em um cálculo do módulo 11 (ENCAT, 2024b), semelhante a como é feito no CPF. O módulo 11 de um número é calculado multiplicando-se cada algarismo pela sequência de multiplicadores [..., 3, 2, 9, 8, 7, 6, 5, 4, 3, 2] …, posicionados da direita para a esquerda (SOUZA; FARIA; PINTO, 2017).

Conforme alerta o mais recente manual da Receita (ENCAT, 2024b, pág. 19), as informações que compõem a chave de acesso podem ser facilmente deduzidas por qualquer pessoa, sendo necessário que o código numérico (cNF) seja uma sequência aleatória, sob o risco de ter os dados da nota acessados por terceiros. Para isso, o programador deve utilizar um bom gerador de números pseudo-aleatórios (PRNG) e escolher convenientemente o valor da semente (*seed*), a fim de não enfraquecer a aleatoriedade do código gerado (BHATTACHARJEE; DAS, 2022). A próxima seção explica o conceito e aplicabilidade dos PRNGs.

3.4. Geração de Números Pseudo-aleatórios

Programas de computador geram números aleatórios utilizando um gerador de Números Pseudo-aleatórios (PRNG), que pode ser definido como um algoritmo que produz uma sequência de números que se aproxima de uma sequência de números aleatórios verdadeiros (CORMEN et al., 2022). Esses números são gerados a partir de uma semente (ou estado inicial) e podem ser reproduzidos se a mesma semente for utilizada novamente. Os PRNGs são amplamente utilizados em diversas áreas, incluindo simulações

computacionais, criptografia, jogos, jogos de azar e reconhecimento de padrões (JOHNSTON, 2018).

Os PRNGs podem ser classificados em três categorias, de acordo com a técnica de construção em-pregada (BHATTACHARJEE; DAS, 2022): (1) Gerador Congruencial Linear (LCG), amplamente usado em linguagens de programação, como na função *rand()* do GNU C; (2) baseados em *Linear Feedback Shift Registers* (LFSRs), a exemplo do bem conhecido Mersenne Twister(MATSUMOTO; NISHIMURA, 1998), e os mais recentes, baseados em Autômatos Celulares.

De acordo com Johnston (2018), um bom PRNG deve ser eficiente, sendo capaz de produzir rapidamente uma grande quantidade de números aleatórios. Também tem que ser reproduzível, sempre gerando a mesma sequência de números para uma mesma semente. Além disso, deve ser possível retornar números com distribuições específicas, como uniforme, norma ou exponencial.

Uma limitação crítica do PRNG é o fato de ser determinístico. Se a semente for conhecida, toda a sequência de números gerada pode ser reproduzida. Essa previsibilidade pode ser um problema em sistemas que o utilizam para fins de segurança, como é o caso do sistema de notas fiscais eletrônicas, como visto anteriormente.

Segundo (MONTEIRO, 2022) um gerador de números pseudo aleatórios (PRNG - *Pseudo Random number Generator*) é uma fórmula matemática que produz uma sequência periódica e determinística de números. Para (BHATTACHARJEE; DAS, 2022), os geradores de números pseudo-aleatórios também são algoritmos simples e determinísticos, mas acrescentam que produzem sequências de números que simulam a aleatoriedade.

Um gerador de números pseudo aleatórios é uma estrutura do tipo

$$(S,\ s_0,\ f,\ U,\ g),$$

com S sendo um conjunto finito de estados, s_0 o estado inicial, f: $S \rightarrow S$ uma função de transformação, U um conjunto finito de símbolos de saída e g: $S \rightarrow U$ uma função de saída. O estado inicial s_0, comumente conhecido como semente, é muito importante nos métodos de PRNG, pois ela é uma das principais responsáveis por definir a sequência de números que serão gerados.

No entanto, pela sequência de valores dos PRNGs serem gerados a partir de uma condição específica (semente) que pode ser determinada, a segurança dos sistemas criptográficos reduz, tornando os projetos suscetíveis a ataques. Desse modo, alguns algoritmos de PRNG não são indicados para o uso de projetos e trabalhos que necessitem de algoritmos mais robustos em relação à segurança. Em contrapartida, de acordo com (MAYER, 2022), os PRNG são importantes para simulações, permitindo repetir um mesmo processo tendo o conhecimento da semente, além de ser útil em jogos e testes estatísticos. Pode-se acrescentar também o fato desses métodos serem mais eficientes para produzir grandes sequências em pouco tempo, terem um baixo custo e facilidade de implementação e utilização.

É interessante ressaltar, como já foi citado por (MONTEIRO, 2022), que por definição, os algoritmos de PRNG são periódicos, ou seja, a sequência de números se repete a longo prazo. No entanto, atualmente os algoritmos possuem períodos tão grandes que esse fato pode ser desprezado.

Ainda de acordo com (BHATTACHARJEE; DAS, 2022) um algoritmo de PRNG é considerado bom quando cumpre as seguintes propriedades: uniformidade, independência, período longo, reprodutibilidade, consistência, sequências disjuntas, permutações, portabilidade, eficiência, cobertura, característica espectral e segurança criptográfica. No entanto, o autor ressalta que a maioria dos PRNGs não cumprem todas as propriedades. Segundo ele, a maioria dos algoritmos falham nas propriedades de sequências distintas, indicando que deveria haver pouca ou nenhuma correlação entre as sequências geradas por diferentes sementes; de permutações, ou seja, cada permutação de um número gerado por PRNG é esperado que sejam igualmente prováveis, caso contrário poderia ser tendencioso e ajudaria a prever números sucessivos; e, por fim, de segurança criptográfica, os números gerados deveriam ser criptograficamente seguros.

3.5. Situação das Empresas de Campina Grande em 2017

Como citado previamente na Introdução (1), em 2017 o autor deste projeto o professor e cientista da computação Elmano Ramalho Cavalcanti realizou um levantamento, em caráter informal, analisando as NFC-e emitidas por mais de 140 estabelecimentos comerciais, dos mais diversos setores, no município de Campina Grande (PB). Os dados encontrados foram alarmantes: quase 75% das empresas estavam vulneráveis por utilizar sistemas que não geravam a chave de acesso da maneira indicada na documentação técnica do projeto NF-e. Ressalta-se que os resultados encontrados não foram divulgados em nenhum tipo de mídia ou veículo de publicação. Devido à grande dimensão da tabela, não ficou adequada sua inserção neste documento. Entretanto, a mesma pode ser acessada no link: https://shorturl.at/dOq1d

Durante a investigação, percebeu-se que vários programas de emissão de NFC-e adotavam formas alternativas e não seguras de gerar o cNF (Quadro 1). Alguns usavam um valor fixo, enquanto outros igualavam o valor do cNF ao valor do número da nota fiscal (nNF), que naturalmente deve assumir um valor incremental. Então, após analisar a chave de acesso de diversas NFC-e, foi possível constatar os seguintes algoritmos comumente usados pelos desenvolvedores na hora de gerar o cNF:

- **Fixo (F)**: quando o valor do cNF é sempre o mesmo, independente da nota. Como exemplo, a chave de acesso 25170921138765000243650010000237261**12345678**4, referente a uma loja varejista de cosméticos, produtos de perfumaria e de higiene pessoal (CNAE 4772500) usava sempre o valor 12345678 para o cNF;
- **Espelho (E)**: quando o valor do cNF é igual ao valor da nNF. Como exemplo, tem-se a chave 25170770097530000690650011**00003415210003415**20, referente a uma farmácia;
- **Espelho+K (E+K)**: equivalente ao método anterior, mas somando-se o valor da nNF a uma constante K. Inclusive, algumas empresas de grande porte estavam enquadradas nessa categoria;
- **Incremental (I)**: quando o valor do cNF cresce de forma incremental, mas não aparenta ter relação direta ao valor da nNF;

- **Aleatória (A)**: quando o valor do cNF aparenta ter sido gerado por um PRNG.

Importante ressaltar que até mesmo as chaves de acesso que possuem geração do cNF de forma aleatória podem, em alguns casos, serem “quebradas”, a depender do algoritmo utilizado para gerar o número pseudo aleatório e da forma como é escolhido o valor da semente (TSAKALAKI, 2022).

Quadro 1- Levantamento sobre as vulnerabilidades na geração do código numérico da nota fiscal usado por alguns sistemas em 2017.

Nome do Sistema	**(Empresa que desenvolveu)**	**Tipo de cNF**
DataSync (v. 1.0.12)	DataSync	Aleatório
Softcom	Softcom	Aleatório
CGS_1.0.62.0	CG Sistema	Espelho
ACBrNFe	Portal ACB	Espelho
Genesis PDV	RBA Sistemas	Espelho
GNF-e (.Net)	PointSystems	Aleatório
Bematech NFC-e	Bematech	Aleatório
b2click-PDV	REDESOFT	Incremental
UNIF-E 6.1.4	Sistema Hiper	Espelho
Linx MID NFC-e	Linx	Aleatório
Stock.Net:3.2.1511	gueppardo.net	Aleatório
NDDigital NFCe 4.5.6	ndd.com.br	Espelho
Comanda Ideal 5.1	PLSoft	Incremental
gdoor	gdoor	Aleatório

Inteligence	inteligence.com.br	Incremental
cupomfiscalsistemas	Cupom Fiscal Sistemas	Espelho
Maximus	Microtecnologia	Aleatório
DJPDV	DJ System	Espelho
ITEC PDV	ITEC	Incremental
Digifarma	Digifarma	Aleatório
NBS	NBS	Espelho
CHShop	Chianca	Espelho

4. Metodologia

As etapas metodológicas desta pesquisa foram executadas com o intuito de atender de maneira eficaz a cada um dos objetivos específicos estabelecidos. A primeira fase consistiu em realizar um levantamento abrangente no município de Campina Grande, focando no tipo de código numérico utilizado nos sistemas de emissão de NFC-e. Para isso, visitamos empresas de diversos setores, empregando um dos três métodos a seguir: (1) realizando compras em estabelecimentos, solicitando as notas fiscais como clientes; (2) procurando por locais de descarte de papel onde consumidores frequentemente descartam DANFEs; (3) ou abordamos gerentes e funcionários de lojas, explicando os objetivos do projeto e solicitando acesso a quaisquer DANFEs disponíveis. Para otimizar a coleta de dados, o coordenador do projeto, já cadastrado no portal da SEFAZ-PB, obteve o histórico de notas emitidas para seu CPF, facilitando a obtenção de um maior volume de informações.

Durante o levantamento, registramos dados relevantes, como CNPJ, razão social e nome fantasia das empresas, além do número e código das notas fiscais, a chave de acesso e a versão do software de emissão. Com esses dados em mãos, prosseguimos com a comparação entre as informações obtidas e os resultados do levantamento realizado em 2017, utilizando *software* de planilha para proceder com a análise dos dados.

Em seguida, desenvolvemos um serviço web baseado na linguagem *python* e utilizando o framework *flask*, voltado para a verificação da qualidade das chaves de acesso. Essa API foi projetada para aceitar uma lista de chaves de acesso como entrada e retornar informações sobre o método utilizado no código da nota fiscal.

Por fim, para atingir nosso objetivo de analisar projetos de comunicação de NF-e/NFC-e, procuramos os projetos mais populares na plataforma GitHub. Nosso critério de seleção incluiu aqueles com maior número de estrelas e *"forks"*, pois isso indicava um envolvimento ativo da comunidade. Demos

prioridade a linguagens de programação como *Java, C++/C#, Delphi* e PHP, amplamente utilizadas no setor do varejo.

5. Levantamento e Análise dos Dados

O levantamento dos dados mais recentes, realizado em 2024, foi conduzido pelos alunos Estevão Holanda e Jonathan Saturnino, com a orientação do professor Elmano Ramalho. As atividades de coleta de dados duraram aproximadamente 60 dias, nos quais foram visitados diversos estabelecimentos para obter informações sobre a emissão de Notas Fiscais de Consumidor Eletrônica (NFC-e). No total, foram consultadas 204 empresas, sendo que 90 delas faziam parte do levantamento anterior de 2017. Deste conjunto, identificamos que 22 empresas haviam encerrado suas atividades, enquanto 9 não puderam ser localizadas durante as buscas. Isso resultou em 115 empresas ativas em 2024 que foram analisadas para a nossa avaliação de um total de 146.

Os dados coletados em 2024 foram organizados em 23 setores diferentes, abrangendo áreas como grandes comércios locais, supermercados, entre outros destacados no Quadro 2. Durante a análise, exploramos a vulnerabilidade das empresas em relação à segurança na emissão do código da NFC-e, classificando-as nos níveis: seguro, alta vulnerabilidade ou baixa vulnerabilidade (Gráfico 1).

Quadro 2 – Levantamento do total de empresas agrupadas por setor nos anos de 2017 e 2024.

Total de empresas por setor		
Setor	**Ano**	
	2017	**2024**
Açougue	3	5

Artigos de escritório e papelaria	3	4
Cama, mesa e banho	1	3
Concessionárias, Autopeças e serviços	4	15
Cosméticos e Beleza	11	9
Eletrodomésticos, etc.	4	7
Farmácias	12	12
Maiores Empresas da Paraíba, (emissão de NF-e)	0	10
Joalheria e bijuteria	1	6
Loja de Brinquedos	2	5
Lojas de informática e celular	9	6
Óticas	1	6
Outros	5	11
Padaria	7	10
Postos de combustível	15	5
Restaurantes, lanchonetes	18	14

Home center e Móveis	5	18
Sapataria	4	5
Supermercados e Mercadinhos	19	19
Varejo e Utilidades	9	8
Vestimenta e Artigos esportivos	14	26
Total	147	204

Fonte: Dados da pesquisa

Adicionalmente, cada empresa foi avaliada quanto ao porte (Quadro 3). O objetivo desse levantamento foi verificar se empresas de menor ou maior porte estariam mais ou menos vulneráveis. A hipótese que tínhamos era de que empresas menores, por restrições orçamentárias, utilizariam soluções de código aberto ou sistemas desenvolvidos por empresas de TI menores, e possivelmente por programadores menos experientes que iriam adotar algum método não seguro de geração do cNF. Por outro lado, imaginávamos que empresas maiores, por utilizar sistemas próprios de emissão ou sistemas de grandes empresas de TI, estariam consequentemente mais seguras. Entretanto, os resultados mostraram que estávamos enganados (Gráfico 3).

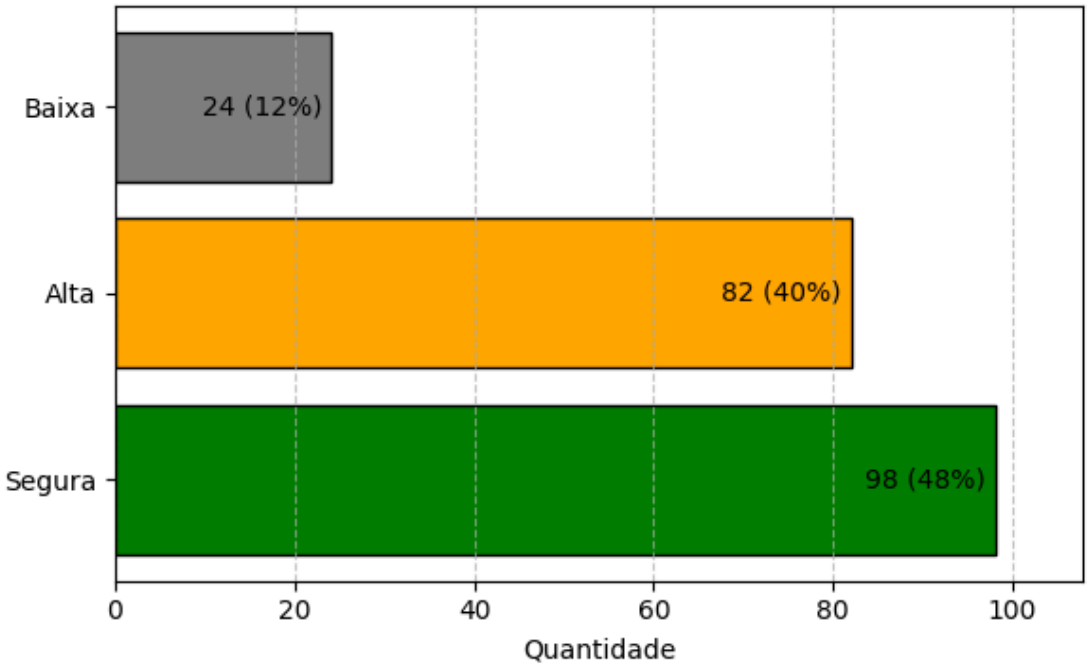
Outro dado coletado importante foi o nome e versão do software emissor. Essa informação pode aparecer no campo *procEmi* da nota fiscal, que indica o ‘processo de emissão’. Entretanto, muitas vezes a empresa informa neste campo apenas a versão do sistema, impossibilitando que encontrássemos qual a empresa que de fato desenvolveu o sistema.

Com base nos dados coletados, nossa análise permitiu identificar cinco métodos/algoritmos não seguros de geração do cNF (Gráfico 3):

- Fixo: quando a empresa utiliza um número (aparentemente aleatório) no cNF. Porém, todas as notas emitidas possuem esse mesmo cNF. Esse método de geração representa uma vulnerabilidade alta;
- Espelho: como atualmente as SEFAZs não permitem mais fazer o cNF=nNF, vários programadores “resolveram” esse “problema” simplesmente fazendo com que cNF=nNF+K, sendo K às vezes igual a 1 ou 2. Esse método de geração representa uma vulnerabilidade alta;
- Incremental: quando o cNF cresce sequencialmente assim como a nNF, porém com valor diferente desta. Esse método de geração representa uma vulnerabilidade alta;
- Parcialmente aleatória: notamos que alguns cNF aparentam ter apenas alguns dos oito dígitos gerados aleatoriamente. Consideramos que isso aumenta a chance de descoberta do código por força bruta e, por isso, classificamos esse tipo de método como vulnerabilidade baixa;
- Baseada no horário: uma das empresas que analisamos utilizava o horário de emissão da nota (hhmmss) na construção do cNF. Entendemos que isso aumenta a chance de descoberta por força bruta e, por isso, classificamos esse tipo de método como vulnerabilidade baixa.

Essa classificação foi importante para compreender as diferentes “soluções” que os programadores costumam utilizar na hora de gerar o cNF. O Gráfico 2 mostra a quantidade de ocorrências dos diferentes tipos de métodos não seguros de geração do cNF. O tipo Incremental tem a maior quantidade, com 42 ocorrências, seguido por Parcialmente Aleatório (23), Fixo (21) e Espelho (19). O tipo Baseado no horário foi encontrado em apenas uma empresa.

Gráfico 1 - Níveis de vulnerabilidade das empresas em 2024



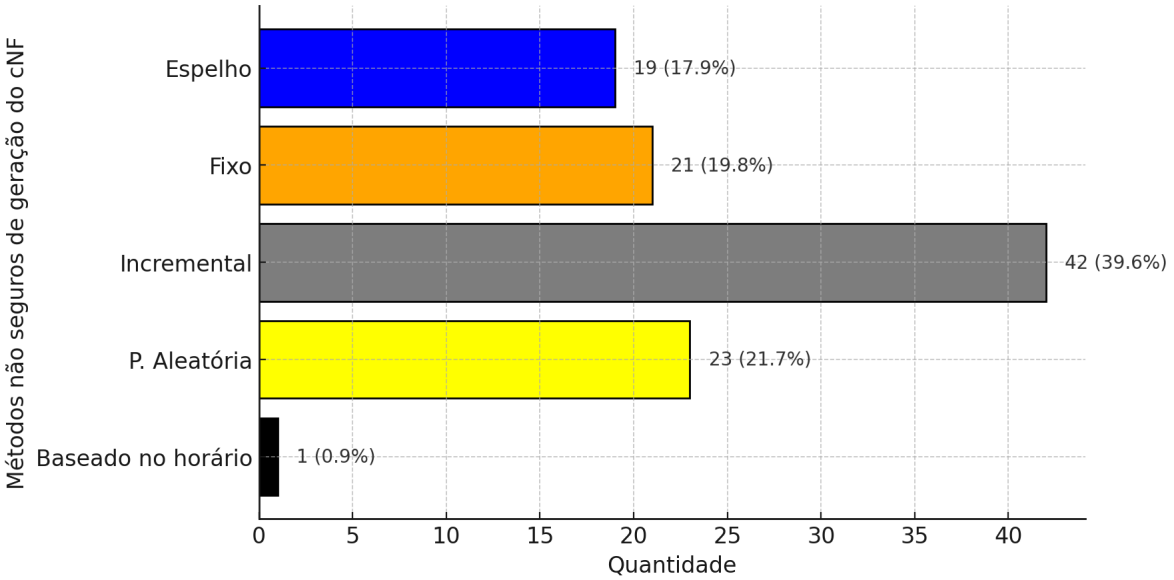
Fonte: Dados da pesquisa

Quadro 3 - Total de empresas por porte em 2024.

Número de empresas por porte	
Micro	98
Pequeno	47
Médio	12
Grande	46
Demais	1

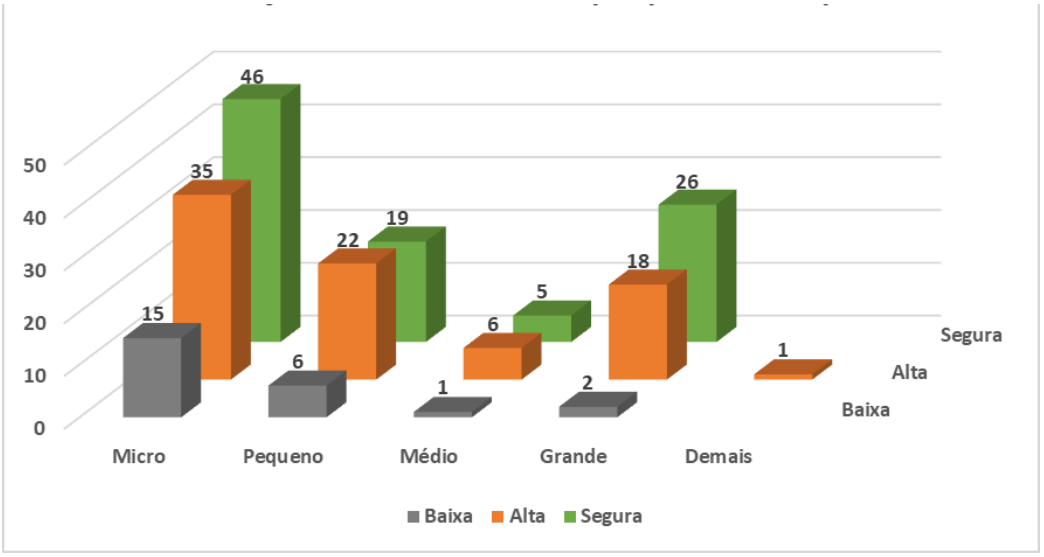
Fonte: Dados da pesquisa

Gráfico 2 - Distribuição dos métodos não seguros de geração do cNF.



Fonte: Dados da pesquisa

Gráfico 3 - Distribuição de vulnerabilidade por porte da empresa



Fonte: Dados da pesquisa

No cenário de 2017, o levantamento indicou que haviam 146 empresas ativas, porém foram consideradas apenas 115 na reanálise de 2024, pela razão de que 22 empresas haviam encerrado suas atividades e, adicionalmente, 9 empresas não puderam ser localizadas durante o novo processo de reanálise. Dentre essas, 86 foram classificadas como vulneráveis, refletindo uma preocupação com a segurança das operações. Por outro lado, 29 empresas

foram consideradas seguras, o que indicava que, apesar da maioria estar enfrentando desafios de vulnerabilidade, havia uma parte do mercado já com práticas de segurança mais eficazes.

Dentre as 86 empresas que eram vulneráveis em 2017, conseguimos localizar e analisar 66 em 2024. O resultado dessa reanálise mostrou avanços importantes: 25 empresas deixaram de ser vulneráveis e agora estão gerando códigos NFC-e de forma aleatória, o que é um indicativo positivo de melhoria na segurança. Entretanto, 36 empresas continuam a apresentar uma alta vulnerabilidade, e cinco empresas, embora com uma vulnerabilidade considerada baixa, têm códigos NFC-e que são parcialmente aleatórios ou se baseiam no horário de emissão, sugerindo a necessidade de correções no software para reduzir as chances de acesso indevido e vazamento de dados sensíveis.

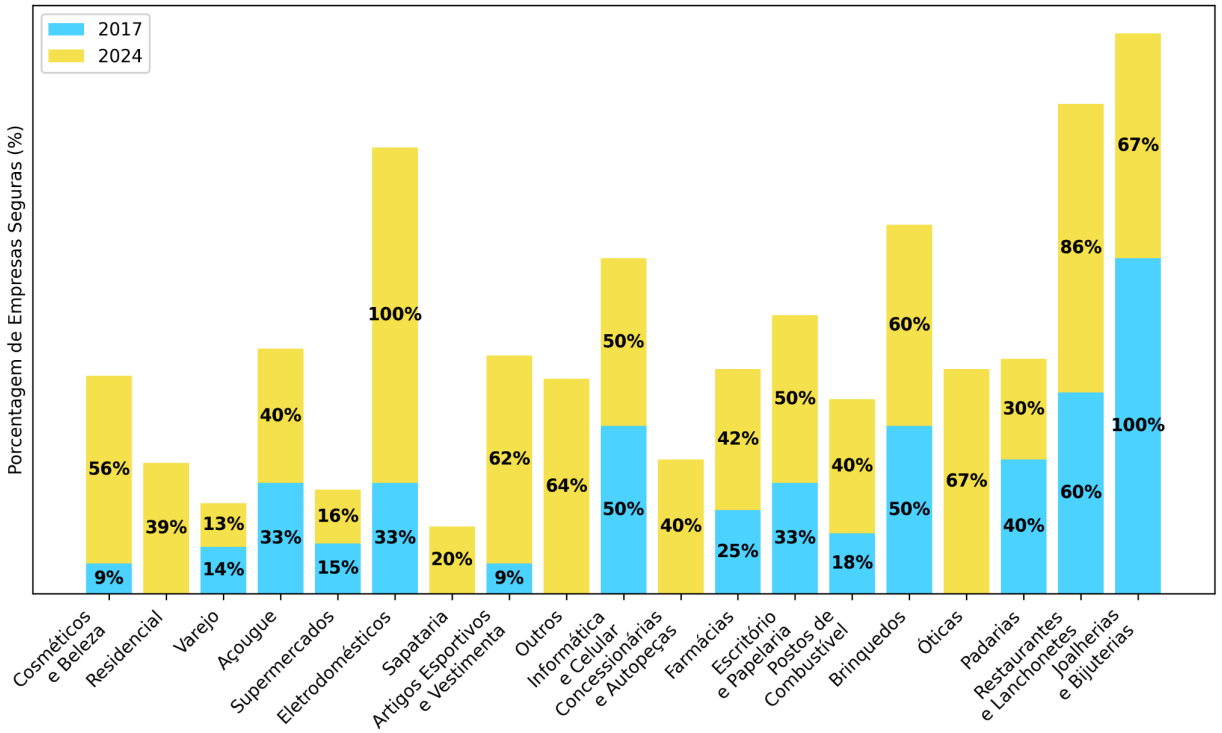
No que diz respeito às 29 empresas que, em 2017, foram catalogadas como seguras, em 2024 analisamos novamente 24 dessas organizações. Os resultados destacaram que 18 empresas conseguiram manter seu status de segurança, reafirmando a eficácia das práticas implementadas ao longo do tempo. No entanto, outras cinco empresas passaram a apresentar uma alta vulnerabilidade, o que destaca que a segurança não é uma condição permanente. E também, apenas uma empresa foi classificada com uma vulnerabilidade baixa, apresentando um código parcialmente aleatório, o que indica que mesmo as empresas consideradas seguras têm áreas que podem ser melhoradas.

Com essa análise podemos perceber a importância do monitoramento contínuo das práticas de segurança, notando que houve mudanças significativas, onde algumas empresas conseguiram alcançar melhorias, mesmo em um ambiente desafiador. Porém, algumas empresas demonstram a necessidade de melhores estratégias para que possa garantir uma segurança de fato eficaz.

Uma análise comparativa dos dados encontrados em 2017 e 2024 sobre segurança na geração da chave de acesso para Notas Fiscais de Consumidor Eletrônica (NFC-e) em Campina Grande, Paraíba, revelou uma evolução

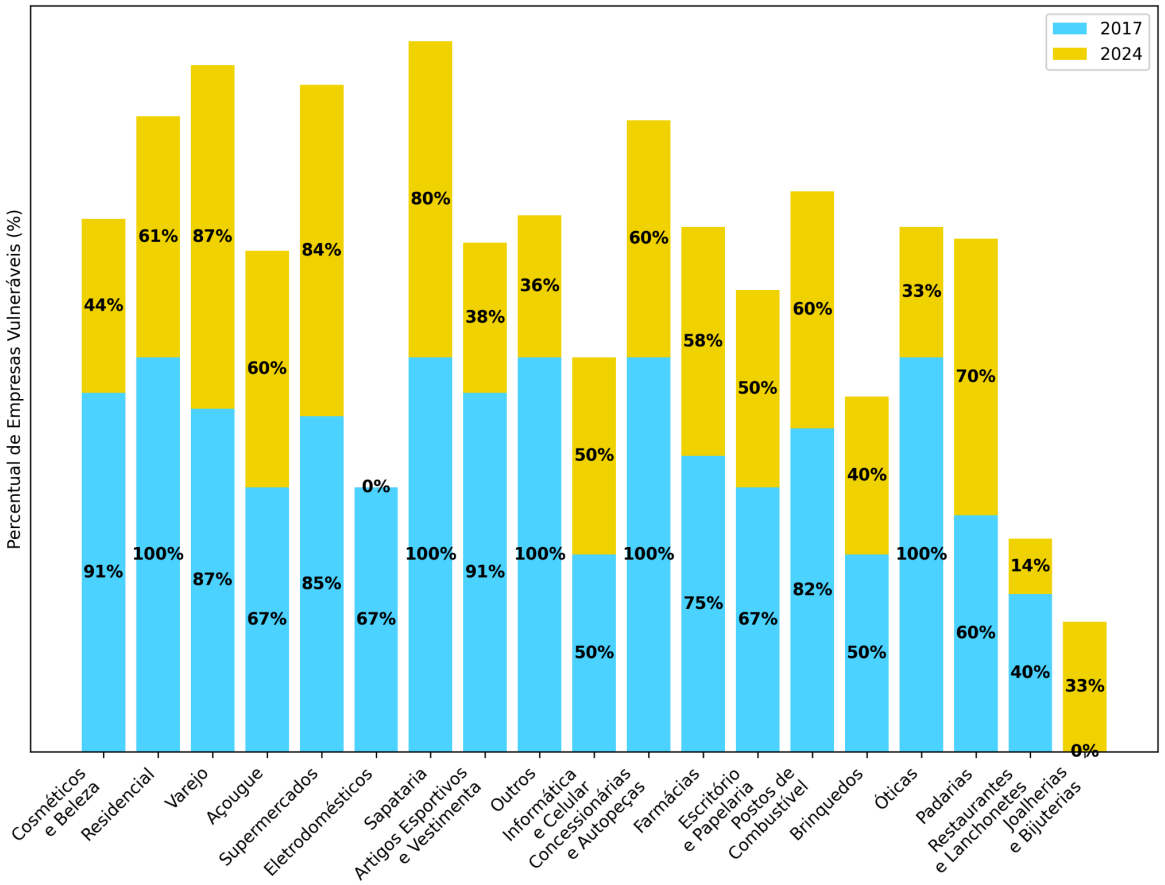
notável na maioria dos setores, como demonstrado no Gráfico 4 e no Gráfico 5. Em 2017, um estudo abrangente foi conduzido envolvendo aproximadamente 140 empresas da região. Os resultados desta análise inicial foram preocupantes, pois mostraram que 75% dessas empresas tinham vulnerabilidades críticas na geração do código NFC-e, que é crucial para a segurança e integridade do NFC-e. Essas vulnerabilidades podem ser atribuídas a práticas que são prejudiciais à segurança da informação, como o uso de algoritmos que geram números sequencialmente predeterminados ou a falha em atualizar os sistemas de segurança. (Silva & Almeida, 2018; Santos et al., 2019).

Gráfico 4 - Comparação de empresas seguras na emissão da NFC-e por setor nos anos de 2017 e 2024



Fonte: Dados da pesquisa

Gráfico 5 - Comparação de empresas vulneráveis na emissão da NFC-e por setor nos anos de 2017 e 2024



Fonte: Dados da pesquisa

Esses dados demonstram uma evolução na segurança das empresas em diversos setores entre 2017 e 2024. Embora alguns setores tenham mostrado melhorias significativas na redução da vulnerabilidade, 14 deles, outros ainda enfrentam desafios consideráveis, 4 setores. Essa análise ressalta a importância de monitorar constantemente as práticas de segurança e assim poder implementar melhorias onde necessário, garantindo um ambiente de comércios mais seguro e confiável.

Os resultados mostraram uma melhoria específica na segurança da geração de NFC-e, com 48% das empresas utilizando métodos seguros e eficazes. Essa evolução pode ser atribuída a vários fatores, como o aumento da conscientização sobre a importância da segurança da informação, regulamentações mais rigorosas implementadas pelas autoridades financeiras

e avanços tecnológicos que possibilitaram o desenvolvimento de soluções de segurança mais sofisticadas e acessíveis.

Levando em consideração os dados obtidos e a taxa de vulnerabilidade no comércio local de Campina Grande apresentados acima, se mostra o quanto cada vez mais a área de segurança da informação vem conseguindo mais espaço ao passar dos anos, tendo em vista que cada vez mais os dados pessoais se tornam cada vez mais sensíveis e críticos em ambientes virtuais, para garantir a confidencialidade e, além do mais, a integridade dos aspectos legais sobre eles, como é o caso da Lei Geral de Proteção aos Dados Pessoais (LGPD), tendo em vista o alto risco de prejuízo às empresas ao se demonstrarem vulneráveis pela escolha do seu software de emissão da NFC-e, por não seguir as normas estabelecidas quanto a estes dados.

como também aos clientes destes determinados comércios que por terem seus dados vazados, tais como: nome completo, endereço, CPF, etc., podem ser vítimas de fraudes e tentativas de golpes por pessoas mal intencionadas que conhecem a vulnerabilidade na chave de acesso.

Além disso, a melhoria observada pode ser explicada pelo investimento em treinamento e capacitação dos profissionais de TI nas empresas, o que levou à adoção de práticas de segurança mais robustas, como o uso de geradores de chaves criptográficas e a implementação de políticas de segurança da informação mais rigorosas. (Gomes & Ribeiro, 2022).

Em conclusão, uma comparação dos dados de 2017 e 2024 mostra uma evolução positiva na segurança da geração da chave de acesso NFC-e em Campina Grande, indicando um avanço notável na modernização das práticas de segurança da informação entre as empresas da região. Esse desenvolvimento não só fortalece a confiança no sistema de emissão de notas fiscais eletrônicas, mas também ajuda a reduzir a fraude financeira e protege os interesses econômicos de todas as partes envolvidas. (Rodrigues & Silva, 2024).

6. Ferramenta para detecção do nível de segurança da NF-e/NFC-e

A ferramenta web para análise de chaves de NFC-e foi implementada utilizando uma arquitetura cliente-servidor, seguindo o padrão REST (*Representational State Transfer*). A aplicação foi desenvolvida utilizando Python com o framework Flask no backend, e HTML, CSS e JavaScript no frontend.

O sistema implementa validações de formato e tratamento de erros para garantir a integridade dos dados, que é a verificação do tamanho da chave (44 dígitos), validação de caracteres numéricos e a verificação de CNPJ consistente entre chaves. Já para o tratamento de erros, o sistema busca encontrar e tratar erros de formato, validação e erros internos do servidor.

O serviço também conta com medidas de segurança na hora de processar a entrada, tais como a organização dos dados, tratamento adequado de erros e a proteção contra injeção de código:

Devido à arquitetura modular existente nesta ferramenta, é permitido a fácil expansão do sistema para a adição de novos padrões de análise, implementação de novos endpoints, integração com outros sistemas e a expansão das funcionalidades de análise.

6.1. Tecnologias Utilizadas

No desenvolvimento do serviço, foi empregada no Backend a linguagem de programação *python* na versão 3.11.9 como linguagem principal de desenvolvimento. Também foi implementada com o framework web *Flask* na versão 3.1.0, e o módulo RE (Regular Expressions) para a validação de padrões.

Já na parte do Front-end, foi utilizado *HTML5* para a estruturação da interface, *CSS3* para se ter uma estilização com o *Bootstrap* 5.1.3, que é um Framework CSS para design responsivo, e *JavaScript* para interatividade e comunicação com o Backend.

6.2. Implementação do Backend

O backend é estruturado de maneira modular, com a classe NFCeAnalyzer(Figura 3) atuando como o elemento principal na análise das chaves. Esta classe incorpora procedimentos para a extração de elementos da chave, verificação do formato e reconhecimento de padrões no cNF.

O backend foi organizado seguindo uma estrutura modular(Figura 2).

Figura 2: Estrutura do Projeto

```
projeto/
├── app.py                # Aplicação principal Flask
├── static/               # Arquivos estáticos
│   └── script.js         # JavaScript do cliente
└── templates/            # Templates HTML
    └── index.html        # Interface principal
```

Fonte: autoria própria

Figura 3: Classe NFCeAnalyzer

```
class NFCeAnalyzer:
    def __init__(self):
        self.vulnerability_levels = {
            'F': 'ALTO',       # Fixo
            'E': 'ALTO',       # Espelho
            'E+K': 'ALTO',     # Espelho + Constante
            'I': 'MÉDIO',      # Incremental
            'A': 'BAIXO',       # Aleatório
            'PA': 'BAIXO'       #Parcialmente Aleatório
        }
```

Fonte: autoria própria.

6.3. Endpoints da API

O serviço expõe dois endpoints principais: O primeiro endpoint, chamado "Endpoint Principal," é acessado usando uma requisição do tipo GET na rota raiz do serviço, indicada por uma barra (/). Este endpoint tem a função de renderizar a interface principal do sistema, fornecendo ao usuário uma

página HTML que atua como uma porta de entrada para o serviço. Quando o servidor recebe essa rota, ele responde com um template HTML que possui todos os elementos necessários para que o usuário insira as chaves de acesso NFC-e/NF-e que deseja analisar. Utilizando tecnologias como HTML, CSS e Bootstrap, esta interface é construída para ser responsiva e visualmente consistente em diversos dispositivos.

O segundo endpoint, chamado "Endpoint de Analyze"(Figura 4), é acessado usando uma requisição POST na rota /analyze. Como é responsável por receber, processar e analisar as solicitações de acesso feitas pelos usuários, este ponto de entrada é essencial para o funcionamento principal do serviço. O cliente envia um payload JSON com uma lista de chaves de acesso, que é então submetido a um procedimento de análise no servidor. Durante este processo, o serviço decompõe as chaves em suas partes componentes, verifica o formato e determina o padrão de geração do código numérico (cNF) usando a classe NFCeAnalyzer. Após a análise, o servidor retorna ao cliente um objeto JSON que contém os resultados, incluindo o padrão identificado e o nível de vulnerabilidade associado a esse padrão. Este retorno permite ao usuário entender a segurança das chaves analisadas e tomar medidas apropriadas, se necessário.

Figura 4: Endpoint de Análise (POST)

```python
@app.route('/analyze', methods=['POST'])
def analyze_keys():
    try:
        data = request.get_json()
        analyzer = NFCeAnalyzer()
        result = analyzer.analyze_pattern(data['keys'])
        return jsonify(result)
    except ValueError as e:
        return jsonify({'error': str(e)}), 400
```

Fonte: autoria própria

6.4. Funcionalidades de Análise

Na implementação do serviço web para análise de chaves de acesso NFC-e, duas funções principais foram desenvolvidas para garantir a eficiência e precisão do processo de análise. A primeira função, denominada *extract_components*, desempenha um papel crucial ao decompor a chave de acesso em seus componentes individuais. Quando uma chave é inserida pelo usuário, esta função verifica seu formato e, em seguida, extrai informações essenciais, como a UF (cUF), o CNPJ, o modelo, a série, o número da nota fiscal (nNF), o tipo de emissão (tpEmis), o código numérico (cNF) e o dígito verificador (cDV). Este processo de decomposição é fundamental para a análise subsequente, pois permite que cada elemento da chave seja avaliado individualmente.

A segunda parte do processo envolve a classe NFCeAnalyzer, que é responsável por implementar a lógica de identificação dos padrões de geração do cNF. Esta classe analisa os valores de cNF extraídos e determina se eles seguem um dos padrões predefinidos. O padrão "Fixo" (F) é identificado quando o cNF tem o mesmo valor para todas as notas. O padrão "Espelho" (E) ocorre quando o cNF é idêntico ao número da nota (nNF). O padrão "Incremental" (I) é reconhecido quando os valores de cNF formam uma sequência com um incremento entre eles. Quando nenhum padrão claro pode ser determinado, o cNF é classificado como "Aleatório" (A), indicando que não há um padrão identificável. Além disso, um padrão "Parcialmente Aleatório" (PA) pode ser identificado quando há alguma previsibilidade, mas não o suficiente para classificá-lo como um dos padrões anteriores.

6.5. Implementação do Frontend

A interface do usuário foi implementada utilizando Bootstrap para garantir responsividade e consistência visual. Os principais componentes incluem: Formulário de entrada para chaves de acesso, área de exibição de resultados, indicadores visuais de vulnerabilidade e detalhamento dos componentes das chaves.

6.6. Comunicação com o Backend

A comunicação cliente-servidor é realizada através de requisições AJAX utilizando a API Fetch(Figura 5). A comunicação com o backend é iniciada quando o usuário interage com a interface do sistema, ao submeter o formulário de análise de chaves. O formulário captura as chaves de acesso inseridas pelo usuário e, ao ser submetido, dispara um evento que é tratado por um script JavaScript no frontend. Este script utiliza a API Fetch para enviar uma requisição POST para o endpoint /analyze do servidor. A requisição POST inclui um payload JSON contendo as chaves de acesso que o usuário deseja analisar. A escolha do formato JSON é motivada pela sua simplicidade e facilidade de manipulação tanto no cliente quanto no servidor. O uso de AJAX para esta comunicação permite que a aplicação seja mais dinâmica, proporcionando uma experiência de usuário mais fluida, já que as respostas do servidor podem ser processadas e exibidas na interface sem a necessidade de recarregar a página inteira.

Figura 5: Função de integração com API de análise de chaves

```
async function analyzeKeys(keys) {
    const response = await fetch('/analyze', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({ keys })
    });
    return await response.json();
}
```

Fonte: autoria própria

6.7. Formatos de Dados

Para a formatação dos dados de entrada e saída, o serviço aceita requisições POST com payload JSON no formato de lista, já na saída, o serviço retorna um objeto JSON contendo o *pattern*, que indica o padrão identificado na geração do cNF, *vulnerability_level* que indica o nível de

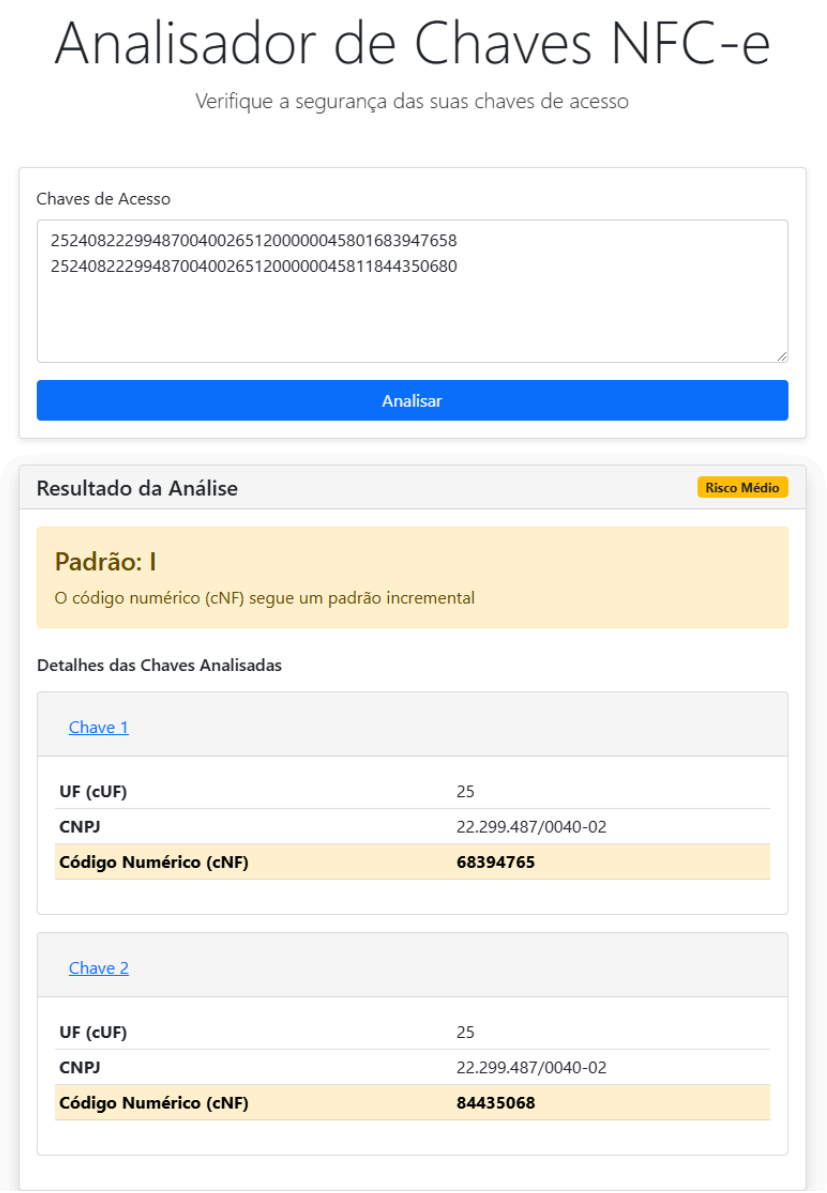
vulnerabilidade identificado de acordo com o padrão detectado e classifica como "ALTO, "MÉDIO" ou "BAIXO e o *description* é a descrição natural do padrão encontrado com uma explicação clara e objetiva (Figura 6).

6.8. Resultado da Análise do Serviço

Ao utilizar o serviço web desenvolvido para análise de chaves de acesso NFC-e, o usuário é guiado por uma interface intuitiva que facilita a entrada e análise das informações. Nesta interface, o usuário insere uma ou mais chaves de acesso no campo designado e aciona o botão "Analisar". O sistema, então, processa essas chaves para identificar o padrão de geração do código numérico (cNF).

Após a submissão, o serviço rapidamente realiza a decomposição das chaves em seus componentes, como UF (cUF), CNPJ, e o próprio cNF. O backend, utilizando a classe NFCeAnalyzer, valida o formato das chaves e compara os valores do cNF para detectar padrões.

Figura 6: Tela da ferramenta de análise



Fonte: autoria própria

No exemplo apresentado, o resultado da análise indica que o cNF segue um padrão incremental. Isso significa que os códigos numéricos das chaves analisadas aumentam de forma consistente, sugerindo um incremento entre eles. Este padrão é visualizado na interface, onde cada chave é listada com seus detalhes, incluindo o cNF destacado.

O sistema classifica o risco associado a este padrão como "Médio", alertando o usuário sobre a previsibilidade potencial dos próximos códigos, o

que pode representar uma vulnerabilidade moderada à segurança. A interface utiliza cores e ícones para comunicar claramente o nível de risco, garantindo que o usuário compreenda as implicações de segurança.

Assim, o serviço não apenas realiza a análise técnica, mas também apresenta os resultados de maneira acessível e informativa, permitindo que os usuários tomem decisões informadas sobre a segurança de suas operações fiscais.

7. Análise técnica de sistemas de NFC-e de código aberto

Para a escolha dos sistemas de *NF-e/NFC-e* no *GitHub* foram selecionados quatro projetos para serem analisados. O foco principal é analisar a forma como é realizada a geração do código numérico de 8 dígitos que compõe a chave de acesso das NF-e/NFC-e.

Iniciando a análise pelos projetos criados na linguagem Java, foi observado que ambos os projetos utilizam a classe *java.util.Random* que utiliza do método de LCG para a geração dos números pseudo-aleatórios. De acordo com o que já vem sendo exposto, sabe-se que o método LCG não é o mais seguro para utilizar em métodos com a necessidade maior de segurança, como é o caso da geração dos números aleatórios para as NF-e e NFC-e [3].

Aprofundando um pouco mais a análise, no projeto intitulado *nfe* foi possível identificar o código que de fato está realizando a geração do código numérico aleatório. O método *geraCodigoRandomico* pertence a classe *NFGeraChave*. Essa função-método gera um número aleatório de 8 dígitos do tipo string, preenchendo com zeros à esquerda no caso do número aleatório gerado possuir menos de 8 dígitos.

Figura 7 - Função-método geraCodigoRandomico do projeto nfe

```
public String geraCodigoRandomico() {
    final Random random = new Random(this.nota.getInfo().getIdentificacao().getDataHoraEmissao().toInstant().toEpochMilli());
    return StringUtils.leftPad(String.valueOf(random.nextInt(100000000)), 8, "0");
}
```

Fonte: Captura do projeto https://github.com/wmixvideo/nfe.git

É possível observar que a semente utilizada para o gerador é definida como o valor em milissegundos do timestamp da data de emissão de uma nota fiscal. Isso ocorre pelo uso dos métodos em sequência *.toInstant().toEpochMilli()* que convertem um objeto da classe tempo para um objeto do tipo *java.time.Instante*, que representa um ponto na linha do tempo com precisão de nanossegundos desde a Epoch (01 de janeiro de 1970 às

[3] Já são conhecidos alguns métodos de quebra do LCG como pode ser visto no repositório https://github.com/lachlan2k/pot-lcg-cracker e no domínio https://www.elttam.com/blog/cracking-randomness-in-java/.

00:00:00), e, em seguida, converte esse valor para um número longo que representa o Epoch em milissegundos. Dessa forma, os números gerados serão os mesmos para uma mesma data e hora.

Essa relação no código gera uma vulnerabilidade de previsibilidade que poderia ser explorada. Usando o valor do timestamp do momento da emissão como semente para o gerador, sabendo que o padrão de emissão segue horários previsíveis, qualquer pessoa que conheça ou consiga estimar o horário de emissão pode recriar a mesma semente e gerar o mesmo código numérico, reproduzindo a mesma lógica usada no sistema.

Um exemplo da fragilidade nesse método são estabelecimentos, como restaurantes e supermercados, que em determinado dia e horário acabam emitindo um grande volume de notas fiscais em um curto período de tempo. Desse modo, mesmo utilizando um gerador de números pseudo-aleatório com o *timestamp* como valor da semente, conhecendo uma nota fiscal, existem grandes chances de identificar notas anteriores e posteriores utilizando métodos de tentativa e erro. Esse fato é possível de ser observado na Figura 8, no histórico de emissões de um restaurante de grande nome*, em Campina Grande,* no dia 19/01/2025 (domingo) em um de seus horários de maior movimento. Nessa imagem é possível identificar a emissão de sete notas fiscais em menos de um minuto e as cinco notas fiscais seguintes com diferença na emissão de cinco segundos entre elas.

Figura 8 - Histórico de emissão de notas fiscais do restaurante de renome em Campina Grande

☑	1581736	2024-01-19 12:00:04	Autorizada	R$ 189,80
☐	1581737	2024-01-19 12:00:09	Autorizada	R$ 35,89
☐	1581738	2024-01-19 12:00:13	Autorizada	R$ 71,80
☐	1581739	2024-01-19 12:00:18	Autorizada	R$ 43,80
☐	1581740	2024-01-19 12:00:22	Autorizada	R$ 35,89
☐	1581741	2024-01-19 12:00:27	Autorizada	R$ 35,89
☐	1581742	2024-01-19 12:00:31	Autorizada	R$ 89,90
☐	1581743	2024-01-19 12:01:35	Autorizada	R$ 35,89
☐	1581744	2024-01-19 12:01:40	Autorizada	R$ 35,89
☐	1581745	2024-01-19 12:01:45	Autorizada	R$ 35,90
☐	1581746	2024-01-19 12:01:50	Autorizada	R$ 48,90
☐	1581747	2024-01-19 12:01:54	Autorizada	R$ 69,80
☐	1581748	2024-01-19 12:08:55	Autorizada	R$ 127,60
☐	1581749	2024-01-19 12:09:43	Autorizada	R$ 84,70

Fonte: SEFAZ-PB

Em contrapartida, existem fatores que podem dificultar o processo de recriação da semente, como o sistema não seguir intervalos fixos, utilizar outros dados para a semente, misturar o timestamp com outros dados únicos (como um identificador de máquina ou outro número aleatório) e validar cada tentativa com os sistemas de emissões para confirmar se os códigos gerados são válidos.

Algumas recomendações de melhoria para o código seria o uso do *SecureRandom* da biblioteca padrão Java, pois é um gerador criptograficamente seguro, não é determinístico e não permite a previsibilidade com base na semente, e a combinação de diferentes fontes para a semente, incorporando além do timestamp outros fatores imprevisíveis, como um identificador único ou uma sequência derivada de um algoritmos mais complexo.

No projeto *Java_NFe* não foi possível identificar de fato o local onde está sendo gerado o código numérico aleatório. No entanto, em conversa com o autor, ele destacou que de fato utilizou a classe *java.util.Random.* Apesar de não localizar o método de geração randômica, foi possível localizar uma função-método chamada *completarComZerosAEsquerda,* pertencente a classe *ChaveUtil*, que executa uma função semelhante a da Figura 7, completando com zeros a esquerda quando o número aleatório gerado tiver menos de 8 dígitos.

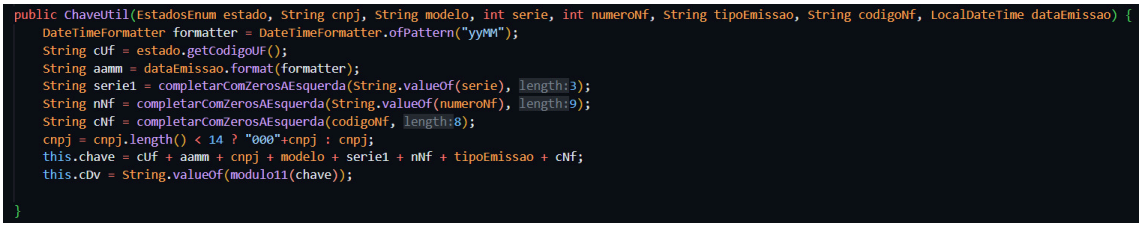
Figura 9 - Função-método *completarComZerosAEsquerda* do projeto *Java_NFe*

```
public static String completarComZerosAEsquerda(String value, int length) {
    int tam = value.length();
    StringBuilder result = new StringBuilder(value);

    for (int i = tam; i < length; i++) {
        result.insert(0, "0");
    }
    return result.toString();

}
```

Fonte: Captura do projeto https://github.com/Samuel-Oliveira/Java_NFe.git

A função-membro da Figura 8 está sendo utilizada no construtor da classe *ChaveUtil*, como pode ser visto na Figura 9.

Figura 10 - Construtor da classe *ChaveUtil* do projeto *Java_NFe*



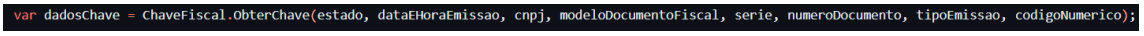
```
public ChaveUtil(EstadosEnum estado, String cnpj, String modelo, int serie, int numeroNf, String tipoEmissao, String codigoNf, LocalDateTime dataEmissao) {
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyMM");
    String cUf = estado.getCodigoUF();
    String aamm = dataEmissao.format(formatter);
    String serie1 = completarComZerosAEsquerda(String.valueOf(serie), length:3);
    String nNf = completarComZerosAEsquerda(String.valueOf(numeroNf), length:9);
    String cNf = completarComZerosAEsquerda(codigoNf, length:8);
    cnpj = cnpj.length() < 14 ? "000"+cnpj : cnpj;
    this.chave = cUf + aamm + cnpj + modelo + serie1 + nNf + tipoEmissao + cNf;
    this.cDv = String.valueOf(modulo11(chave));

}
```

Fonte: Captura do projeto https://github.com/Samuel-Oliveira/Java_NFe.git

Apesar de ambos os projetos utilizarem um método de PRNG considerado não seguro para aplicações que precisam de maior segurança, foi possível notar o cuidado em seguir o recomendado pelo Ministério da Fazenda, gerando o código numérico de forma aleatória ao invés do uso de métodos alternativos. Para maior segurança, o recomendado seria utilizar a classe *java.security.SecureRandom* como citado na seção *Tipos de PRNG.*

Dando continuidade, agora analisando o projeto intitulado *DFe.NET* feito na linguagem *C#* foi possível identificar na pasta *NFe.Utils* a classe *NFe.Utils.NFe* que utiliza a função *ObterChave* (Figura 10), evidenciando a criação da chave de acesso da NF-e.

Figura 11 - Utilização da função para gerar a chave de acesso do projeto *DFe.NET*



```
var dadosChave = ChaveFiscal.ObterChave(estado, dataEHoraEmissao, cnpj, modeloDocumentoFiscal, serie, numeroDocumento, tipoEmissao, codigoNumerico);
```

Fonte: Captura do projeto https://github.com/ZeusAutomacao/DFe.NET.git

Essa função foi gerada na classe *DFe.Utils,* destacando que o campo chamado de *cNF* é o código numérico que compõe a chave de acesso e que esse número é gerado pelo emitente (Figura 11). Desse modo, foi possível compreender que o projeto não usa nenhum tipo de PRNG para de fato gerar o código numérico, deixando a cargo do usuário fornecê-lo, o que permite brechas para que os códigos gerados não sejam seguros e, por muitas vezes, não sejam gerados de modo aleatório.

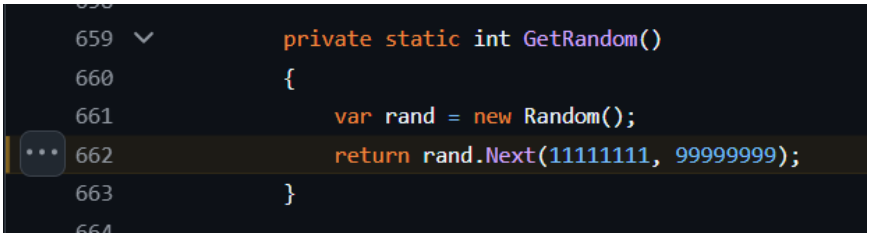
Figura 12 - Criação da função *ObterChave* do projeto *DFe.NET*

```
/// <summary>
/// Obtém a chave do documento fiscal
/// </summary>
/// <param name="ufEmitente">UF do emitente do DF-e</param>
/// <param name="dataEmissao">Data de emissão do DF-e</param>
/// <param name="cnpjEmitente">CNPJ do emitente do DF-e</param>
/// <param name="modelo">Modelo do DF-e</param>
/// <param name="serie">Série do DF-e</param>
/// <param name="numero">Número do DF-e</param>
/// <param name="tipoEmissao">Tipo de emissão do DF-e. Informar inteiro conforme consta no manual de orientação do contribuinte para o DF-e</param>
/// <param name="cNf">Código numérico que compõe a Chave de Acesso. Número gerado pelo emitente para cada DF-e</param>
/// <returns>Retorna um objeto <see cref="DadosChaveFiscal"/> com os dados da chave de acesso</returns>
1 reference
public static DadosChaveFiscal ObterChave(Estado ufEmitente, DateTimeOffset dataEmissao, string cnpjEmitente, ModeloDocumento modelo, int serie, long numero, int tipoEmissao, int cNf)
{
    var chave = new StringBuilder();

    if (cnpjEmitente.Length < 14)
    {
        cnpjEmitente = cnpjEmitente.PadLeft(14, '0');
    }

    chave.Append(((int)ufEmitente).ToString("D2"))
         .Append(dataEmissao.ToString("yyMM"))
         .Append(cnpjEmitente)
```

Fonte: Captura do projeto https://github.com/ZeusAutomacao/DFe.NET.git

No entanto, na pasta *CTe.AppTeste.NetCore,* criado pelo autor para demonstração de uso CTe, foi possível encontrar em *Program.cs* o código *GetRandom* (Figura 12) que gera um valor aleatório de 11111111 a 99999999. Apesar de ser apenas para demonstração do código e não de fato para a geração do código aleatório de NFe para o projeto, uma modificação possível seria considerar o intervalo de geração de valores de 00000000 a 99999999, abrangendo uma maior escolha de valores. No entanto, um problema possível que poderia surgir, seria o uso repetido desse método em um curto período de tempo, a semente gerada poderia ser a mesma, resultando em números repetidos. Uma forma de solucionar esse problema, seria declarar o objeto random como uma variável estática ou utilizar um gerador de números aleatórios criptograficamente seguro, como o uso da classe *RandomNumberGenerator*.

Figura 13 - Criação da função *GetRandom* do projeto *DFe.NET*



```
659     private static int GetRandom()
660     {
661         var rand = new Random();
662         return rand.Next(11111111, 99999999);
663     }
```

Fonte: Captura da classe Program.cs do projeto ZeusAutomacao/DFe.NET

Por fim, em relação ao projeto *sped-nfe* desenvolvido na linguagem PHP foi identificado o uso do método *Random* pertencente a classe *Keys* (Figura 13). Esse método gera um número de 8 dígitos aleatório para o código numérico utilizando o *mt_rand*(Figura 14) para gerar um número inteiro pseudo

aleatório entre 0 e 99.999.999. Caso o número gerado não possua os 8 dígitos, são acrescentados zeros à esquerda.

Em seguida, é feita uma validação que verifica se o valor está na lista de valores inválidos definidos para o código numérico aleatório e, caso o número do documento fiscal seja passado, ele assegura que o número gerado não seja igual a ele, atendendo à norma NT2019.001. Se o número gerado for inválido, o processo é repetido até que um valor válido seja gerado.

Figura 14 - Método *random* do projeto *sped-nfe*

```php
/**
 * Generate and return a 8 digits random number
 * for cNF tag
 * @param string|null $nnf
 * @return string
 */
public static function random($nnf = null)
{
    $loop = true;
    while ($loop) {
        $cnf = str_pad(strval(mt_rand(0, 99999999)), 8, '0', STR_PAD_LEFT);
        $loop = !self::cNFIsValid($cnf);
        if (!empty($nnf)) {
            if (intval($cnf) === intval($nnf)) {
                $loop = true;
            }
        }
    }
    return $cnf;
}
```

Fonte: Captura do projeto https://github.com/nfephp-org/sped-da.git

Como já exposto, o uso do *mt_rand* não é indicado por não ser criptograficamente seguro. Alternativas para melhorar a segurança seria o uso das funções *random_int* ou *random_bytes.* Apesar do uso de um método com baixa segurança, o projeto tenta cumprir com o recomendado pelas normas de segurança, utilizando um método de geração de números aleatórios para gerar o código numérico de 8 dígitos da NF-e e NFC-e.

8. Trabalhos Relacionados

Muitas das pesquisas relacionadas à nota fiscal eletrônica são das áreas de administração pública ou economia. Em relação à primeira, Machado e Bioni (2016) mapearam as políticas de proteção de dados pessoais nos programas de nota fiscal em 12 estados do Brasil, coletando as informações que as Secretarias Estaduais da Fazenda publicaram em seus portais. O objetivo era entender como os dados coletados pelo programa seriam usados, com quem poderiam ser compartilhados, quais medidas de segurança existem para proteger essas informações, por quanto tempo elas seriam guardadas e se há diretrizes sobre como descartá-las. Também se buscou saber como os cidadãos podem exercer controle sobre seus dados, especialmente o direito de deletá-los. Com relação ao Programa Nota Fiscal Paulista, a pesquisa revelou um cenário preocupante, com a falta de informações sobre as políticas de proteção de dados, mostrando que o foco está apenas nas vantagens do sistema, sem garantias de privacidade. Um questionário detalhado sobre esse programa revelou a ausência de políticas claras para a coleta e uso de dados, indicando que muitas informações pessoais dos cidadãos estão em uma "zona cinza", sem supervisão ou regulamentação.

Podemos encontrar, entretanto, alguns trabalhos que envolvem conceitos de tecnologia da informação. Como exemplo, Neto (2023) analisa o potencial da inteligência artificial (IA) no sistema público de escrituração digital (SPED) e sua aplicação na fiscalização tributária no Brasil. O objetivo principal foi discutir como a IA pode ser utilizada no contexto do SPED, enquanto os específicos abordaram mudanças na ação fiscal do governo, impactos do SPED na arrecadação e defesa do contribuinte, a introdução da IA no setor e os benefícios, desafios e questões éticas de sua implementação. A pesquisa foi embasada em uma revisão da literatura especializada.

Um trabalho mais técnico na área de computação foi apresentado por Araújo (2022), em que foi proposta uma API para comparação de preços

através de notas fiscais de produtos emitidas por estabelecimentos comerciais. A API foi projetada para melhorar a comparação de preços, funcionando através da leitura de códigos QR nos DANFEs. Para entender melhor as necessidades dos consumidores e quais soluções existem, foram aplicados questionários sobre hábitos de compra e realizada uma pesquisa de mercado. Entre as funcionalidades do aplicativo estão o cadastro das notas fiscais, a pesquisa e comparação de preços, sugestões de itens mais comprados por área, montagem de listas de compras e o histórico de preços.

9. Considerações Finais

O objetivo deste estudo foi comparar dados de 2017 e 2024 para examinar a segurança na geração da chave de acesso para Notas Fiscais de Consumidor Eletrônicas (NFC-e) em Campina Grande. (PB). O estudo encontrou uma modesta melhora na adoção de práticas seguras, com um aumento de 25% para 48% no uso de métodos adequados de geração do código numérico.

Apesar dos avanços, ainda há uma porcentagem significativa de empresas (52%) que utilizam práticas inseguras, tornando não apenas a empresa, mas também os seus consumidores vulneráveis de terem seus dados acessados indevidamente.. Como propostas para solucionar ou amenizar o problema apresentado neste trabalho, recomenda-se uma fiscalização mais rigorosa das SEFAZs em detectar chaves de acesso vulneráveis, a realização de campanhas de conscientização nas mídias e, por parte das empresas desenvolvedoras de software, a correção imediata dos sistemas de emissão que não atendem ao critério de aleatoriedade do campo da nota fiscal.

Devido ao tamanho reduzido da equipe do projeto e ao investimento significativo de tempo e recursos necessários para realizar visitas aos estabelecimentos, limitamos a amostra da pesquisa a aproximadamente 200 empresas. Procuramos diversificar os setores de atividade dos estabelecimentos para tentar garantir uma representação estatística adequada do varejo no município de Campina Grande-PB.

O serviço web desenvolvido e descrito na Seção 6, com objetivo de detectar vulnerabilidades nas chaves de acesso, pode ser usado para que as empresas tenham ciência se o software de emissão de NFC-e que utilizam segue ou não a norma técnica de segurança de criação do código da nota fiscal. Outra contribuição vital que trouxemos (Seção 7), foi a análise técnica de quatro projetos de código aberto disponíveis na plataforma GitHub, apontando algumas falhas empregadas no código e sugerindo formas mais apropriadas de utilizar geradores de número pseudo-aleatórios.

Importante ressaltar que mesmo após a correção do sistema, todo o histórico de notas fiscais já emitidas continuarão passíveis de serem acessadas indevidamente. Nesse ponto, uma sugestão é que as próprias SEFAZ desenvolvam algum mecanismo de proteção dos dados sensíveis presentes nas notas.

Com relação aos desdobramentos desta pesquisa, podemos vislumbrar os seguintes trabalhos futuros:

a) Expandir a análise para um número maior de estabelecimentos, com o intuito de alcançar uma visão mais ampla das práticas de emissão de NFC-e. Além disso, seria interessante repetir o estudo em outro município ou região do país, permitindo comparações regionais que possam mostrar diferenças significativas nas práticas e o uso das notas fiscais eletrônicas.
b) Analisar outros projetos no GitHub, a exemplo do PyNFe, desenvolvido em python, que possui um grande número de *forks* e *stars*;
c) Investigar e propor mecanismos de garantir um acesso seguro às notas fiscais vulneráveis que já foram emitidas e encontram-se nos bancos de dados das secretarias das fazendas.
d) Realizar uma pesquisa com as empresas de TI sobre o que as equipes de desenvolvimento conhecem sobre os problemas discutidos neste trabalho, contribuindo com o desenvolvimento profissional dessas equipes.

Por fim, acreditamos que este estudo fornece uma base sólida para futuras pesquisas e melhorias na segurança do acesso à NF-e e NFC-e, ajudando a criar um ambiente fiscal mais seguro e confiável.

Referências

AHN, L. V.; MAURER, B.; MCMILLEN, C.; ABRAHAM, D.; BLUM, M. recaptcha: human based character recognition via web security measures. *Science*, American Association for the Advancement of Science, v. 321, n. 5895, p. 1465–1468, 2008.

ARAÚJO, Adam Cordeiro. *Ludiiprice: API para comparação de preços através de notas fiscais de produtos emitidas por estabelecimentos comerciais.* 2022. Monografia (Graduação em Ciência da Computação) – Departamento de Computação, Centro de Ciências Exatas e Tecnologia, Universidade Federal de Sergipe, São Cristóvão, SE, 2022.

BHATTACHARJEE, Kamalika; DAS, Sukanta. A search for good pseudo-random number generators: survey and empirical studies. *Computer Science Review*, v. 45, p. 100471, 2022.

BONFIM, D. P. et al. Nota fiscal eletrônica: uma mudança de paradigma sob a perspectiva do fisco estadual. *ConTexto – Contabilidade em Texto*, v. 12, n. 21, p. 17–28, 2012.

BRASIL. Receita Federal. *Informações sobre Estatísticas.* 2024. Disponível em: https://www.nfe.fazenda.gov.br/portal/infoEstatisticas.aspx.

CAVALHEIRO, J. S. *Percepção de contadores sobre a implementação da Nota Fiscal de Consumidor Eletrônica em Santa Catarina e seus reflexos para escritórios de contabilidade.* 2022. 49 f. Monografia (Graduação) – Universidade Federal de Santa Catarina, Florianópolis, 2022.

COLOMBO, F. J.; NETO, B. B.; BARROS, L. d. J. R. de. Princípio de funcionamento da nota fiscal eletrônica. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, v. 8, n. 5, p. 2092–2102, 2022.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to algorithms.* [S.l.]: MIT Press, 2022.

CUNHA, J. P. P. da. *Nota Fiscal Eletrônica: arquitetura tecnológica e contábil.* 2022. 48 f. Monografia (Graduação) – Universidade Federal de Uberlândia, Uberlândia, 2022.

ENCAT. *Nota Técnica 2019.001: Criação e Atualização de Regras de Validação. Versão 1.60 – Dezembro de 2023.* [S.l.], 2023. Disponível em: https://www.nfe.fazenda.gov.br/portal/exibirArquivo.aspx?conteudo=pGfPgyEaNGg=.

ESTATÍSTICAS | NFC-e. 2024. Disponível em: http://nfce.encat.org/estatisticas/.

GONZAGA, L. M. *Percepções dos consumidores finais sobre dados informacionais com base na NFC-e (Nota Fiscal de Consumidor Eletrônica).* 25 f. Monografia (Graduação) – Universidade do Extremo Sul Catarinense, Criciúma, 2017.

GUERAR, M.; VERDERAME, L.; MIGLIARDI, M.; PALMIERI, F.; MERLO, A. Gotta captcha'em all: a survey of 20 years of the human-or-computer dilemma. *ACM Computing Surveys (CSUR)*, ACM, New York, NY, v. 54, n. 9, p. 1–33, 2021.

HOLOHAN, Naoise. **Random number generators and seeding for differential privacy.** arXiv preprint arXiv:2307.03543, 2023. IBM Research Europe - Ireland.

JOHNSTON, D. *Random number generators—principles and practices: a guide for engineers and programmers.* [S.l.]: Walter de Gruyter GmbH & Co KG, 2018.

KATZGRABER, Helmut G. **Random numbers in scientific computing: An introduction.** arXiv preprint arXiv:1005.4117, 2010. Lecture given at the International Summer School Modern Computational Science.

KNUTH, D. E. **The Art of Computer Programming**: Seminumerical Algorithms. 3. ed. Reading, MA: Addison-Wesley, 1997.

KUMAR, M.; JINDAL, M.; KUMAR, M. A systematic survey on captcha recognition: types, creation and breaking techniques. *Archives of Computational Methods in Engineering*, Springer, v. 29, n. 2, p. 1107–1136, 2022.

MACHADO, Jorge; BIONI, Bruno Ricardo. A proteção de dados pessoais nos programas de Nota Fiscal: um estudo de caso do "Nota Fiscal paulista". *Liinc em Revista*, Rio de Janeiro, v. 12, n. 2, p. 350-364, nov. 2016. Disponível em: https://revista.ibict.br/liinc/article/view/3734. Acesso em: 29 de jan. 2025.

NETO, Venicius Landulpho Magalhães. *O potencial do Sistema Público de Escrituração Digital com a aplicação da inteligência artificial na fiscalização tributária*. 2023. Dissertação (Mestrado Profissional em Direito) – Instituto Brasileiro de Ensino, Pesquisa e Desenvolvimento, Brasília, 2023. Disponível em: http://52.186.153.119/bitstream/123456789/5115/1/Disserta%C3%A7%C3%A3o_Venicius%20Landulpho%20Magalh%C3%A3es%20Neto_Mestrado%20Profissional%20em%20Direito.pdf. Acesso em: 29 de jan. 2025.

MANOHARAN, A. P.; INGRAMS, A.; KANG, D.; ZHAO, H. Globalization and worldwide best practices in e-government. *International Journal of Public Administration*, Taylor & Francis, v. 44, n. 6, p. 465–476, 2021.

MANUAL DE ORIENTAÇÃO AO CONTRIBUINTE - MOC - versão 7.0 - NF-e e NFC-e. [S.l.], 2024. Disponível em: https://www.nfe.fazenda.gov.br/portal/infoEstatisticas.aspx.

MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, ACM, New York, NY, USA, v. 8, n. 1, p. 3–30, 1998.

MATSUMOTO, Makoto; NISHIMURA, Takuji. **Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator.** ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1):3–30, 1998.**MAYER, Fernando.** Geração de números aleatórios uniformes, 2022. Disponível em: http://cursos.leg.ufpr.br/ce089/slides/02_RNG_uniforme.html#1. Universidade Federal do Paraná. Último acesso em: 13 nov. 2024.

MAYER, Fernando. **Geração de números aleatórios uniformes**, 2022. Disponível em: http://cursos.leg.ufpr.br/ce089/slides/02_RNG_uniforme.html#1. Universidade Federal do Paraná. Último acesso em: 13 nov. 2024.

MICROSOFT. **RandomNumberGenerator Class**, 2024. Disponível em: https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography.randomnumbergenerator?view=net-9.0 Acesso em: 16 dez. 2024.

MINAS GERAIS. SECRETARIA DA TRIBUTAÇÃO. *Boas práticas no desenvolvimento de emissor do NFC-e.* 2018. Disponível em: http://www.nfe.fazenda.gov.br/portal/listaConteudo.aspx?tipoConteudo=33ol5hhSYZk=. Acesso em: 13 jan. 2025.**MONTEIRO, Pedro Alexandre Gândara.** *Gerador de números aleatórios integrado em tecnologia CMOS.* 2022. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores).

OLIVEIRA, L. C.; TEODORO, P.; VASCONCELLOS, Y.; BONFIM, M. P. Nota fiscal eletrônica (NF-e): benefícios propagados pelos responsáveis pelo projeto versus benefícios percebidos por profissionais de contabilidade. *Revista de Administração, Contabilidade e Economia da Fundace*, v. 12, n. 2, 2021.

ORACLE. **Java Cryptography Architecture (JCA) Reference Guide**: Standard Names. Disponível em: https://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#SecureRandom. Acesso em: 05 dez. 2024.

PHP. **mt_rand - Manual.** Disponível em: https://www.php.net/manual/pt_BR/function.mt-rand.php. Acesso em: 16 dez. 2024.

PHP. **random_bytes - Manual.** Disponível em: https://www.php.net/manual/pt_BR/function.random-bytes.php. Acesso em: 16 dez. 2024.

PHP. **random_int - Manual.** Disponível em: https://www.php.net/manual/pt_BR/function.random-int.php. Acesso em: 16 dez. 2024.

PICCOLI, M. S. d. Q.; STECANELA, N. Popularização da ciência: uma revisão sistemática de literatura. *Educação e Pesquisa*, Faculdade de Educação, v. 49, 2023.

RAPÔSO, C. F. L.; LIMA, H. M. de; JUNIOR, W. F. de O.; SILVA, P. A. F.; BARROS, E. E. de S. LGPD-lei geral de proteção de dados pessoais em tecnologia da informação: revisão sistemática. *RACE-Revista de Administração do Cesmac*, v. 4, p. 58–67, 2019.

RECEITA FEDERAL. *Manual de Orientação ao Contribuinte: DANFE NFC-e e QR Code.* 2024. Disponível em: http://moc.sped.fazenda.pr.gov.br/DanfeQrCodeNFCe.html.

RECEITA FEDERAL. *Sobre a NF-e.* 2024. Disponível em: https://www.nfe.fazenda.gov.br/portal/sobreNFe.aspx

RIO GRANDE DO NORTE. SECRETARIA DA TRIBUTAÇÃO. *Nota Fiscal de Consumidor Eletrônica: um novo modelo de documento fiscal para o varejo.* Natal: Encat, 2016. 61 slides, color. Disponível em: http://www.set.rn.gov.br/contentProducao/aplicacao/set_v2/nfce/gerados/obrigatoriedades.asp. Acesso em: 13 jan. 2025.

RODRIGUES, Luan Thomaz P. et al. A nota fiscal do consumidor eletrônica: uma revolução no varejo brasileiro. *Revista de Trabalhos Acadêmicos Universo São Gonçalo*, v. 1, n. 2, p. 56-71, 2016.

SEFAZ-PB. Receita Estadual antecipa obrigatoriedade de emissão da Nota Fiscal de Consumidor Eletrônica para varejistas. 2017. Disponível em: https://www.sefaz.pb.gov.br/9-posts/1918-receita-estadual-antecipa-obrigatoriedade-de-emissao-da-nota-fiscal-de-consumidor-eletronica-para-varejistas.

SIVAKORN, S.; POLAKIS, J.; KEROMYTIS, A. D. I'm not a human: Breaking the google recaptcha. *Black Hat*, v. 14, p. 1–12, 2016.

SOUZA, N. P. de; FARIA, L.; PINTO, P. D. Sistemas de dígitos verificadores ótimos baseados em aritmética modular. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 5, n. 1, 2017.

System.Random Class, 2024. Disponível em: https://learn.microsoft.com/pt-br/dotnet/fundamentals/runtime-libraries/system-random. Acesso em: 13 dez. 2024.

TANG, M.; GAO, H.; ZHANG, Y.; LIU, Y.; ZHANG, P.; WANG, P. Research on deep learning techniques in breaking text-based captchas and designing image-based captcha. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 13, n. 10, p. 2522–2537, 2018.

TECH, N. Obrigação de uso do CPF na nota: uma análise comparativa entre os estados brasileiros. 2023. Disponível em: https://ndd.tech/blog/compliance-fiscal/saiba-em-quais-estados-e-obrigatorio-informar-o-cpf-na-nota-fiscal-e-como-essa-medida-podecombater-a-son.

TSAKALAKI, E. Ensure your random number generator is properly seeded. 2022. Disponível em: https://wiki.sei.cmu.edu/confluence/display/cplusplus/MSC51-CPP.+Ensure+your+random+number+generator+is+properly+seeded.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Campina Grande - Código INEP: 25137409
R. Tranqüilino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Restrito

Versão Final do TCC

Assunto:	Versão Final do TCC
Assinado por:	Jonathan Saturnino
Tipo do Documento:	Requerimento
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Direito Autoral (Art. 24, III, da Lei no 9.610/1998)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Jonathan Saturnino Pontes de Souto**, **ALUNO (201921250037) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 18/03/2025 16:04:28.

Este documento foi armazenado no SUAP em 18/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1423951
Código de Autenticação: 96aad1799f