

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, utilizada atualmente, vem enfrentando grandes desafios os quais não são fisicamente possíveis de serem solucionados devido ao nível de recursos necessários para solucionar tais requerimentos. (Em desenvolvimento)

**ABSTRACT**

(Em desenvolvimento)

Pendente – ajustes da tabela de imagens e legendas

**SUMÁRIO**

[1. INTRODUÇÃO 4](#_Toc134373935)

[2. OBJETIVOS 4](#_Toc134373936)

[2.1 Objetivos Específicos 4](#_Toc134373937)

[3. JUSTIFICATIVA 5](#_Toc134373938)

[4. DESENVOLVIMENTO 5](#_Toc134373939)

[4.1 Computação Quântica 5](#_Toc134373940)

[4.2 Cloud Computing 5](#_Toc134373941)

[4.2.1 Tipos de serviço 6](#_Toc134373942)

[4.2.2 Modelos de Nuvem 7](#_Toc134373943)

[4.3 Azure 7](#_Toc134373944)

[4.3.1 Azure Quantum 8](#_Toc134373945)

[4.4 Linguagem Q# 8](#_Toc134373946)

[4.5 MySQL 8](#_Toc134373947)

[4.5.1 Origem 8](#_Toc134373948)

[4.5.2 Instalação 9](#_Toc134373949)

[4.5.3 Configuração: 9](#_Toc134373950)

[4.5.4 Extensões no VSCODE 12](#_Toc134373951)

[4.6 GitHub 15](#_Toc134373952)

[4.6.1 Projeto para planejamento de horário no GitHub 15](#_Toc134373953)

[4.7 Implementação do Algoritmo de Deutsch-Josza 18](#_Toc134373954)

[4.8 Protocolos Quânticos e Algoritmo de Shor na Segurança da Informação 20](#_Toc134373955)

[4.9 Machine Learning 22](#_Toc134373956)

[5. METODOLOGIA 23](#_Toc134373957)

[6. RESULTADOS 24](#_Toc134373958)

[7. CRONOGRAMA 24](#_Toc134373959)

[8. REFERÊNCIAS BIBLIOGRÁFICAS 27](#_Toc134373960)

# INTRODUÇÃO

(Em desenvolvimento)

# 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

- Pesquisar plataformas, softwares, bibliotecas e extensões que possibilitam a programação quântica em computadores clássicos;

- Pesquisar plataformas que suportem tarefas computacionais clássicas e quânticas;

- Pesquisar softwares híbridos e tarefas com possibilidades de processamento híbrido;

- Pesquisar linguagens computacionais que possibilitem a migração de algoritmos clássicos para algoritmos quânticos;

- Elencar as possíveis aplicabilidades da plataforma Microsoft Azure e das linguagens de programação Python e Q#, na computação quântica;

- Elencar a usabilidade da computação quântica em um banco de dados clássico;

- Elencar e definir conceitos relacionados à Computação Quântica e Banco de dados;

- Construir um código "Frankeinstein", utilizando as linguagens Q#, Python e MySQL, implementando recursos quânticos dentro da programação clássica, demonstrando dessa maneira a usabilidade das soluções quânticas em processos de manipulação de um banco de dados.

# JUSTIFICATIVA

(Em desenvolvimento)

# DESENVOLVIMENTO

Em 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, foi obtido os conceitos de Computação Quântica, Computação em Nuvem, Banco de Dados, Microsof Azure e Azure Quantum, Linguagens de Programação Python, Q#, MySQL, a plataformas VSCode e o repositório GitHub.

## Computação Quântica

Tecnologia emergente revolucionária, que se destaca por seu processamento de dados baseado em princípios da mecânica quântica, com capacidade de processamento exponencialmente maior quando comparado à computação clássica.

## Cloud Computing

Através das evoluções tecnológicas, a IoT, entre outros, o número de dados gerados por cada indivíduo tem crescido exponencialmente. 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.

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.

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.

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.

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.

## 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.

- 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.

- 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.

## 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.

- 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.

- 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.

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.

## Azure

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 altoo desempenho em cada uma de suas funcionalidades e aprimorar a segurança dos dados confiados à plataforma.

## 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.

## Linguagem Q#

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.

## 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.

## Origem

O MySQL 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

Para iniciar, abra seu navegador e acesse o link no navegador: <https://dev.mysql.com/downloads/mysql/>, para fazer a instalação do MySQL Server:

**Passo 1:**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Passo 2:**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## Configuração

Todas as configurações são padrões, menos essas;

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Insira a senha de sua preferência:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Faça o teste de conexão do banco:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## Extensões no VSCODE

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 da extensão:**

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

**Passo 1:**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Passo 2:**

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

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

**Passo 3:**

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

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

**Extensão configurada:**

Texto

Descrição gerada automaticamente

**Testes:**

Banco de dados criado:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Tabela criada:

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

Descrição gerada automaticamente

Inserindo valores na tabela:



Fazendo pesquisa no banco;

Tela de celular

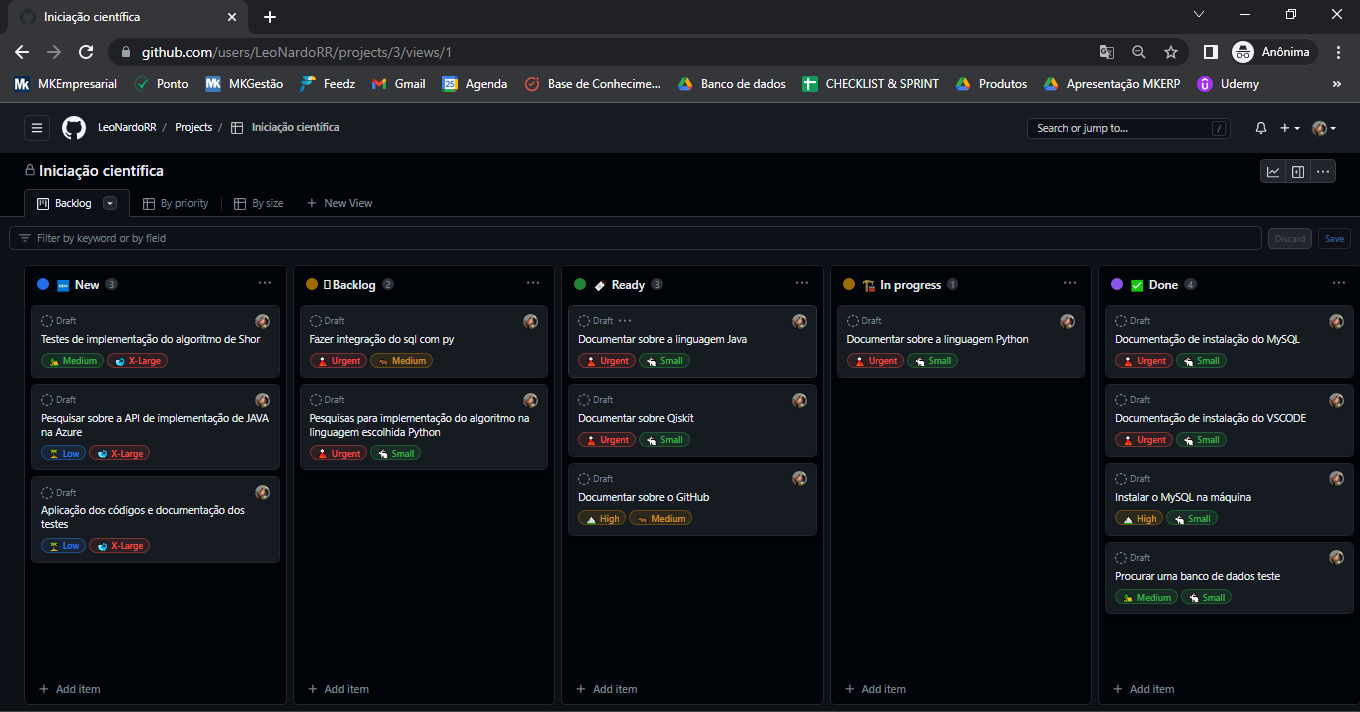
Descrição gerada automaticamente

## GitHub

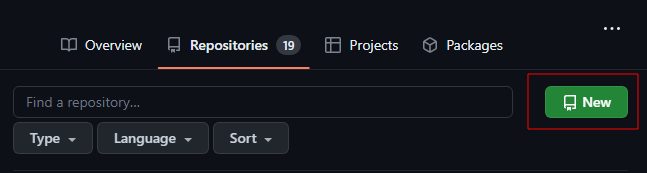
GitHub é uma plataforma de hospedagem de código para controle de versão e colaboração. Permite que você e outras pessoas trabalhem em conjunto em projetos de qualquer lugar.

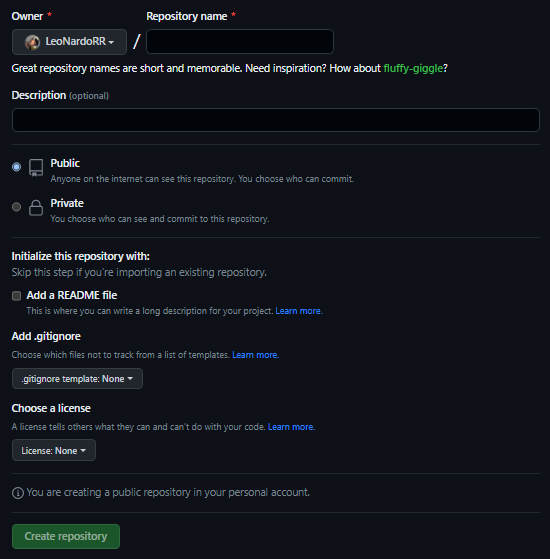
Este tutorial ensina os princípios básicos de GitHub como, por exemplo, repositórios, branches, commits e pull requests. Você criará seu próprio repositório Hello World e aprenderá o fluxo de trabalho de pull request de GitHub, uma maneira popular de criar e revisar o código.

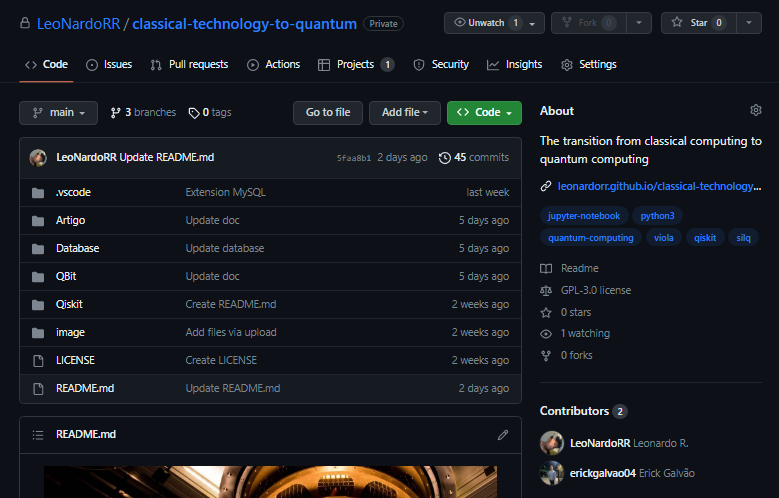
## Projeto para planejamento de horário no GitHub



**Hospedando código:**







## Implementação do Algoritmo de Deutsch-Josza

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.

## Protocolos Quânticos e Algoritmo de Shor na 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 decriptografadas 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.

## Machine Learning

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, científica e open-source.

# RESULTADOS

# 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 |
| 01/05 | Elaboração do escopo do artigo, definindo principais tópicos a serem abordados | Erick Galvão  Stheffani Emboava |
| 05/05 | Pesquisa e teste de criptografia em dados no MySQL | 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 |
| 20/05 | Documentação e estudo da linguagem Silq para implementação nos códigos funcionais em Python | Erick Galvão  Leonardo Ribeiro  Stheffani Emboava |
| 27/05 | Verificação das pesquisas e conteúdos estudados, buscando erros no que foi produzido do artigo até o momento | Erick Galvão  Stheffani Emboava |
| 31/05 | Implementação de Silq dentro dos códigos implementados anteriormente | Leonardo Ribeiro  Erick Galvão |

# REFERÊNCIAS BIBLIOGRÁFICAS

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.

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

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.

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

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

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

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.

CANTU, Ana. **A História e o Futuro da Computação em Nuvem**. DELL. Disponível em: <https://www.dell.com/learn/br/pt/brbsdt1/sb360/social_cloud#:~:text=Em%201997%2C%20o%20termo%20%E2%80%9Ccomputa%C3%A7%C3%A3o,sistemas%20de%20informa%C3%A7%C3%A3o%2C%20Ramnath%20Chellappa>. 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.

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.

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

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