

KICK-OFF

API PRIMEIRO SEMESTRE BANCO DE DADOS

2025-2

Professor M2: Leônidas

Professor P2: Sabha



Introdução ao Projeto

Título do Projeto

Desenvolvimento
de Aplicação de
Cálculo de
Sequências
Lógicas





Objetivos do projeto

Melhoria nas Habilidades de Lógica

Promover o desenvolvimento das habilidades lógicas dos participantes para melhorar a resolução de problemas na programação.

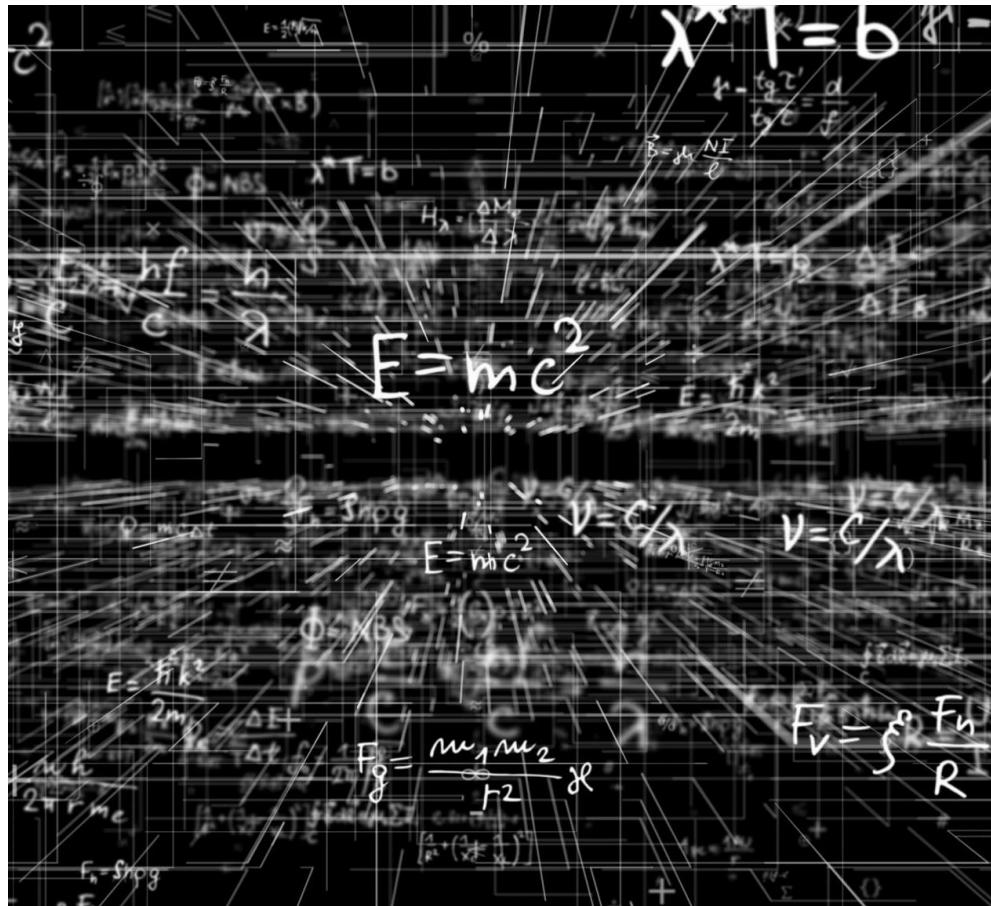
Entendimento de Padrões Numéricos

Aprofundar o entendimento de padrões em sequências numéricas, essencial para a programação e a análise de dados.

Aplicação em Problemas Reais

Aplicar conceitos aprendidos em problemas de programação do mundo real para facilitar a aprendizagem prática.

Importância do estudo de sequências numéricas



Desenvolvimento do Raciocínio Lógico

O estudo de sequências numéricas é essencial para aprimorar o raciocínio lógico e analítico, habilidades fundamentais em diversas áreas.

Aplicações em Matemática

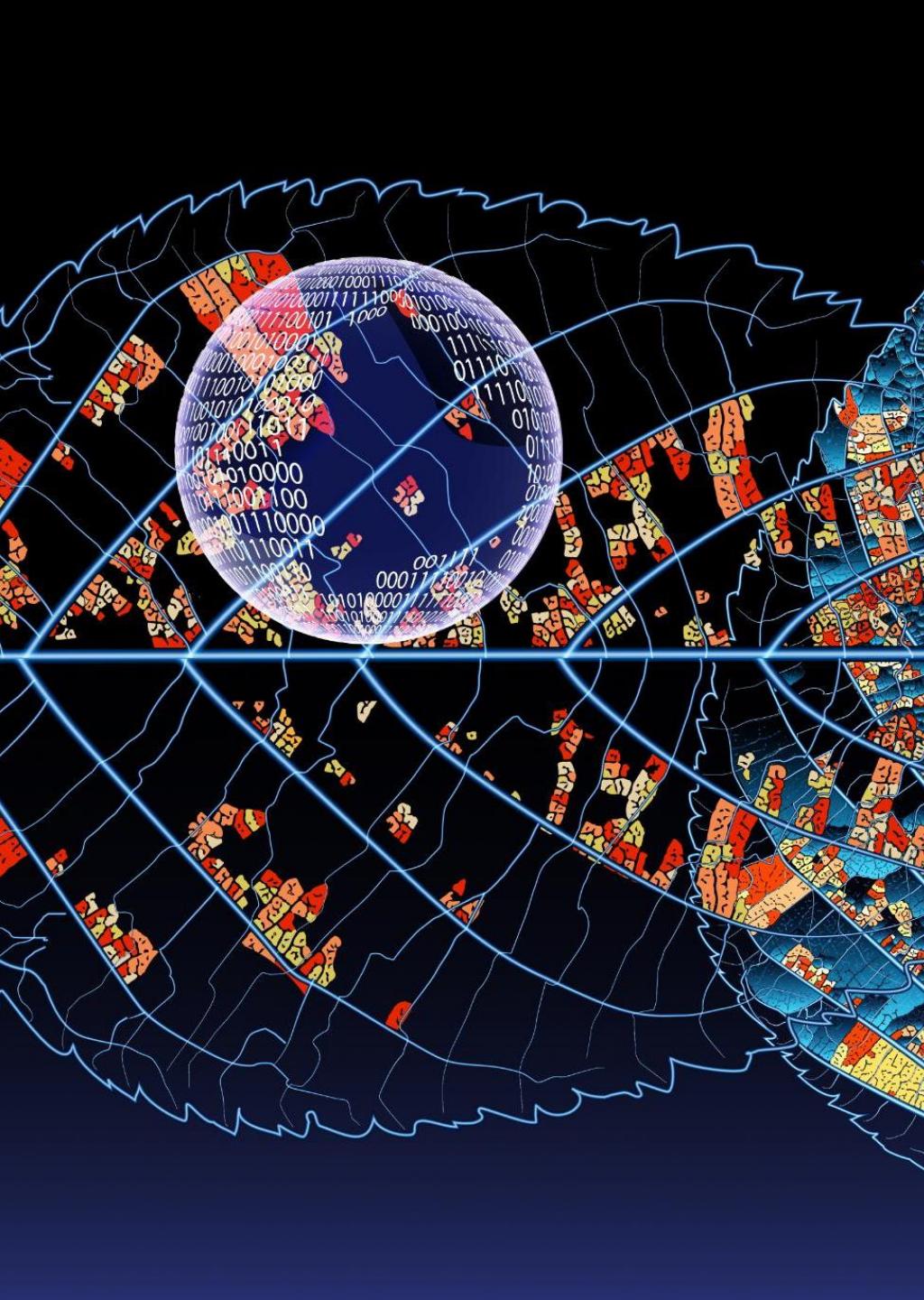
Sequências numéricas são amplamente utilizadas na matemática para resolver problemas complexos e entender padrões numéricos.

Relevância na Ciência da Computação

Na ciência da computação, sequências numéricas são fundamentais para algoritmos, estruturas de dados e modelagem de fenômenos.

Requisitos Funcionais

Sequência de Fibonacci



Padrão da sequência

Início da sequência

A sequência de Fibonacci começa com os números 0 e 1, estabelecendo a base para o resto da sequência.

Padrão crescente

A sequência avança como 0, 1, 1, 2, 3, 5, 8, 13, mostrando um padrão de crescimento específico.

Aplicações diversas

A sequência de Fibonacci é fundamental em biologia, computação e outras áreas, demonstrando sua importância em diferentes campos.

Exemplo e cálculo do n-ésimo termo

Cálculo Recursivo

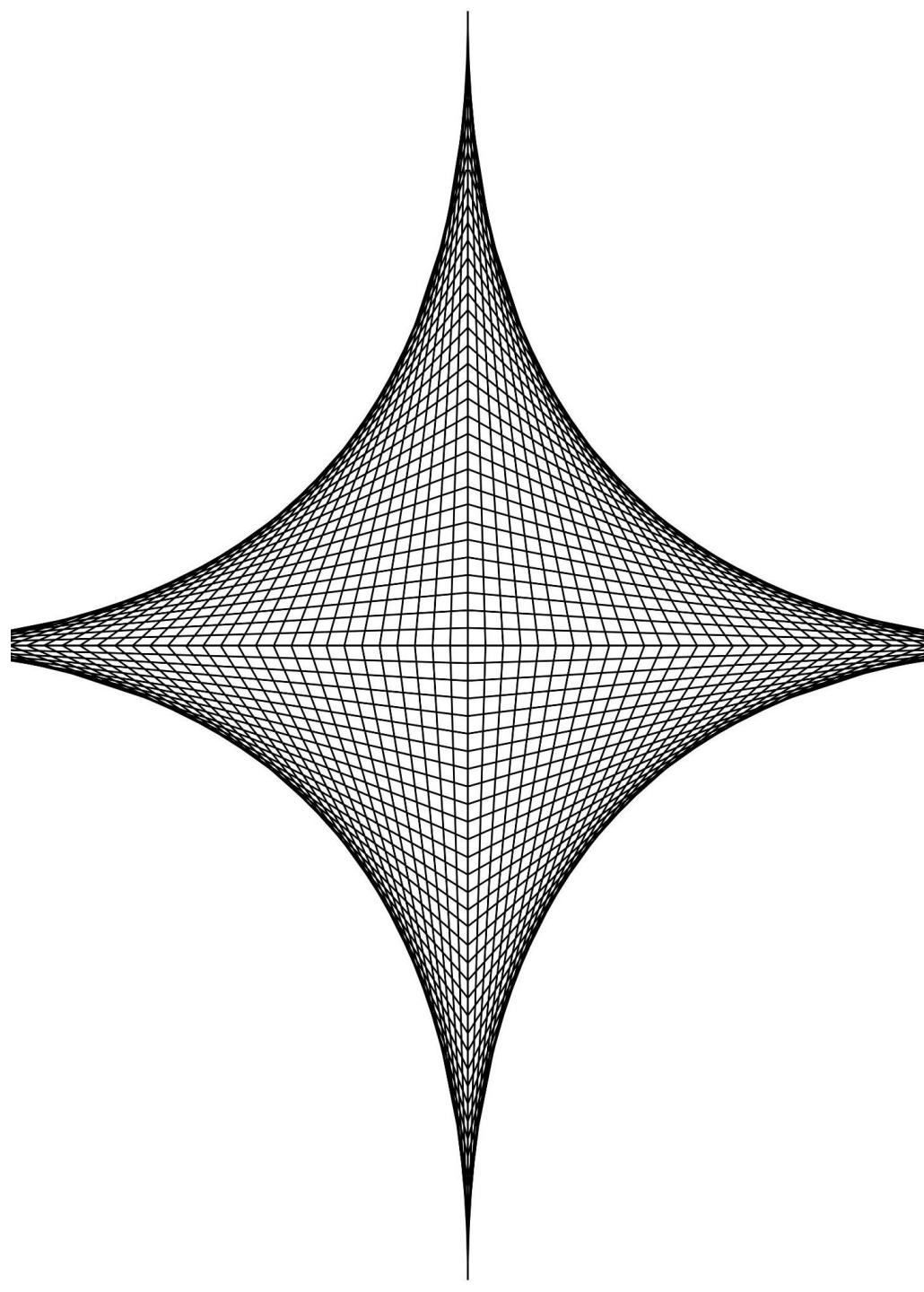
O cálculo do n-ésimo termo da sequência de Fibonacci pode ser realizado usando uma função recursiva, que chama a si mesma até atingir o caso base.

Cálculo Iterativo

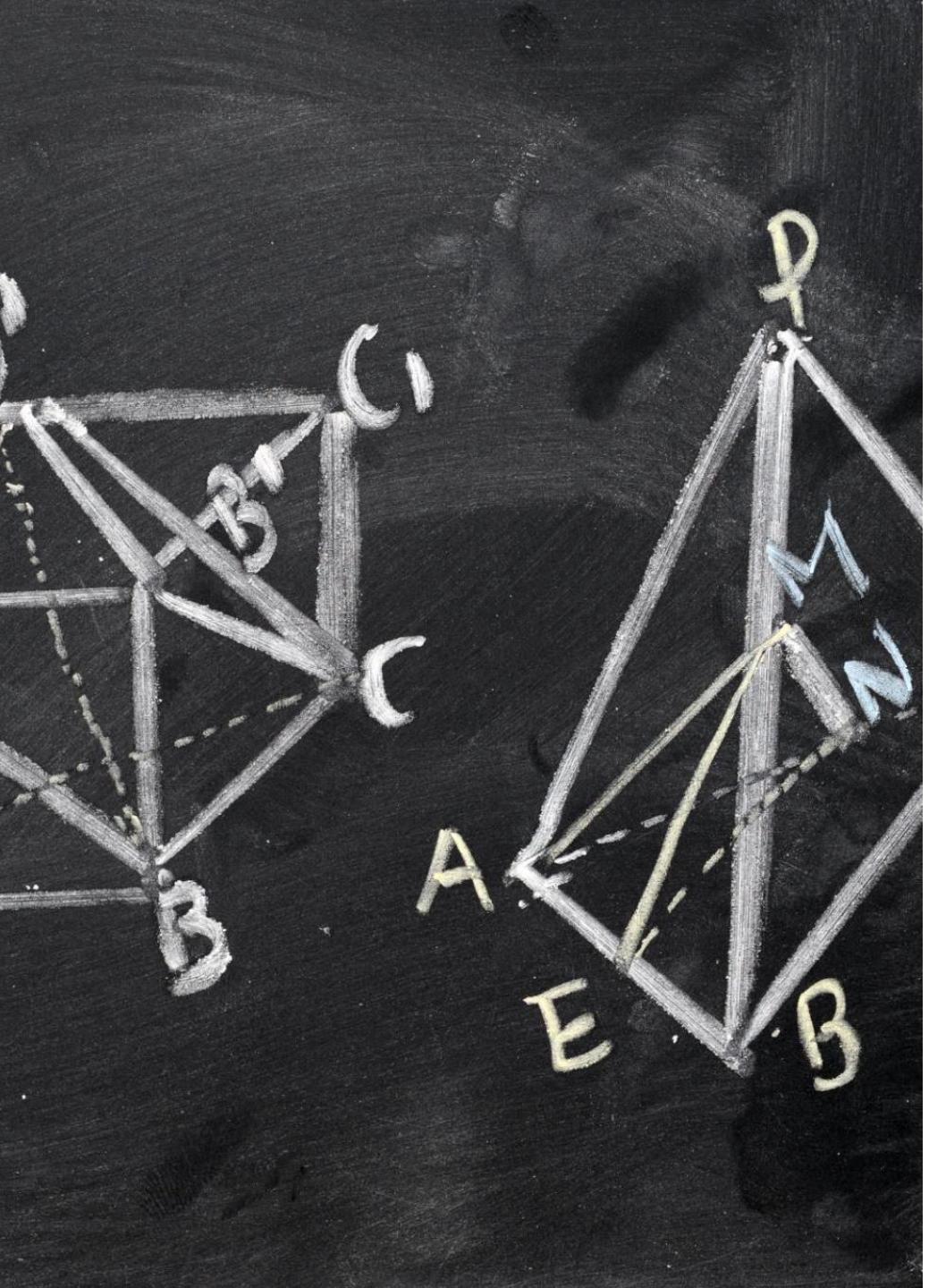
Outra abordagem é o cálculo iterativo, que utiliza um loop para calcular cada termo, resultando em maior eficiência em termos de tempo de execução.

Complexidade do Algoritmo

A complexidade do algoritmo recursivo é exponencial, enquanto a abordagem iterativa possui complexidade linear, o que a torna mais eficiente.



Números Triangulares



Padrão da sequência

Definição de Números Triangulares

Os números triangulares são representações geométricas que formam triângulos e são fundamentais na matemática.

Fórmula para Cálculo

A fórmula $T(n) = n(n + 1)/2$ é utilizada para calcular os números triangulares, começando a sequência com 1, 3, 6, 10, 15, entre outros, assim por diante.

Exemplos da Sequência

A sequência de números triangulares inclui 1, 3, 6, 10, 15, entre outros, demonstrando seu crescimento.

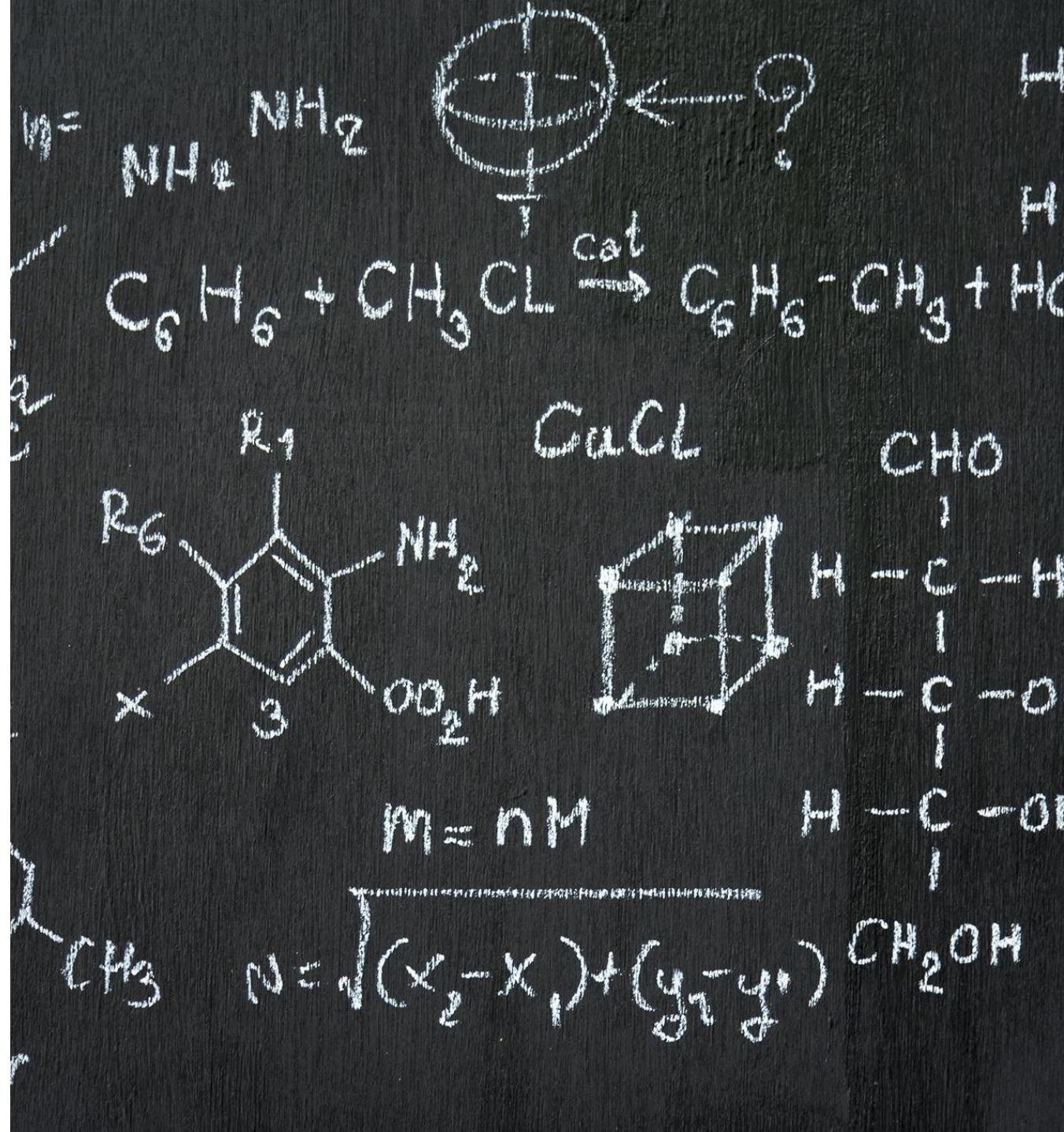
Exemplo e cálculo do número triangular na posição N

Fórmula do Número Triangular

A fórmula para calcular o número triangular na posição N é $N^*(N+1)/2$, que permite determinar rapidamente seu valor.

Implementação em Programa

Vamos analisar um programa de exemplo que utiliza a fórmula para calcular o número triangular na posição N usando uma linguagem de programação.



Sequência de Números Primos

Definição e identificação de números primos

Definição de Números Primos

Um número primo é definido como um número natural maior que 1 que não pode ser formado pela multiplicação de dois menores.

Algoritmos de Identificação

Existem algoritmos eficientes, como o Crivo de Eratóstenes, que ajudam na identificação de números primos em um intervalo.



Exemplo e listagem dos primeiros N primos

Implementação do Código

Vamos criar um código que gera e lista os primeiros N números primos, utilizando um algoritmo eficiente.

Eficácia do Algoritmo

Analisaremos a eficácia do algoritmo utilizado para listar números primos, focando na sua precisão e eficiência.

Complexidade de Tempo

Exploraremos a complexidade de tempo do algoritmo e como isso afeta o desempenho na geração de números primos.

Sequência Fatorial



Padrão da sequência

Definição do Fatorial

O fatorial de um número N é o produto de todos os inteiros positivos até N, representado como N!.

Sequência Fatorial

A sequência de fatoriais começa com 1, 2, 6, 24, 120, demonstrando o crescimento exponencial dos valores.

Exemplo de Cálculo

Por exemplo, $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$, ilustrando como calcular o fatorial de um número.

Exemplo e cálculo do fatorial de um número



Cálculo do Fatorial

O fatorial de um número é o produto de todos os inteiros positivos até esse número, essencial em matemática e programação.

Abordagem Iterativa

A abordagem iterativa usa um loop para calcular o fatorial, sendo mais eficiente em termos de uso de memória.

Abordagem Recursiva

A abordagem recursiva chama a função dentro de si mesma, proporcionando um método mais simples, mas com maior uso de memória.

Implicações de Desempenho

Discutir as diferenças de desempenho entre as abordagens iterativa e recursiva é crucial para entender a eficiência da programação.

Sequência de Quadrados Perfeitos

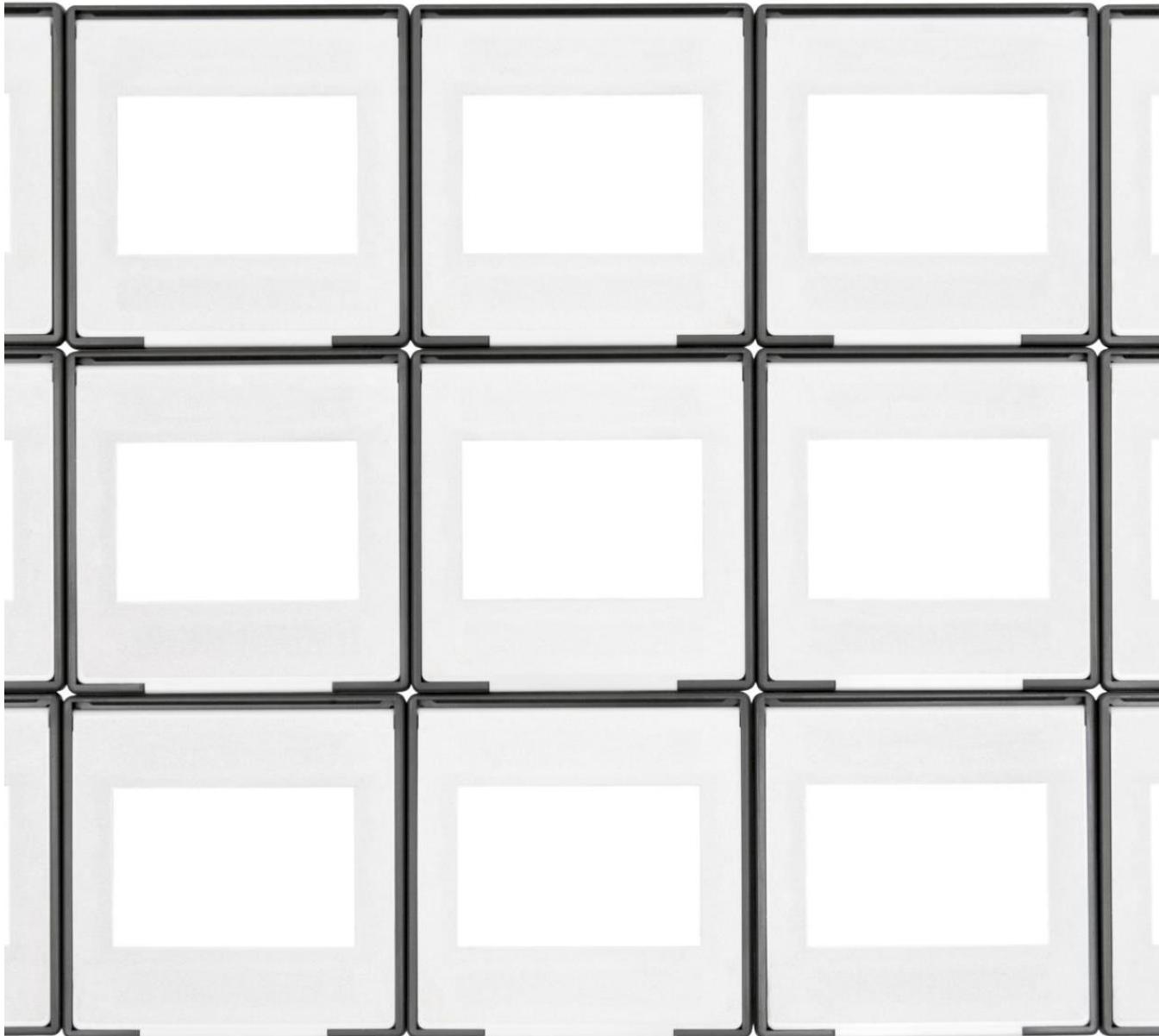
Padrão da sequência

Sequência de Quadrados Perfeitos

A sequência é formada por números que são quadrados de inteiros, como 1, 4, 9, e assim por diante.

Geração da Sequência

Cada termo da sequência é obtido elevando o índice ao quadrado, demonstrando um padrão matemático claro.



Sequência de Cubos

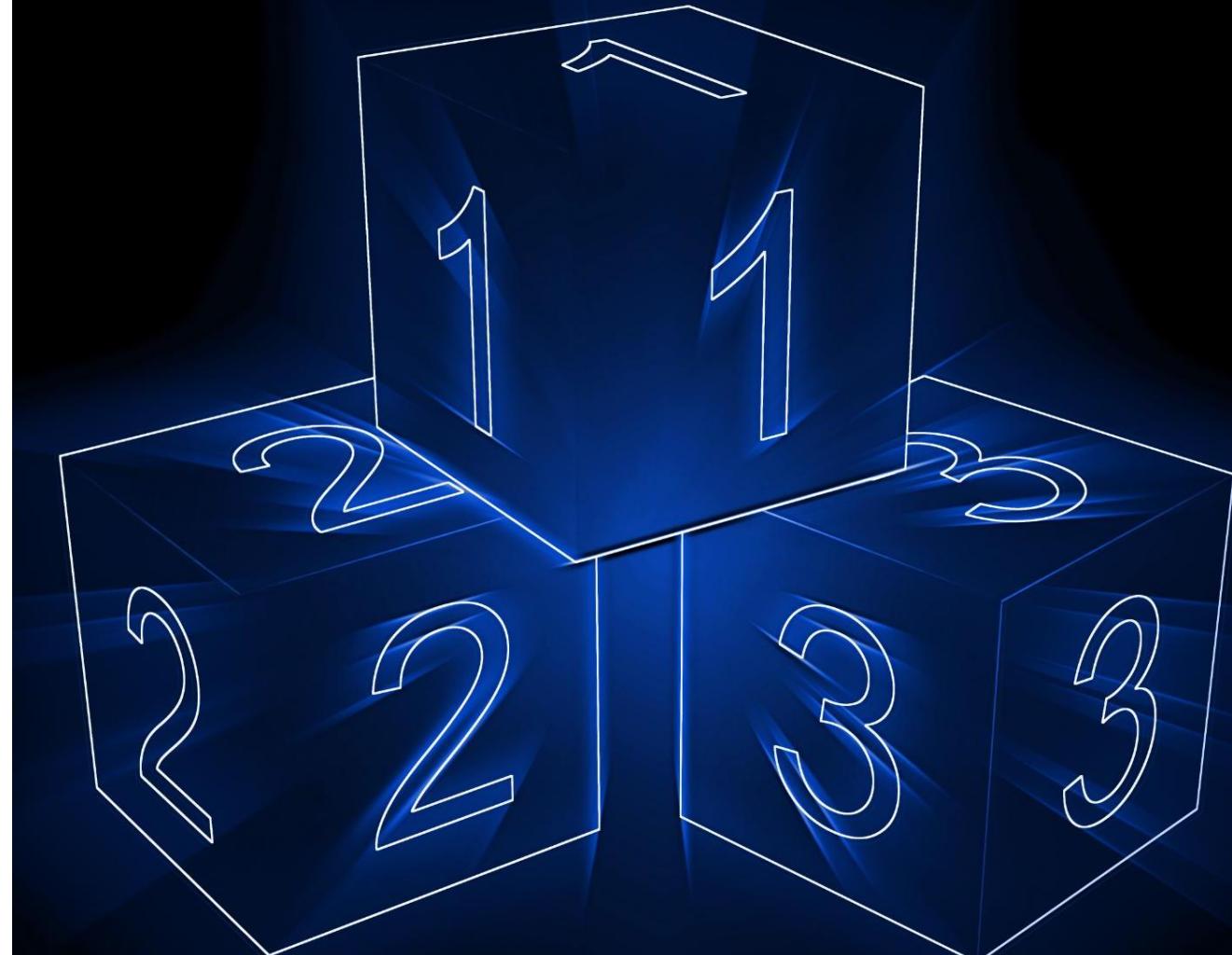
Padrão da sequência

Sequência de Cubos

A sequência de cubos é uma série de números que resulta da elevação de cada índice ao cubo, começando por 1.

Cálculo dos Cubos

Cada número na sequência é calculado elevando o índice correspondente ao cubo, demonstrando a relação entre números e suas potências.



Sequência Geométrica

Padrão da sequência

Fórmula da Sequência Geométrica

A fórmula $a = a_1 * r^{(n-1)}$ é essencial para entender como as sequências geométricas são calculadas e utilizadas.

Primeiro Termo e Razão

O primeiro termo (a_1) e a razão (r) são fundamentais para definir uma sequência geométrica específica e suas propriedades.

Exemplos Práticos

Vamos explorar exemplos práticos para entender melhor como aplicar a sequência geométrica em diferentes contextos.

Sequência Alternada

Padrão da sequência

Definição de Sequências Alternadas

Sequências alternadas são formadas por números que mudam seguindo um padrão específico de adição e subtração.

Padrões Diversos

Essas sequências podem seguir diferentes padrões, permitindo a exploração de diversas aplicações matemáticas e práticas.

Aplicações Práticas

Sequências alternadas têm aplicações em áreas como finanças, ciências e computação, aumentando sua relevância prática.



Sequência Tribonacci

Padrão da sequência

Início da Sequência

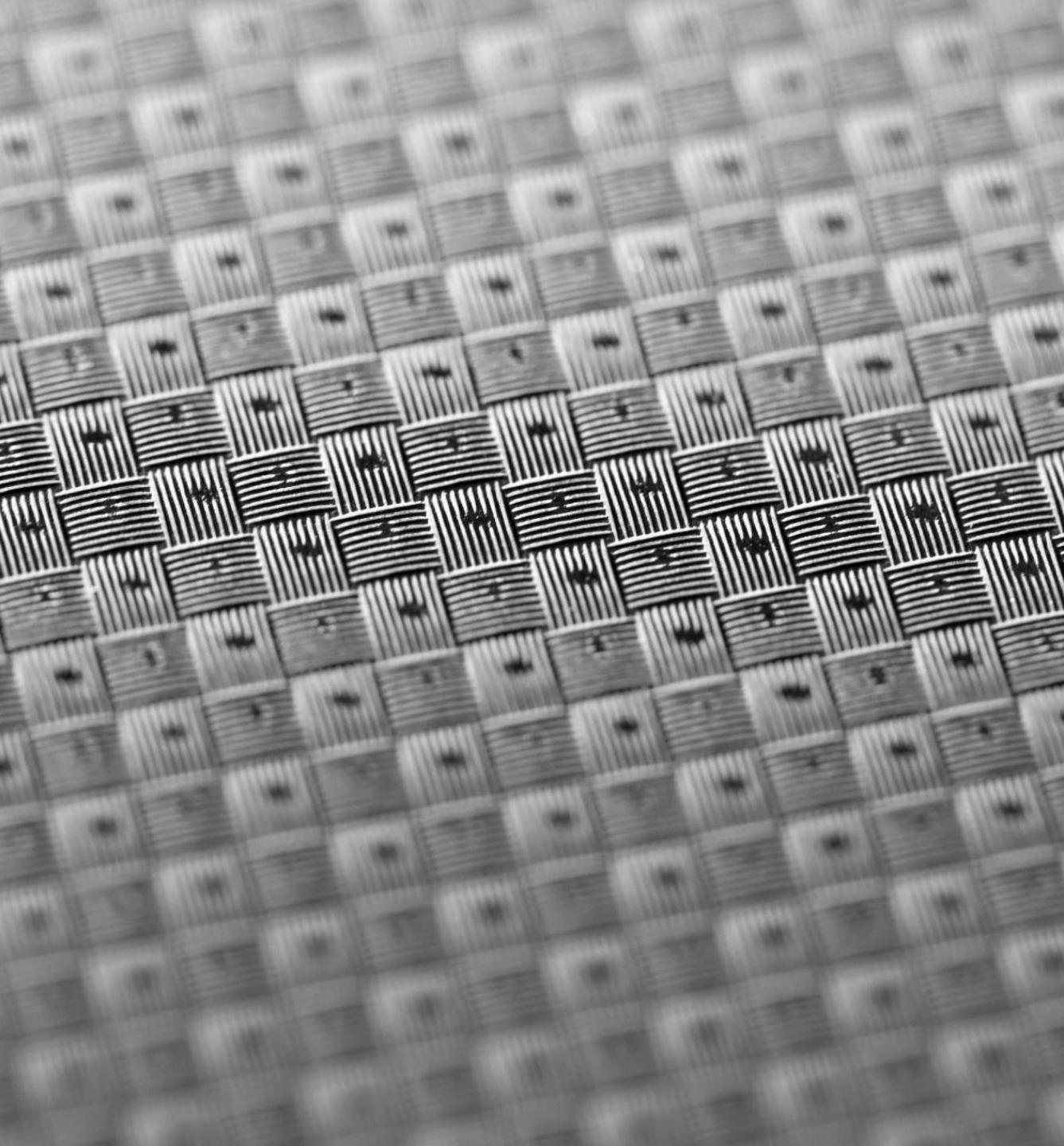
A sequência de Tribonacci começa com os números 0, 0 e 1, estabelecendo a base para os próximos termos.

Geração dos Próximos Termos

Os próximos termos da sequência são gerados pela soma dos três números anteriores, criando um padrão único.

Exemplos de Termos

A sequência continua como 0, 0, 1, 1, 2, 4, 7, 12, mostrando a progressão dos números.



Exemplo e cálculo dos termos

O que é a Sequência de Tribonacci?

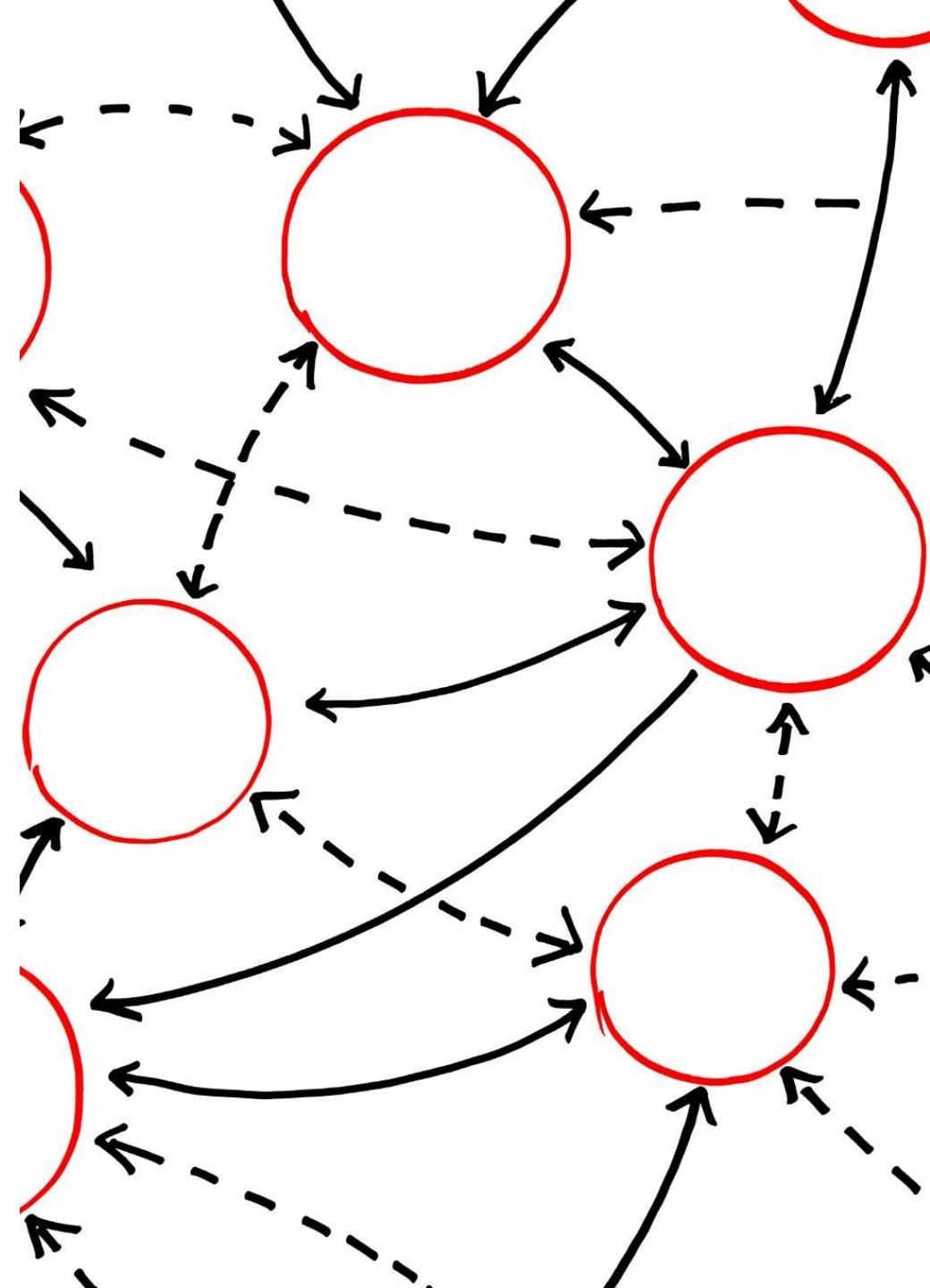
A sequência de Tribonacci é uma série numérica onde cada termo é a soma dos três termos anteriores. É uma generalização da sequência de Fibonacci.

Algoritmos Recursivos

Os algoritmos recursivos calculam os termos da sequência chamando a si mesmos, o que pode ser mais intuitivo, mas menos eficiente para grandes valores.

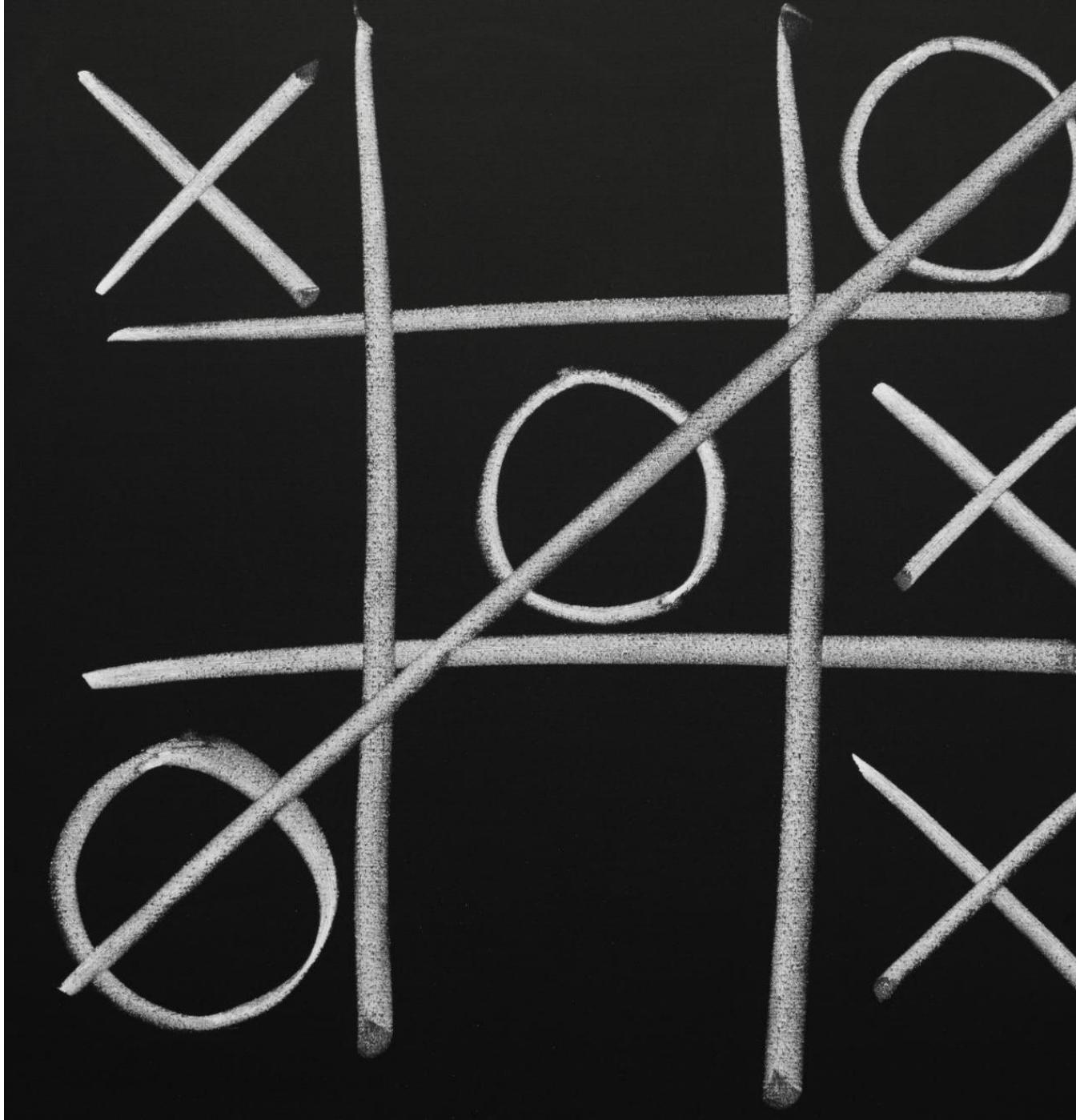
Algoritmos Iterativos

Os algoritmos iterativos utilizam loops para calcular os termos da sequência de forma mais eficiente, especialmente para grandes números.



Mais Requisitos Funcionais

- Deve possuir um interface TEXTO
- Deve apresentar ao usuário um Menu
- Deve permitir ao usuário a execução de apenas 1 cálculo de sequência por vez, porém deve permitir que ao finalizar 1 cálculo o usuário continue solicitando novos cálculos, até indicar que deseja encerrar a aplicação



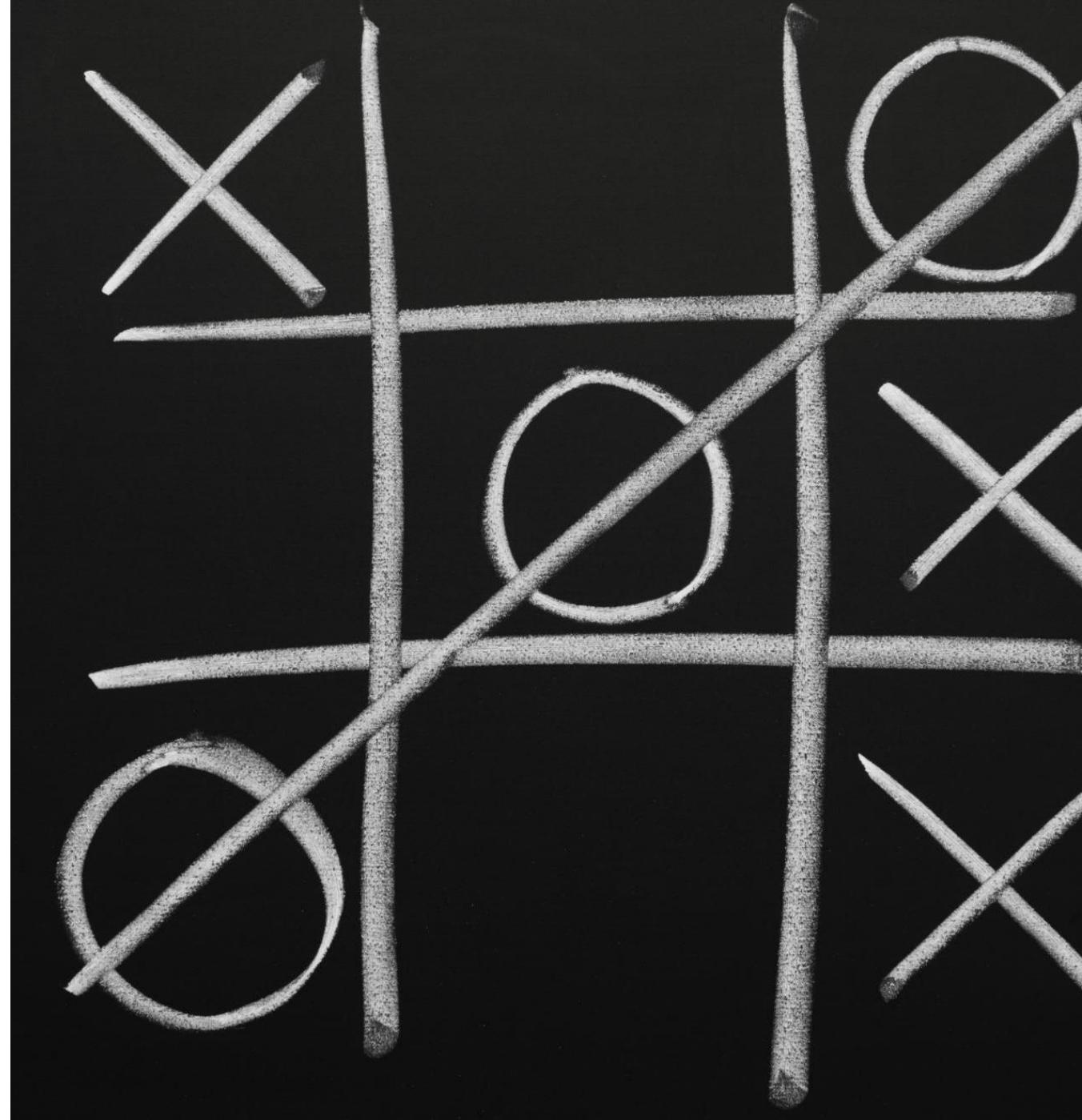
Requisitos Não Funcionais & Limitação

Requisitos Não-Funcionais

Obrigatório uso do VisualG

Documentação – Manual do Usuário

Uso do GIT





LIMITAÇÃO

**Não é permitido o uso nenhum tipo de função
“pronta”.**

Toda a lógica deve ser desenvolvida pela equipe.

Conclusão

Lógica de Programação

A lógica de programação é essencial para entender como os programas funcionam e como resolver problemas de maneira eficaz.

Importância da Prática

A prática contínua é fundamental para adquirir proficiência em programação e aplicar a lógica em projetos reais.

Exploração e Aprendizado

Sinta-se motivado a explorar mais sobre programação e a experimentar diferentes sequências e algoritmos em seus projetos.

PERGUNTAS ?

