Resumo - Pensamento Computacional

Pensamento Computacional (Universidade Virtual do Estado de São Paulo)

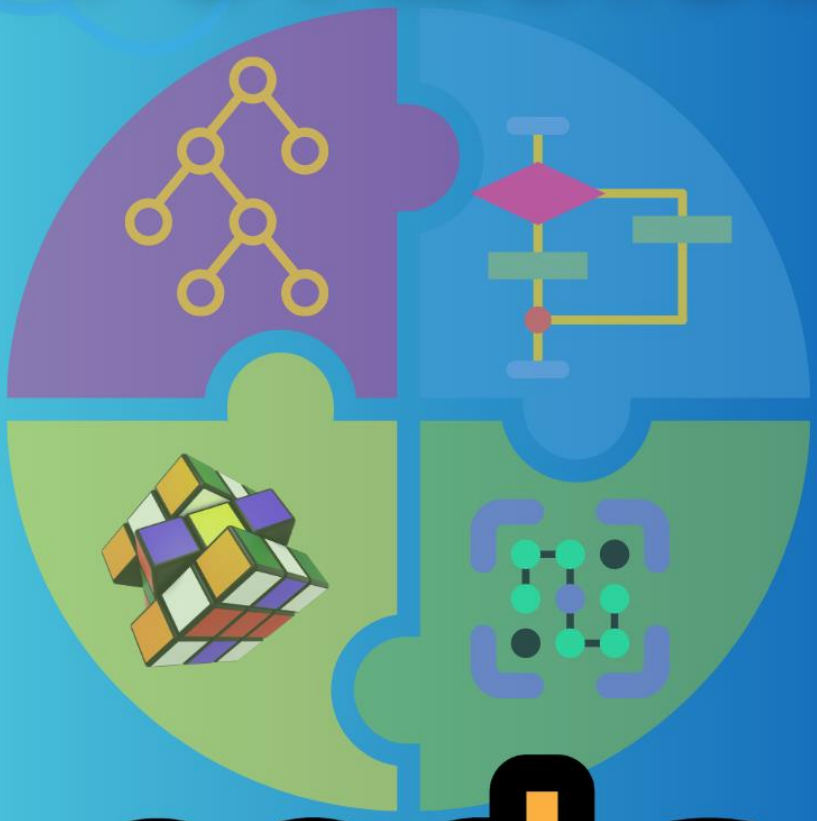
Scan to open on Studocu

Studocu is not sponsored or endorsed by any college or university
Downloaded by vinicius rangel (viniciuslelele@gmail.com)

eadvideos.com.br

RESUMO

PENSAMENTO COMPUTACIONAL



EAD VÍDEOS

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Resumo

Semana 1 ..... **2**
Introdução 2
Pilares do pensamento computacional 5
Como e quando ensinar o Pensamento Computacional 6
Semana 2 ..... **7**
Algoritmo e Lógica de Programação 7
Representação dos Algoritmos 10
Semana 3 ..... **13**
Scratch 13
Scratch - Parte 2 16
Semana 4 ..... **18**
Scratch - Parte 3 18
String 19
Uso de procedimentos para reutilização de código 19
Uso de listas para armazenar múltiplos valores do mesmo tipo 19
Semana 5 -Letramento digital **20**
Semana 6 - Pensamento computacional na Educação Básica **26**
Semana 7 - Resolução de problemas **29**

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Semana 1

Introdução

O pensamento computacional é a nova alfabetização, juntamente com a leitura, a escrita e a aritmética, e deve ser aprendido por todos, independentemente da profissão que venham a escolher.

É uma disciplina incluída com o objetivo de ensinar os conceitos básicos da forma de pensar de forma estruturada e organizada para resolver problemas. Seu grau de importância pode ser comprovado de muitas formas, uma delas é observando as dez habilidades essenciais do profissional do futuro segundo o Fórum Econômico Mundial:

1. Resolução de problemas complexos

2. Pensamento crítico

3. Criatividade

4. Gestão de pessoas

5. Coordenação

6. Inteligência Emocional

7. Capacidade de julgamento e de tomada de decisões

8. Orientação para servir

9. Negociação

10. Flexibilidade cognitiva

O Fórum Econômico Mundial é uma importante organização internacional localizada em Genebra (Suíça), responsável pela organização de encontros anuais com a participação e colaboração das maiores e principais empresas do mundo. Seu principal objetivo é “melhorar a situação do mundo”, através de ações tomadas e executadas por líderes mundiais, grandes economistas, investidores e empresários.

Dessa forma, vemos que as considerações emitidas por essas reuniões são de grande relevância mundial. Entre as 10 habilidades listadas, as destacas (habilidades 1, 2, 3 3 10) são ou podem ser desenvolvidas pelo pensamento computacional.

Segundo o documento da base comum curricular do MEC, o pensamento computacional envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos. É útil para descrever, explicar e modelar o universo e seus processos complexos.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Existem ainda alguns autores que dão definições importantes do que vem a ser o pensamento computacional:

> *"Pensamento Computacional é uma habilidade fundamental para todos, não somente para os cientistas da computação"*

Jeannette Wing (2006)

Essa definição levanta a questão de que adicionar o Pensamento Computacional na habilidade analítica de cada criança (juntamente com a leitura, a escrita e a aritmética) pode ser proveitoso para o seu futuro como profissional e cidadão.

> *"É o processo de reconhecer aspectos da computação em um mundo que nos cerca e, aplicar ferramentas e técnicas da Ciência da Computação para entender e argumentar sobre sistemas e processos naturais e artificiais"*

(FURBER, 2012)

> *"É uma abordagem usada para solução de problemas utilizando o que se sabe sobre Computação."*

(GOOGLE FOR EDUCATION, 2015)

Essas duas definições destacam o quanto o raciocínio computacional depende, naturalmente, da computação e suas técnicas de resolução de problemas. A computação é então a fonte do pensamento dito computacional.

O estudo do pensamento computacional é impulsionado por duas principais motivações. Em primeiro lugar, a inclusão social é um fator importante, pois a habilidade de lidar com soluções computacionais tornou-se essencial para a cidadania plena na sociedade digital atual, presente em todas as áreas do conhecimento.

Em segundo lugar, a produtividade é significativamente beneficiada pelo pensamento computacional. Através dele, é possível executar tarefas com agilidade e velocidade, além de resolver problemas de forma mais rápida e eficiente. Essa abordagem também permite lidar com demandas mais complexas, aumentando a escala dos problemas abordados com maior eficácia.

Devido a sua abrangência aparente, pode parecer a princípio que o pensamento computacional engloba habilidades múltiplas de outras disciplinas mais avançadas de computação ou que a

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

disciplina é exclusivamente baseada em técnicas de computação, dessa forma é importante destacar que:

- Pensamento Computacional não é saber navegar na internet, enviar e-mail, publicar um blog, operar um processador de texto ou planilha eletrônica;
- Pensamento Computacional pressupõe a utilização do computador como um instrumento capaz de aumentar o poder cognitivo e operacional humano;
- Não envolve apenas conceitos de Computação para solução de problemas em suas raízes, pois também agrega práticas de projetar sistemas, entender o comportamento humano e o pensamento crítico (WING, 2010);
- Não é apenas uma atividade de programação de computadores (Computação Desplugada).

Além disso, destacam se as habilidades que se relacionam diretamente com o pensamento computacional:

- **Coleta de dados:** capacidade de coletar informações de forma adequada;
- **Análise de dados:** dar sentido aos dados encontrando padrões e obtendo conclusões;
- **Representação de dados:** exibir dados através de gráficos, imagens e tabelas;
- **Decompor problemas:** separar uma tarefa em partes menores e gerenciáveis;
- **Abstração:** diminuir a complexidade do problema para poder identificar o elemento principal;
- **Algoritmos e procedimentos:** definir um conjunto de passos para resolver um problema ou tarefa;
- **Automação:** fazer uso de computadores e máquinas para execução de tarefas repetitivas;
- **Paralelização:** organizar recursos com o fim de realizar tarefas simultaneamente com o intuito de alcançar um objetivo comum;
- **Simulação:** representar ou modelar um processo.

Por fim, é importante pontuar as contribuições que o pensamento computacional traz para o indivíduo:

- **Pensamento Algorítmico:** Capacidade de projetar e seguir sequências lógicas de passos para resolver problemas, levando a soluções estruturadas e eficientes.
- **Aprendizagem Colaborativa:** Estimula o trabalho em equipe, permitindo que várias mentes contribuam para a solução de problemas complexos.
- **Resolução de Problemas:** Fomenta uma abordagem sistemática para enfrentar desafios, identificando padrões e aplicando estratégias adequadas.
- **Criatividade:** O pensamento computacional incentiva a busca por soluções inovadoras e criativas, ampliando a capacidade de encontrar alternativas únicas.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

- **Raciocínio Lógico:** Desenvolve habilidades para análise e argumentação coerente, essenciais para tomar decisões informadas e sustentadas.
- **Interpretação Textual:** Ajuda na compreensão e análise de informações escritas, permitindo que as pessoas sejam mais críticas e reflexivas ao ler e interpretar textos.

Pilares do pensamento computacional

Decomposição **Reconhecimento de padrões** **Abstração** **Algoritmo**

Sendo uma abordagem sistemática (organizada de acordo com uma certa ordem, com critérios). O pensamento computacional funciona identificando um problema e o decompondo em elementos menores, que são analisados individualmente, focando apenas em detalhes importantes e assim criar soluções. Podemos dizer que ele se baseia em quatro pilares que orientam o processo de solução de problemas:

Decomposição: é o processo que divide os problemas em partes menores para facilitar a resolução, desenvolvimento e gerenciamento. Analisa os problemas para identificar as partes que podem ser separadas e formas como podem ser reconstituídas para solucionar um problema como um todo. Possibilita resolver problemas complexos de forma mais simples e permite projetar sistemas de grande porte.

Reconhecimento de padrões: os padrões são similaridades ou características que alguns problemas compartilham. O hábito de identificar padrões nos acompanha desde a infância, é uma construção continuada e o nosso repertório de padrões não para de crescer e de se reconstruir. O processo de reconhecimento de padrões permite encontrar similaridades ou padrões entre pequenos problemas decompostos. Quanto mais padrões encontrarmos, mais fácil e rápida será a nossa tarefa geral de solução de problemas.

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Abstração: consiste na filtragem a classificação dos dados, criando mecanismos que permitam separar apenas os elementos essenciais em determinado problema, ignorando detalhes irrelevantes. Permite criar uma representação, uma ideia do que está se tentando resolver. Neste processo, essencial é escolher o detalhe a ser ignorado para que o problema seja mais fácil de ser compreendido sem perder nenhuma informação que seja importante para tal.

Algoritmo: é uma sequência finita de etapas (passos), cada qual executável em um tempo finito, por um agente computacional, natural (humano) ou sintético (computador). Um algoritmo é um plano, uma estratégia ou um conjunto de instruções ordenadas para a solução de um problema ou execução de uma tarefa. A formulação de um algoritmo passa pelo processo de decomposição, reconhecimento de padrões e abstração. Na execução seguirão os passos pré-definidos, não havendo a necessidade de criar um novo algoritmo para cada uma de suas execuções posteriores. É o pilar que agrega os demais pilares.

Como e quando ensinar o Pensamento Computacional

O aprendizado da programação de computadores é essencial, pois ensina a pensar de forma estruturada e lógica. Além disso, as habilidades do pensamento computacional estão entre as dez competências-chave do profissional do futuro, independente da área de atuação. Essa capacidade é considerada a nova alfabetização, ao lado da leitura, escrita e aritmética, e seu ensino tem o potencial de impactar positivamente tanto a sociedade quanto o desenvolvimento pessoal das pessoas.

A abordagem de ensino do pensamento computacional está em constante evolução, sendo predominantemente baseada em processos em vez de conteúdos fixos. Essa discussão, embora antiga, permanece relevante, explorando a relação entre tecnologia e educação. O computador pode tanto ser utilizado como uma ferramenta para ensinar quanto como um objeto a ser ensinado no contexto educacional.

Instrucionismo é uma abordagem de ensino que envolve o uso do computador como uma "máquina de ensinar". Nesse método, os métodos tradicionais de ensino são informatizados e implementados no computador, proporcionando aos alunos informações na forma de tutoriais, exercícios de prática ou jogos. O objetivo é utilizar a tecnologia para automatizar e aprimorar os processos de ensino já existentes, oferecendo aos alunos uma experiência de aprendizagem guiada e estruturada.

Construcionismo é uma abordagem educacional em que o estudante assume o papel de protagonista da aprendizagem, utilizando o computador como uma "máquina para ser ensinada". Nesse contexto, o aluno constrói algo de seu interesse, interagindo diretamente com o computador. O processo de construção é impulsionado por ações específicas no computador, que se mostram altamente efetivas no desenvolvimento do conhecimento.

Essa abordagem tem suas raízes nas ideias de Seymour Papert e está associada à linguagem de programação Logo, que permite que os alunos criem e manipulem objetos gráficos no computador. O construcionismo valoriza a aprendizagem ativa, a criatividade e a resolução de

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

problemas, possibilitando que os estudantes desenvolvam suas habilidades de forma significativa e autônoma

O ensino do pensamento computacional já é adotado em diversos países para crianças e adolescentes. Países como Japão, Finlândia, Inglaterra, Estados Unidos, Espanha e outros estão investindo na inserção do pensamento computacional na educação básica.

Essa abordagem permite que alunos com aptidão para a programação desenvolvam suas habilidades desde cedo, o que pode ser vantajoso para a descoberta precoce de talentos nessa área. Além disso, mesmo para os alunos que não se destacam na programação, o ensino do pensamento computacional proporciona uma melhor compreensão do mundo digital em que vivem, capacitando-os a enfrentar desafios tecnológicos de forma mais eficiente e tornando-os cidadãos mais informados e preparados para a sociedade digital contemporânea.

Semana 2

Algoritmo e Lógica de Programação

Um algoritmo é como uma receita de cozinha ou um conjunto de passos para fazer algo. Pense, por exemplo, que você tem uma receita para fazer um bolo - essa receita é o algoritmo. Ele te diz o que fazer em cada etapa, como misturar os ingredientes e assar o bolo.

Mas os algoritmos não são apenas para cozinhar. Eles também ajudam a resolver problemas ou realizar tarefas, como organizar listas de coisas ou encontrar a melhor rota para chegar a um lugar. Esses passos são como um plano lógico. É como seguir um caminho para resolver um quebra-cabeça ou qualquer outro problema que possa ser resolvido com uma sequência de passos. Os algoritmos também podem ser seguidos por diferentes agentes inteligentes, como computadores ou mesmo pessoas.

Criar um algoritmo envolve dividir um grande problema em partes menores (decomposição), reconhecer padrões para entender como lidar com eles e, finalmente, simplificar tudo para torná-lo mais fácil de seguir (abstração), como quebrar um grande quebra-cabeça em pedaços menores e mais fáceis de encaixar.

Quando criamos um algoritmo, precisamos escrever as instruções de forma clara para que possamos seguir o plano. Existem diferentes maneiras de fazer isso:

1. **Diagrama**: Um diagrama é uma representação visual das etapas do algoritmo. Pode ser um fluxograma, que usa formas geométricas e setas para mostrar as ações e decisões em sequência. Também podem ser blocos conectados, onde cada bloco representa uma ação específica.

2. **Pseudocódigo**: O pseudocódigo é uma maneira de escrever o algoritmo usando uma linguagem mais próxima da humana do que da linguagem de programação real. Ele se

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

concentra nas ações que precisam ser executadas e nas lógicas envolvidas, mas não se preocupa com detalhes específicos de programação.

3. **Linguagem de Programação**: Se quisermos que um computador execute o algoritmo, precisamos escrevê-lo em uma linguagem que a máquina possa entender. As linguagens de programação, como Python, C++, Java, entre outras, são usadas para traduzir nossas instruções em comandos que o computador pode seguir.

Resolver um problema envolve pensar de maneira organizada e lógica, o que significa seguir um conjunto de regras para chegar a conclusões corretas. Essa maneira de pensar é chamada de "lógica".

A lógica é como uma ferramenta que nos ajuda a pensar da melhor maneira possível. Ela lida com o "bem pensar", ou seja, com a forma como pensamos de maneira razoável e consistente. É como ter um manual para garantir que nossos pensamentos façam sentido.

Imagine que a lógica é como um conjunto de regras para montar um quebra-cabeça. Cada peça precisa se encaixar corretamente para formar uma imagem completa. Da mesma forma, a lógica nos ajuda a organizar nossos pensamentos de maneira que façam sentido e levem a conclusões verdadeiras.

Um dos principais trabalhos da lógica é diferenciar entre argumentos válidos (aqueles que têm uma estrutura lógica correta) e argumentos inválidos (aqueles que não seguem uma estrutura lógica correta). Isso nos ajuda a chegar ao conhecimento verdadeiro, porque podemos confiar que nossas conclusões são baseadas em razão e coerência, em vez de serem simplesmente aleatórias ou confusas.

A compreensão dos mecanismos da lógica pode se dar por meio das suas partes fundamentais:

1. **Proposição**: Uma proposição é uma frase que afirma algo como verdadeiro ou falso. É uma maneira de expressar uma ideia que pode ser julgada como correta ou errada. Por exemplo, "O sol é uma estrela" é uma proposição, pois podemos verificar se é verdadeira ou falsa.

2. **Premissas**: As premissas são proposições que usamos como base para provar ou apoiar alguma coisa. São como os tijolos que sustentam uma construção. Quando estamos defendendo uma ideia ou argumento, usamos premissas para dar suporte lógico a essa ideia. Por exemplo, se estamos argumentando que "Estudar é importante para aprender", podemos usar premissas como "O estudo ajuda a adquirir conhecimento" para fundamentar essa ideia.

3. **Conclusão**: A conclusão é a proposição que segue necessariamente das premissas. Se concordarmos com as premissas e seguirmos a lógica corretamente, chegaremos à conclusão. É como a parte final de um raciocínio. Usando o exemplo anterior, a conclusão pode ser "Portanto, devemos dedicar tempo ao estudo para aprender".

Essas três partes trabalham juntas para formar um argumento lógico. Começamos com premissas, que são as afirmações de suporte. Em seguida, seguimos a lógica para chegar a uma

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

conclusão que decorre das premissas. A ideia é que, se as premissas são verdadeiras e a lógica é sólida, a conclusão também deve ser verdadeira com base nas informações dadas.

Lógica de programação é a técnica de ligar ideias em uma ordem organizada para alcançar um objetivo específico. É como criar um plano passo a passo para resolver um problema ou realizar uma tarefa.

A sequência lógica é a série de passos que seguimos para atingir um objetivo ou resolver um problema. São como instruções num manual. Por exemplo, se o objetivo for fazer um sanduíche, a sequência lógica inclui passos como pegar duas fatias de pão, espalhar manteiga, adicionar recheios, etc.

Essa sequência de passos é um conjunto de instruções. Na lógica de programação, estas instruções são as responsáveis por fazer um programa de computador funcionar. Cada instrução é uma parte crucial do processo para alcançar o resultado desejado.

A ordem da sequência de passos é fundamental porque determina como as ações se relacionam entre si e afetam o resultado final de uma tarefa. Considere que:

1. **Uma gaveta está fechada**: essa é a situação inicial. Não é possível obter nenhum objeto que eventualmente esteja dentro da gaveta. É necessário abrir a gaveta primeiro.

2. **Existe uma caneta dentro da gaveta**: temos por objetivo pegar a caneta. Mas antes disso, há um passo necessário para atingir esse objetivo (que não pode ser alcançado diretamente ou imediatamente antes que outra ação ocorra).

3. **Precisamos primeiro abrir a gaveta para depois pegar a caneta**: Esta afirmação enfatiza a importância da ordem. Se a ordem dos passos for invertida, não será possível pegar a caneta antes de abrir a gaveta, pelo simples fato da caneta estar inacessível até que a gaveta seja aberta.

Este exemplo destaca por que a sequência correta dos passos é crucial. A lógica por trás disso é que cada ação prepara o terreno para a próxima ação. Se não seguirmos a ordem adequada, as ações podem não ser bem-sucedidas ou causar resultados indesejados.

Na programação de computadores, a importância da ordem da sequência de passos é a mesma. Ao escrever um programa com a ordem das instruções trocada, isso pode resultar em erros ou resultados inesperados. A ordem correta garante que cada ação ocorra no momento adequado e que o programa funcione conforme o esperado.

Instruções são conjuntos de regras ou normas que guiam a realização ou execução de algo. Em programação, uma instrução é uma ação básica a ser executada. No entanto, é essencial entender que uma única instrução não é suficiente para completar todo um processo. Para alcançar um resultado desejado, é necessário ter um conjunto de instruções organizadas em uma ordem sequencial lógica.

Um algoritmo deve ser traduzido para uma linguagem de programação para que um computador possa entendê-lo e executá-lo. Essa linguagem é uma forma escrita e formal que contém um conjunto de instruções e regras. Ela é usada para criar programas (softwares) ao

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

definir como o computador deve realizar tarefas específicas seguindo as etapas estabelecidas pelo algoritmo.

Programas de computador (softwares) são conjuntos de instruções escritas em uma linguagem de programação específica. Eles são como guias detalhados que dizem ao computador o que fazer.

O processo é a ação de executar um programa. Isso também se relaciona à execução de um algoritmo, que é um plano lógico para resolver um problema.

Um software pode ser criado para ser usado em diferentes tipos de dispositivos, como computadores, dispositivos móveis ou qualquer equipamento que possa executar as instruções do programa. Isso significa que os programas podem ser adaptados para funcionar em diferentes ambientes, desde que esses ambientes possam interpretar e seguir as instruções fornecidas pelo software.

Representação dos Algoritmos

As diversas formas de representar algoritmos, como a descrição narrativa, o pseudocódigo e o fluxograma (ou diagrama de blocos), oferecem maneiras distintas de expressar os passos necessários para resolver problemas ou realizar tarefas. A descrição narrativa utiliza linguagem natural, enquanto o pseudocódigo é uma linguagem intermediária mais formal que a linguagem comum. Por outro lado, o fluxograma é uma representação gráfica que utiliza símbolos e setas para ilustrar a sequência lógica dos passos. Cada abordagem é selecionada com base na situação e nas preferências do programador, mas todas têm em comum a finalidade de transmitir de maneira clara a lógica do algoritmo.

A descrição narrativa é uma forma de expressar algoritmos usando a linguagem comum que utilizamos para nos comunicar diariamente. No entanto, essa abordagem tem uma desvantagem, já que pode resultar em ambiguidades e falta de precisão. Um exemplo conhecido é um livro de receitas, que pode conter passos para cozinhar pratos, mas nem sempre fornece instruções detalhadas o suficiente.

O pseudocódigo é uma abordagem para descrever conjuntos de instruções que não requerem a sintaxe específica de uma linguagem de programação formal. Embora não seja uma linguagem de programação em si, é semelhante à escrita em linguagens de programação. Também é conhecido como "Português Estruturado" ou "Portugol".

No pseudocódigo, as instruções frequentemente são escritas em maiúsculas, as variáveis em minúsculas e as mensagens em maiúsculas, facilitando a diferenciação. Essa técnica visa fornecer uma representação mais compreensível e clara das ações a serem executadas em um algoritmo, servindo como uma etapa intermediária entre a linguagem natural e as linguagens de programação formais.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

O pseudocódigo apresenta várias vantagens, como o uso do português como base, permitindo definir a estrutura dos dados e a forma como serão manipulados, além de possibilitar uma transição relativamente rápida do algoritmo para uma linguagem de programação real.

No entanto, apresenta desvantagens como a necessidade de criar uma linguagem não real para trabalhar e a falta de padronização. Embora seja uma ferramenta útil para expressar algoritmos de maneira mais legível e compreensível do que as linguagens de programação formais, é importante considerar essas vantagens e desvantagens ao escolher o pseudocódigo como método de representação.

As regras para escrever pseudocódigo são as seguintes:

1. Usar apenas um verbo por frase, no modo imperativo.
2. Utilizar sentenças compreensíveis por pessoas sem conhecimento técnico no assunto.
3. Empregar frases curtas e simples.
4. Ser direto e objetivo nas instruções.
5. Evitar palavras com interpretação ambígua.

Um fluxograma, também conhecido como diagrama de blocos, é uma representação gráfica de um conjunto de instruções. Ele utiliza símbolos geométricos para representar diferentes ações ou comandos em um algoritmo. Essa abordagem gráfica facilita a compreensão dos algoritmos e permite visualizar claramente os passos a serem seguidos.

Os fluxogramas podem ser subdivididos em várias etapas para fornecer detalhes mais específicos sobre o que está ocorrendo em cada passo. Além disso, é possível simplificar fluxogramas para combinar várias etapas em uma única etapa, tornando a visualização mais clara e concisa.

O fluxograma, ou diagrama de blocos, possui uma estrutura bem definida em termos de sintaxe e semântica. A sintaxe trata da aplicação correta de seus elementos, enquanto a semântica se refere ao significado dos símbolos utilizados. É importante que apenas uma linha de fluxo se conecte a um terminal ou conector. Além disso, cada ação no fluxograma deve ser claramente descrita, evitando ambiguidades.

Nesta disciplina seguiremos as seguintes definições de símbolos para as ações de um fluxograma:


Retângulo com cantos arredondados para definir o início e o fim do fluxograma;

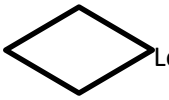

Paralelogramo ou retângulo com canto cortado para representar entrada de dados;



Downloaded by vinicius rangel (viniciuslelele@gmail.com)

 Retângulo para representar instruções genéricas de processamento;

 Símbolo de documento para representar a saída de dados;

 Losango para representar um ponto de decisão;

 Seta para representar o fluxo de dados e conexão entre os demais elementos;

 Círculo para conectar as demais partes do fluxograma.

O fluxograma oferece três formas diferentes de ordem de execução:

1. **Sequencial**: Nesse tipo, as atividades são executadas em ordem, uma após a outra, seguindo uma sequência lógica. É como seguir uma lista de tarefas em que uma ação ocorre após a conclusão da anterior.

2. **Por seleção**: Aqui, ocorre uma bifurcação no processo. Uma via de processamento é escolhida com base em uma decisão ou condição. Cada via leva a um processamento diferente. É semelhante a tomar decisões ramificadas em um caminho.

3. **Por repetição**: Esse tipo envolve a execução em ciclos de processamento até que uma condição de finalização seja alcançada. É como realizar uma ação repetidamente até que um objetivo seja atingido ou uma condição seja satisfeita.

Um algoritmo sempre deverá ser escrito considerando 3 fases:

1. ENTRADA: Essa é a etapa em que os dados de entrada são obtidos para o algoritmo. Geralmente identificada por verbos como "Ler", "Receber" e assim por diante, essa fase envolve coletar as informações necessárias para realizar os cálculos ou ações subsequentes.

2. PROCESSAMENTO: Aqui, estão os procedimentos necessários para alcançar o resultado final do algoritmo. Essa fase abrange os cálculos, manipulações ou transformações realizadas nos dados de entrada para produzir o resultado desejado.

3. SAÍDA: Nesta etapa, os dados processados e o resultado final são apresentados ou retornados. Geralmente identificada por verbos como "Retornar", "Imprimir", "Mostrar" e outros semelhantes, essa fase conclui o algoritmo ao fornecer o resultado ou a saída desejada.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Semana 3

Scratch

Um programa de computador é composto por um conjunto de instruções que orientam as ações a serem executadas pelo computador. Essas instruções são criadas por meio de linguagens de programação, que possuem regras de sintaxe, podendo ser desafiadoras para iniciantes. Uma abordagem acessível é o Scratch, um ambiente de programação visual desenvolvido pelo MIT, que utiliza blocos gráficos para representar as instruções. No Scratch, a vantagem é que não é necessário lidar com comandos complexos, o que torna o aprendizado de programação mais simples e prazeroso.

Os objetivos da plataforma incluem a criação de projetos interativos abrangendo jogos, histórias, simulações científicas, arte e música, bem como a geração de Objetos de Aprendizagem. A plataforma promove o desenvolvimento de criatividade, raciocínio lógico e habilidades matemáticas, fomentando o aprimoramento das capacidades de resolução de problemas através do Pensamento Computacional.

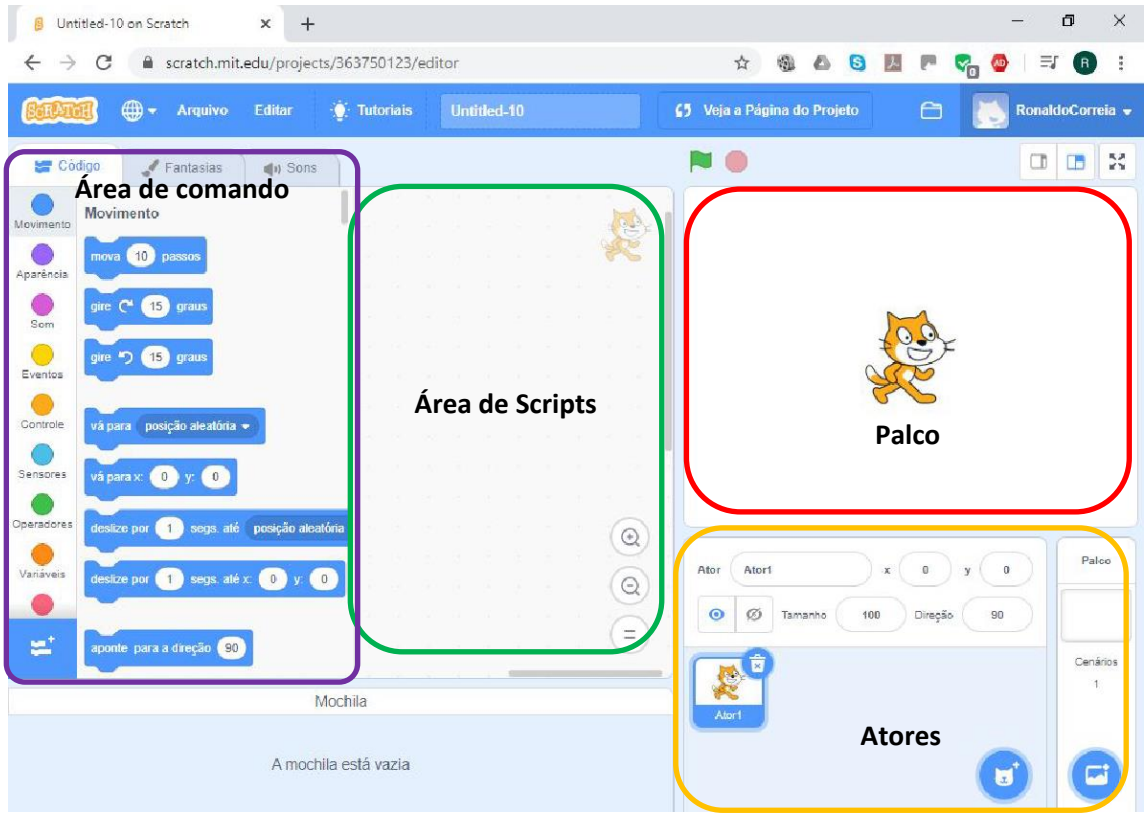
A motivação para incorporar o Scratch em cursos de licenciatura é sua capacidade de se integrar de maneira contextualizada ao processo de ensino e aprendizagem. Isso se traduz em uma valiosa ferramenta para a criação de recursos didáticos, onde professores podem desenvolver e aplicar materiais, ou os alunos podem produzir seus próprios conteúdos.

Além disso, a programação encontra aplicação em várias disciplinas e configurações escolares, o que a torna relevante em diferentes contextos educacionais. O objetivo subjacente é impulsionar a inovação nas práticas pedagógicas dos futuros educadores, enriquecendo sua abordagem de ensino.

A plataforma está disponível em vários idiomas e pode ser instalada localmente (pelo link: https://scratch.mit.edu/download) ou acessada em sua versão online (pelo link: https://scratch.mit.edu/).

Vejamos os componentes do ambiente de programação da plataforma:

Downloaded by vinicius rangel (viniciuslelele@gmail.com)



O Palco, ou Stage, no Scratch, é o ambiente onde ocorre a apresentação do resultado da execução do programa criado. Funciona como um simulador onde os Atores, também conhecidos como Sprites, se movimentam e interagem. Uma característica interessante é a capacidade de adicionar cenários, chamados de Panos de Fundo, ao Palco, permitindo que os projetos tenham diferentes ambientes ou configurações visuais.

O Palco tem dimensões padrão de 480 passos de largura (no eixo X) por 360 passos de altura (no eixo Y), sendo que o ponto central desse espaço é a coordenada 0,0. Isso significa que as coordenadas dos objetos no Palco podem variar de -240 a 240 no eixo X e de -180 a 180 no eixo Y, com o ponto central localizado em (0,0).

A aba Cenários, ou Backdrops, no Scratch, oferece a funcionalidade de criar e editar cenários ou imagens de fundo para o Palco. Isso pode ser feito usando o Paint Editor, uma ferramenta incorporada que permite desenhar, pintar e editar imagens diretamente no ambiente Scratch. A aba Cenários é onde o usuário organiza todas as imagens de fundo que serão usadas no projeto.

No Scratch, os Atores, também conhecidos como Sprites, desempenham um papel fundamental nos projetos. Cada projeto pode incluir um ou mais Atores, e todos esses Atores são listados na chamada "Lista de Atores". Cada um dos Atores no projeto possui suas próprias características individuais, incluindo códigos, que são os scripts que definem suas ações e comportamentos, e fantasias, que representam sua aparência visual.

This document is available on studocu
Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Essa abordagem permite aos criadores de projetos no Scratch dar vida a personagens ou objetos distintos em suas criações. Cada ator pode ser programado de forma independente, o que possibilita a criação de interações complexas e a personalização de cada elemento do projeto, desde personagens em um jogo até elementos em uma animação ou simulação.

A aba Fantasias, ou Costumes, no Scratch, oferece a funcionalidade de criar e editar fantasias para os Atores (Sprites) do projeto. Os Atores podem ter várias fantasias que representam suas diferentes aparências visuais ao longo do projeto. Essa aba é essencial para a customização visual dos Atores, permitindo que os criadores de projetos personalizem a aparência de seus personagens, objetos ou elementos interativos. Ao organizar as fantasias de um ator, é como se estivesse gerenciando um guarda-roupa virtual para esse personagem, possibilitando transições suaves e animações visuais interessantes ao longo do projeto.

A aba Blocos ou Código no Scratch é o espaço onde os criadores de projetos usam elementos gráficos chamados de "Blocos" para construir os programas. Esses Blocos representam comandos e ações que serão executados pelo ator (Sprite) ou pelo Palco, dependendo do contexto.

Os Blocos são essenciais para organizar as instruções de um programa, e eles estão divididos em 10 categorias, cada uma identificada por uma cor específica. Essas categorias representam diferentes tipos de comandos, como movimento, aparência, som, sensores, controle, entre outros, facilitando a busca e seleção dos comandos necessários para criar a lógica do programa. Uma característica importante dos Blocos no Scratch é que eles estão associados ao ator ou ao Palco que está selecionado no momento, o que permite programar de forma específica cada elemento do projeto.

A Área de Scripts, também conhecida como Área de Programação, é o espaço no Scratch onde os Blocos de comandos são arrastados e conectados para construir os projetos. Nesta área, os comandos são organizados em uma ordem de execução específica, determinando a sequência das ações que serão realizadas pelo ator (Sprite) ou pelo Palco. Os Blocos de comandos na Área de Scripts estão sempre associados ao ator ou ao Palco que está selecionado no momento, permitindo programar de forma direta e específica cada elemento do projeto.

Além disso, na Área de Scripts, os criadores de projetos podem testar os Blocos à medida que são criados, o que ajuda a depurar e verificar o comportamento do programa em tempo real. Essa funcionalidade é útil para identificar possíveis erros ou ajustar o funcionamento do projeto de acordo com as necessidades.

A aba Sons no Scratch permite que os Palcos e Atores reproduzam sons em projetos. Os usuários podem inserir, editar ou até mesmo gravar áudio diretamente no projeto, associando-o ao ator ou palco selecionado. Essa funcionalidade enriquece a experiência do projeto, adicionando elementos sonoros que podem ser usados para criar efeitos sonoros, músicas ou narrativas interativas.

A Mochila, ou Backpack, no Scratch, é uma funcionalidade que permite aos usuários compartilhar seus projetos e utilizar atores e códigos de outros projetos existentes. No entanto, a Mochila está disponível apenas para usuários logados no sistema. O conteúdo da Mochila é

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

armazenado no servidor do Scratch, o que facilita o compartilhamento e a reutilização de elementos de projetos entre diferentes usuários e criações. Isso promove a colaboração e a criatividade, permitindo que os usuários aproveitem recursos compartilhados para enriquecer seus próprios projetos.

Na programação Scratch, a "Inicialização e movimentos" se refere a como um programa começa a ser executado e como ele pode ser interrompido.

1. **Bandeira Verde:** A bandeira verde é um ícone que marca o início da execução do programa. Quando você clica na bandeira verde, o projeto é iniciado, e todas as instruções que você programou começam a ser executadas. É o ponto de partida para testar e ver o funcionamento do seu programa ou jogo no Scratch.

2. **Botão Vermelho:** O botão vermelho, por outro lado, é usado para encerrar a execução do programa a qualquer momento. Se você clicar no botão vermelho, o programa será interrompido imediatamente, independentemente de onde ele estiver no momento. Isso é útil quando você deseja parar a execução do programa antes que ele termine naturalmente ou se algo não estiver funcionando como o esperado.

Scratch - Parte 2

No Scratch, os Blocos de comandos são organizados em categorias para facilitar a identificação e seleção dos comandos necessários para construir programas. Essas categorias são distinguidas por cores e agrupam comandos com funcionalidades semelhantes. Existem 10 categorias de comandos mais utilizadas no Scratch, incluindo:

1. **Movimento (Azul):** Contém comandos relacionados ao movimento dos Atores no Palco, como mover-se para frente, para trás, girar, entre outros.

2. **Aparência (Roxo):** Inclui comandos para modificar a aparência dos Atores ou Palco, como alterar o tamanho, mudar de fantasia ou definir cores. Também possibilita a comunicação do ator com os comandos “diga” e “pense”.

3. **Som (Magenta):** Contém comandos para lidar com sons, como reproduzir, parar ou alterar o volume de áudio.

4. **Eventos (Amarelo):** Inclui comandos que permitem detectar eventos que podem estar condicionados à outras ações ou eventos.

5. **Controle (Laranja):** Contém comandos de controle de fluxo, como loops (repetições) e condicionais (se-então).

6. **Sensores (Azul turquesa):** Inclui comandos que permitem aos Atores reagir a eventos ou condições, como detecção de colisões ou cliques do mouse.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

7. **Operadores (verde):** permite aplicar operadores lógicos, relacionais, matemáticos, textuais entre outros.

8. **Variáveis (Laranja escuro):** Permite criar e gerenciar variáveis para armazenar e manipular dados durante a execução do programa.

9. **Meus blocos (Rosa):** são blocos de comandos personalizados que podem ser criados para agrupar sequências de comandos frequentemente usadas em um único bloco.

Existem quatro tipos principais de blocos no Scratch:

1. **Blocos de Comando:** São usados para dar instruções e podem ser empilhados como uma pilha de ações. Eles frequentemente requerem entradas, como texto editável ou seleções de menu.

2. **Blocos de Função:** São blocos personalizados que realizam ações específicas. Eles podem ter entradas personalizadas e são reutilizáveis em diferentes partes do projeto.

3. **Blocos de Trigger:** Executam blocos que estão abaixo deles na sequência. Eles respondem a eventos específicos, como cliques de mouse ou pressionamentos de tecla.

4. **Blocos de Controle:** São usados para organizar e controlar o fluxo de execução de blocos em um projeto. Eles não realizam ações diretamente, mas armazenam outros blocos e controlam como esses blocos são executados.

No Scratch, as variáveis são estruturas usadas para armazenar valores na memória do computador. Elas são fundamentais para armazenar e manipular informações durante a execução de um programa. Aqui estão alguns aspectos importantes sobre variáveis no Scratch:

Tipos de Dados: No Scratch, as variáveis podem ser de três tipos principais de dados:

1. **Numérico:** Essas variáveis armazenam números, que podem ser inteiros ou decimais (reais).

2. **Textual:** Variáveis de texto (ou strings) são usadas para armazenar sequências de caracteres, como palavras ou frases.

3. **Lógico:** Variáveis lógicas, ou booleanas, têm apenas dois valores possíveis: verdadeiro (true) ou falso (false).

Escopo das Variáveis: As variáveis no Scratch têm escopo local por padrão. Isso significa que elas são acessíveis apenas naquele bloco de código ou naquele bloco de função onde foram criadas. No entanto, você também pode criar variáveis globais que são acessíveis em todo o projeto.

Atribuição de Novos Valores: Você pode atribuir um novo valor a uma variável a qualquer momento no código. Por exemplo, para definir uma variável chamada "pontuação" com o valor 10, você usaria o bloco "definir pontuação para 10".

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Operações com Variáveis: Além de definir valores, você pode realizar operações com variáveis, como adicionar ou subtrair um valor específico ao valor atual. Isso é útil para rastrear contagens, pontuações, e outras variáveis que podem mudar durante a execução do projeto.

A entrada de dados no Scratch envolve a leitura de informações digitadas pelo usuário usando os blocos "Pergunte e Espere". Esses dados são armazenados no "Bloco Resposta", permitindo que você utilize as respostas do usuário em seu programa. Isso é útil para criar interações com o usuário em projetos, como jogos que pedem nomes de jogadores ou programas de simulação que solicitam parâmetros personalizados.

A categoria "Operadores" no Scratch permite realizar uma variedade de operações, incluindo cálculos aritméticos, avaliações lógicas e funções matemáticas mais avançadas. Os "Operadores Aritméticos" são símbolos que representam operações matemáticas básicas e podem ser usados com variáveis ou valores fixos. As "Expressões Lógicas" são combinações de operadores relacionais, lógicos e identificadores que resultam em valores lógicos (verdadeiro ou falso). Os "Operadores Relacionais" comparam valores e retornam valores lógicos com base na comparação, podendo ser usados com variáveis ou valores fixos. Os "Operadores Lógicos" servem para combinar expressões lógicas e avaliar se são verdadeiras ou falsas. Em conjunto, essas ferramentas são fundamentais para criar lógica e funcionalidade em projetos no Scratch.

A categoria "Controle" no Scratch oferece comandos pré-definidos que são responsáveis pela estrutura lógica de conexão entre outros comandos em um projeto. Ela é fundamental para controlar o fluxo de execução dos comandos em seu projeto. Dentro dessa categoria, encontramos duas estruturas principais:

1. **Estruturas de Seleção (Decisão):** Essas estruturas são usadas para representar desvios no fluxo normal de comandos. Elas permitem que um bloco de comandos seja executado quando determinadas expressões lógicas são satisfeitas. Em outras palavras, você pode criar ramificações no seu código para lidar com diferentes situações com base em condições lógicas, como "se" uma determinada condição for verdadeira, execute um conjunto específico de comandos.

2. **Estruturas de Repetição:** As estruturas de repetição são usadas quando um conjunto de ações precisa ser executado várias vezes. Elas permitem que você crie loops (laços) que repetem comandos até que uma condição seja atendida. Isso é útil para automatizar tarefas repetitivas ou criar animações baseadas em movimentos repetidos.

Essas estruturas de controle são fundamentais para criar programas mais complexos e interativos no Scratch, permitindo que você tome decisões com base em condições e execute ações repetitivas, tornando seus projetos mais dinâmicos e funcionais.

Semana 4

Scratch - Parte 3

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

String

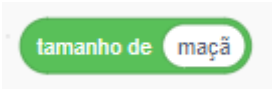
Strings são sequências de caracteres, como palavras ou frases. No Scratch, você pode criar variáveis do tipo string para armazenar e manipular esses dados de texto. Isso é útil para lidar com informações como nomes, mensagens e muito mais.

Com strings, você pode manipular caracteres individualmente. Isso permite que você acesse letras específicas ou realize operações em caracteres específicos de uma string.

O operador *letra* permite que você acesse letras específicas em uma string. Isso é alcançado usando índices para representar a posição de cada caractere na string, o índice do primeiro caracter é 1 do segundo é 2 e assim por diante.



Você pode usar o operador *tamnho de* para encontrar o tamanho de uma string.



Uso de procedimentos para reutilização de código

Os procedimentos são essenciais para reutilizar código. Você pode criar um bloco de código que realize uma tarefa específica e, em seguida, chamá-lo sempre que precisar executar essa tarefa novamente.

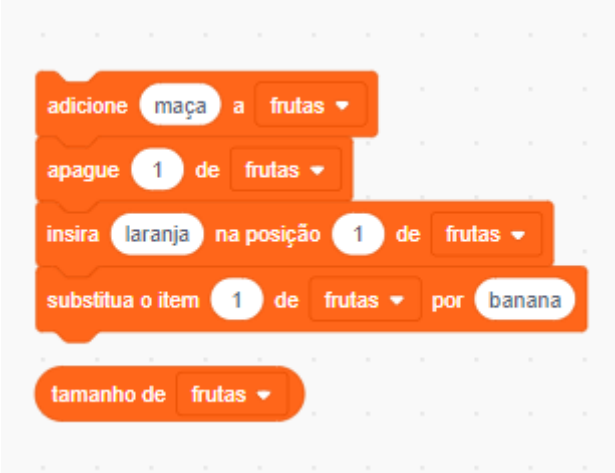


Uso de listas para armazenar múltiplos valores do mesmo tipo

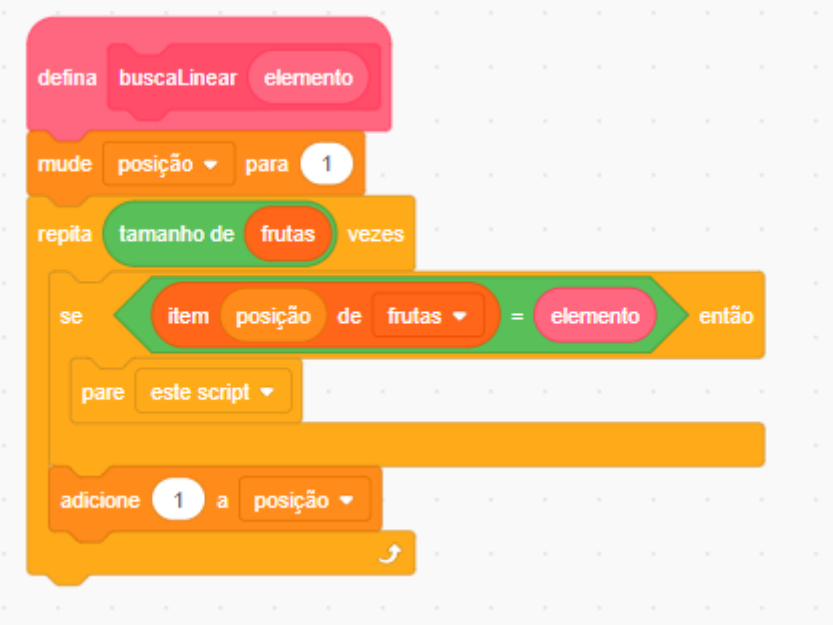
Listas permitem que você armazene vários valores do mesmo tipo em uma única estrutura. Isso é útil para lidar com coleções de dados, como uma lista de nomes, números ou outros itens.

As listas podem ser manipuladas usando comandos como adicionar, remover, substituir elementos e encontrar o tamanho da lista. Isso é crucial para trabalhar com conjuntos de dados de maneira eficaz.

Downloaded by vinicius rangel (viniciuslelele@gmail.com)



A busca linear é um método simples para encontrar um elemento específico em uma lista, percorrendo-a item por item. Isso pode ser útil para localizar informações em listas não ordenadas.



Semana 5 - Letramento digital

Computação

A computação é a ciência que se dedica ao estudo da resolução de problemas através da criação e aplicação de algoritmos. Algoritmos são sequências de instruções que orientam um computador na realização de tarefas específicas. O pensamento computacional, um pilar fundamental da computação, envolve a habilidade de decompor problemas complexos em partes menores, identificar padrões e desenvolver algoritmos eficazes para resolvê-los. Os principais pilares do pensamento computacional incluem:

This document is available on studocu
Downloaded by vinicius rangel (viniciuslelele@gmail.com)

- **Algoritmo:** A capacidade de criar sequências lógicas de passos para resolver um problema.
- **Abstração:** A habilidade de simplificar um problema, isolando as informações essenciais e ignorando detalhes irrelevantes.
- **Decomposição:** A capacidade de dividir um grande problema em subproblemas menores e mais gerenciáveis.
- **Reconhecimento de Padrões:** Identificação de tendências e regularidades em dados e informações.

A computação também nos permite explorar o funcionamento do mundo digital, que se tornou uma parte integrante de nossa sociedade. Desde a comunicação online até a coleta e análise de dados, o conhecimento sobre como o mundo digital opera é fundamental para uma participação efetiva na era digital.

Informação e Dados

Em computação, é crucial distinguir entre dados e informação. Dados são representações brutas de fatos, números ou caracteres. Por outro lado, a informação é o significado extraído desses dados, o conhecimento ou contexto que eles representam. Por exemplo, a temperatura 38°C é um dado, enquanto "está muito quente" é uma informação derivada desse dado.

Componentes de um Computador

Os computadores são compostos por dois elementos principais: hardware e software. O hardware é a parte física do computador, incluindo componentes como o processador (CPU), memória RAM, armazenamento (HD ou SSD) e dispositivos de entrada e saída, como teclados, mouses e monitores. O software, por outro lado, é a parte lógica do computador, que inclui o sistema operacional, responsável por gerenciar os recursos do hardware, e os programas de aplicação, que são desenvolvidos para realizar tarefas específicas, como processamento de texto ou edição de fotos.

Software

O software pode ser categorizado em duas principais classes: sistema operacional e programas de aplicação. O sistema operacional é uma interface que controla todas as operações do computador e permite que os programas de aplicação funcionem. Exemplos de programas de aplicação incluem navegadores da web, processadores de texto, planilhas eletrônicas, reprodutores de mídia e muito mais.

Processamento de Dados

O processamento de dados é o núcleo da computação. Ele envolve três etapas principais: entrada, processamento e saída. A entrada consiste na coleta de dados ou informações do ambiente. O processamento envolve a execução de operações e cálculos nos dados. A saída é a apresentação dos resultados do processamento para o usuário.

Armazenamento de Dados

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Os dados são armazenados em unidades de medida como bytes, kilobytes, megabytes e assim por diante. As diferenças entre discos rígidos (HDs) e unidades de estado sólido (SSDs) são importantes. Os HDs são mecânicos, mais lentos e usam discos rotativos para armazenamento. Por outro lado, os SSDs são mais rápidos, pois usam chips de memória flash, não possuem partes móveis e estão se tornando cada vez mais populares devido à sua velocidade e confiabilidade.

Processamento de Dados:

No cerne da revolução digital estão os dados. Armazenamento de dados refere-se ao ato de guardar informações em dispositivos, como discos rígidos ou servidores, para uso futuro. O processamento de dados, por outro lado, envolve a manipulação dessas informações para gerar insights ou realizar tarefas específicas, como cálculos complexos ou análise de dados.

Redes de Computadores e a Internet

Redes de computadores são sistemas interconectados que permitem a troca de informações entre dispositivos. A internet é um exemplo notável, servindo como uma "rede das redes" que conecta milhões de redes em todo o mundo. Isso possibilita a comunicação instantânea e o compartilhamento de recursos em escala global.

Acesso à Internet Através de Provedores de Serviço

Para acessar a internet, a maioria das pessoas depende de provedores de serviço de internet (ISP). Essas empresas fornecem conexões de internet e podem oferecer acesso móvel ou local, dependendo das necessidades do usuário.

Tipos de Conexões, Incluindo Conexão Móvel e Local

As conexões à internet podem ser móveis, usando tecnologias como 3G e 4G, ou locais, usando infraestrutura de cabo ou fibra óptica. A escolha depende da disponibilidade e das preferências do usuário.

Roteadores e Acesso Físico à Internet

Os roteadores são dispositivos que permitem que vários dispositivos se conectem a uma rede local e, por meio dela, à internet. Eles fazem a ponte entre a rede local e a conexão física à internet, geralmente por meio de cabos ou conexões sem fio.

Largura de Banda

A largura de banda se refere à capacidade de uma conexão para transportar dados. Quanto maior a largura de banda, maior será a velocidade de conexão. Isso é fundamental para garantir uma experiência de internet rápida e eficaz.

Navegação na Web

A web é acessada por meio de navegadores da web, como Chrome, Firefox e Safari. Os protocolos HTTP e HTTPS governam a forma como os dados são transmitidos da web para o navegador. O HTTPS é uma versão segura do HTTP, garantindo a criptografia dos dados durante a transmissão.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Diferenças Entre Sites, Blogs e Wikis:

Sites são páginas da web que podem conter informações diversas. Blogs são sites focados em postagens de artigos e opiniões. Wikis são plataformas colaborativas onde os usuários podem editar e contribuir com o conteúdo.

Registros de Domínio

Registros de domínio são nomes únicos que identificam um site na internet, como "www.exemplo.com". Eles desempenham um papel fundamental na identificação e acessibilidade dos sites.

Ferramentas de Busca e Busca Avançada no Google

As ferramentas de busca, como o Google, ajudam os usuários a encontrar informações na web. A busca avançada permite refinar os resultados com base em critérios específicos.

Operadores de busca, como aspas para pesquisar frases específicas ou o sinal de menos para excluir termos, permitem refinar ainda mais os resultados da pesquisa.

Os mecanismos de busca também permitem filtrar os resultados por tipo de documento, como PDFs, e limitar a pesquisa a sites específicos para encontrar informações relevantes com mais eficiência.

E-mail

O correio eletrônico, ou e-mail, desempenha um papel fundamental no mundo digital atual. Ter um e-mail pessoal é de extrema importância, pois ele serve como sua identidade digital e meio de comunicação universal. Ao escolher um nome de e-mail, é essencial optar por algo significativo e memorável. Evite combinações complexas de números e letras e, se possível, utilize seu nome real. Isso facilita a identificação e a memorização para você e para os outros.

Além disso, a estruturação adequada de e-mails é crucial para a comunicação eficaz. Comece com um assunto claro e conciso que resuma o conteúdo da mensagem. Em seguida, mantenha o corpo do e-mail organizado, dividindo-o em parágrafos curtos e informativos. Evite o uso excessivo de jargões técnicos e seja claro em sua comunicação. Lembre-se de que o e-mail é uma forma de comunicação escrita, e a clareza é fundamental.

Computação em nuvem

A computação em nuvem revolucionou a forma como armazenamos e acessamos informações. Ela envolve o fornecimento de serviços de computação pela internet, permitindo que os usuários acessem recursos de processamento e armazenamento em servidores remotos. Uma das principais vantagens da computação em nuvem é a escalabilidade, o que significa que você pode aumentar ou diminuir seus recursos de acordo com as necessidades, evitando gastos excessivos com infraestrutura.

O armazenamento em nuvem é uma parte fundamental da computação em nuvem, permitindo que documentos, fotos, vídeos e outros arquivos sejam armazenados de forma segura em servidores remotos. Serviços populares como Google Drive e Dropbox oferecem espaço de

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

armazenamento acessível e conveniente, permitindo o acesso a partir de qualquer dispositivo conectado à internet.

Organizar seus dados é crucial para a eficiência e a produtividade. Ao armazenar documentos e arquivos, crie uma estrutura de pastas lógica e intuitiva. Por exemplo, você pode organizar documentos por projetos, datas ou categorias. Isso facilita a localização rápida e eficaz de informações quando necessário.

Para a edição de documentos, ferramentas como Microsoft Word, Google Docs e LibreOffice são amplamente utilizadas. Elas oferecem recursos de formatação avançados, permitindo a criação de documentos bem elaborados, com índices, tabelas e imagens. Essas ferramentas simplificam a formatação e a colaboração em documentos compartilhados.

Quando se trata de apresentações, ferramentas como Google Apresentações e PowerPoint são ideais para criar slides visualmente atraentes. Além disso, você pode criar gráficos e usar lógica de programação em planilhas para apresentações mais dinâmicas.

O compartilhamento de documentos online é uma prática valiosa. Ele permite a colaboração em tempo real e a manutenção de históricos de versões para rastrear alterações feitas por diferentes colaboradores. Isso aumenta a produtividade e a eficácia das equipes de trabalho.

Pesquisas online são comuns na coleta de dados e opiniões. O Google Forms é uma ferramenta útil para criar formulários de pesquisa e coletar informações de forma organizada e eficiente.

Para a edição de vídeos, existem softwares como o Vegas, que oferecem recursos avançados, incluindo edição, inserção de filtros, recorte e adição de texto. Isso é essencial para criar vídeos de alta qualidade.

Redes Sociais

As redes sociais desempenham um papel significativo na vida moderna, com uma variedade de objetivos que vão além do simples compartilhamento de momentos pessoais. Elas se tornaram plataformas versáteis que abrangem aspectos pessoais, profissionais e comerciais.

No âmbito pessoal, as redes sociais proporcionam um espaço para indivíduos compartilharem suas experiências, fotos, pensamentos e conexões com amigos e familiares. É um ambiente onde a expressão pessoal e a manutenção de relacionamentos são fundamentais.

Do ponto de vista profissional, as redes sociais se transformaram em ferramentas poderosas para networking, recrutamento e desenvolvimento de carreira. Profissionais podem criar perfis detalhados que destacam suas habilidades e experiências, conectando-se com colegas de trabalho e potenciais empregadores.

Além disso, as redes sociais desempenham um papel crucial no cenário comercial. Empresas utilizam essas plataformas para promover seus produtos, alcançar públicos-alvo e construir relacionamentos com clientes. A publicidade direcionada é uma das principais estratégias comerciais nas redes sociais, aproveitando os dados de perfil dos usuários.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Para alcançar com sucesso esses objetivos, é fundamental materializar as redes sociais em sites e aplicativos eficazes. Plataformas como Facebook, Twitter e WhatsApp se destacam como os líderes populares nesse espaço. Cada uma delas oferece recursos exclusivos e oportunidades de interação, atendendo a diferentes necessidades e preferências dos usuários.

No entanto, o uso das redes sociais requer certos cuidados para maximizar sua utilidade e segurança. A manutenção regular e a atualização das contas são essenciais para garantir que as informações e conexões estejam sempre atualizadas. Compartilhar conteúdo de forma adequada é vital, evitando informações sensíveis e respeitando a privacidade dos outros.

A segurança nas redes sociais também é um ponto crítico. É importante aceitar conexões apenas de pessoas conhecidas e confiáveis, evitando interações com desconhecidos que possam representar riscos. A configuração do perfil, incluindo nome, foto e informações pessoais, deve ser cuidadosamente considerada para equilibrar a exposição pessoal com a privacidade.

Em relação à segurança na Internet em geral, os usuários devem estar atentos a potenciais ameaças, como acesso a conteúdo impróprio e compras online inseguras. A prevenção contra o furto de dados e identidade é fundamental, bem como a consciência sobre plágio e o uso apropriado de referências ao utilizar informações da web.

A disseminação de fake news é outra preocupação importante. Os usuários devem verificar a veracidade das informações antes de compartilhá-las, contribuindo para a promoção de uma comunicação precisa e confiável.

Internet das Coisas

A Internet das Coisas (IoT) representa uma revolução tecnológica em que objetos cotidianos se conectam à internet. Geladeiras, carros e diversos outros dispositivos estão interligados, permitindo automação e controle remoto de várias tarefas.

Sensores desempenham um papel fundamental na IoT, fornecendo dados para a tomada de decisões e melhorando a eficiência em casas inteligentes e cidades inteligentes. O uso de placas como Arduino tornou acessível a criação de dispositivos IoT personalizados.

Big Data e Aprendizado de Máquina

O Big Data, por sua vez, refere-se à análise de grandes volumes de dados que são coletados na IoT e em outras fontes. Os cinco "Vs" do Big Data (volume, variedade, velocidade, veracidade e valor) destacam os desafios e oportunidades associados a esses dados massivos. A inteligência artificial (IA) é utilizada para analisar esses dados, identificar padrões e melhorar a tomada de decisões em várias áreas, como correção ortográfica, diagnóstico de imagem e jogos.

O Aprendizado de Máquina é uma subcategoria da IA que envolve sistemas que podem aprender e se adaptar a partir de dados. Essa tecnologia é aplicada em reconhecimento de padrões, robótica, segurança e prevenção de fraudes. Os carros autônomos são um exemplo notável de aplicação de IA, onde os veículos podem operar sem intervenção humana, com base na aprendizagem de máquina e na análise de dados em tempo real.

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Semana 6 - Pensamento computacional na Educação Básica

O Pensamento Computacional na Educação Básica é um tema que ganha cada vez mais relevância no contexto educacional atual. Essa abordagem visa preparar os estudantes para enfrentar os desafios de um mundo cada vez mais digital e tecnológico.

A primeira questão a ser abordada é a necessidade premente de incluir o Pensamento Computacional na educação básica. Vivemos em uma era em que a tecnologia permeia todos os aspectos da nossa vida, e é crucial que os alunos adquiram habilidades relacionadas à computação desde cedo.

Dois conceitos que fundamentam o Pensamento Computacional são a lógica e a abstração. A lógica permite aos estudantes desenvolverem raciocínio crítico, enquanto a abstração os capacita a simplificar problemas complexos em termos mais simples e compreensíveis

Apesar da importância do Pensamento Computacional, muitos educadores enfrentam dificuldades ao tentar aplicá-lo na prática. A falta de recursos, treinamento adequado e materiais didáticos apropriados podem ser obstáculos significativos.

Para superar essas dificuldades, é fundamental a criação de diretrizes claras que orientem escolas e professores na implementação do Pensamento Computacional no currículo. Essas diretrizes devem fornecer orientação sobre conteúdo, metodologias de ensino e avaliação.

O Pensamento Computacional envolve diversas competências, incluindo:

- Conhecimento: Compreensão dos princípios básicos da computação.
- Habilidades Práticas: Capacidade de aplicar o conhecimento computacional na resolução de problemas do mundo real.
- Habilidades Cognitivas: Desenvolvimento do raciocínio lógico e abstrato.
- Competências Socioemocionais: Trabalho em equipe, comunicação e pensamento crítico.

Os algoritmos desempenham um papel fundamental no Pensamento Computacional, pois são sequências de passos lógicos usados para resolver problemas. Eles podem ser considerados os pilares desse pensamento, uma vez que capacitam os alunos a decompor problemas complexos em passos menores e mais gerenciáveis.

Importância da Tecnologia na Educação Básica:

A tecnologia é uma ferramenta poderosa para melhorar o processo de ensino-aprendizagem. Ela oferece a oportunidade de acesso a recursos educacionais globais, interatividade e personalização do aprendizado.

No entanto, é crucial que o conteúdo tecnológico seja apresentado de forma concisa e adaptada à faixa etária dos estudantes. Isso evita sobrecarregar os alunos com informações complexas demais para sua compreensão.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

O computador não deve ser apenas uma ferramenta de entretenimento, mas também um instrumento de aprendizado. Isso envolve o uso de aplicativos e programas educacionais que podem auxiliar no desenvolvimento do Pensamento Computacional.

Aplicação na Educação Básica:

O Pensamento Computacional pode ser aplicado em diversas áreas do conhecimento, desde matemática e ciências até humanidades e artes. Ele oferece uma abordagem multidisciplinar para a solução de problemas.

Relação com o Aumento do Poder Político:

O uso eficaz da tecnologia e do Pensamento Computacional pode aumentar a participação cívica e o poder político dos cidadãos, capacitando-os a compreender e influenciar questões relacionadas à tecnologia e à sociedade.

Currículo e Formação Humana Integral

O currículo educacional deve ser direcionado não apenas para o desenvolvimento acadêmico, mas também para a formação humana integral. Isso inclui a definição de 10 competências gerais que abrangem conhecimento, conceitos, procedimentos, habilidades práticas, habilidades cognitivas, socioemocionais, atitudes e valores.

Aplicação do Pensamento Computacional

O Pensamento Computacional pode ser aplicado em diversas áreas do conhecimento, e o uso de algoritmos na resolução de problemas é uma abordagem eficaz. Exemplos de aplicação incluem a resolução de problemas em ciência, tecnologia, engenharia e matemática.

Futuro do Pensamento Computacional

No futuro, é provável que atividades cognitivas sejam cada vez mais automatizadas, o que torna o Pensamento Computacional uma habilidade essencial. Ensinar essa habilidade desde a educação básica é fundamental para preparar os alunos para os desafios do mercado de trabalho do século XXI.

Além disso, o Pensamento Computacional não se limita apenas ao uso de tecnologia, mas também desenvolve habilidades importantes, como trabalho em equipe, criatividade e resolução de problemas.

Ciência da Computação: Reconhecimento de Soluções e Resolução de Problemas

A ciência da computação não se limita apenas a aprender a programar; ela envolve o reconhecimento de problemas complexos e a aplicação de soluções por meio do pensamento algorítmico. Os alunos devem ser capazes de identificar problemas e abordá-los com uma mentalidade computacional.

Resolver e Analisar Problemas: Aplicação de Projetos

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

A capacidade de resolver problemas é uma habilidade essencial. Através da aplicação de projetos, os alunos podem desenvolver suas habilidades de resolução de problemas, identificando questões, planejando soluções e executando projetos que envolvam a computação.

Compreender e Transformar o Mundo: Processo na Resolução dos Problemas, Mundo Digital, Máquinas Computam

Compreender como a computação afeta o mundo ao nosso redor é fundamental. Isso inclui a compreensão do funcionamento do mundo digital e como as máquinas processam informações. Os alunos devem ser capazes de relacionar a computação com os desafios da vida cotidiana.

Representação: Técnica de Resolução de Problemas, Criação de Modelos Computacionais

A representação envolve a capacidade de traduzir problemas do mundo real em termos que podem ser manipulados por computadores. Isso inclui a criação de modelos computacionais que permitem simular situações e testar soluções antes da implementação real.

Avaliação Crítica: Identificação de Soluções e Propósitos

Os alunos devem desenvolver habilidades de avaliação crítica para discernir a qualidade das soluções computacionais e entender os propósitos por trás delas. Isso inclui a capacidade de analisar a ética e a moral no mundo digital.

Para adquirir essas competências, os alunos precisam de conhecimentos sólidos, incluindo a compreensão da computação, algoritmos, programação, segurança digital e tecnologia em geral. Além disso, devem utilizar a tecnologia como ferramenta para coletar informações e solucionar problemas, promovendo a análise crítica e a argumentação sobre questões tecnológicas e éticas.

Os eixos principais na educação que devem ser abordados são o pensamento computacional, a cultura digital e a tecnologia. Esses elementos constituem a base para a integração da computação na educação.

Os métodos de ensino podem variar, mas uma abordagem transversal, que integre conceitos de computação em várias disciplinas, pode ser eficaz. Além disso, a criação de componentes curriculares específicos ou o uso de atividades práticas podem ser úteis. É crucial envolver a comunidade escolar na elaboração e implementação dessas estratégias, garantindo uma abordagem holística e alinhada com os objetivos educacionais.

Semana 7 - Resolução de problemas

Problema do Elevador Lento

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

No cenário de um prédio de escritórios, um problema comum surgiu: a lentidão do elevador. Os frequentes relatos de reclamações por parte dos usuários tornaram esse dilema uma questão crítica a ser resolvida. Diante desse desafio, várias possíveis soluções foram consideradas.

Uma das abordagens discutidas foi a substituição do elevador atual por um modelo mais moderno e eficiente, equipado com um motor mais forte e rápido. Essa opção, embora promissora, também levantou preocupações sobre os custos e a viabilidade da implementação.

Outra solução considerada envolveu a atualização do algoritmo responsável pelo funcionamento do elevador. A ideia era tornar o sistema mais inteligente, capaz de otimizar a alocação de recursos e reduzir o tempo de espera dos passageiros.

No entanto, uma das propostas mais interessantes e inovadoras foi a instalação de espelhos no interior do elevador. Essa ideia se baseia na psicologia humana, pois muitos indivíduos tendem a se distrair ao olhar para algo interessante, como seus reflexos nos espelhos. Essa solução não apenas oferece uma experiência mais agradável aos passageiros, mas também ajuda a mascarar a percepção do tempo, fazendo com que as pessoas sintam que o elevador está se movendo mais rápido do que realmente está.

Resolução de Problemas

A resolução de problemas é uma habilidade intrinsecamente humana e tem uma longa história como parte integrante do nosso desenvolvimento como espécie. Ao longo dos séculos, a humanidade aprimorou essa capacidade de enfrentar desafios e encontrar soluções eficazes para uma ampla variedade de problemas.

Uma das abordagens históricas para a resolução de problemas é o método cartesiano, proposto por René Descartes no século XVII. Esse método enfatiza a importância de dividir problemas complexos em partes menores e mais gerenciáveis. Ele estabeleceu quatro etapas essenciais para a resolução de problemas:

- Compreensão Do Problema
- Planejamento
- Execução
- Retrocesso
-

Pensamento Computacional

O pensamento computacional é uma abordagem moderna para a resolução de problemas, que se baseia em princípios da ciência da computação. Ele se apoia em quatro pilares fundamentais:

- Abstração
- Reconhecimento De Padrões
- Algoritmos
- Decomposição

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

A abstração é a capacidade de simplificar a representação de problemas complexos, concentrando-se apenas nos aspectos essenciais. Isso permite que um problema seja tratado de forma mais eficiente.

O reconhecimento de padrões envolve a identificação de tendências, regularidades e características que podem ser aplicadas para solucionar problemas.

A criação de algoritmos é a formulação de sequências de passos que, quando seguidos, levam à resolução de um problema específico. Esses algoritmos podem incluir decisões e estruturas de repetição para lidar com várias situações.

Técnicas de Resolução de Problemas

Dentro do contexto da resolução de problemas, várias técnicas desempenham um papel importante. Isso inclui a decomposição, que envolve a divisão de um problema maior em problemas menores e mais gerenciáveis. A recursão, por sua vez, permite que uma solução seja aplicada repetidamente em problemas semelhantes.

A generalização é a capacidade de criar soluções genéricas que podem ser aplicadas a diferentes problemas, enquanto a transformação envolve a adaptação de soluções existentes para resolver novos desafios.

É fundamental destacar a importância da avaliação das soluções propostas. A eficácia de uma solução deve ser cuidadosamente examinada para garantir que ela atenda aos critérios de sucesso e resolva o problema de maneira satisfatória.

Paralelismo e Processamento Paralelo

Imagine que você está preparando um jantar com várias etapas, como cozimento de arroz, legumes e carne. Em vez de cozinhar cada item sequencialmente, você decide usar todas as bocas do fogão ao mesmo tempo. Enquanto o arroz está cozinhando em uma panela, você pode preparar os legumes na outra e a carne na terceira. Essa abordagem paralela permite que você conclua o jantar muito mais rapidamente, pois várias tarefas estão sendo executadas simultaneamente.

Da mesma forma, na computação, podemos utilizar múltiplos processadores ou núcleos de CPU para resolver tarefas em paralelo. Isso é especialmente útil para lidar com problemas computacionais intensivos, como simulações complexas, processamento de grandes volumes de dados e renderização de gráficos em jogos. O uso eficiente do paralelismo pode levar a um aumento significativo na velocidade e na eficiência da resolução de problemas.

Recursividade

A recursividade é uma técnica fundamental na resolução de problemas complexos. Ela envolve a capacidade de um algoritmo chamar a si mesmo para resolver partes menores de um problema. Isso é semelhante a resolver um quebra-cabeça gigante dividindo-o em pequenas peças e resolvendo cada uma delas.

This document is available on studocu

Downloaded by vinicius rangel (viniciuslelele@gmail.com)

Um exemplo clássico de algoritmo recursivo é o cálculo do fatorial de um número. Para calcular o fatorial de um número "n", você pode usar a seguinte fórmula recursiva: fatorial(n) = n * fatorial(n-1), com uma condição de parada de fatorial(0) = 1. Neste caso, o algoritmo se chama a si mesmo repetidamente, reduzindo o problema original em partes menores até atingir a condição de parada.

A recursividade é particularmente útil quando você está lidando com problemas que podem ser subdivididos em partes menores e idênticas ao problema original. No entanto, é importante definir uma condição de parada clara para evitar que o algoritmo entre em um loop infinito.

Torre de Hanói

Um desafio interessante que envolve recursividade é a Torre de Hanói, um quebra-cabeça que requer o cálculo do número mínimo de movimentos para transferir um conjunto de discos de um pino para outro, seguindo regras específicas. O desafio é calcular quantos movimentos são necessários para resolver a Torre de Hanói com um determinado número de discos.

A solução desse problema requer a aplicação da recursividade, uma vez que cada movimento pode ser visto como uma instância menor do problema original. A demonstração do processo de resolução envolve a identificação de padrões e a subdivisão do problema em partes menores, eventualmente chegando à solução final.

Downloaded by vinicius rangel (viniciuslelele@gmail.com)