

Computação na Educação Básica



Leila Ribeiro

UFRGS

Diretora de Ensino de Computação na Educação Básica/SBC

leila@inf.ufrgs.br

Rozelma Soares de França

UFRPE

rozelma.franca@ufrpe.br

Thaís Batista

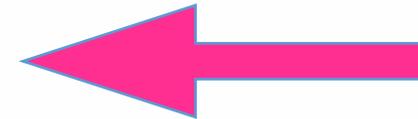
UFRN

thais.batista@ufrn.br



Organização do minicurso

- Dia 1: Por que ensinar Computação na Educação Básica?
- Dia 2: Atividades desplugadas
- Dia 3: Mundo digital e cultura digital
- **Dia 4: Pensamento computacional e PNED**



Eixo Pensamento Computacional

O **Pensamento Computacional** está diretamente ligado com a **compreensão**, **definição**, **modelagem** e, principalmente, **solução de problemas**. Esta se dá por meio da **construção de algoritmos**, envolvendo **abstrações** e **técnicas** necessárias para a **descrição** e **análise** de **dados** e **processos**, assim como para a **automação** de **soluções** de qualquer tipo.



Pensamento Computacional

■ Pensamento Computacional está relacionado:

- À capacidade de **compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas e soluções** de forma **metódica e sistemática**, através da **construção de algoritmos**.
- Ao **intelecto humano**, juntamente com a **leitura**, a **escrita** e a **aritmética**, pois serve como forma de **descrever, explicar e modelar o universo** e seus **processos**.



Pensamento Computacional

■ Pensamento Computacional é importante para:

- Ajudar na construção de **confiança** e **persistência** ao lidar com **problemas amplos** e **complexos**, assim como para criar **tolerância** à **ambiguidades** e a desenvolver **habilidades** de **comunicação** e de **trabalho em grupo**.
- Desenvolver **disposições** e **atitudes essenciais** para os **profissionais** do **século XXI**, por meio de uma **mudança** de **paradigma** na forma como os **alunos enxergam problemas**, proporcionando **pensamento crítico**, **colaboração**, **flexibilidade**, **adaptabilidade** e **argumentação**.



Pensamento Computacional

Abstração

RESOLUÇÃO DE PROBLEMAS, REPRESENTAÇÃO DA INFORMAÇÃO E PROCESSOS: Compreender e utilizar modelos e representações adequadas para organizar informação e descrever processos. Usar diferentes técnicas (decomposição, generalização, transformação, reuso, recursão, refinamento) para construir soluções para problemas, identificando as vantagens do uso destas técnicas e as condições para utilizá-las, aperfeiçoando e articulando saberes escolares.

Automação

PROGRAMAS E DADOS: Identificar problemas de diversas áreas do conhecimento e criar soluções, de forma individual e colaborativa, usando computadores, celulares e /ou outras máquinas processadoras de instruções para resolver todo ou partes do problema. Criar modelos computacionais para simular e fazer previsões sobre diferentes fenômenos e processos.

Análise

ANÁLISE QUANTITATIVA E QUALITATIVA DE PROBLEMAS/SOLUÇÕES: Analisar criticamente os problemas e soluções para identificar não somente se existem soluções que podem ser automatizadas, mas também ser capaz de avaliar a eficiência e a correção destas soluções e de justificar tanto a adequação das soluções aos requisitos quanto as escolhas frente às diferentes possibilidades de soluções.

Ensino Fundamental I:

Objetos de conhecimento do Pensamento Computacional



1º ano	Organização de objetos		Conceito de algoritmo	
2º ano	Modelagem de objetos		Algoritmos com repetições simples	
3º ano		Decomposição	Lógica computacional	Algoritmos com repetições e condicionais
4º ano	Matrizes e registros			Algoritmos com repetições aninhadas
5º ano	Listas e grafos		Lógica computacional	Algoritmos com seleção

Ensino Fundamental II: Objetos de conhecimento do Pensamento Computacional



6º ano

Tipos de dados

Generalização

Decomposição

Linguagem de programação

7º ano

Propriedades de grafo

Programação com registros e matrizes

Reuso

Análise de programas

8º ano

Programação com listas e recursão

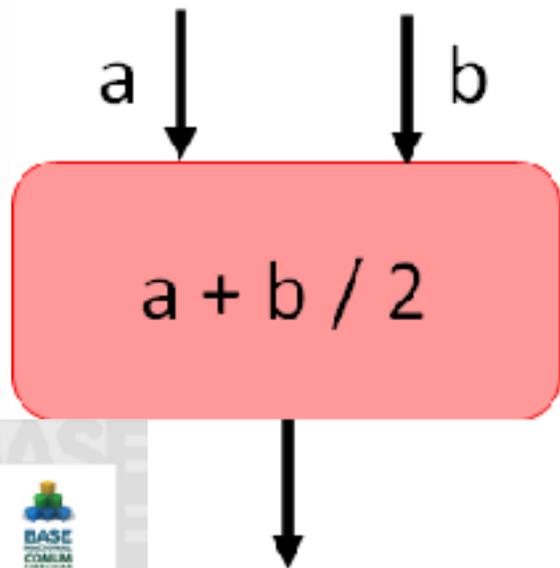
9º ano

Programação usando gratos e árvores

Autômatos e linguagens baseadas em eventos



Ano	Objetos de Conhecimento	Habilidades
6	Generalização	<p>Identificar os recursos ou insumos necessários (entradas) para a resolução de problemas, bem como os resultados esperados (saídas), determinando os respectivos tipos de dados, e estabelecendo a definição de problema como uma relação entre entrada e saída.</p> <p>Comparar diferentes casos particulares (instâncias) de um mesmo problema, identificando as semelhanças e diferenças entre eles, e criar um algoritmo para resolver todos, fazendo uso de variáveis (parâmetros) para permitir o tratamento de todos os casos de forma genérica.</p>



Idealmente, um **algoritmo** é uma **solução genérica**: ele resolve várias instâncias de um problema. Por exemplo, um algoritmo que calcula a média aritmética de 2 números resolve este problema para qualquer par de números (que são as instâncias do problema). Para descrever um algoritmo de forma genérica, é necessário dar nomes às entradas do algoritmo. Esses nomes são chamados de variáveis ou parâmetros do algoritmo.

Ano	Objetos de Conhecimento	Habilidades
6	Linguagem visual de programação	Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.
		Descrever com precisão a solução de um problema, construindo o programa que implementa a solução descrita.



MIT
APP INVENTOR



Blockly

SCRATCH



Existem inúmeras linguagens (várias delas visuais) que podem ser usadas para descrever algoritmos.



Ano	Objetos de Conhecimento	Habilidades
6	Linguagem visual de programação	Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

Se o ponteiro do mouse **tocar** no gatinho então o gato **andar** 10 passos 10 vezes seguidas.

Descrever com precisão a solução de um problema, construindo o programa que implementa a solução descrita.

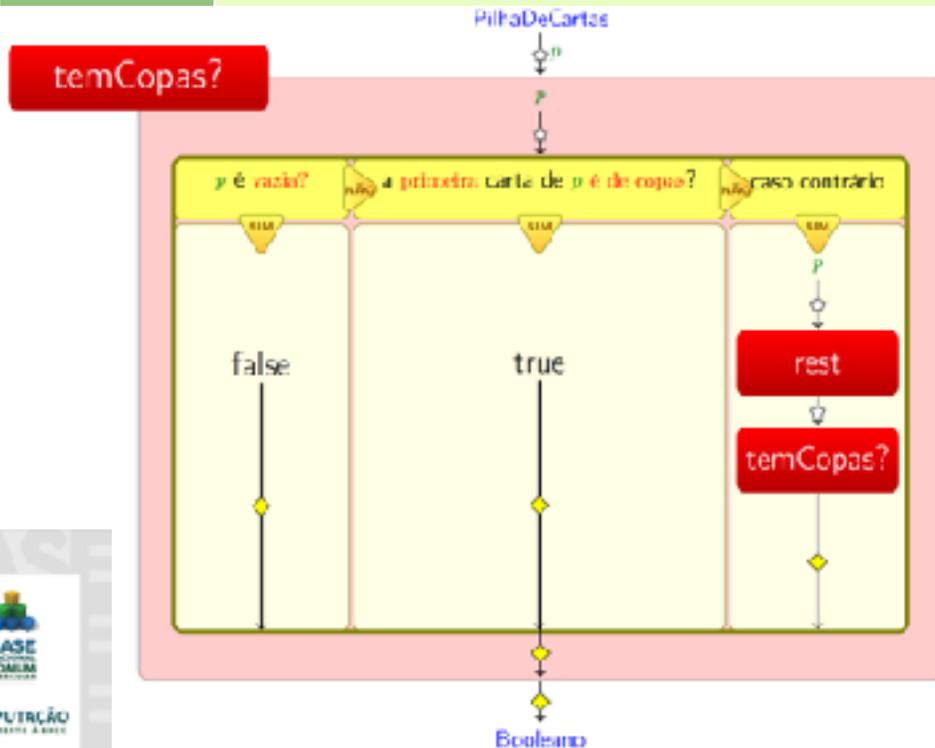
É importante que se consiga explicar em português o que os programas descritos nessas linguagens fazem.

```

se tocando em ponteiro do mouse ? então
  repita 10 vezes
    mova 10 passos
  
```



Ano	Objetos de Conhecimento	Habilidades
6	Linguagem visual de programação	<p>Elaborar algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.</p> <p>Descrever com precisão a solução de um problema, construindo o programa que implementa a solução descrita.</p>



Dada uma pilha de cartas p :

- Se a pilha p está vazia, devolver **false** (não tem carta de copas)
- Se o primeiro elemento da pilha p é de copas, devolver **true** (tem carta de copas)
- Senão, verificar se tem carta de copas no resto da pilha p

Ano	Objetos de Conhecimento	Habilidades
7	Análise de programas	Analisar programas para detectar e remover erros, ampliando a confiança na sua correção.



Imagem de DarkAthena por Pixabay

Deve-se estimular a análise crítica do programa construído. Uma das formas é através da depuração, que consiste em uma análise detalhada do código e realização de testes para identificar erros. Depuração é uma das formas de desenvolver a habilidade do pensamento crítico. A realização de testes é uma forma prática de exercitar o pensamento indutivo.

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las usando uma linguagem de programação, empregando ou não a recursão como uma técnica de resolver o problema.</p>

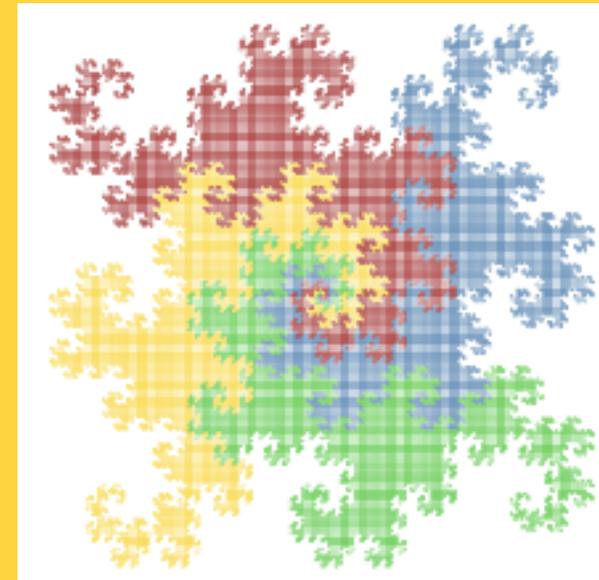
Recursão, em Computação, se refere a resolver um problema através da **decomposição** em (sub)problemas menores, onde pelo menos um dos subproblemas é do **mesmo tipo do problema original**. A ideia de recursão, porém, não se restringe à Computação...

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las</p>

Recursão em Artes Visuais:

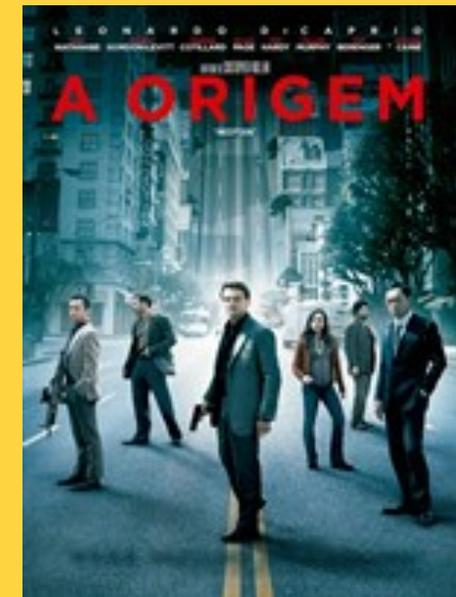
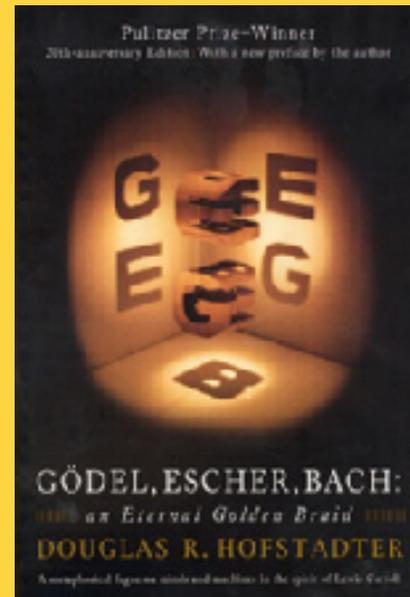


Imagem de Pedro Ribeiro Simões, CC BY 2.0, Flickr



Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las usando uma linguagem de programação, empregando ou não a</p>

Recursão em Literatura:



Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las</p>

Recursão na Natureza:



Imagem de Alexandre Duret-Lutz, CC BY-SA 2.0, Wikimedia Commons

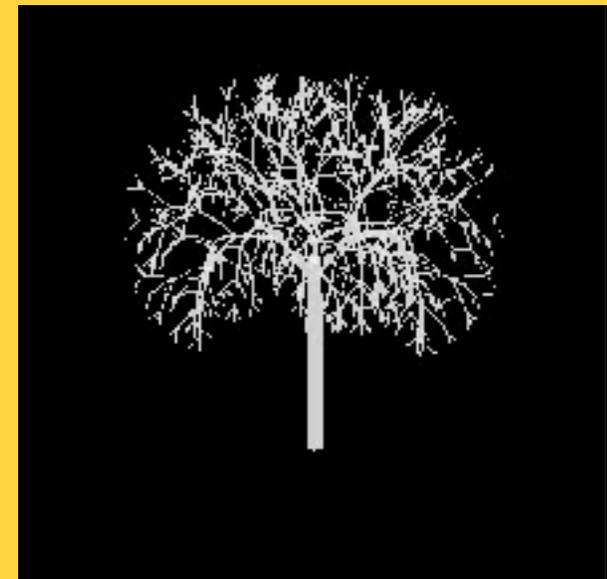
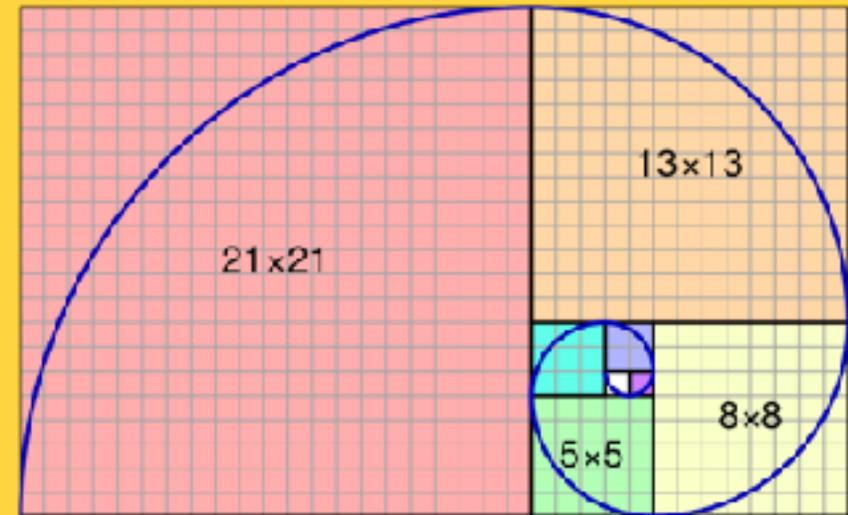


Imagem de Nevit Dilmen, CC BY-SA 3.0, via Wikimedia Commons

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las usando uma linguagem de programação, empregando ou não a</p>

Recursão na Matemática:

$$\begin{aligned}
 \text{fib } 0 &= 0 \\
 \text{fib } 1 &= 1 \\
 \text{fib } n &= \text{fib } (n-1) + \text{fib } (n-2)
 \end{aligned}$$



Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las</p>

Recursão na Matemática:

Algoritmos de Euclides para calcular o maior divisor comum (MDC) :

Sejam a e b dois números naturais tal que $a \geq b$:

$$MDC(a, b) = \begin{cases} b, & \text{se } b \text{ divide } a \\ MDC(b, r), & \text{caso contrário, onde } r \text{ é o resto da divisão inteira de } a \text{ por } b \end{cases}$$

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las usando uma linguagem de programação, empregando ou não a</p>

Recursão na Matemática:

Definição da função fatorial:

$$n! = \begin{cases} 1, & \text{se } n = 0 \\ n \times (n-1)!, & \text{se } n > 0 \end{cases}$$

Execução: fatorial de 3

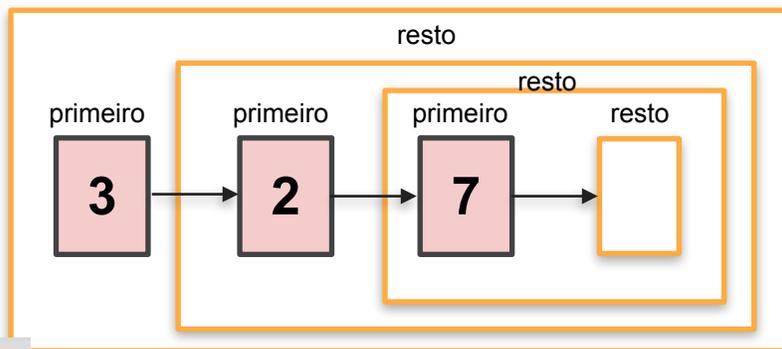
$$3! = \begin{cases} 1, & \text{se } 3 = 0 \\ 3 \times 2!, & \text{se } 3 > 0 \end{cases}$$

$$\begin{cases} 1, & \text{se } 2 = 0 \\ 2 \times 1!, & \text{se } 2 > 0 \end{cases} = 6$$

$$\begin{cases} 1, & \text{se } 1 = 0 \\ 1 \times 0!, & \text{se } 1 > 0 \end{cases} = 2$$

$$\begin{cases} 1, & \text{se } 0 = 0 \\ 1 \times 0!, & \text{se } 0 > 0 \end{cases} = 1$$

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las usando uma linguagem de programação, empregando ou não a recursão como uma técnica de resolver o problema.</p>



Representar **listas** usando as **construções de uma linguagem de programação**, e construir programas para manipular essas estruturas.

Ano	Objetos de Conhecimento	Habilidades
	Programação com listas e recursão	Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.

Exemplo: Ordenação de listas:

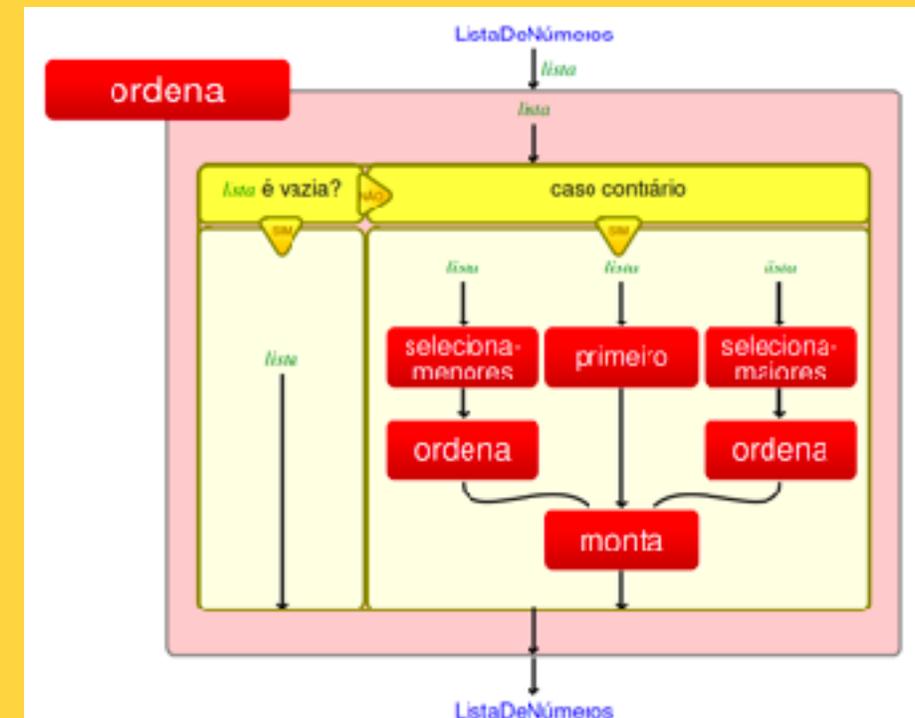
Linguagem natural:

Se a **lista** for vazia
 então devolver a própria **lista**
 senão **montar** uma lista contendo:
 e lista dos números **menores** que o primeiro, **ordenada**;
 e **primeiro** elemento; e
 e lista dos números **maiores** que o primeiro, **ordenada**.

Linguagem de programação (Racket):

```
(define (ordena lista)
  (cond
    [(vazia? lista) lista]
    [else (monta
            (ordena [seleciona-menores lista]);
            (primeiro lista); e
            (ordena [seleciona-maiores lista]))]))
```

Linguagem visual:



Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las usando uma linguagem de programação, empregando ou não a recursão como uma técnica de resolver o problema.</p>

Fazer **projetos** e construir **soluções** usando listas e **recursão**. É importante salientar a importância da análise crítica de programas recursivos identificando a existência de um caso base (fim) e de chamadas recursivas que fazem o programa convergir (se aproximar do fim) - caso contrário os programas podem não terminar.

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las.</p>

Um **baralho sem cartas repetidas** está dividido em **duas pilhas - A e B.**

Cada **pilha** está **ordenada**:

- ▶ Ordem dos **naipes** (mais importante): Ouros, espadas, copas e paus.
- ▶ Ordem dos **valores** (menos importante): A, 2, ..., 10, J, Q, K.

Como juntar as duas pilhas de forma que o baralho todo continue ordenado?



Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso</p>

Juntar pilhas de cartas: Dadas duas pilhas A e B



Linguagem natural:

Se a pilha A estiver vazia, retornar a pilhaB
 Se a primeira carta da pilha A for menor que a da pilha B
 colocar a primeira carta da pilha A na frente da
 junção do resto da pilha A com a pilha B
 Se a primeira carta da pilha A for maior que a da pilha B,
 colocar a primeira carta da pilha B na frente da
 junção da pilha A com o resto da pilha B

Linguagem de programação (Racket):

```
(define (junta-pilhas pilhaA pilhaB)
  (cond
    [(empty? pilhaA) pilhaB]
    [(menor (first pilhaA) (first pilhaB))
     (cons (first pilhaA)
           (junta-pilhas (rest pilhaA) pilhaB))]
    [(maior (first pilhaA) (first pilhaB))
     (cons (first pilhaB)
           (junta-pilhas pilhaA (rest pilhaB)))]))
```

Ano	Objetos de Conhecimento	Habilidades
8	Programação com listas e recursão	<p>Construir soluções de problemas usando a técnica de recursão e automatizar tais soluções usando uma linguagem de programação.</p> <p>Criar soluções de problemas para os quais seja adequado o uso de listas para descrever suas informações e automatizá-las</p>

Problema das Torres de Hanoi:

- ▶ Existem **3 pilares** nas Torres de Hanoi: inicial, intermediário e final.
- ▶ Existem **n discos ordenados** no **pilar inicial**, cada um com um **tamanho diferente** dos demais.
- ▶ Como mover os **n discos** para o **pilar final**?
- ▶ Não pode haver em momento algum um disco maior acima de outro menor!

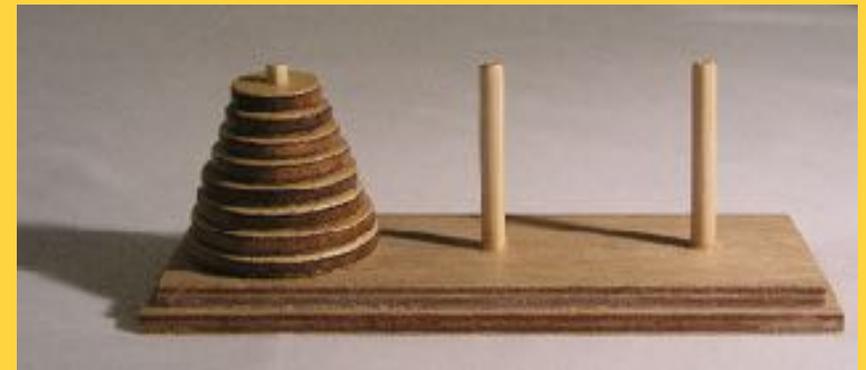
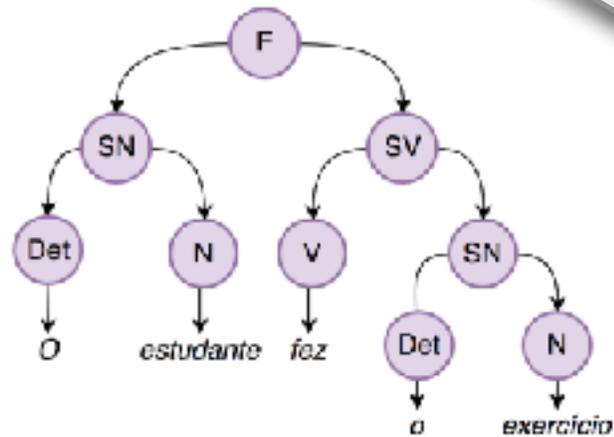
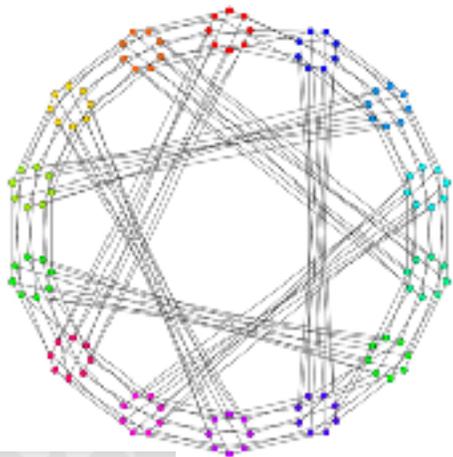


Imagem de Ævar Arnfrjó Bjarmason, CC BY 3.0, Wikipedia Commons

Ano	Objetos de Conhecimento	Habilidades
9	Estruturas de dados: grafos e árvores	Criar soluções de problemas para os quais seja adequado o uso de árvores e grafos para descrever suas informações e automatizá-las usando uma linguagem de programação.

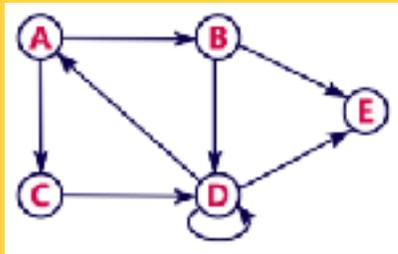


Grafos e árvores podem ser usados para **representar** uma gama enorme de **informações**. Para que possamos construir programas de computador, essas estruturas precisam ser formalizadas e descritas em linguagens de programação.

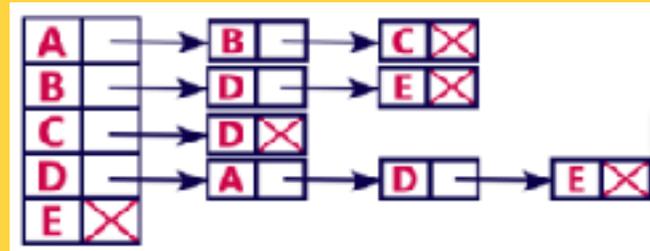
Ano	Objetos de Conhecimento	Habilidades
9	Estruturas de dados: grafos e árvores	Criar soluções de problemas para os quais seja adequado o uso de árvores e grafos para descrever suas informações e automatizá-las usando uma linguagem de programação.

Grafo:

Representação visual:



Formalização usando listas encadeadas:



Representação em linguagem Racket:

```
(list (make-nodo 'A (list 'B 'C))
      (make-nodo 'B (list 'D 'E))
      (make-nodo 'C (list 'D))
      (make-nodo 'D (list 'A 'D))
      (make-nodo 'B '()))
```

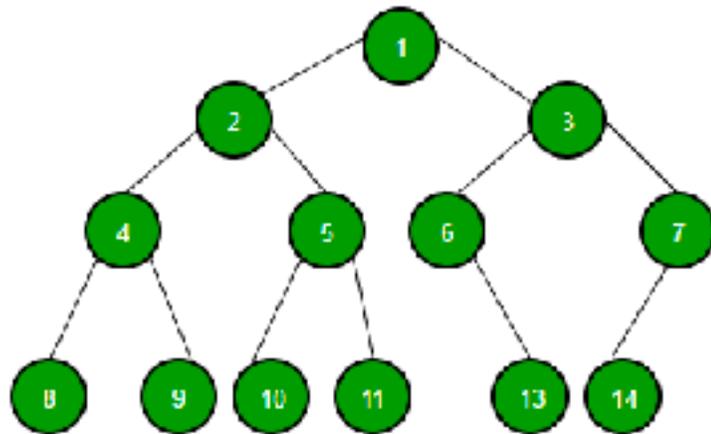
Formalização usando matriz de incidência:

	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	1	1
E	0	1	0	1	0

Representação em linguagem Python:

```
[ [ 0,1,1,1,0 ],
  [ 1,0,0,1,1 ],
  [ 1,0,0,1,0 ],
  [ 1,1,1,1,1 ],
```

Ano	Objetos de Conhecimento	Habilidades
9	Estruturas de dados: grafos e árvores	Criar soluções de problemas para os quais seja adequado o uso de árvores e grafos para descrever suas informações e automatizá-las usando uma linguagem de programação.



Uma **árvore** é um grafo com elementos organizados hierarquicamente. Há uma raiz, o elemento inicial, ligada a outros elementos, seus galhos ou filhos. Estes filhos levam a outros elementos que também podem possuir outros filhos. O elemento que não possui filhos é conhecido como folha ou nó terminal.

Ano	Objetos de Conhecimento	Habilidades
9	Estruturas de dados: grafos e árvores	Criar soluções de problemas para os quais seja adequado o uso de árvores e grafos para descrever suas informações e automatizá-las usando uma linguagem de programação.

Árvore:



Imagem de dandelion_tea - Pixabay

Dado

```
(define-struct pessoa (nome ano olhos pai mãe))
;; Um elemento do conjunto Pessoa pode ser
;; 1. empty, representando a falta de informação, ou
;; 2. (make-pessoa n a o p m)
;; onde:
;; n : String, representa o nome da pessoa
;; a : Número, representa o ano de nascimento da pessoa
;; o : String, representa a cor dos olhos da pessoa
;; p : Pessoa, representa o pai da pessoa
;; m : Pessoa, representa a mãe da pessoa
```

Programa

```
;; ancestralOlhosAzuis?: Pessoa -> Booleano.
""
(define (ancestralOlhosAzuis? p)
  (cond
    [(empty? p) falsa]
    [(string=? (pessoa-olhos p) "azuis") true]
    [else
     (or
      (ancestralOlhosAzuis? (pessoa-mae p))
      (ancestralOlhosAzuis? (pessoa-pai p)))]))
```

Ensino Fundamental II: Objetos de conhecimento do Pensamento Computacional

6º ano

Tipos de dados

Generalização

Decomposição

Linguagem de programação

7º ano

Propriedades de grafo

de programas

8º ano

9º ano

os e linguagens
s em eventos

Programação deve ser vista como **MEIO**
e não como **FiM!**

A programação provê uma forma muito efetiva de trabalhar abstração, análise e resolução de problemas **na prática!**

Ensino Médio: Competências e habilidades para a etapa



COMPUTAÇÃO - ENSINO MÉDIO



COMPETÊNCIAS

1. Considerar as possibilidades e os limites da computação para resolver problemas, tanto em termos de eficiência quanto de eficiência, propondo e analisando soluções computacionais para diversos domínios de conhecimento, considerando diferentes aspectos.
2. Analisar criticamente aspectos computacionais tendo como referência as possibilidades dos processos e dos recursos computacionais buscando garantir a integridade, privacidade, sigilo e segurança dos dados.
3. Analisar situações de mundo contemporâneo, selecionando fontes computacionais relevantes para a situação em análise.
4. Construir conhecimento usando teórico e baseado comparando, provando, criando e analisando de forma crítica, sem negligenciar questões éticas e legais, que possam ser relevantes para a sociedade.
5. Desenvolver projetos para investigar desafios do mundo contemporâneo, tendo em suas etapas de planejamento, execução, documentação e apresentação, utilizando diferentes ferramentas, linguagens e tecnologias de computação de forma flexível, crítica, ética, significativa, reflexiva e ética.
6. Promover e participar de eventos locais, utilizando a soluções computacionais utilizando diferentes softwares, ferramentas, linguagens e tecnologias de computação de forma flexível, crítica, ética, significativa, reflexiva e ética.
7. Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, reconhecendo aos conhecimentos da computação e suas tecnologias, tendo as condições de diversidade, segurança.

COMPETÊNCIA ESPECÍFICA	HABILIDADE	EXPLICAÇÃO DA HABILIDADE	EXEMPLOS	
	(EM13DC001) Explorar e construir a solução de problemas por meio da reutilização de partes de soluções existentes.	Considerando que esta habilidade já vem sendo trabalhada desde o Ensino Fundamental, nesta etapa serão trabalhados principalmente dois tópicos: (i) pelo lado mais prático, deve ser enfatizado o reuso de códigos de bibliotecas; (ii) pelo lado conceitual, deve-se trabalhar a construção de soluções através da decomposição (fragmentação) do problema e ser resolvido com outros problemas já solucionados, e assim reusando (e eventualmente adaptando) as soluções existentes.	Utilização de bibliotecas de código com ferramentas de programação textual, uso de Git/Lab para elaborar soluções colaborativamente.	
	(EM13DC002) Explorar e construir a solução de problemas por meio de refinamentos, utilizando diversos níveis de abstração desde a especificação até a implementação.	Aplicar boas práticas de Engenharia de Software, tanto para construir uma solução usando níveis de abstração diferentes partindo da definição dos requisitos, especificação, projeto e implementação (refinamento vertical), quanto para fazer a evolução do sistema partindo de um protótipo e tratando gradualmente as funcionalidades desejadas até chegar ao sistema completo (refinamento horizontal).	Representar problemas delimitados em conjunto com outras áreas, como na Biologia, e partir do código geral para níveis crescentes de detalhamento.	
	Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de validade quanto de eficiência, propondo e analisando soluções computacionais para diversos domínios de conhecimento, considerando diferentes aspectos.	(EM13DC003) Identificar o comportamento dos algoritmos no que diz respeito ao consumo de recursos como tempo de execução, espaço de memória e energia, entre outros.	Conhecer os princípios de complexidade de algoritmos, identificando as principais classes de funções que descrevem o consumo de recursos (tempo, espaço, energia) por algoritmos. Essas classes são caracterizadas por funções estudadas na Matemática no Ensino Médio (polinomial, logarítmica, exponencial). É importante compreender, por exemplo, que um algoritmo que executa em tempo polinomial é mais eficiente que um que executa em tempo exponencial. Esse tema é relevante na prática, pois várias técnicas de criptografia usadas em bancos, sistemas de eleição etc., se baseiam no fato de que determinadas problemas não têm solução algorítmica eficiente conhecida.	Testes de programas com soluções corretas, mas que geram tempo inaceitável de execução, ou utilizam memória em quantidade maior do que disponível na máquina. Exemplificar programas com tempo de execução exponencial, linear, quadrático e logarítmico. Mostrar esquemas de empilhamento recursivo que usam fatoração de números grandes (produto de dois números primos grandes), ou seja, com mais de 30 dígitos.
	(EM13DC004) Reconhecer o conceito de metaprogramação como uma forma de generalização na construção de programas, permitindo que algoritmos sejam entrada ou saída para outros algoritmos.	Podem-se construir programas que manipulam ou geram outros programas. Isto se chama metaprogramação. Com isto, se pode obter programas muito mais flexíveis, que podem ser utilizados em diversos contextos. Do ponto de vista técnico, o conceito é importante para analisarmos os limites do que pode ser resolvido com programas de computador (teoria da Computabilidade).	Construção de scripts em um sistema operacional capazes de gerar outros scripts de execução. Outro exemplo seria um programa que cria um outro programa (calcular tamanho, mostrar nome etc.) em vários arquivos de uma mesma pasta.	
	(EM13DC005) Identificar os limites da Computação para determinar o que pode ou não ser automatizado, buscando uma compreensão mais ampla dos limites dos processos mentais envolvidos na resolução de problemas.	Existem problemas que não têm solução computacional, o exemplo clássico é o Problema da Parada. A existência de uma solução para este problema levaria a um paradoxo, isto mostra que os computadores não são (e nunca serão) capazes de resolver qualquer tipo de problema. Vários problemas chamados não-Computáveis têm relação com programação, como o problema da parada de Turing, os problemas de decisão, etc.	Possui relação direta com (EM13MAT315). Mostrar paradoxos como o Paradoxo de Mentiroso, para explicar o que é um paradoxo e relacionar com o problema de Parada (que gera o paradoxo que um programa para se somente se ele próprio não para). Uma consequência da não existência de solução computacional para o problema da parada é que não é possível construir um programa que analise se um programa para ou não para.	
	(EM13DC006) Avaliar software levando em consideração diferentes características e métricas associadas.	A ideia é que os alunos possam realizar avaliação de software, através da adoção de características (eficiência, usabilidade, portabilidade, correção, segurança, privacidade, referência a ética, entre outras) e métricas associadas, embasando criticamente as suas escolhas, em contextos diversos de uso dessas ferramentas computacionais.	Dados os exemplos propostos por diferentes grupos de uma turma de estudantes do Ensino Médio, definir critérios relevantes, classifica-los em níveis de importância e avaliar os sistemas e fazer uma discussão crítica comparando os resultados das avaliações dos sistemas.	

Competência

Habilidade

Explicação

Exemplos



Ensino Médio: Competências



A ETAPA DO ENSINO MÉDIO



1. **Compreender as possibilidades e os limites da Computação para resolver problemas**, tanto em termos de viabilidade quanto de eficiência, **propondo e analisando soluções computacionais** para diversos domínios do conhecimento, considerando diferentes aspectos.

2. **Analisar criticamente artefatos computacionais**, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.

3. **Analisar** situações do mundo contemporâneo, **selecionando técnicas computacionais apropriadas para a solução de problemas**.

4. **Construir conhecimento usando técnicas e tecnologias computacionais**, produzindo conteúdos e artefatos de forma criativa, com respeito às questões éticas e legais, que proporcionem experiências para si e os demais.



Ensino Médio: Competências



A ETAPA DO ENSINO MÉDIO



5. **Desenvolver projetos** para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, **articulando conceitos, procedimentos e linguagens próprias da Computação** preferencialmente de maneira colaborativa.

6. **Expressar e partilhar** informações, ideias, sentimentos e soluções computacionais **utilizando diferentes plataformas, ferramentas, linguagens e tecnologias da Computação** de forma fluente, criativa, crítica, significativa, reflexiva e ética.

7. **Agir** pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, **recorrendo aos conhecimentos da Computação e suas tecnologias frente às questões de diferentes naturezas.**



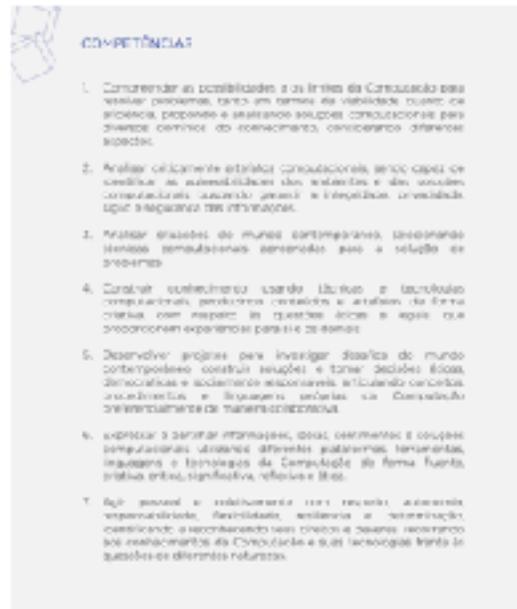
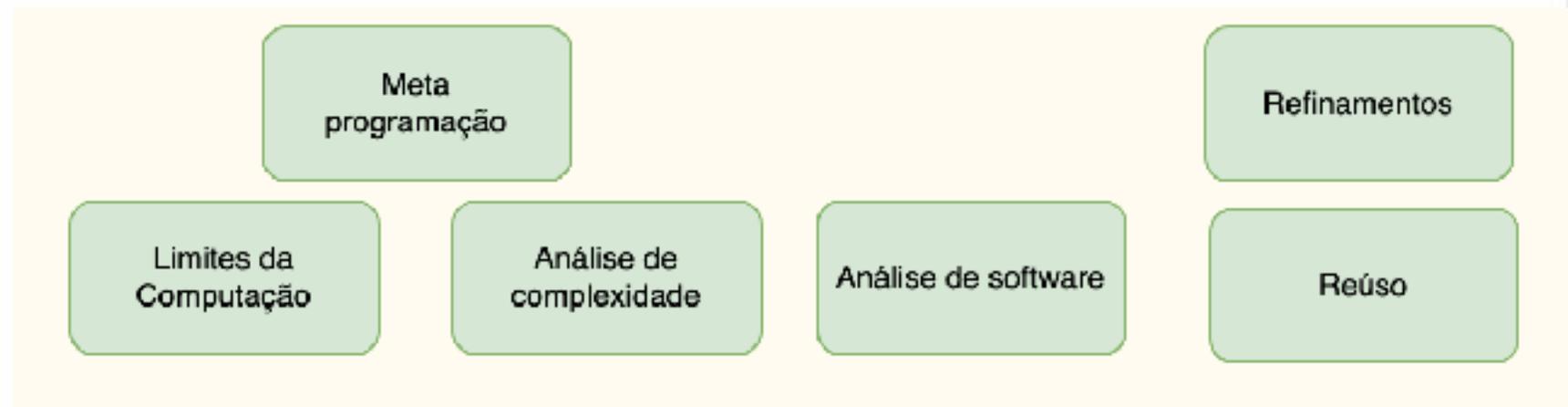
Ensino Médio: Competências e habilidades



A ETAPA DO ENSINO MÉDIO

1. **Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e analisando soluções computacionais** para diversos domínios do conhecimento, considerando diferentes aspectos.

Tópicos abordados nas habilidades:



Ensino Médio: Competências e habilidades

A ETAPA DO ENSINO MÉDIO

2. **Analisar criticamente artefatos computacionais**, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.

Tópicos abordados nas habilidades:

Análise de redes

Segurança cibernética

COMPETÊNCIAS

1. Compreender as habilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e avaliando soluções computacionais para diversos domínios do conhecimento, considerando diferentes aspectos.
2. Analisar criticamente artefatos computacionais, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.
3. Analisar desafios do mundo contemporâneo, identificando cenários tecnológicos associados para a solução de problemas.
4. Controlar o conhecimento usando técnicas e ferramentas computacionais, permitindo controlar e analisar de forma crítica com respeito às questões éticas e legais que proporcionam experiências para a sociedade.
5. Desenvolver projetos para investigar desafios do mundo contemporâneo controlando recursos e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos acadêmicos e tecnológicos relativos ao desenvolvimento orientado para a transferência.
6. Explorar e gerar informações, ideias, conceitos e soluções computacionais utilizando diferentes paradigmas tecnológicos, linguagem e técnicas da Computação de forma fluente, prática, crítica, significativa, reflexiva e ética.
7. Agir pessoal e coletivamente com recursos, autonomia, responsabilidade, flexibilidade, resiliência e determinação, avaliando e reconhecendo seus pontos e bases, reconstruindo os conhecimentos da Computação e suas técnicas frente às questões e diferentes situações.

Ensino Médio: Competências e habilidades

A ETAPA DO ENSINO MÉDIO

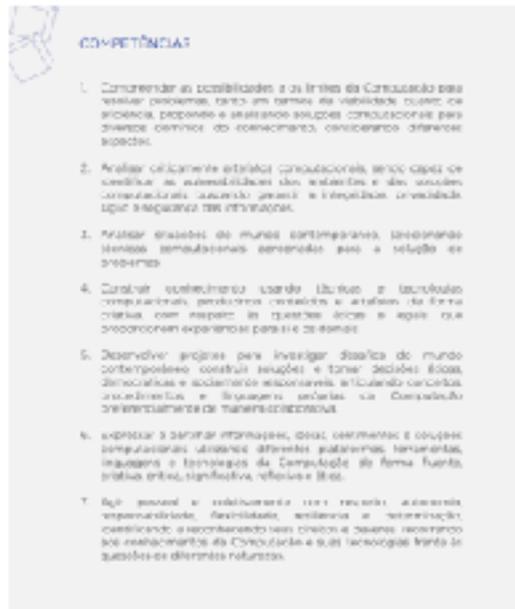
3. **Analisar** situações do mundo contemporâneo, **selecionando técnicas computacionais apropriadas para a solução de problemas.**

Tópicos abordados nas habilidades:

Inteligência artificial

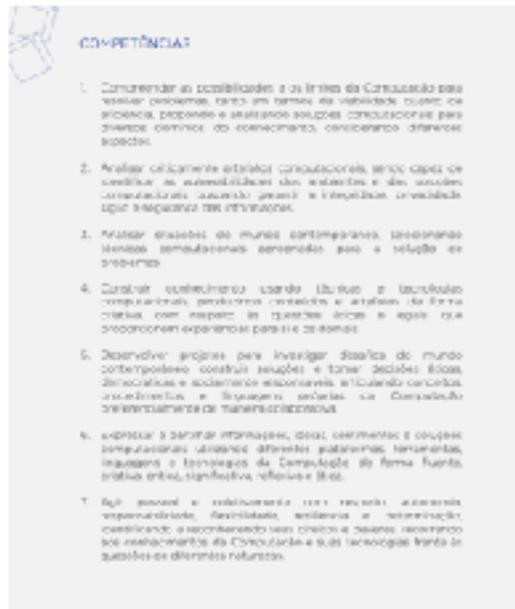
Modelos computacionais

Tecnologias digitais no mundo do trabalho



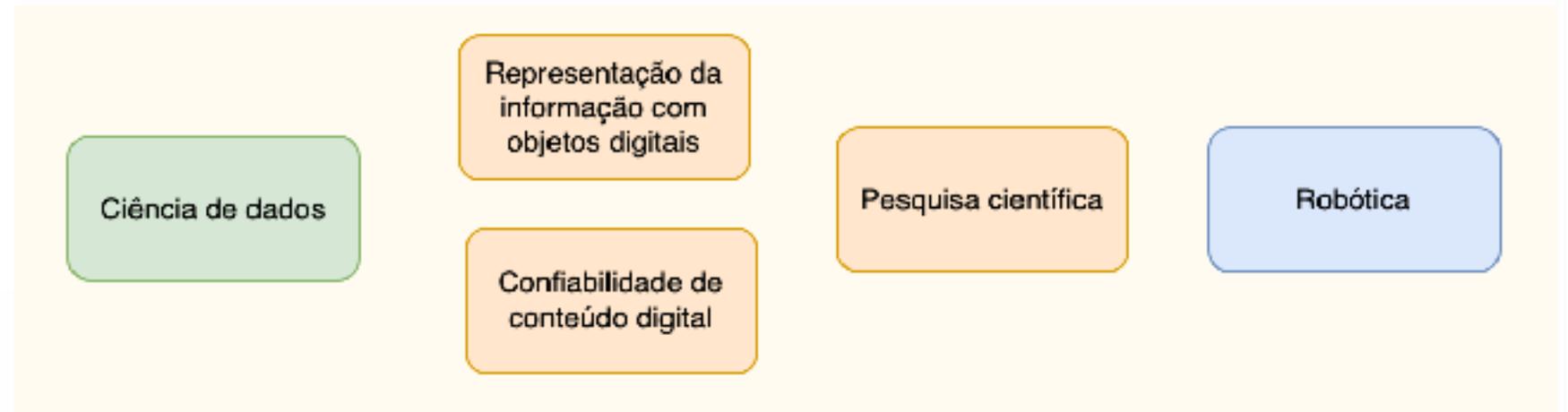
Ensino Médio: Competências e habilidades

A ETAPA DO ENSINO MÉDIO



4. Construir conhecimento usando técnicas e tecnologias computacionais, produzindo conteúdos e artefatos de forma criativa, com respeito às questões éticas e legais, que proporcionem experiências para si e os demais.

Tópicos abordados nas habilidades:

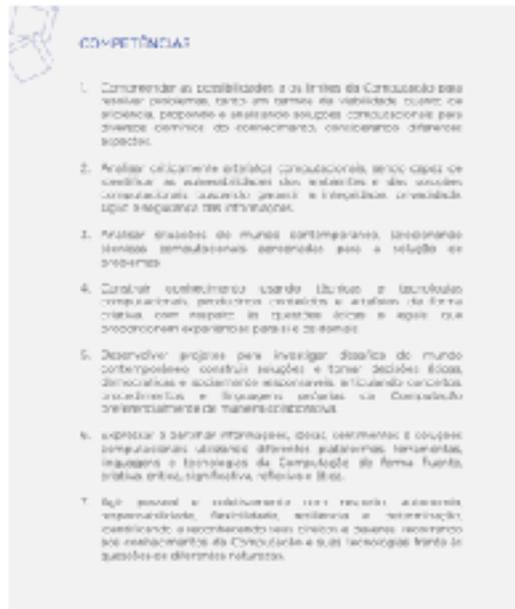


Ensino Médio: Competências e habilidades



A ETAPA DO ENSINO MÉDIO

5. **Desenvolver projetos** para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, **articulando conceitos, procedimentos e linguagens próprias da Computação** preferencialmente de maneira colaborativa.



Tópicos abordados nas habilidades:

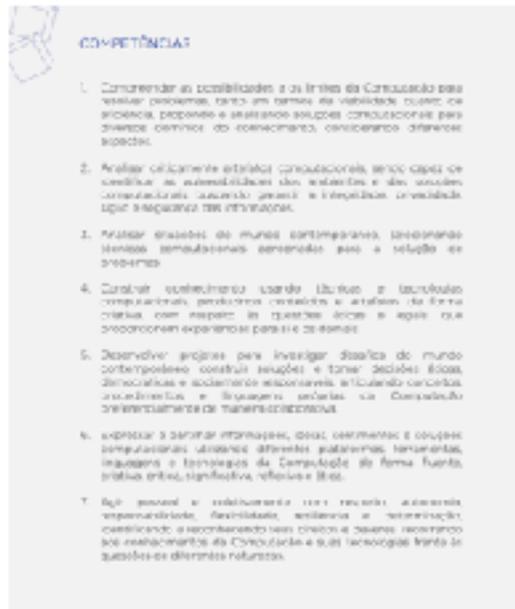
Redes virtuais e trabalho colaborativo

Projetos usando artefatos computacionais



Ensino Médio: Competências e habilidades

A ETAPA DO ENSINO MÉDIO



6. **Expressar e partilhar** informações, ideias, sentimentos e soluções computacionais **utilizando diferentes plataformas, ferramentas, linguagens e tecnologias da Computação** de forma fluente, criativa, crítica, significativa, reflexiva e ética.

Tópicos abordados nas habilidades:

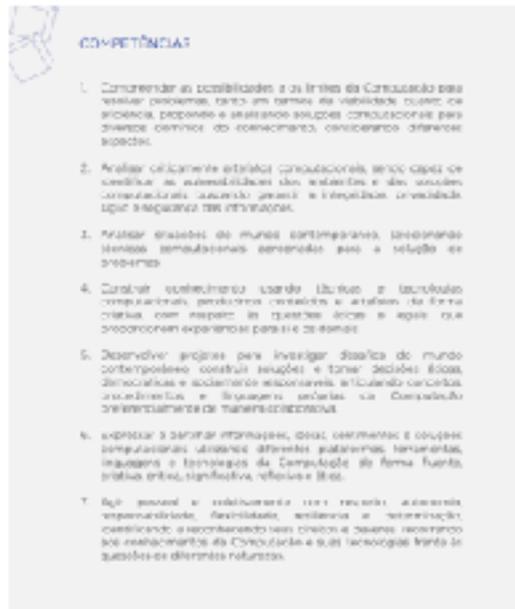
Uso de ferramentas digitais para exposição de ideias

Produção de conteúdo em mídias digitais

Criação e análise crítica de conteúdo digital

Ensino Médio: Competências e habilidades

A ETAPA DO ENSINO MÉDIO



7. **Agir** pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, **recorrendo aos conhecimentos da Computação e suas tecnologias frente às questões de diferentes naturezas.**

Tópicos abordados nas habilidades:

Comportamento em comunidades virtuais

Impactos de artefatos computacionais na vida moderna

Segurança e privacidade em ambientes virtuais

Direito digital

Desafios

- **Como entender o Anexo de Computação?**
- **Como formar os professores?**
- **Como adaptar currículos?**
- **Como implantar esses novos currículos?**
- **Como engajar os professores e as escolas no processo?**



A questão de infra-estrutura física (computadores, tablets, laboratórios, acesso à internet, etc) é relevante, mas bem menos complexa de solucionar do que os desafios listados acima...

Política Nacional de Educação Digital (PNED)

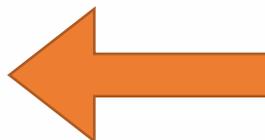


LEI N° 14.533, DE 11 DE JANEIRO DE 2023

Eixos estruturantes:

I - Inclusão Digital;

II - Educação Digital Escolar;



III - Capacitação e Especialização Digital;

IV - Pesquisa e Desenvolvimento (P&D) em Tecnologias da Informação e Comunicação (TICs).

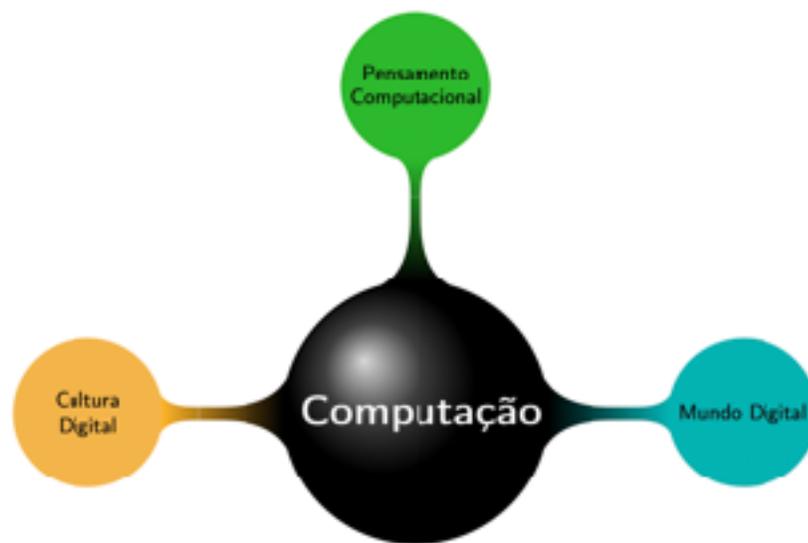
Política Nacional de Educação Digital (PNED)



LEI Nº 14.533, DE 11 DE JANEIRO DE 2023

Educação Digital Escolar: tem como objetivo garantir a inserção da educação digital nos ambientes escolares, em todos os níveis e modalidades, a partir do estímulo ao letramento digital e informacional e à aprendizagem de computação, de programação, de robótica e de outras competências digitais, englobando:

- I - Pensamento computacional
- II - Mundo digital
- III - Cultura digital;
- IV - Direitos digitais;
- V - Tecnologia assistiva.



Política Nacional de Educação Digital (PNED)

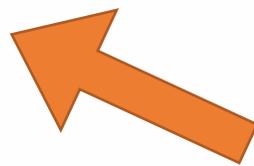


LEI Nº 14.533, DE 11 DE JANEIRO DE 2023

§ 1º Estratégias prioritárias da Educação Digital Escolar::

- Desenvolvimento de competências digitais na Educação Básica;
- Promoção de práticas pedagógicas para o domínio de lógica, algoritmos, ética digital, ...
- Incentivo a parcerias e acordos de colaboração;
- Promoção de formação inicial de professores de EB e ES com competências digitais
- ...

§ 2º O eixo Educação Digital Escolar deve estar em consonância com a base nacional comum curricular e com outras diretrizes curriculares específicas.



Política Nacional de Educação Digital (PNED)

LEI Nº 14.533, DE 11 DE JANEIRO DE 2023



§ 7º Os arts 4º e 26 da [Lei nº 9.394, de 20 de dezembro de 1996 \(Lei de Diretrizes e Bases da Educação Nacional\)](#), passam a vigorar com as seguintes alterações:

Art. 4º O dever do Estado com educação escolar pública será efetivado mediante a garantia de:

I - educação básica obrigatória e gratuita ...;

II - educação infantil gratuita ...;

...

XII - **educação digital**, com a garantia de conectividade de todas as instituições públicas de educação básica e superior à internet em alta velocidade, adequada para o uso pedagógico, com o desenvolvimento de competências voltadas ao letramento digital de jovens e adultos, criação de conteúdos digitais, comunicação e colaboração, segurança e resolução de problemas.

*Parágrafo único. Para efeitos do disposto no inciso XII do **caput** deste artigo, as relações entre o ensino e a aprendizagem digital deverão prever técnicas, ferramentas e recursos digitais que fortaleçam os papéis de docência e aprendizagem do professor e do aluno e que criem espaços coletivos de mútuo desenvolvimento.*

Política Nacional de Educação Digital (PNED)

LEI Nº 14.533, DE 11 DE JANEIRO DE 2023



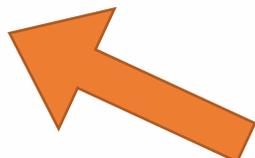
§ 7º Os arts 4º e 26 da [Lei nº 9.394, de 20 de dezembro de 1996](#) (Lei de Diretrizes e Bases da Educação Nacional), passam a vigorar com as seguintes alterações:

Art. 26. Os currículos da educação infantil, do ensino fundamental e do ensino médio devem ter base nacional comum, a ser complementada, em cada sistema de ensino e em cada estabelecimento escolar, por uma parte diversificada, exigida pelas características regionais e locais da sociedade, da cultura, da economia e dos educandos.

§ 1º Os currículos a que se refere o *caput* devem abranger, obrigatoriamente, o estudo da língua portuguesa e da matemática, o conhecimento do mundo físico e natural e da realidade social e política, especialmente do Brasil.

...

§ 11º A educação digital, com foco no letramento digital e no ensino de computação, programação, robótica e outras competências digitais, será componente curricular do ensino fundamental e do ensino médio.



Como inserir Computação na EB?



Quem sabe ensinar os fundamentos da Computação?

Quem sabe ensinar alunos de Educação Básica?

Professores de cursos de graduação em Computação

Licenciados em Computação

Professores de EB

A forma mais efetiva de enfrentar os desafios rapidamente e com qualidade seria através de **cooperação** entre esses grupos!

Governo

Empresas

SBC, ReLiC e ONGs

Escolas

Considerações finais



A norma aprovada foi uma construção coletiva, contém habilidades extremamente relevantes para a formação do cidadão do século XXI.

A homologação da resolução que insere Computação na BNCC foi um passo importante, mas é apenas o início...

Para ter sucesso, políticas públicas visando inclusão de Computação precisam envolver vários setores da sociedade!

A inserção de Computação nas escolas pode transformar a Educação no Brasil!

Computação na Educação Básica



Leila Ribeiro

UFRGS

Diretora de Ensino de Computação na Educação Básica/SBC

leila@inf.ufrgs.br

Rozelma Soares de França

UFRPE

rozelma.franca@ufrpe.br

Thaís Batista

UFRN

thais.batista@ufrn.br

