TÍTULO DO ARTIGO

TITLE OF THE ARTICLE

Fonte Times New Roman, tamanho 14, negrito, centralizado, todas as letras maiúsculas e espaçamento simples entre linhas.

Erick Galvão da Silva, Faculdade de Tecnologia de Americana,

[erick.silva82@fatec.sp.gov.br](mailto:erick.silva82@fatec.sp.gov.br)

Stheffani Gonçalves Rocha Emboava, Faculdade de Tecnologia de Americana,

[sthefani.emboava@fatec.sp.gov.br](mailto:sthefani.emboava@fatec.sp.gov.br)

Leonardo Rodrigues Ribeiro, Faculdade de Tecnologia de Americana,

[leonardo.ribeiro16@fatec.sp.gov.br](mailto:leonardo.ribeiro16@fatec.sp.gov.br)

**Resumo**

**Palavras-chave**: De três a cinco, separadas uma da outra por vírgula, tamanho da fonte 11.

***Abstract***

***Keywords****: Three to five, separated from each other by a comma, font size 11.*

# **Introdução**

A computação quântica é um campo revolucionário que promete transformar radicalmente a forma como processamos e armazenamos informações. (Preskill, 2018). Essa tecnologia tem o potencial de resolver problemas complexos de maneira mais eficiente do que os computadores clássicos convencionais, e promete ser a resolução dos problemas enfrentados pela tecnologia atual, sanando problemas como o processamento de dados, de maneira eficiente e otimizada, além de ser o ponto chave para o desenvolvimento de outras tecnologias como a Inteligência Artificial, IoT, Machine Learning, Cloud Computing e Banco de Dados.

Com o avanço da computação quântica, a segurança inevitavelmente é afetada. Assim como Mosca e Stebila (2018) previam, "Os algoritmos criptográficos convencionais, amplamente utilizados hoje em dia, são vulneráveis a ataques por meio de computadores quânticos", uma vez que os algoritmos criptográficos são baseados na fatoração de grandes números primos, o que era considerado um problema complexo para a computação clássica e para áreas da matemática e que poderia levar centenas de anos para ser solucionado, com a computação quântica, dependendo da complexidade dos números, a resolução da fatoração pode levar segundos, dias ou semanas, um tempo muito menor comparado a computação clássica, ameaçando a segurança antes estabelecida.

O presente artigo apresenta pesquisas técnicas e aplicações práticas, com o objetivo de identificar desafios que poderão ser enfrentados na iminente migração da computação clássica para a quântica. Com o uso do ambiente quântico desenvolvido pela Microsoft e da linguagem quântica Q#, juntamente com a linguagem clássica multiparadigma Python, demonstram-se os efeitos da computação quântica sobre a clássica, destacando seu impacto na segurança proporcionada pela criptografia em um banco de dados relacional, e contribui-se na busca de plataformas e linguagens que possibilitem uma migração segura e de baixa complexidade.

Para isso, a pesquisa contou com um código onde foram implementadas bibliotecas e métodos de criptografia utillizados em larga escala, e depois foi realizada uma análise comparativa dos resultados de decifragem dos dados por meio de processos quânticos e clássicos.

# **Referencial Teórico**

**2.1 Computação Quântica**

A Computação Quântica é uma tecnologia emergente da Ciência da Computação, que utiliza de conceitos e fundamentos da Física, Matemática e Mecânica Quântica, sendo esses o tripé da Computação Quântica, e se destaca por sua capacidade de processamento de dados exponencialmente maior quando comparado à computação clássica. Enquanto os computadores clássicos operam com bits clássicos, de 0 e 1, os bits quânticos, ou qubits, operam com a superposição de estados, onde o bit pode estar em zero, um, ou simultaneamente entre os dois estados. Os qubits também podem estar entrelaçados, e através disso, e possível realizar as correlações quânticas, que não são possíveis em sistemas clássicos. (LACAVA & MIANO, 2018).

Existem diversas vantagens potenciais no uso da Computação Quântica em relação à Computação Clássica, como a velocidade e eficiência na resolução de determinados problemas, por exemplo a fatoração de números grandes, em que na computação clássica esse resultado pode levar anos a ser obtidos (e por conta disso se tornou a base da criptografia clássica), já com a criptografia quântica, essa fatoração leva segundos, minutos ou dias, dependendo do tamanho do número e sua complexidade. Outra vantagem, seria a simulação de sistemas quânticos complexos e a otimização de problemas em grande escala, como o processamento de dados através do uso de tecnologias como algoritmos de Machine Learning para o processamento de uma massa de dados extensa, que demanda de velocidade e grandes recursos. (MIANO, 2020).

**2.1.1 Algoritmo de Shor**

O algoritmo de Shor é um algoritmo quântico desenvolvido por Peter Shor em 1994. É conhecido por sua capacidade de fatorar números inteiros grandes de forma significativamente mais rápida do que qualquer algoritmo clássico conhecido.

A fatoração de números inteiros grandes em seus fatores primos é um problema desafiador para a computação clássica, especialmente quando os números envolvidos têm centenas ou milhares de dígitos. A segurança de muitos sistemas criptográficos modernos, como o RSA, depende da dificuldade de fatoração.

O algoritmo de Shor utiliza princípios da computação quântica, aproveitando a capacidade dos qubits de existirem em estados de superposição e entrelaçamento. Em um computador quântico com capacidade suficiente, o algoritmo de Shor pode ser usado para fatorar um número inteiro N em seus fatores primos em tempo polinomial, o que representa um avanço significativo em relação aos métodos clássicos exponenciais. É um algoritmo quântico revolucionário que pode fatorar números inteiros grandes de forma eficiente. Sua descoberta levantou questões sobre a segurança dos sistemas criptográficos atuais e impulsionou a pesquisa em criptografia pós-quântica.

Segundo MIANO (2020), o objetivo do algoritmo é achar o período de uma função, e na sequência, encontrar os fatores do valor solicitado.

O algoritmo de Shor tem uma aplicação fundamentalmente importante na área de criptografia, especificamente na quebra de criptografia baseada no problema de fatoração de números inteiros grandes. O RSA (Rivest-Shamir-Adleman), um dos algoritmos de criptografia mais utilizados, é vulnerável à fatoração de números primos grandes, e o algoritmo de Shor pode ser utilizado para realizar essa fatoração de forma muito mais eficiente em um computador quântico.

A implicação direta disso é que, se um computador quântico com poder de processamento adequado for construído, ele poderá fatorar rapidamente os números primos envolvidos no RSA, comprometendo a segurança de comunicações e transações protegidas por esse algoritmo.

**2.2 Plataformas e Simuladores Quânticos**

As plataformas e simuladores quânticos são ambientes que contém ferramentas desenvolvidas para simular e estudar sistemas quânticos, através das máquinas clássicas. Elas fornecem a possibilidade de investigar o comportamento de sistemas quânticos, realizar experimentos e testes de algoritmos quânticos virtualmente, oferecendo a possibilidade de trabalhar com tecnologias híbridas, o que auxilia na realização dos estudos e experimentos para desenvolvimento deste trabalho, pois podemos, através delas, simular ambientes quânticos através das máquinas atuais.

**2.2.1 Microsoft Azure Quantum**

O Microsoft Azure Quantum foi a plataforma de simulação escolhida para o desenvolvimento do projeto. Devido à maturidade da Microsoft em serviços baseados em nuvem, seu simulador quântico mostrou-se muito eficiente para o desenvolvimento do projeto. Dentro da plataforma contamos com um conjunto de ferramentas de desenvolvimento QDK (Microsoft Quantum Development Kit) e a linguagem de programação Q# (Q Sharp), nativos da plataforma, oferecendo suporte à simulação, programação e integração com hardware quântico.

**2.3 Linguagens de Programação**

Considerando os objetivos do trabalho e a área de conhecimento enfatizada nesse artigo (Tecnologia da Informação), mostrou-se imprescindível a construção de um programa, demonstrando na prática os resultados almejados.

**2.3.1 Python**

Python é uma linguagem de programação de alto nível, interpretada e orientada a objetos, conhecida por sua sintaxe clara e legível. Sua semântica dinâmica permite que os desenvolvedores escrevam código de maneira rápida e eficiente, sem a necessidade de declarações explícitas de tipos de variáveis. A digitação dinâmica do Python facilita a prototipagem e o desenvolvimento ágil de aplicativos, tornando-o uma escolha popular para uma ampla gama de projetos.

Uma das características distintivas do Python é a presença de estruturas de dados incorporadas, como listas, dicionários e conjuntos, que facilitam a manipulação e gerenciamento de informações complexas. Além disso, a ligação dinâmica do Python permite que as chamadas de função e os métodos sejam resolvidos em tempo de execução, proporcionando flexibilidade aos desenvolvedores.

A popularidade do Python também se deve à sua vasta coleção de bibliotecas e frameworks, que abrangem desde desenvolvimento web até aprendizado de máquina e processamento de dados. Essas bibliotecas fornecem funcionalidades adicionais e permitem aos desenvolvedores aproveitar soluções pré-existentes, acelerando o tempo de desenvolvimento e aumentando a eficiência. Com sua comunidade ativa e dedicada, o Python continua a evoluir e se adaptar às demandas da indústria de software, consolidando sua posição como uma linguagem poderosa e versátil.

**2.3.2 Q# (Q Sharp)**

Q# é uma linguagem de programação específica para domínio usada para expressar algoritmos quânticos. Foi lançada ao público pela Microsoft como parte do Kit de Desenvolvimento Quântico (QDK). Q# é uma linguagem de alto nível, independente de hardware, que permite focar no nível de algoritmo e aplicação para criar programas quânticos. Q# também introduz novas estruturas e operações específicas para quântica, como repetir até sucesso e estimativa de fase adaptativa, que permitem a integração de computações quânticas e clássicas. (MICROSOFT, 2023e)

**2.4 Criptografia**

A criptografia é uma ciência que se dedica a proteger a comunicação e os dados em face de adversários. Com base em algoritmos matemáticos, ela possui diversas aplicações práticas, desde transações comerciais online até segurança militar. Algoritmos clássicos, como DES, AES, RSA e ECC, são amplamente utilizados para garantir a confidencialidade e a integridade das informações. A criptografia desperta grande interesse entre pesquisadores, agências governamentais, defensores dos direitos civis e usuários comuns que valorizam sua privacidade e buscam a proteção de seus dados.

A criptografia moderna é um campo em constante evolução, adaptando-se aos avanços tecnológicos e às novas ameaças digitais. Seu papel é crucial na proteção de informações sensíveis e na manutenção da segurança dos sistemas digitais. Com a crescente quantidade de dados pessoais circulando pela internet, a necessidade de técnicas criptográficas eficazes e confiáveis é cada vez mais evidente. A pesquisa e o desenvolvimento contínuos nesse campo são essenciais para enfrentar os desafios emergentes e garantir a privacidade e a confiança nas comunicações e transações online.

Desempenha um papel fundamental na proteção da privacidade e segurança das informações em um mundo digital. É uma disciplina em constante avanço, impulsionada pelas demandas da sociedade moderna e pela necessidade de proteger dados sensíveis de adversários mal-intencionados. Através de algoritmos criptográficos robustos, a criptografia oferece meios eficazes de proteger a confidencialidade, a integridade e a autenticidade dos dados. A compreensão e o avanço contínuo nesse campo são essenciais para enfrentar os desafios emergentes e garantir a segurança dos sistemas e das comunicações digitais.

## **2.4.1 Tipos de criptografia**

A criptografia pode ser separada em dois tipos, simétrica e assimétrica. Na criptografia simétrica, a mesma chave é usada tanto para criptografar quanto para descriptografar os dados. A chave é compartilhada entre as partes autorizadas de antemão. A criptografia simétrica é rápida e eficiente, adequada para grandes volumes de dados. Exemplos de algoritmos simétricos são o AES (Advanced Encryption Standard) e o DES (Data Encryption Standard). Na criptografia assimétrica ou criptografia de chave pública, são usadas duas chaves diferentes: uma chave pública e uma chave privada. A chave pública é usada para criptografar os dados, enquanto a chave privada correspondente é usada para descriptografá-los. As chaves são geradas em pares e são matematicamente relacionadas. A criptografia assimétrica é mais segura em termos de compartilhamento de chaves, mas é computacionalmente mais intensiva do que a criptografia simétrica. O RSA e o ECC (Elliptic Curve Cryptography) são exemplos de algoritmos assimétricos amplamente utilizados.

## **2.4.2 Blowfish - criptografia simétrica**

Criado por Bruce Schneier em 1993, o Blowfish é um algoritmo de criptografia de bloco simétrico que tem sido amplamente utilizado em várias aplicações e sistemas. Uma das principais características do Blowfish é a sua operação em blocos fixos de dados de 64 bits, utilizando uma chave de criptografia variável de 32 a 448 bits. Essa flexibilidade no tamanho da chave permite adaptar a segurança do algoritmo de acordo com as necessidades específicas. Além disso, o Blowfish utiliza uma estrutura de rondas de transformações matemáticas para processar os dados de entrada. Ele é conhecido por seu desempenho rápido e eficiente, sendo otimizado para diferentes plataformas, incluindo implementações em hardware dedicado. Sua velocidade é um fator importante em muitos sistemas que exigem criptografia eficiente em tempo real.

O Blowfish é uma opção viável para aplicações que exigem uma camada adicional de segurança, como a proteção de dados em trânsito ou a geração de assinaturas digitais. No entanto, antes de implementar o algoritmo, é importante considerar cuidadosamente o contexto de uso, as necessidades de segurança e as melhores práticas atuais em relação à criptografia.

## **2.4.3 RSA - Criptografia Assimétrica**

## A criptografia RSA, como citado anteriormente, é um modelo baseado em chaves públicas que, conforme observação de A. Budiman, é composto por três principais estágios: geração de chave, criptografia e descriptografia. Esse tipo de criptografia é baseado na fatoração de números primos extensos, e desde sua criação na década de 70, por parte dos estudiosos Ron Rivest, Adi Shamir e Leonard Adleman, mostrou-se um método de segurança eficiente.

## O que garante a segurança deste método criptográfico é o fato de que os computadores com os quais estamos acostumados, não possuem capacidade de processamento suficiente para quebrar o algoritmo RSA, de modo que levariam centenas de anos para decifrar uma mensagem. Contudo, com o surgimento da computação quântica, algoritmo passou a estar sob grande ameaça, pois a criptografia antes protegida por centenas de anos de processamento em um computador clássico, com um sistema quântico pode ser quebrada em segundos.

**2.5 Banco de Dados**

Um banco de dados é um sistema organizado e estruturado para armazenar, gerenciar e recuperar informações. Ele é projetado para permitir o armazenamento eficiente, a recuperação rápida e a manipulação segura de dados. Um banco de dados é usado para coletar, armazenar, gerenciar e organizar grandes volumes de informações, de modo que os usuários possam acessar, modificar e analisar esses dados conforme necessário.

Um banco de dados é composto por uma coleção de tabelas ou entidades, que contêm registros individuais com informações específicas. Cada tabela é formada por colunas, que representam os diferentes tipos de dados que podem ser armazenados, e linhas, que contêm os registros individuais com os dados propriamente ditos.

**2.5.1 Banco de Dados Relacional**

## Dentre os tipos de banco de dados, o modelo relacional é amplamente adotado e considerado o mais comum. Esse modelo organizacional baseia-se na estrutura de tabelas compostas por linhas e colunas, como ilustrado de forma básica na figura 1. Cada tabela representa uma entidade do mundo real, como clientes, produtos ou pedidos, e as relações entre essas entidades são estabelecidas por meio de chaves primárias e estrangeiras.

## No modelo relacional, as colunas de uma tabela representam os atributos ou características das entidades, como nome, idade, preço ou quantidade. Por sua vez, as linhas contêm os registros individuais de dados, representando cada instância específica das entidades. Essa abordagem tabular fornece uma representação estruturada e organizada dos dados, facilitando a manipulação e consulta das informações armazenadas.

## Uma das principais vantagens do modelo relacional é sua flexibilidade na organização e recuperação de dados. Por meio da linguagem SQL (Structured Query Language), é possível executar consultas complexas que envolvem junções entre várias tabelas, filtragem de dados com base em condições específicas e ordenação dos resultados. Essa capacidade de consulta poderosa permite que os usuários obtenham insights valiosos dos dados, realizem análises avançadas e tomem decisões informadas com base nas informações armazenadas no banco de dados relacional.

## Figura 1-Banco de Dados Relacional

## 

## Fonte: elaborado pelos autores

**2.5.2 SQL (Structured Query Language)**

SQL (Structured Query Language) é uma linguagem de programação usada para gerenciar e manipular principalmente bancos de dados relacionais. Apesar de ser conhecida como linguagem de programação, o SQL é projetado especificamente para consultas e manipulações de dados em bancos de dados, ela permite a criação de consultas para recuperar, inserir, atualizar e excluir dados, e a definição de esquemas, índices e restrições de integridade. (RAMAKRISHNAN; GEHRKE, 2020).

# **Metodologia**

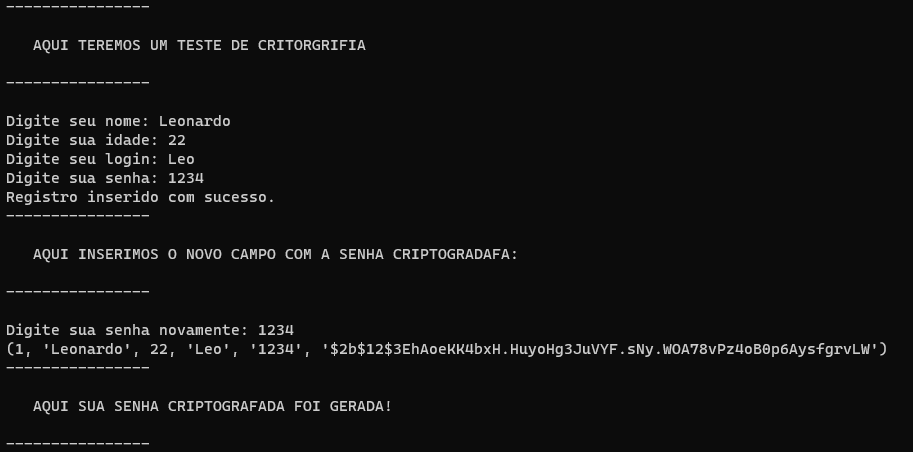
Este trabalho descreve uma pesquisa teórica de caráter experimental na área da computação, cujo intuito é demonstrar a vantagem de soluções quânticas sobre códigos clássicos, aproximando esta tecnologia das soluções computacionais vigentes. Assim sendo, foi desenvolvido um código que demonstrasse de maneira comparativa a superioridade da computação quântica, explorando as vulnerabilidades resultante do aumento da capacidade de processamento de um computador, em especial as que se referem à segurança de um banco de dados.

**3.1 Coleta de Dados**

Para obter os dados necessários para a o resultado final apresentado, foi realizada uma coleta abrangente e sistemática. Inicialmente, a coleta começou com a adição manual das informações relevantes a um banco de dados. Esse processo envolveu a inserção cuidadosa dos dados por meio de um sistema de entrada manual, garantindo a precisão e integridade das informações.

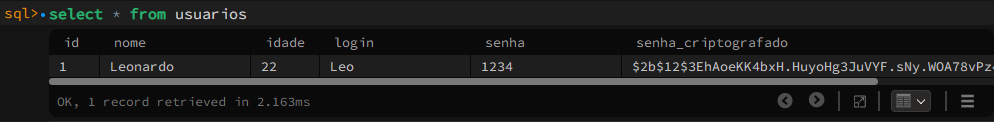
À medida que o projeto avançava, foi implementado um formulário na Web para facilitar e agilizar ainda mais o processo de coleta de dados. Esse formulário, acessível aos usuários por meio de um navegador, permitia a inserção direta das informações no banco de dados. Com essa abordagem, os participantes puderam fornecer suas informações de maneira conveniente e segura, enquanto as informações eram armazenadas no mesmo banco de dados central.

## Figura 1-Simulador Web que recolhe os dados do cliente



## Fonte: elaborado pelos autores

## Figura 2-Dados armazenados no banco de dados central



## Fonte: elaborado pelos autores

Essa combinação de métodos de coleta de dados garantiu a abrangência e diversidade necessárias para o resultado final. A adição manual inicial permitiu uma entrada precisa e detalhada dos dados, enquanto o formulário na Web trouxe uma abordagem mais escalável e eficiente para coletar informações de um número maior de participantes. Juntos, esses métodos asseguraram uma base de dados robusta e confiável para análise e conclusões relevantes.

**3.2 Código**

O código desenvolvido para este projeto apresenta uma abordagem híbrida, combinando duas linguagens distintas: uma quântica, o Q#, e outra clássica, o Python. Essa combinação permite aproveitar as vantagens da computação quântica para as operações matemáticas envolvidas na criptografia, enquanto a manipulação dos dados em suas demais particularidades é realizada por meio do Python.

No contexto da criptografia, foram empregados dois algoritmos amplamente utilizados: Blowfish e RSA. Esses algoritmos foram implementados em Q# para realizar a criptografia dos dados de forma segura. No entanto, para testar a segurança do sistema, foram empregados dois métodos diferentes de quebra de criptografia. Primeiramente, um algoritmo clássico de fatoração, programado em Python, foi utilizado para tentar quebrar a segurança dos dados. Em seguida, o poderoso algoritmo quântico de Shor, implementado em Q#, foi empregado para realizar o mesmo processo.

Para uma análise completa e comparativa do desempenho entre a computação quântica e a computação clássica, o foco foi direcionado especificamente para o processo de descriptografia dos dados criptografados. Ao analisar os resultados obtidos, foi possível avaliar as vantagens e limitações de cada abordagem, bem como compreender o potencial disruptivo da computação quântica no campo da criptografia.

Essa integração entre as linguagens Q# e Python, aliada ao uso dos algoritmos de criptografia e dos métodos de quebra empregados, proporcionou uma análise abrangente e aprofundada do desempenho e da segurança da computação quântica em relação à computação clássica no contexto da criptografia de dados. Essa pesquisa contribuiu para ampliar o conhecimento sobre as capacidades e desafios da computação quântica aplicada à segurança da informação.

**3.3 Simulador Quântico**

O Microsoft Azure Quantum foi o ambiente escolhido para o processamento quântico, utilizando os créditos fornecidos pela Microsoft para credenciais acadêmicas, a plataforma permitiu a exploração de todo o potencial da computação quântica, indo além das capacidades dos computadores clássicos. Com acesso à infraestrutura quântica do Azure Quantum, foram implementados algoritmos, executado simulações e até mesmo realizado experimentos em hardware quântico real.

# **Resultados e Discussões**

# Os resultados e discussões giram em torno da análise comparativa proporcionada pelo código desenvolvido para este artigo. Serão apresentadas a seguir conclusões e discussões teóricas sobre o estudo desenvolvido, e os principais resultados obtidos nesta análise.

# **4.1 Bibliotecas e métodos de criptografia**

# Por meio de uma minuciosa pesquisa, foram realizadas análises aprofundadas dos algoritmos de criptografia Blowfish e RSA. O Blowfish é uma cifra simétrica que divide as informações em blocos de 64 bits e criptografa cada um deles individualmente. Por sua vez, o RSA opera por meio da geração de um par de chaves distinto, composto por uma chave pública e outra privada, permitindo a cifragem das mensagens com a chave pública e sua decifragem exclusivamente com a chave privada. Nesse contexto, foram exploradas minuciosamente as vantagens e desafios inerentes a cada um desses algoritmos, visando compreender em profundidade suas aplicações e mecanismos de segurança.

# O Blowfish é mais rápido e eficiente, pois usa menos recursos computacionais. Em contrapartida, o RSA é mais seguro e flexível, pois permite o uso de assinaturas digitais e a troca de chaves sem a necessidade de um canal seguro. O RSA foi o algoritmo priorizado pelo fator de segurança, que se mostrou, portanto, mais inerente ao projeto desenvolvido, e ideal para a proteção dos dados de um banco de dados relacional.

# **4.2 Linguagens de Programação**

# Foi dedicada atenção especial às linguagens de programação empregadas nesse contexto, abrangendo tanto a linguagem clássica, como o Python p a linguagem quântica Q#, e identificou-se a oportunidade de conceber o algoritmo de Shor. O algoritmo foi desenvolvido com o fim último de utilizar a eficiência quântica para a decifragem da criptografia. Conforme o item 2.2.1 deste mesmo artigo, o algoritmo de Shor pode ser usado para fatorar um número inteiro N em seus fatores primos em tempo polinomial. Essa análise comparativa permitiu avaliar as diferenças de tempo e eficiência entre essas duas abordagens.

# Era essencial para o desenvolvimento do código que ele possuísse uma linguagem de programação quântica e outra clássica, e isso era claro desde o início de sua programação. Python mostrou-se uma linguagem muito versátil, de modo que a interação das diferentes partes do código. Uma das vantagens de usar o Q# para programar algoritmos quânticos é que ele pode se comunicar facilmente com outras linguagens populares, como o Python. O Python é uma linguagem de alto nível, amplamente usada para ciência de dados, aprendizado de máquina e desenvolvimento web. O Q# é uma linguagem específica de domínio, projetada para expressar operações quânticas de forma clara e concisa. Para integrar essas duas linguagens. Esse é um diferencial que facilita a criação de aplicações híbridas, que combinam o poder do processamento quântico com a flexibilidade do Python.

# **Considerações Finais**

# Tendo em vista a pesquisa apresentada ao longo deste artigo, conclui-se que a computação quântica apresenta potencial superior em termos de eficiência e processamento em comparação à computação clássica, o que pôde ser apresentado de maneira precisa por emio da utilização do algoritmo de Shor para a decifragem de dados criptografados por um algoritmo clássico.

# Desse modo, o Algoritmo mostra-se promissor não somente para um ataque, mas também para a mitigação de incidentes de segurança, pois pode ser utilizado para desbloquear arquivos criptografados em ataques de ransomware, por exemplo. Nesse sentido, o fato esperado de que esse avanço tecnológico favorece aqueles que agem dentro e fora dos limites éticos, pode ser considerado verídico.

# Contudo, é importante ressaltar que a superioridade da computação quântica em termos de eficiência e processamento se aplica especificamente a determinadas tarefas, como a descriptografia de dados. Para outras aplicações e desafios computacionais, a computação clássica ainda possui sua relevância e utilidade. Portanto, é fundamental considerar cuidadosamente o contexto e a natureza dos problemas a serem abordados ao avaliar a eficácia das abordagens computacionais.

# Em suma, com base nas análises realizadas, constatou-se que a computação quântica apresenta uma vantagem considerável em relação à eficiência e ao processamento de dados, especialmente na descriptografia de informações. Essa descoberta ressalta a importância contínua da pesquisa e do desenvolvimento nesse campo promissor, com o intuito de explorar todo o potencial da computação quântica para aplicações futuras.

# **Referências**

(ajustar)

ACEVEDO, C.; NOHARA, J. Metodologia no curso de administração: guia completo de conteúdo e forma. São Paulo: Atlas, 2004. Citado na página 2.

ANDERSON, C.; LONGA, A. C. do mercado de massa para o mercado de nicho. Rio De Janeiro, RJ: Elsevier, 2006. Citado na página 6.

BOULIC, R.; RENAULT, O. 3d hierarchies for animation. In: MAGNENAT-THALMANN, N.; THALMANN, D. (Ed.). New Trends in Animation and Visualization. [S.l.]: John Wiley & Sons ltd., 1991. Citado na página 3.

COHEN, S. Folk devils and moral panics: the creation of the mods. Criminology: A Reader, SAGE, p. 130, 2002. Citado na página 6.

KNUTH, D. E. The TEX Book. 15th. ed. [S.l.]: Addison-Wesley, 1984. Citado na página 3.

MIANO, Mariana V. Aplicação de protocolos quânticos e algoritmo de Shor para a segurança da informação. 2020. Revista Tecnológica da Fatec Americana, vol. 8 n. 01, 2020. Disponível em: <https://fatec.edu.br/revista/index.php/RTecFatecAM/article/view/233>. Acesso em 22 fev 2023.

MICROSOFT. O que é computação em nuvem. MICROSOFT, 2023a. Disponível em: https://learn.microsoft.com/pt-br/training/modules/describe-cloud-compute/3-what-cloud-compute. Acesso em 19 abr 2023.

MICROSOFT. Descrever a computação em nuvem. MICROSOFT, 2023b. Disponível em: https://learn.microsoft.com/pt-br/training/modules/describe-cloud-compute/5-define-cloud-models. Acesso em 19 abr 2023.

MICROSOFT. O que é o Azure. MICROSOFT, 2023c. Disponível em: https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-azure/. Acesso 19 abr 2023.

MICROSOFT. Serviço de nuvem do Azure Quantum. MICROSOFT, 2023d. Disponível em: https://azure.microsoft.com/pt-br/products/quantum/. Acesso em 20 abr 2023.

MICROSOFT. What are Q# and the Quantum Development Kit? MICROSOFT, 2023e. Disponível em: https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk. Acesso em 1 mai 2023.

MILANI, Andre. MySQL: guia do programador. 1ª edição. São Paulo: Novatec Editora Ltda, dezembro de 2006.

MySQL Documentation. Disponível em: https://dev.mysql.com/doc/. Acesso em 21 abr 2023.

NIELSEN, Michael A.; CHUANG, Isaac L. Quantum Computation and Quantum Information. 10th Edition. Editora: Cambridge University Press, 2010. p. 1–161.

PACIEVITCH, Yuri. MySQL InfoEscola. Disponível em: https://www.infoescola.com/informatica/mysql/. Acesso em 24 abr 2023.

PRESKILL, J. (2018). Quantum Computing in the NISQ era and beyond. Quantum, 2, 79.

RAMAKRISHNAN, R.; GEHRKE, J. (2020). Database Management Systems. McGraw-Hill Education.

SMITH, A.; JONES, B. On the complexity of computing. In: SMITH-JONES, A. B. (Ed.). Advances in Computer Science. [S.l.]: Publishing Press, 1999. p. 555–566. Citado na página 3.

SOARES, M. C. d. P. Althusser, poulantzas, buci-glucksmann: desenvolvimentos ulteriores do conceito gramsciano de estado integral. Crítica Marxista, n. 29, p. 97–121, 2009.