

CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA

CURSO SUPERIOR DE TECNOLOGIA EM SEGURANÇA DA INFORMAÇÃO

ERICK GALVÃO DA SILVA

LEONARDO RODRIGUES RIBEIRO

STHEFFANI GONÇALVES ROCHA EMBOAVA

ANÁLISE DE PLATAFORMAS E LINGUAGENS COMPUTACIONAIS PARA A MIGRAÇÃO DA TECNOLOGIA CLÁSSICA PARA A TECNOLOGIA QUÂNTICA

AMERICANA – SP

2023



CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA

CURSO SUPERIOR DE TECNOLOGIA EM SEGURANÇA DA INFORMAÇÃO

ERICK GALVÃO DA SILVA

LEONARDO RODRIGUES RIBEIRO

STHEFFANI GONÇALVES ROCHA EMBOAVA

Iniciação científica, sob a orientação da Prof. Dra. Mariana Godoy Vazquez

Área de concentração: Ciência da computação

AMERICANA – SP

2023

**RESUMO**

A computação clássica, tecnologia utilizada nos dias de hoje, vem enfrentando grandes desafios, os quais não são fisicamente possíveis de serem solucionados devido ao nível altíssimo de recursos necessários para solucionar tais requerimentos. Tarefas como o processamento de dados, por exemplo, têm se tornado cada vez mais complexa devido o aumento exponencial da quantidade de dados, graças ao fato de que todo aparelho eletrônico da atualidade necessita de se conectar à internet, receber um IP e gerar dados através dos protocolos de comunicação de rede, desde relógios e smartphones, até geladeiras, televisões e dispositivos eletrodomésticos. Tal fator teve suas consequências, de modo que o processamento dos dados e outras tarefas computacionais estão perdendo sua eficiência, levando cada vez mais tempo para encontrar uma solução, o que nos leva a entender que a computação clássica está chegando em seu limite. Mas afinal, é questionável como podemos continuar progredindo, considerando os recentes avanços e a exigência de processamento cada vez mais complexa que aplicações de Inteligência Artificial, IoT, Machine Learning, Cloud Computing e Banco de Dados.

A Computação Quântica promete a resolução dos problemas mencionados, e possibilita a evolução e desenvolvimento tecnológico em áreas como meio ambiente, energia, saúde e agricultura, tendo grande impacto não somente no âmbito tecnológico, mas também no social. O objetivo é que os computadores quânticos sejam capazes de processar informações computacionais complexas e impossíveis para computação clássica, não apenas uma maneira mais rápida de fazer o que já é feito através da computação atual, mas é uma abordagem diferente que promete resolver os problemas que a computação clássica nunca pôde resolver efetivamente.

Com isso, a atual pesquisa, visa o estudo de linguagens computacionais, plataformas e bibliotecas que possibilitam o desenvolvimento de tarefas e algoritmos clássicos para serem executado em ambiente quântico virtual, e posteriormente, transcrever os algoritmos para suas versões quânticas, e sua aplicação na criptografia de um banco de dados. Sendo assim, para o desenvolvimento da pesquisa, foram realizadas revisões bibliográficas, à respeito dos conceitos de Computação Quântica, Computação em Nuvem, Banco de Dados, Linguagens e Plataformas que permitem o desenvolvimento híbrido, além da criação e implementação de algoritmos clássicos em ambientes quânticos.

Palavras-chave: Computação Quântica, Computação em Nuvem, Banco de Dados, Criptografia Clássica, Algoritmos.

**ABSTRACT**

Classical computing, currently used, has been facing significant challenges that are physically impossible to solve due to the level of resources required to address such requirements. Tasks such as data processing have become increasingly complex due to the exponential increase in data production, considering that every electronic device today needs to connect to internet networks, receive an IP, and generate data through it. From clocks and smartphones to refrigerators, televisions, and household appliances, all of them contribute to this data generation. As a result, the time required for data processing or any other task has exponentially grown, unable to meet this high demand quickly or with the resources available through machine and equipment processors. Many other advancements in the fields of Artificial Intelligence, IoT, Machine Learning, Cloud Computing, and Database management require more and more resources and speed.

Quantum computing promises to solve the aforementioned problems and enables advancements and innovations in various fields such as the environment, energy, healthcare, agriculture, and more. The goal is for quantum computers to be capable of processing complex computational information that is impossible for classical computing, offering not just a faster way of performing existing computations, but a fundamentally different approach that promises to effectively solve problems that classical computing could never tackle.

Therefore, the current research focuses on the study of computational languages, platforms, and libraries that enable the development of classical tasks and algorithms to be executed in a virtual quantum environment. The aim is to subsequently transcribe these algorithms into their quantum versions and apply them in the encryption of a database. To carry out this research, literature reviews were conducted regarding the concepts of Quantum Computing, Cloud Computing, Databases, Languages, and Platforms that allow for hybrid development. Additionally, classical algorithms were created and implemented in quantum environments.

Keywords: Quantum Computing, Cloud Computing, Databases, Classical Cryptography, Algorithms.

**SUMÁRIO**

[1. INTRODUÇÃO 8](#_Toc139411965)

[2. OBJETIVOS 9](#_Toc139411966)

[2.1 Objetivos Específicos 9](#_Toc139411967)

[3. JUSTIFICATIVA 10](#_Toc139411968)

[4. DESENVOLVIMENTO 11](#_Toc139411969)

[4.1 Computação Quântica 11](#_Toc139411970)

[4.2 Computação em Nuvem 12](#_Toc139411971)

[4.2.1 Origem 12](#_Toc139411972)

[4.2.2 Definição 13](#_Toc139411973)

[4.2.3 Tipos de serviço 13](#_Toc139411974)

[4.2.4 Modelos de Nuvem 14](#_Toc139411975)

[4.3 Plataformas e Simuladores Quânticos 15](#_Toc139411976)

[4.3.1 Microsoft Azure 15](#_Toc139411977)

[4.3.2 Azure Quantum 15](#_Toc139411978)

[4.4 Banco de Dados 16](#_Toc139411979)

[4.4.1 Sistema de Gerenciamento de Banco de Dados (SGBD) 16](#_Toc139411980)

[4.4.2 Banco de Dados Relacional 16](#_Toc139411981)

[4.4.3 MySQL 17](#_Toc139411982)

[4.4.3.1 Origem 17](#_Toc139411983)

[4.4.3.2 Instalação do SGBD 17](#_Toc139411984)

[4.4.3.3 Configuração 18](#_Toc139411985)

[4.4.4 Funções de Banco de Dados 20](#_Toc139411986)

[4.5 Linguagens de Programação 20](#_Toc139411987)

[4.5.1 Python 20](#_Toc139411988)

[4.5.2 Q# (QSharp) 20](#_Toc139411989)

[4.5.3 Quantum Development Kit (QDK) 20](#_Toc139411990)

[4.5.4 Qiskit 21](#_Toc139411991)

[4.5.5 SQL 21](#_Toc139411992)

[4.6 Criptografia 21](#_Toc139411993)

[4.6.1 Criptografia Simétrica 22](#_Toc139411994)

[4.6.1.1 AES 22](#_Toc139411995)

[4.6.1.2 DES 22](#_Toc139411996)

[4.6.2 Criptografia Assimétrica 22](#_Toc139411997)

[4.6.2.1 RSA 22](#_Toc139411998)

[4.6.2.2 ECC 22](#_Toc139411999)

[4.6.3 MD5 22](#_Toc139412000)

[4.6.4 SHA-1 23](#_Toc139412001)

[4.6.5 SHA-2 23](#_Toc139412002)

[4.6.6 SHA-3 23](#_Toc139412003)

[4.7 Extensões 23](#_Toc139412004)

[4.7.1 Instalação de extensão MySQL - VsCode 23](#_Toc139412005)

[4.7.1.1 Extensão configurada 25](#_Toc139412006)

[4.7.1.2 Testes 25](#_Toc139412007)

[4.8 GitHub 26](#_Toc139412008)

[4.8.1 Repositórios 27](#_Toc139412009)

[4.8.1.1 Projeto para planejamento de horário no GitHub 27](#_Toc139412010)

[4.9 Revisão Bibliográfica 29](#_Toc139412011)

[4.9.1 Implementação do algoritmo quântico Deutsch-Jozsa em linguagem funcional e no simulador IBM Q Experience 29](#_Toc139412012)

[4.9.2 Aplicação de protocolos quânticos e algoritmo de Shor para a Segurança da Informação 30](#_Toc139412013)

[4.9.3 Desempenho de algoritmos quânticos e clássicos em treinamento de Machine Learning supervisionado 33](#_Toc139412014)

[5. METODOLOGIA 34](#_Toc139412015)

[6. RESULTADOS 35](#_Toc139412016)

[6.1. Implementação de Q# e Python 35](#_Toc139412017)

[6.1.1. Criação de um projeto em Q# 35](#_Toc139412018)

[7. CRONOGRAMA 36](#_Toc139412019)

[7.1. Cargos da Equipe 39](#_Toc139412020)

[8. REFERÊNCIAS BIBLIOGRÁFICAS 39](#_Toc139412021)

**LISTA DE FIGURAS**

[Figura 1 - Instalação do MySQL Server – Passo 1 17](#_Toc136777798)

[Figura 2- Instalação do MySQL Server – Passo 2 17](#_Toc136777799)

[Figura 3 - Configuração do MySQL Server 18](#_Toc136777800)

[Figura 4 - Configuração do MySQL Server - Inserção de Senha 19](#_Toc136777801)

[Figura 5 - Configuração do MySQL Server - Teste de Conexão 19](#_Toc136777802)

[Figura 6 - Instalação do MySQL no VsCode – Passo 1 21](#_Toc136777803)

[Figura 7 - Instalação do MySQL no VsCode – Passo 2 21](#_Toc136777804)

[Figura 8 - Instalação do MySQL no VsCode – Passo 3 22](#_Toc136777805)

[Figura 9 - Instalação do MySQL no VsCode – Passo 4 22](#_Toc136777806)

[Figura 10 - Criação do banco de dados 22](#_Toc136777807)

[Figura 11 - Criação de tabela no banco de dados 23](#_Toc136777808)

[Figura 12 - Inserção de dados na tabela 23](#_Toc136777809)

[Figura 13 - Pesquisa no banco de dados 23](#_Toc136777810)

[Figura 14 - Planejamento de Tarefas - GitHub 24](#_Toc136777811)

[Figura 15 - Criando um repositório - GitHub – Passo 1 24](#_Toc136777812)

[Figura 16 - Criando um repositório - GitHub – Passo 2 25](#_Toc136777813)

[Figura 17 - Criando um repositório - GitHub – Passo 3 25](#_Toc136777814)

# 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 e rápida 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.

Então, faz-se necessário o estudo de plataformas e linguagens que auxiliem e facilite na migração da computação clássica para quântica, e esse estudo permitirá que pesquisadores e profissionais da área compreendam as ferramentas disponíveis e as dificuldades encontradas, e possam explorar todo o potencial da computação quântica para resolver problemas complexos em busca de impulsionar a inovação tecnológica em diversos setores.

Porém, com o avanço da computação quântica, a segurança vai sendo 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, derrubando a segurança que era estabelecida.

Portanto, a maneira como os dados gerados (que está em contante crescimento) são armazenados, impacta diretamente na segurança dos mesmos. Segundo Elmasri e Navathe (2016), "A criptografia é uma ferramenta crucial para proteger informações confidenciais em sistemas de banco de dados, garantindo que apenas pessoas autorizadas possam acessá-las", garantindo assim a integridade dos dados. O tripé da segurança de dados, nada mais é, do que disponibilidade, confiabilidade e integridade das informações e uma vez, que um dos fatores principais para assegurar a segurança (integridade, alcançada através da criptografia) está em xeque, faz-se necessário o estudo de novas formas de proteção, através do uso da computação quântica, que é a tecnologia que vem sido estudada e desenvolvida, para que quando ocorrer a migração das mesmas, não venha ter um impacto direto nessa área.

Em suma, o presente projeto de pesquisa poderá contribuir para um avanço significativo na área da tecnologia, uma vez que irá buscar através de pesquisas técnicas e aplicações práticas, identificar quais os desafios que podem ser enfrentados na migração da computação clássica para a quântica, o desenvolvimento de um ambiente de testes que busca facilitar a transição das tecnologias, desde simuladores e ambientes de desenvolvimento, até linguagens de programação, bibliotecas e etc, a realização de uma avaliação sistemática e uma comparação entre diferentes plataformas e linguagens disponíveis para a computação quântica que contribuirá para o avanço da computação quântica de maneira geral e aplicabilidade na segurança dentro de um banco de dados, através do uso da criptografia.

# OBJETIVOS

Em suma, o objetivo geral deste projeto é demonstrar o efeito das mudanças ocasionadas pela migração da tecnologia clássica para quântica, destacando seu impacto, principalmente na segurança de um banco de dados, e buscar plataformas e linguagens que possibilitem a migração da computação clássica para a computação quântica, de maneira segura e com baixa complexidade.

## 2.1 Objetivos Específicos

* Buscar soluções quânticas, principalmente dentro da plataforma Azure, que contemplem os requisitos necessários para o desenvolvimento do projeto, mesclando computação clássica com a quântica;
* Demonstrar a volatilidade da segurança de um banco de dados clássico quando se considera a capacidade de processamento exponencialmente mais avançada da computação quântica.
* Comprovar a usabilidade da computação quântica dentro de sistemas clássicos atuais, de maneira prática, sem abstrações que a computação quântica não seja uma abstração.
* Testar e verificar a eficiência da computação quântica em um banco de dados clássico, especificamente em algoritmos criptográficos usados largamente em bancos de dados;
* Elencar as possíveis aplicabilidades da plataforma Microsoft Azure e das linguagens de programação Python e Q#, relacionando as linguagens e os recursos quânticos a bancos de dados relacionais.
* Construir um código híbrido utilizando as linguagens Q#, Python e MySQL, implementando recursos quânticos que interajam com a programação clássica, demonstrando dessa maneira a usabilidade das soluções quânticas em processos de manipulação de um banco de dados, destacando a quebra da criptografia de dados e comparando os resultados de um algoritmo quântico e de um clássico.

# JUSTIFICATIVA

A computação quântica é uma área que tem despertado grande interesse e expectativas na atualidade, desde grandes empresas de tecnologia e profissionais da área, até entusiastas em busca de inovações tecnológicas. Isso se dá graças ao seu imenso portencial de revolucionar o campo da tecnologia da informação, além de diversas outras áreas em que seu uso pode ser aplicado.

A importância da computação quântica se dá através dos problemas que ela pode solucionar, sendo eles problemas complexos, de maneira mais eficiente e otimizada do que os sistemas clássicos utilizados atualmente. Enquanto os computadores clássicos usam o paradigma de computação baseada em bits, a computação quântica utiliza do qubits, que podem estar em estados simultâneos entre 0 e 1. Essa propriedade permite que os problemas sejam solucionados de maneira mais rápida e eficiênte, que abrirá portas para avanços e soluções tecnológicas significativas. A utilização de qubits, explora os princípios da superposição e emaranhamento quântico, que permite a resolução eficiente de problemas complexos, como a fatoração de números inteiros grandes e a otimização de algoritmos, que implicam diretamente na segurança da criptografia e no desenvolvimento de novos métodos de aprendizado de máquina.

Assim como Nielsen e Chuang (2010) apontam, os “computadores quânticos vieram para resolver problemas que são impossíveis de serem resolvidos em computadores clássicos, não porque sejam insolúveis, mas sim pela grande quantidade de recursos necessários para a sua solução”.

A computação quântica representa um avanço tecnológico que desempenha um papel fundamental na Quarta Revolução Industrial, a qual é caracterizada pela união de tecnologias digitais, físicas e biológicas. Nesse contexto, a computação quântica assume uma posição central como um catalisador da inovação, proporcionando novas oportunidades para a transformação dos processos produtivos e sistemas de informação. A Quarta Revolução Industrial é marcada pela integração de tecnologias disruptivas, como a Internet das Coisas (IoT), inteligência artificial, big data e computação em nuvem. Nesse contexto, a computação quântica é a força que impulsiona a inovação e resolve problemas complexos que desafiam a capacidade dos sistemas computacionais tradicionais. Ela também tem o potencial de promover avanços em áreas de sistemas físicos complexos, descoberta de materiais e modelagem molecular, o que impacta diretamente em setores de medicina, energia, manufatura avançada e logística, além de auxiliar com a resolução eficiente de problemas, o que irá gerar de produtividade e competitividade.

Assim como as descobertas e inovações do passado foram cruciais para o desenvolvimento econômico, social, cultural e de diversos setores da atualidade, a computação quântica e as áreas que a envolvem serão cruciais para inovações e desenvolvimentos futuros.

Portanto, faz-se necessário a busca por linguagens, plataformas e bibliotecas que possibilitem e facilite a transição das tecnologias, de clássica para quântica, e a busca para identificar quais os desafios que podem ser enfrentados no processo de migração. E através disso, determinar aplicação da parte de segurança e criptografia, dentro de um banco de dados em um simulador quântico.

# DESENVOLVIMENTO

Na busca de atender os objetivos de elencar e definir os conceitos das áreas relacionadas à Computação Quântica, através das pesquisas bibliográficas em livros e nos meios de publicações técnicas e científicas, foram obtidos os conceitos de Computação Quântica, Computação em Nuvem, Banco de Dados, Microsof Azure, algumas Linguagens de Programação, como Python, Q# e SQL, e a busca por Segurança nos bancos de dados, através du uso de Criptografia.

## 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 para ser obtido (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).

Porém, a Computação Quântica ainda enfrenta desafios, como a correção de erros quânticos, a instabilidade dos qubits, a necessidade de melhoria da coerência e escalabilidade dos sistemas quânticos, a falta de um hardware quântico mais robusto e algoritmos quânticos otimizados. Entretanto, muitas das grandes empresas de tecnologia, como a Microsoft, IBM e Sansung, tem investido muito em pesquisa e desenvolvimento para atender as demandas para o desenvolvimento completo e possibilidade de uso e migração para a Computação Quântica. (MIANO & OLIVEIRA, 2021).

## Computação em Nuvem

Através das evoluções tecnológicas, a IoT, entre outros, o número de dados gerados por cada indivíduo tem crescido exponencialmente. (IDC, 2021). Tudo o que fazemos, cada página da web que acessamos, cada dispositivo que utilizamos se conecta através de uma rede, compartilha pacotes através dela e necessita de armazenamento suficiente para transitar e armazenar as informações que são geradas e na maioria das vezes confidenciais. Desde os computadores, smartphones, impressoras, até as nossas geladeiras e televisões são geradores de informação. Por conta dessa alta demanda da necessidade de recursos, iniciou-se uma procura por alternativas que fossem viáveis, seguras, disponíveis e "mais baratas" para se manter, visando que um aparelho que tem os recusos mais próximos do necessário, é caro, pouco durável, e pensando em alta escala, inviável. Além do equipamento, tem-se o gasto com rede, infraestrutura, segurança, refrigeração, manutenção, upgrade e inúmeros outros fatores que podem ser geradores de custos. (MICROSOFT, 2023a).

## Origem

O termo computação em nuvem ou Cloud Computing, foi utilizado pela primeira vez, no ano de 1997, por Ramnath Chellappa, um professor de sistemas de informação. Em meados de 1950 esse conceito de computação em nuvem já existia, quando John McCarthy havia proposto o uso da computação como uma utilidade. Porém só em 1997 o termo é proferido durante uma palestra do professor Ramnath, na qual ele dizia que as empresas iriam passar a trocar a sua infraestrutura de hardware por serviços disponíveis em nuvem. Após alguns anos, as empresas realmente começaram a utilizar os serviços em nuvem por conta dos benefícios, como redução de custos e simplificações de processos. (BESSA, 2023; CANTU, 2011)

Atualmente, utilizamos de serviços em nuvem sem nem nos darmos conta, como o uso de serviços de streaming de música e vídeos, serviços de compartilhamento e armazenamento de arquivos, e diversos outros serviços que são hospedados ou necessitam da nuvem para seu funcionamento.

Através do uso de de soluções em Cloud, existe a possibilidade de se utilizar recursos de armazenamento e computação de forma compartilhada e delegar a manutenção de infraestrutura, reduzindo drasticamente os gastos. Algumas das vantagens obtidas é a redução de gastos com manutenção e disponibilidade de infraestrutura de TI, otimização do trabalho, maior velocidade de processamento, armazenamento e agilidade nos processos, e os provedores de cloud são os responsáveis pela implementação e garantia de redundância de serviços, realização de backups de segurança e disponibilidade do serviço 24 horas por dia, 7 dias por semana. (BESSA, 2023)

Algumas das desvantagens seria a dependência de rede e conexão com a internet. Outras variáveis seriam em caso de falhas nas estruturas dos data centers físicos, a ocorrência de interferências no uso dos recursos oferecidos pela nuvem, falhas de acesso devido o uso compartilhado de recursos entre diferentes empresas, e os riscos relacionados à segurança, como perda ou sequestro de dados.

## Definição

A computação em nuvem, nada mais é do que a entrega de serviços de computação pela internet, como infraestrutura de TI comum, máquinas virtuais, armazenamento, bancos de dados e rede. Os serviços de nuvem também podem incluir itens como IoT (Internet das Coisas), ML (machine learning) e IA (inteligência artificial). Essa entrega é feita sob demanda, com definição de preço de pagamento conforme o uso, e todo esse acesso é fornecido e gerenciado através do uso de um provedor de nuvem. (MICROSOFT, 2023a; AWS, 2023).

## Tipos de serviço

Existem 3 tipos de serviços de compuatação em nuvem:

- Infraestrutura como serviço (IaaS): o qual contém os componentes básicos da TI na nuvem. Oferece acesso à recursos de rede, computadores e espaço para armazenamento de dados, e é o tipo de computação mais semelhante aos recursos existentes de TI. Irá oferecer a infraestrutura veirtualizada que será acessado pelo uauário através da nuvem. (AWS, 2023).

- Plataforma como serviço (PaaS): não é necessário mais o gerenciamento dos recursos de infraestrutura, como atualizações de sistema operacional, etc. O foco seria apenas na implantação e gerenciamento dos aplicativos, e o provedor irá gerenciar a infraestrutura. É criado um ambiente para a programação e o desenvolvimento de alguma solução. (AWS, 2023).

- Software como serviço (SaaS): É a solução mais completa, pois todo o produto será executado e gerenciado pelo provedor de serviços. não será necessário pensar sobre a manutenção do serviço ou o gerenciamento da infraestrutura subjacente, apenas em como utilizará esse software específico, o mesmo é executado em um servidor, tirando a necessidade do usuário tê-lo instalado em sua máquina. (AWS, 2023).

## Modelos de Nuvem

- Nuvem privada: oferece uma infraestrutura alocada para uma única organização. É a evolução natural de um datacenter corporativo. Ela fornece um controle muito maior para a empresa e o departamento de TI. Porém tem mais custos e menos benefícios para a implementação, em relação à nuvem pública. A nuvem privada pode ser hospedada em um datacenter local, em um datacenter dedicado externo, e até mesmo por terceiros que tenham dedicado esse datacenter à prganização. (MICROSOFT, 2023b; AWS, 2023)

- Nuvem pública: é criada, controlada e mantida por um provedor de nuvem de terceiros. Qualquer pessoa que queira comprar serviços de nuvem pode acessar e usar os recursos. A disponibilidade pública geral é uma diferença fundamental entre nuvens públicas e privadas. Os serviços são mantidos por um determinado provedor de serviços, que possuem uma série de data centers com uma gama de servidores interconectados e que estão compartilhando recursos e serviços pelos utilizadores da nuvem. (MICROSOFT, 2023b; AWS, 2023)

- Nuvem híbrida: é um ambiente de computação que usa nuvens públicas e privadas em um ambiente interconectado. Um ambiente de nuvem híbrida pode ser usado para permitir que uma nuvem privada escale para atender a uma demanda maior temporária implantando recursos de nuvem pública. A nuvem híbrida pode ser usada para fornecer uma camada adicional de segurança. Por exemplo, os usuários podem escolher com flexibilidade quais serviços manter na nuvem pública e quais implantar na infraestrutura de nuvem privada. Um cenário muito comum seria o de usar uma nuvem pública para manter a execução de algum sistema e manter a fonte de dados da aplicação na nuvem privada. (MICROSOFT, 2023b; AWS, 2023).

Principais provedores de Cloud: AWS ou Amazon Web Services, Azure (que será utilizado na realização dos testes da pesquisa), Google Cloud, IBM, o Oracle e o VMWare, e outros.

## 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 do presente projeto, pois podemos, através delas, operar em ambientes quânticos através das máquinas atuais.

Existem várias plataformas e simuladores que oferecem esse benefício, como por exemplo a IBM Quantum Experience (plataforma baseada na nuvem oferecida pela IBM), Qiskit (plataforma de software de código aberto desenvolvida pela IBM), Cirq (biblioteca de código aberto desenvolvida pelo Google), e diversas outras.

No presente projeto, utilizaremos o Azure Quantum, com o conjunto de ferramentas de desenvolvimento QDK (Microsoft Quantum Development Kit) e a linguagem de programação Q# (Q Sharp), todas fornecidas pela Microsoft, que oferece recursos para computação quântica ser explorada, oferecendo suporte à simulação, programação e integração com hardware quântico. As ferramentas foram escolhidas por mais familiaridade ao uso das ferramentas da Microsoft e pela parceria da Microsoft com a FATEC, que possibilita o uso dessas ferramentas de forma gratuita, através das contas institucionais e o crédito de $100 oferecidos para uso na plataforma do Azure.

## Microsoft Azure

A Microsoft Azure é uma plataforma em nuvem desenvolvida pela Microsoft, que oferece mais de 200 produtos e serviços - muitos deles oferecidos de maneira gratuita. A plataforma foi escolhida pelo grupo para desenvolvimento do projeto, e logo de início mostrou-se intuitiva e de fácil compreensão, demonstrando também durante os testes uma ótima capacidade de processamento de dados e desenvolvimento de algoritmos quânticos. Ademais, a Microsoft faz questão de estar em constante desenvolvimento, investindo anualmente bilhões de dólares para proporcionar um alto desempenho em cada uma de suas funcionalidades e aprimorar a segurança dos dados confiados à plataforma. (MICROSOFT, 2023c)

## Azure Quantum

O Azure Quantum é um serviço oferecido pelo Azure que funciona como um ecossistema em nuvem aberto, permitindo a interação com software, hardware e outras soluções de nível quântico. Dentre as linguagens suportadas pela plataforma estão: qiskit, Cirq e Q#, além da linguagem multiparadigma Python, que será utilizada na maior parte dos testes desta pesquisa. A interação dos códigos com a plataforma é realizada dentro de Workspaces por meio de um Jupyter Notebook criado pela Microsoft, que pode ser acessado de maneira gratuita por qualquer um que crie uma conta no Azure. (MICROSOFT, 2023d).

## Banco de Dados

Banco de dados é projetado para lidar com grandes volumes de dados e oferecer mecanismos para inserção, atualização, exclusão e consulta de dados. Os bancos de dados são amplamente utilizados em diversos setores, para armazenar e acessar dados estruturados de forma confiável. Desempenham um papel fundamental em várias áreas e são usados para armazenar informações estruturadas de forma confiável, garantindo a integridade e a consistência dos dados.

Os bancos de dados oferecem benefícios significativos, como a capacidade de armazenar grandes quantidades de dados de forma organizada, a possibilidade de realizar consultas complexas e a garantia da consistência e integridade dos dados. (RAMAKRISHNAN & GEHRKE, 2020).

## Sistema de Gerenciamento de Banco de Dados (SGBD)

Os sistemas de gerenciamento de banco de dados (SGBDs) são usados para controlar e administrar o acesso aos dados armazenados, garantindo a segurança e a privacidade das informações. Os SGBDs fornecem uma interface para que os usuários possam interagir com o banco de dados por meio de consultas e comandos específicos da linguagem de consulta, como a linguagem SQL.

## Banco de Dados Relacional

Dentre os tipos de banco de dados, o mais comum é o banco de dados relacional. Nesse modelo, os dados são organizados em tabelas compostas por linhas e colunas. Cada tabela representa uma entidade do mundo real, e as relações entre as entidades são estabelecidas por meio de chaves primárias e estrangeiras. As colunas representam os atributos ou características das entidades, enquanto as linhas contêm os registros individuais de dados. O modelo relacional oferece uma estrutura flexível para a organização e consulta de dados.

A linguagem SQL (Structured Query Language) é a mais utilizada para interagir com bancos de dados relacionais, pois permite a criação de consultas para recuperar, inserir, atualizar e excluir dados, e a definição de esquemas, índices e restrições de integridade.

## MySQL

O MySQL é um servidor e gerenciador de banco de dados relacional de código aberto, de licença dupla (Oracle e GNU) projetado inicialmente para aplicações de pequenos e médios portes, porém atualmente atende a aplicações de grande porte. Na criação de aplicações web abertas e gratuitas, o conjunto de aplicações mais usado é o LAMP, um acrônimo para Linux, Apache, MySQL e Perl/PHP/Python. Nesse conjunto de aplicações, inclui-se, respectivamente, um sistema operacional, um servidor web, um sistema gerenciador de banco de dados e uma linguagem de programação. Assim, o MySQL é um dos componentes centrais da maioria das aplicações públicas da Internet. (MySQL, 2023).

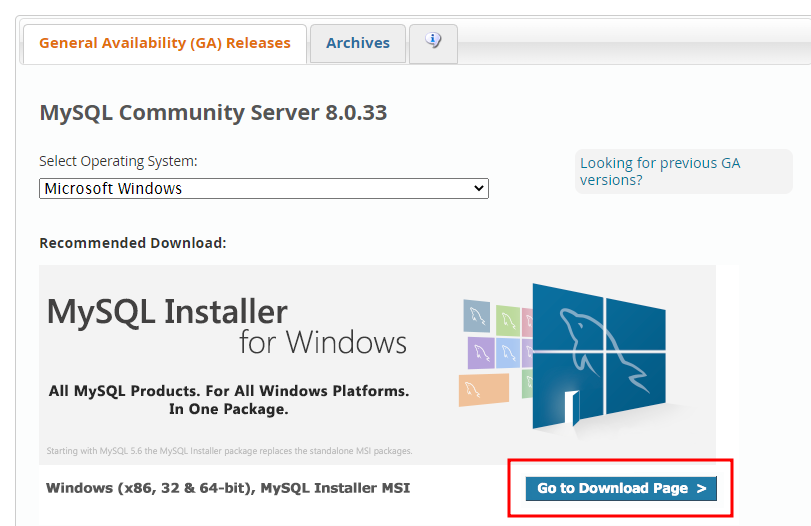
## Origem

Foi criado na Suécia, por David Axmark, Allan Larsson e o finlandês Michael Widenius. Eles começaram o projeto em 1980. O MySQL é um SGBD, um Sistema de gerenciamento de [banco de dados](https://www.infoescola.com/informatica/banco-de-dados/), que usa a linguagem SQL como interface. Este banco de dados é conhecido por sua facilidade de uso, sendo ele usado pela NASA, HP, Bradesco, Sony, e muitas outras empresas. Sua interface simples, e sua capacidade de rodar em vários sistemas operacionais, são alguns dos motivos para este programa ser tão usado atualmente, e seu uso estar crescendo cada vez mais.

## Instalação do SGBD

Primeiramente, para fazer a instalação, deve-se acessar o site oficial do MySQL e fazer a instalação do MySQL Server seguindo os passos abaixo:

Figura 1 - Instalação do MySQL Server – Passo 1



Fonte: Autores

Figura 2- Instalação do MySQL Server – Passo 2

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: Autores

## Configuração

As configurações são definidas por padrão, com exceção da configuração abaixo, que deve ser realizada manualmente:

Figura 3 - Configuração do MySQL Server

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Autores

Insira uma senha de acesso, conforme sua preferência:

Figura 4 - Configuração do MySQL Server - Inserção de Senha

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Autores

Realize um teste de conexão com o Banco:

Figura 5 - Configuração do MySQL Server - Teste de Conexão

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: Autores

## Linguagens de Programação

## 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.

## Q# (QSharp)

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)

## SQL

SQL (Structured Query Language) é uma linguagem de programação usada para gerenciar e manipular bancos de dados relacionais. Embora seja chamada de linguagem de programação, o SQL é especificamente projetado para consultas e manipulações de dados em bancos de dados. Essa linguagem é a mais utilizada para interagir com bancos de dados relacionais, pois 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).

A linguagem SQL é padronizada e possui uma sintaxe semelhante em diferentes sistemas de gerenciamento de banco de dados (SGBDs), como MySQL, Oracle, Microsoft SQL Server, PostgreSQL, entre outros. No entanto, cada SGBD pode ter suas próprias extensões e recursos específicos.

## Criptografia

A criptografia é o processo de transformar informações legíveis em um formato ilegível chamado de texto cifrado, usando algoritmos matemáticos específicos. Ela é amplamente utilizada para proteger a confidencialidade e a integridade dos dados, garantindo que apenas as partes autorizadas possam acessar e compreender as informações.

Existem dois tipos principais de criptografia: a criptografia simétrica e a criptografia assimétrica. Além desses tipos básicos de criptografia, também existem outras técnicas, como a criptografia de hash, que produz um resumo fixo de tamanho fixo de uma mensagem, e a criptografia de ponta a ponta, que protege a comunicação entre dois pontos específicos.

A segurança da criptografia depende da robustez do algoritmo criptográfico, do tamanho da chave utilizada e das práticas adequadas de gerenciamento de chaves. Além disso, a criptografia por si só não protege contra outros tipos de ataques, como ataques de força bruta, ataques de engenharia social ou falhas de segurança em sistemas subjacentes. Portanto, é essencial adotar uma abordagem abrangente de segurança que inclua criptografia juntamente com outras medidas de proteção.

## Criptografia Simétrica

Nesse tipo de criptografia, 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).

## Criptografia Assimétrica

Nesse tipo de criptografia, também conhecida como 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.

## MD5

MD5 (Message Digest Algorithm 5) é um algoritmo de hash criptográfico amplamente utilizado para produzir um resumo fixo de tamanho de 128 bits (16 bytes) de uma mensagem ou dados. Foi desenvolvido por Ronald Rivest em 1991.

O objetivo principal do MD5 é verificar a integridade dos dados, fornecendo um valor de hash único para uma determinada entrada de dados. O algoritmo MD5 gera um hash que é altamente improvável de ser reproduzido a partir de outro conjunto de dados, o que o torna útil para verificar se os dados foram modificados ou corrompidos durante a transmissão ou armazenamento.

No entanto, é importante ressaltar que o MD5 não é considerado seguro para criptografia ou autenticação de dados em ambientes sensíveis atualmente. Isso ocorre porque várias vulnerabilidades foram descobertas no algoritmo ao longo dos anos, tornando-o suscetível a ataques de colisão, nos quais dois conjuntos de dados diferentes podem produzir o mesmo hash.

Devido a essas vulnerabilidades, o uso do MD5 para fins de segurança é desencorajado. Em vez disso, são recomendados algoritmos mais robustos e seguros, como o SHA-256 (Secure Hash Algorithm 256 bits) ou o SHA-3.

No entanto, o MD5 ainda é usado em certos casos em que a segurança não é uma preocupação principal, como na verificação de integridade de arquivos, na geração de identificadores únicos ou na validação de senhas em sistemas legados. Para aplicativos que exigem segurança adequada, é fundamental adotar algoritmos de hash mais fortes e considerar outras medidas de segurança.

## Extensões

As extensões, também conhecidas como plugins, são ferramentas externas, desenvolvidas por outras pessoas desenvolvedoras, cujo objetivo é adicionar recursos e auxiliar no ambiente de desenvolvimento do nosso projeto. Ao utilizarmos esses plugins, conseguimos deixar o editor de código totalmente customizado e preparado para suprir as nossas necessidades.

## **Instalação de extensão MySQL - VsCode**

Para fazer a instalação do MySQL no VsCode siga o seguinte caminho:

**- Passo 1:**

Figura 6 - Instalação do MySQL no VsCode – Passo 1

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Fonte: Autores

**- Passo 2:**

Na aba de pesquisa digite MySQL Shell for VsCode e logo após instale a extensão.

Figura 7 - Instalação do MySQL no VsCode – Passo 2

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Fonte: Autores

**- Passo 3:**

Acesse a extensão, clique em DB Connections e crie uma conexão que esteja vinculada a que já foi criada no MySQL Workbench.

Figura 8 - Instalação do MySQL no VsCode – Passo 3

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Fonte: Autores

## **Extensão configurada**

Figura 9 - Instalação do MySQL no VsCode – Passo 4

Texto

Descrição gerada automaticamente

Fonte: Autores

## **Testes**

- Banco de dados criado:

Figura 10 - Criação do banco de dados

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Autores

- Tabela criada:

Figura 11 - Criação de tabela no banco de dados

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Fonte: Autores

- Inserindo valores na tabela:

Figura 12 - Inserção de dados na tabela



Fonte: Autores

- Fazendo pesquisa no banco:

Figura 13 - Pesquisa no banco de dados

Tela de celular

Descrição gerada automaticamente

Fonte: Autores

## GitHub

O GitHub é uma plataforma de hospedagem de código para controle de versão e colaboração. Permite que diversas pessoas trabalhem em conjunto em projetos de qualquer lugar.

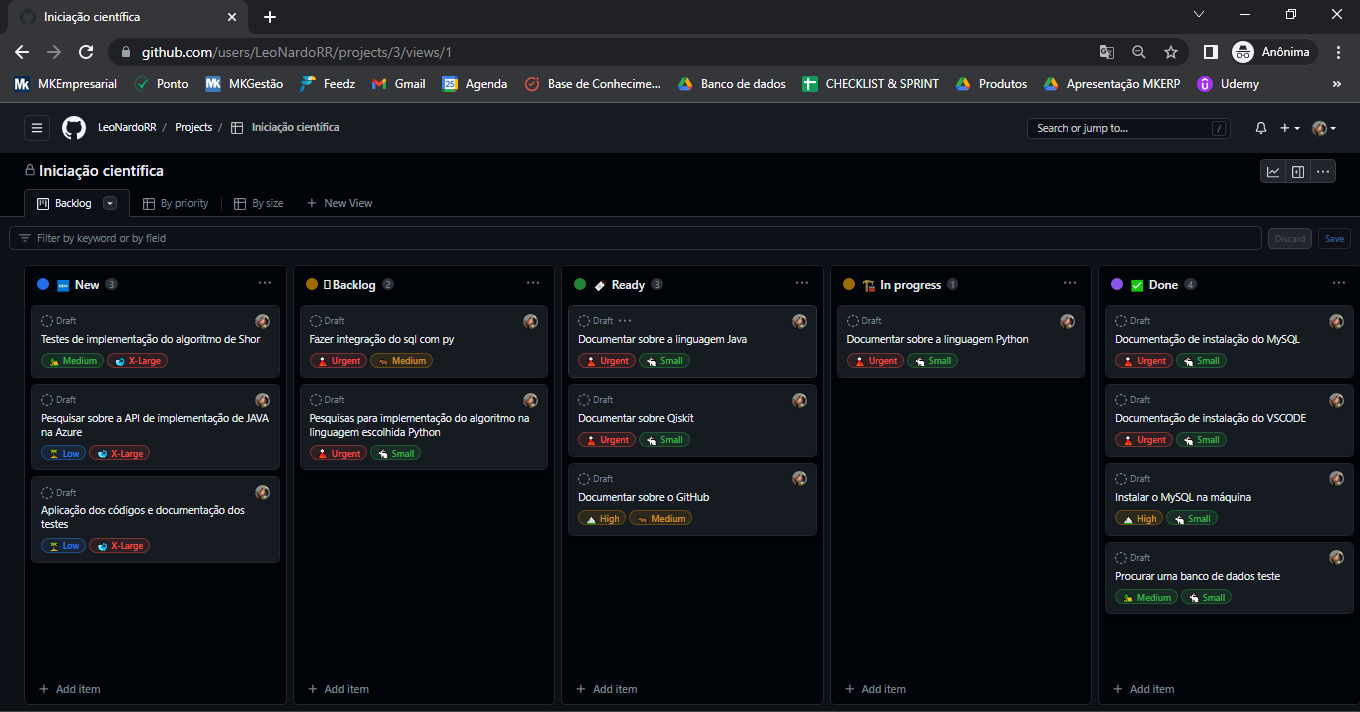
Este tutorial ensina os princípios básicos de GitHub como, por exemplo, repositórios, branches, commits e pull requests. Você criará seu próprio repositório Hello World e aprenderá o fluxo de trabalho de pull request de GitHub, uma maneira popular de criar e revisar o código.

(REVISAR TOPICO)

## Repositórios

## Projeto para planejamento de horário no GitHub

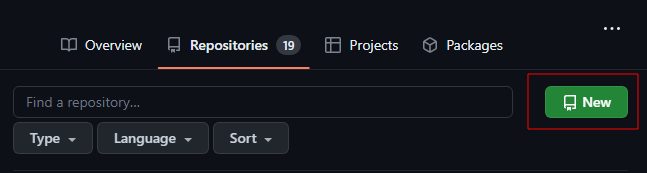
Figura 14 - Planejamento de Tarefas - GitHub



Fonte: Autores

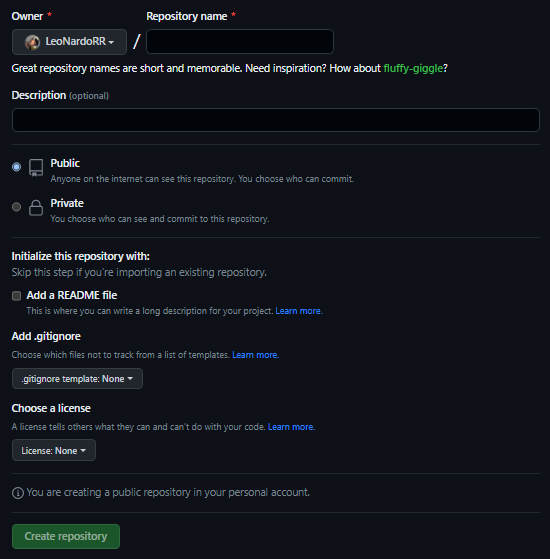
**- Hospedando código:**

Figura 15 - Criando um repositório - GitHub – Passo 1



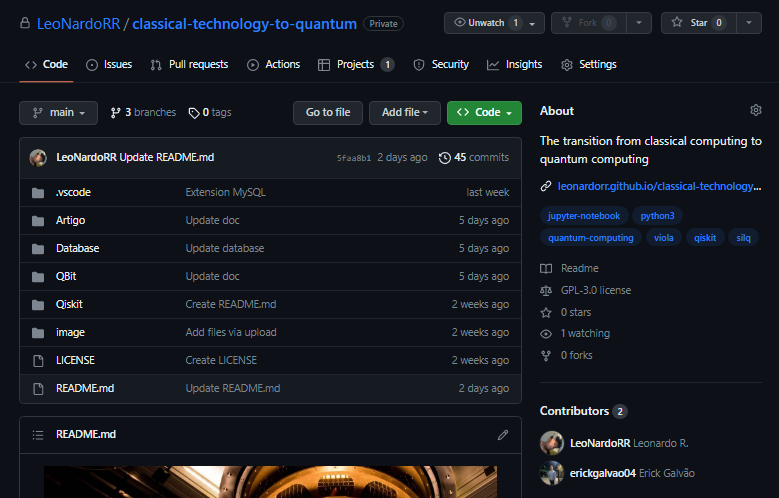
Fonte: Autores

Figura 16 - Criando um repositório - GitHub – Passo 2



Fonte: Autores

Figura 17 - Criando um repositório - GitHub – Passo 3



Fonte: Autores

## Revisão Bibliográfica

Antes do desenvolvimento do projeto ser realizado, por instrução da orientadora Prof. Dra. Mariana Godoy Vazquez, foi realizado uma revisão bibliográfica de três artigos, e através deles, realizamos o resumo e uma apresentação oral de cada um deles, para introdução e fixação do conteúdo, e a partir disso, demos início aos trabalhos de pesquisa e desenvolvimento.

## Implementação do algoritmo quântico Deutsch-Jozsa em linguagem funcional e no simulador IBM Q Experience

Com a constante evolução da Física e Computação Quântica é necessário compreender a lógica de programação deste novo paradigma, tendo ele três etapas: preparação dos estados iniciais, realização das transformações unitárias e execução das medições. Portanto abordaremos os conceitos de Computação Quântica e diferentes linguagens para implementações no simulador IBM Q Experience.

Para compreender a álgebra linear é necessário conhecer alguns conceitos matemáticos, como Espaço Vetorial Real, Subespaço Vetorial, Combinação Linear de Vetores, Produto Interno em Espaço Vetorial, Coordenadas Polares, Números complexos, Forma algébrica e Representação geométrica.

A Lei de Moore foi determinada pelo co-fundador da Intel Corporation, Gordon Earl Moore, formado em química e física, em um artigo escrito pelo mesmo e publicado pela revista Eletronic Magazine no ano de 1965. No artigo de sua autoria, ele conjectura sobre a evolução da tecnologia e a previsão para os anos futuros. Moore previu que a cada dezoito meses, a complexidade para a criação de processadores seria maior, a capacidade de processamento aumentaria em 100% e o custo permaneceria constante. Ou seja, o número de transitores em um processador dobraria, mantendo o mesmo custo e espaço. Os processadores passariam a ter capacidades maiores e melhores em um único “chip”. Essa previsão deu início a uma jornada de investimento em pesquisas e desenvolvimento, que possibilitou o alcance dessa meta. Porém, a Lei está entrando em obsolescência, pois em determinado momento será impossível aumentar a capacidade do processador e mantê-lo no mesmo espaço, tendo em vista que já estão sendo densenvolvidos processadores com transitores na ordem dos nanômetros.

Já Notação de Dirac ou Bra-ket é a mais usada para descrever sistemas mecânicos quânticos e também registradores quânticos. Espaço de Hilbert é um vetor espacial com um produto interno e uma regra definida pelo produto interno.

Em relação aos conceitos de computação quântica essenciais, temos a Sobreposição de estados que é a possibilidade dos *qubits* poderem existir continuamente em estados entre 0 e 1 e até serem estimulados. O Bit quântico (qubit) que pode ser encontrado nos estados 0, 1 ou em uma superposição entre os dois estados, sendo assim, pode existir nos dois estados ao mesmo tempo. No momento da medição do qubit, ele “se torna um bit clássico” pois não se pode ter os dois estados na medição, ele tem 50% de possibilidade de ser zero e 50% de possibilidade de ser um. É como uma foto de algo em movimento, mesmo que o objeto continue se movimentando, quando a foto é tirada, o objeto se mantém em apenas um lugar. Assim como o qubit quando é medido.

A Esfera de Bloch é a representação da tridimensionalidade de um qubit. Já as Portas quânticas são portas para manipulação de qubits, sendo as mais conhecidas Operadores de Pauli, Portas de Fase e Porta de Hadamard. O Algoritmo de Deutsch-Jozsa descreve um problema onde é necessário definir se uma função 𝑓(𝑥) é constante ou balanceada, e caso seja constante, todos resultados serão iguais a 0 ou iguais a 1.

Para finalizar temos os paradigmas de programação e os diferentes tipos de linguagem de programação, que servem para decompor o problema em uma série de funções, que recebem os inputs e produzem output, sem realizar mudança de estado internamente. Temos Haskell como a linguagem funcional mais conhecida e há linguagens multiparadigmas, em que é possível escrever programas que são procedurais, orientadas a objeto ou funcionais, como é o caso de Python, que possui uma sintaxe limpa e reputação de produtividade. Temos também a Quipper que é uma linguagem de programação funcional integrada para Computação Quântica sendo baseada em Haskell, portanto pode ser considerada como uma série de data types, combinadores e bibliotecas de funções Haskell.

## Aplicação de protocolos quânticos e algoritmo de Shor para a Segurança da Informação

A computação quântica é uma área de estudo multidisciplinar, a qual envolve as áreas de computação, física e matemática, e tem se tornado um elemento de estudo de grande destaque, por sua promessa de evolução, inovação e melhorias. As descobertas nas áreas “quânticas” permitem que as “clássicas” continuem se desenvolvendo, tendo em vista que um problema do passado pode ser resolvido através de um novo ponto de vista proporcionado pelos estudos em relação à computação quântica, que antes não se era conhecido.

Nielsen e Chuang (2010) apontam que os “computadores quânticos vieram para resolver problemas que são impossíveis de serem resolvidos em computadores clássicos, não porque sejam insolúveis, mas sim pela grande quantidade de recursos necessários para a sua solução”.

A criptografia clássica, RSA, é baseada na função de Euler, que envolve a fatoração de grandes números primos. Essa fatoração, com a tecnologia clássica pode levar a centenas de anos para ser realizada, e a computação quântica promete solucionar esse problema rapidamente, através de algoritmos, por exemplo o algoritmo de Shor, onde as chaves de criptografia clássica podem ser descriptografadas em uma função de tempo muito menor através da descoberta do resultado da fatoração dos números primos.

Essa descoberta é um risco para segurança atual, pois ao considerarmos que todos os métodos de segurança são baseados na fatoração de grandes números primos por conta da sua complexidade de resolução na computação clássica, com a chegada da computação quântica essa “garantia” de segurança que tínhamos é perdida, e estamos todos sujeitos à diversos tipos de ataques, de modo muito mais rápido.

Para resolver essa deficiência na área de segurança, surge a criptografia quântica, que usa como princípios a Mecânica Quântica e o Princípio da Incerteza de Heisenberg.

A Criptografia Quântica se dá através da Distribuição de Chaves Quânticas (DCQ), que é um protocolo que permite que os bits da chave privada possam ser criados através de um canal público, entre dois parceiros. A segurança da chave que é gerada é garantida pelas propriedades da informação quântica e condicionada às leis da Física. O que impossibilita que a comunicação entre os dois parceiros seja interceptada sem que o espião seja descoberto, ou que o estado entre eles seja perturbado, e mesmo que o invasor consiga obter algum sucesso em alguma das etapas, as informações não podem ser clonadas, e isso é assegurado pelo Teorema de Não-Clonagem. O espião não consegue interceptar a comunicação sem ser percebido, e ao ser detectado (a taxa de ruído no canal seja maior que o limiar estabelecido), ambos os pontos irão alterar o protocolo no qual estão transmitindo.

Os protocolos quânticos de distribuição mais conhecidos são o BB84 e o B92. O BB84 foi o primeiro protocolo de distribuição a ser desenvolvido, sendo criado no ano de 1984, e se aproveita do fato de que é impossível medir uma informação quântica sem interferir em seu estado, impedindo que a comunicação seja interceptada por quem não faz parte da comunicação originalmente. O protocolo utiliza fótons polarizados para a transmissão das chaves, e esses podem ser medidos em três bases: vertical ou horizontal, circular (à esquerda ou à direita) e diagonal. O comunicador A irá escolher aleatoriamente em qual base irá modular o seu fóton: horizontal, vertical, circular à esquerda ou circular à direita, e o comunicador B, escolhe aleatoriamente qual polarização irá usar para medir o fóton que foi recebido, sem ter o conhecimento da base em que foi modulado. Se a base que B escolheu foi a mesma de A, os bits são lidos corretamente, se não, existe 50% de acerto de B. A e B fazem essa comunicação através de um canal público. B irá enviar a sequência que ele utilizou para A, que irá conferir quais medidas estão corretas. É a partir disso que é verificado se há interferência externa em uma comunicação, através de valores incoerentes (polarização correta e valor incorreto), caso não haja, a chave é segura. Se houver incoerências, os resultados são descartados e inicia-se uma nova tentativa de comunicação.

O protocolo B92, é baseado no protocolo BB84, porém utiliza outros estados e bases não ortogonais. Nele, serão utilizados apenas dois estados quânticos, ao invés de 4 como o protocolo anterior. Então, nesse protocolo haverá apenas duas possibilidades para os bits que são recebidos, e dado a sua simplicidade, os bits são considerados individualmente. B irá receber um valor em que o seu resultado poderá ser zero ou um, dependendo da base em que foi gerado. A e B geram duas sequências de bit aleatórios, e o que determina quando será zero e quando será um, depende da base que foi enviada: vertical ou horizontal. Por exemplo, se o resultado for zero, A envia vertical e quando for 1 envia diagonal. E B irá tentar detectar zero para diagonal e 1 para horizontal. Então, B irá verificar quais as posições que o bit foi detectado.

A Transformada de Fourier Quântica, nada mais é do que uma forma de resolver problemas complexos usando operações com resoluções conhecidas. Fourier descobriu que todo sinal poderia ser descrito como a soma de outros sinais que possuem frequência. Existem dois tipos, a Transformada Rápida, que é usada em computadores clássicos, e a Transformada de Fourier Quântica usada em computadores quânticos. A última (quântica), irá alterar as probabilidades de medir certo estado quântico. Ela serve de base para construção de outros algoritmos quânticos.

Em relação ao Algoritmo de Fatoração de Shor, ele é de grande importância para a computação quântica, pois resolve o problema de fatorar grandes números, problema estudado há muito tempo na matemática, tanto que esse método foi utilizado para o sistema de criptografia clássica, no sistema RSA. Sendo implementado na criptografia quântica, o algoritmo é capaz de resolver a fatoração de grandes números em segundos. E essa capacidade causa problemas aos sistemas de criptografia clássica, pois quebra chaves que foram construídas para levar anos para serem quebradas, em segundos. “O objetivo do algoritmo é achar o período de uma função, e na sequência, encontrar os fatores do valor solicitado.”

A criptografia clássica foi desenvolvida através de protocolos que visavam a dificuldade de fatoração de grandes números primos, o que inclusive era tema de estudo de matemáticos. A resolução ou a quebra das chaves geradas nesse princípio, na computação clássica, levaria centenas de anos para ser descoberta. Isso por conta da velocidade de processamento e da incapacidade da realização de vários testes ao mesmo tempo. Já a criptografia quântica possibilita a quebra dessa mesma chave em alguns segundos, o que se torna um grande problema na área de segurança de dados. Ou seja, a computação quântica surgiu através da necessidade de resolver problemas que são impossíveis de serem solucionados na computação clássica, e a criptografia quântica surgiu através da necessidade de segurança que são impossíveis de serem solucionados através da criptografia clássica.

## Desempenho de algoritmos quânticos e clássicos em treinamento de Machine Learning supervisionado

O aumento na quantidade de dados está fazendo com que algoritmos de aprendizagem de máquina cheguem ao seu limite computacional, e é a partir desse limite que a computação quântica se mostra como uma possível solução para a ineficiência de processos realizados em um computador clássico.

Levemos isso em consideração: o melhor algoritmo de ML é, basicamente, aquele que melhor se ajusta aos dados e possui o menor tempo de execução (aprendizagem) e sabemos que dentro do Machine Learning, os dados de entrada são processados e organizados, para que como resultado seja gerado o que chamamos de previsão. O aumento do objeto processado (os dados) nos leva, portanto, ao problema de ineficiência dos algoritmos de ML Clássico devido ao tempo de execução muito alto.

Como solução, surge o Machine Learning Quântico. O QML possui a vantagem da aceleração quântica, que é definida pela complexidade de consulta, que dá nome à quantidade de vezes que o algoritmo consulta a fonte de informação, e pela complexidade de porta, que é a quantidade de operações elementares necessárias para executar o programa. Portanto, para saber se um algoritmo quântico é eficiente, é necessário tomar como base esses dois parâmetros, além de levar em conta os vários tipos de algoritmos de Machine Learning.

Contudo, a migração para o QML tende a ser longa devido aos problemas de entrada e saída de dados. Praticamente todos os dados atuais não estão em estado quântico, então para serem processados é preciso transformá-los. O uso da QRAM tem um custo muito alto e a transformação dos dados é extremamente demorada. Além dos problemas de entrada e saída, temos também o desafio da correção de erros quânticos, pois é preciso fazer ajustes nas portas para que a interferência seja minimizada e a eficiência otimizada.

Uma possível maneira de contornar o problema da transformação de dados, é reduzindo seu tamanho por meio da Análise de Componentes Principais, um método usado para reduzir a massa de dados sem que nenhuma informação seja perdida. A Análise transforma um conjunto de variáveis em outro conjunto de variáveis de mesma dimensão - são esses os componentes principais, e cada componente é uma combinação linear de todas as variáveis originais e mantém o máximo de informação possível.

Os algoritmos quânticos utilizados pelo ML são baseados nas técnicas de estimativa de fase, estimativa de amplitude e simulação hamiltoniana, e podem ser divididos em duas classes: a primeira classe de algoritmos é baseada na transformada quântica de Fourier de Shor e a segunda no algoritmo de Grover, que realiza a pesquisa quântica.

A Transformada de Fourier é essencial para o processamento de dados e é usada na maioria dos algoritmos contribui com a sub-rotina de estimativa de fase, que é usada na maior parte dos algoritmos quânticos pois permite encontrar autovalores de operações unitárias. O algoritmo de Grover é capaz de encontrar e ordenar um conjunto de dados cuja estrutura e organização são desconhecidos.

# METODOLOGIA

O desenvolvimento do projeto percorrerá os seguintes caminhos:

* Pesquisas bibliográficas em livros e nos meios de publicações técnicas e científicas voltadas para conceitos e definições de Computação Quântica, Machine Learning, Banco de Dados e Computação em Nuvem;
* Pesquisas bibliográficas em livros e nos meios de publicações técnicas e científicas voltadas para a criação de algoritmos quânticos para serem rodados em computadores clássicos e suas respectivas versões para computadores quânticos;
* Definição das tecnologias clássicas e quânticas a serem utilizadas no desenvolvimento do projeto;
* Criação e validação de algoritmos de criptografia aplicado em banco de dados para computadores clássicos e suas respectivas versões quânticas;
* Implementação dos algoritmos gerados para verificação das aplicabilidades em computadores clássicos e quânticos;
* Divulgação do material desenvolvido para comunidade local e científica.

# RESULTADOS

## Implementação de Q# e Python

Como parte do objetivo do projeto, a fim de demonstrar a aplicação de algoritmos e linguagens quânticas em códigos construídos majoritariamente nos métodos de computação clássica, foram realizados testes de implementação de Q# juntamente com Python. Nos tópicos a seguir, serão detalhadas as principais etapas no desenvolvimento desses códigos e seu desempenho ao serem executados.

## Criação de um projeto em Q#

1. <https://learn.microsoft.com/en-us/azure/quantum/install-overview-qdk?tabs=tabid-vscode%2Ctabid-conda>

* dotnet new -i Microsoft.Quantum.ProjectTemplates
* dotnet new classlib -lang Q# -o quantum(-o significa a pasta em que o código vai ser gerado, só que para funcionar tudo precisa estar na mesma pasta)

2. <https://learn.microsoft.com/en-us/azure/quantum/how-to-python-qdk-local>

Antes de rodar os comandos abaixo, abrir o terminal do conda

* conda create -n qsharp-env -c microsoft qsharp notebook
* conda activate qsharp-env (O activate serve para entrar dentro do ambiente)

Os comandos acima servem para criar um ambiente de desenvolvimento com pacotes específicos.

**6.1.2 Sobre o Q#**

O Q# é uma linguagem específica de domínio, projetada para expressar operações quânticas de forma clara e concisa. Por isso, foi essencial a utilização da linguagem, pois é possível programar algoritmos quânticos capazes de se comunicar facilmente com outras linguagens populares, como o Python.

**6.1.3 Sobre o Python**

O Python é uma linguagem de alto nível, amplamente usada para ciência de dados, aprendizado de máquina e desenvolvimento web. Por ser uma linguagem multiparadigma, sua interação com o Q# se sucedeu de maneira muito simples, de modo que o desenvolvimento do código híbrido ocorreu sem muita complexidade.

# CRONOGRAMA

|  |  |  |
| --- | --- | --- |
| DATA | ATIVIDADE | RESPONSÁVEL |
| Fevereiro/Março | O grupo se dedicou à revisão Bibliográfica dos 3 artigos (Implementação do algoritmo quântico Deutsch-Jozsa em linguagem funcional e no simulador IBM Q Experience, Aplicação de protocolos quânticos e algoritmo de Shor para a Segurança da Informação e Desempenho de algoritmos quânticos e clássicos em treinamento de Machine Learning supervisionado) e participou de reuniões e seminários semanais com a professora. | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| 29/03 | Apresentação da 1ª versão do cronograma de atividades | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| 04/04 | Pesquisas referente às plataformas, linguagens de programação escolhidas, repositórios e bibliotecas, e a implementação de algoritmo nas linguagens escolhidas (Java e Silq) | Leonardo Ribeiro  Erick Galvão |
| 05/04 | Apresentação da 2ª versão do Cronograma e reunião de alinhamento com a professora. | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| 07/04 | Início dos testes de implementação de circuitos quânticos nas plataformas Azure e IBMQ. Documentação dos testes realizados. | Leonardo Ribeiro |
| 10/04 | Aplicação de algoritmos quânticos em Java e busca por códigos criptográficos simples | Erick Galvão  Leonardo Ribeiro |
| 11/04 | Levantamento das principais propriedades de segurança de um banco de dados e conceitos essenciais para inserção nos relatórios | Stheffani Emboava |
| 13/04 | Procura de um banco de dados experimental | Leonardo Ribeiro  Erick Galvão |
| 17/04 | Preparação de um ambiente com um banco de dados experimental e testes de integração | Leonardo Ribeiro  Erick Galvão |
| 17/04 | Complementação da documentação | Stheffani Emboava |
| 18/04 | Documentação dos testes realizados | Stheffani Emboava |
| 19/04 | Decisão de alteração da linguagem principal de Java para Python | Leonardo Ribeiro  Erick Galvão |
| 21/04 | Documentação sobre GitHub | Leonardo Ribeiro |
| 23/04 | Documentação sobre MySQL | Leonardo Ribeiro |
| 24/04 | Desenvolvimento de um código criptográfico para aplicação no banco de dados | Leonardo Ribeiro  Erick Galvão |
| 25/04 | Implementação de conexão com o banco de dados | Leonardo Ribeiro  Erick Galvão |
| 25/04 | Inserção das documentações elaboradas no relatório | Stheffani Emboava |
| 27/04 | Testes de integração de arquivos da máquina com o Workspace Quantum da Azure para aplicação de códigos criptográficos clássicos | Erick Galvão  Leonardo Ribeiro |
| 05/05 | Pesquisa e teste de criptografia em dados no MySQL. Criação do banco md5 e AES. | Leonardo Ribeiro |
| 10/05 | Aplicação e documentação dos testes do código criptográfico | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| 13/05 | Pesquisa por aplicações de banco de dados que aceitam a migração da computação clássica para a quântica | Stheffani Emboava |
| 19/05 | Pesquisa e documentação de bibliotecas a serem utilizadas no código “Frankenstein” | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| 21/05 | Verificação das pesquisas e conteúdos estudados para início da produção do escopo do artigo | Erick Galvão  Stheffani Emboava |
| 22/05 | Início do desenvolvimento do código “Frankenstein” | Leonardo Ribeiro  Erick Galvão |
| 01/06 | Elaboração do escopo do artigo, definindo principais tópicos a serem abordados | Erick Galvão  Stheffani Emboava |
| 08/06 | Busca por referências relevantes para os tópicos específicos a serem abordados no artigo | Erick Galvão  Stheffani Emboava |
| 12/06 | Correção de erros do código e revisão dos objetivos a serem alcançados com o programa | Leonardo Ribeiro  Erick Galvão |
| 15/06 | Início da produção do artigo | Erick Galvão  Stheffani Emboava |
| 20/06 | Validar funcionamento do código com testes, das mais diversas maneiras | Leonardo Ribeiro  Erick Galvão |
| 27/06 | Revisar o que foi escrito do artigo até então e verificar distoâncias entre a proposta e a execução | Stheffani Emboava |
| 30/06 | Correção de possíveis erros no código e documentação das possíveis alterações | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| Julho | Elaboração do relatório parcial | Stheffani Emboava  Erick Galvão |

## Cargos da Equipe

|  |  |  |
| --- | --- | --- |
| Gestor | Desenvolvedor | Redator |
| Erick Galvão | Leonardo Ribeiro | Stheffani Emboava |

# REFERÊNCIAS BIBLIOGRÁFICAS

AWS. O que é a computação em nuvem? AWS, 2023. Disponível em: https://aws.amazon.com/pt/what-is-cloud-computing/?nc1=f\_cc. Acesso em 19 abr 2023.

BESSA, André. Cloud: o que é, História e Guia da computação em nuvem. Alura, 2023. Disponível em: https://www.alura.com.br/artigos/cloud?gclid=CjwKCAjwov6hBhBsEiwAvrvN6OEPLFRQkoGjnzn5YJcpXaHA\_k5aGY8vhsTPKFGg7Lvz47dlWEQ4shoCNtoQAvD\_BwE. Acesso em 19 abr 2023.

BHARTI, K. (2017). Quantum Computing: Future of Computing. International Journal of Engineering Technology Science and Research, 4(7), 307-310.

CANTU, Ana, DELL. (2011) The History and Future of Cloud Computing. Forbes. Disponível em: https://www.forbes.com/sites/dell/2011/12/20/the-history-and-future-of-cloud-computing/?sh=1fa426c379d9. Acesso em 19 abr 2023.

COLLIER, Michael; SHAHAN, Robin. Fundamentals of Azure: Microsoft Azure Essentials. Editora: Microsoft Press, 2016. p. 1-31.

ELMASRI, R., NAVATHE, S. B. (2016). Fundamentals of Database Systems. Pearson. Disponível em: https://drive.google.com/file/d/1EB3CChnOrcm6TpyqbF\_9CHCw6FMI8o1V/view. Acesso em: 15 abr 2023

GARCIA-MOLINA, H.; ULLMAN, J. D; WIDOM, J. (2020). Database Systems: The Complete Book. Pearson.

GitHub Docs. Sobre versões da Documentação do GitHub. Disponível em: https://docs.github.com/pt/get-started/learning-about-github/about-versions-of-github-docs. Acesso em 23 abr 2023.

IDC, Data Creation and Replication Will Grow at a Faster Rate than Installed Storage Capacity, 2021. Business Wire. Disponível em: https://www.businesswire.com/news/home/20210324005175/en/Data-Creation-and-Replication-Will-Grow-at-a-Faster-Rate-Than-Installed-Storage-Capacity-According-to-the-IDC-Global-DataSphere-and-StorageSphere-Forecasts. Acesso em: 30 abr 2023.

LACAVA, Lucas; MIANO, Mariana V. Implementação do algoritmo quântico Deutsch-Jozsa em linguagem funcional e no simulador IBM Q Experience. 2018. Revista Tecnológica da Fatec Americana, vol. 06 n. 2, abr/set de 2018. Disponível em: https://fatec.edu.br/revista/index.php/RTecFatecAM/article/view/186. Acesso em 14 fev 2023.

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.

MIANO, Mariana V. OLIVEIRA, Aleccheevina S. Desempenho de algoritmos quânticos e clássicos em treinamento de Machine Learning supervisionado. 2022. Revista Tecnológica da Fatec Americana, v. 9 n. 02 (2021). Disponível em: https://fatec.edu.br/revista/index.php/RTecFatecAM/article/view/295. Acesso em 6 mar 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.

MOSCA, M., & Stebila, D. (2018). Post-quantum cryptography. Nature, 555 (7697), 588-589.

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.

SCHWAB, K. (2017). The Fourth Industrial Revolution. Nova York: Crown Business.