

NOME DA AULA:

Pensamento computacional

Tempo de aula: 45–60 minutos Tempo de preparação: 10 minutos

Objetivo principal: apresentar o modelo de “pensamento computacional” como uma forma de preparar problemas do mundo real para a representação digital.

RESUMO

Usando apenas papel e canetas hidrográficas, os alunos vão aprender as quatro etapas do pensamento computacional. Depois de uma breve introdução, os alunos deverão ser divididos em grupos nos quais terão que criar orientações para outros alunos desenharem um monstro específico (a partir de um catálogo de monstros pré-selecionados). A tarefa toda deve ser decomposta, então, as equipes vão analisar todos os monstros do catálogo procurando por padrões, abstrair detalhes semelhantes dos monstros e, por último, usar essas informações para criar um algoritmo (orientações) para que outra equipe desenhe um determinado monstro.

Então, as equipes vão trocar os algoritmos com outro grupo e desenhar o monstro usando como referência aquilo que o algoritmo em questão indica. O desenho foi feito como a equipe original pretendia?

OBJETIVO

Os alunos vão:

- Aprender as quatro etapas do pensamento computacional

- Trabalhar em equipe para solucionar problemas complexos

MATERIAIS

- Catálogo de monstros (1 por grupo)
- Folhas em branco (3 por pessoa)
- Canetas hidrográficas, canetas ou lápis (1 pacote por grupo)
- Tesouras

PREPARAÇÃO

Imprima imagens de monstros para cada grupo.

O rosto do monstro deve ser impresso em papel tradicional, e as “partes” do rosto dele devem ser impressas em transparências (os alunos podem desenhar as partes do rosto do monstro em papel vegetal ou essas partes podem ser impressas em papel tradicional e depois cortadas).

Organize imagens de monstros, tesouras e canetas hidrográficas para cada grupo.

Escreva as palavras do vocabulário na lousa ou projete-as com a câmera de documentos.

3 AULA 3: PENSAMENTO COMPUTACIONAL

VOCABULÁRIO

Pensamento computacional — Um método de resolução de problemas que ajuda cientistas da computação a preparar problemas para soluções digitais.

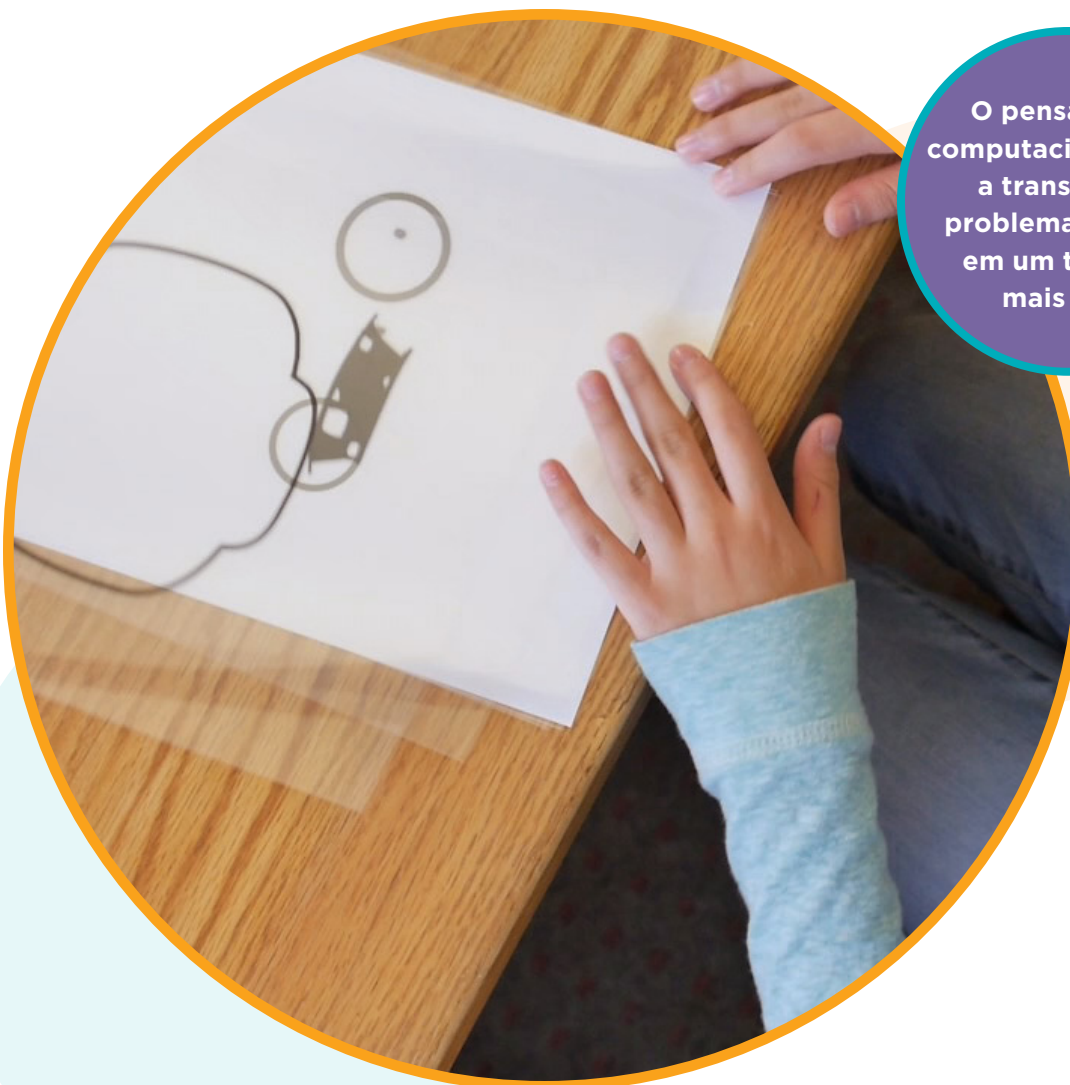
Abstração — Ação de ignorar os detalhes de uma solução de modo que ela possa ser válida para diversos problemas

Algoritmo — Uma lista de etapas que permitem que você complete uma tarefa

Decompor — Dividir um problema difícil em problemas menores e mais fáceis

Padrão — Um tema que se repete diversas vezes

Programa — Instruções que podem ser compreendidas e seguidas por uma máquina



O pensamento computacional ajuda a transformar problemas difíceis em um trabalho mais fácil.

REVISÃO

Esta seção de revisão tem como objetivo fazer a classe se lembrar do conteúdo da última aula. Se você está realizando essas atividades sem seguir a ordem correta, insira seus próprios tópicos a serem revisados aqui.

Perguntas para a participação da classe:

- O que significa “repetir” um “bloco” de código?
- Quando você deve usar um bloco “se”?

Discussão com o parceiro de equipe:

- Você consegue pensar em um motivo pelo qual gostaria de usar um bloco se/senão, em vez de usar apenas o “se”?

INTRODUÇÃO

Hoje, nossa classe vai se transformar em um grupo de elite de cientistas da computação. Cada um de nós vai trabalhar em uma missão do mundo real para obter um “algoritmo” que pode ser usado para “programar” nossa solução. Este é um breve resumo da missão (vocês precisam decidir se a aceitam):

Fomos selecionados para ajudar a identificar alguns monstros encontrados no planeta Zurônio. Precisamos descrever esses seres com base nas descrições que nós já temos de algumas testemunhas oculares. Há vários monstros para descrever, e isso pode parecer bem desafiador, mas darei algumas ferramentas para ajudá-los.

Para ter sucesso nessa operação, nós vamos praticar um método chamado Pensamento Computacional. O pensamento computacional se baseia em quatro etapas para ajudar a resolver vários tipos de problemas diferentes. Muitas pessoas acreditam que o pensamento computacional se resume a obter soluções prontas para serem executadas por uma máquina. Isso é verdade, mas ele também ajuda a transformar problemas difíceis em um trabalho mais fácil.

Frequentemente, os cientistas da computação descobrem que são responsáveis por programar soluções para coisas com as quais as pessoas sequer já sonharam — coisas que nunca foram criadas. Enfrentar um problema que nunca foi solucionado antes pode ser assustador, mas com essas simples ferramentas, tudo é possível.

Etapla 1) Decomposição — Nós não estamos falando de zumbis! Estamos falando de transformar um problema grande e difícil em algo muito mais simples. Geralmente, problemas grandes são apenas diversos problemas pequenos que foram unidos.

Etapla 2) Padrões — Normalmente, quando um problema tem muitas partes menores, você perceberá que essas partes têm algo em comum. Se não tiverem, elas poderão, pelo menos, ter algumas semelhanças evidentes em relação a algumas partes de outro problema solucionado anteriormente. Se conseguir identificar esses padrões, compreender as partes ficará muito mais fácil.

Etapla 3) Abstração — Depois de reconhecer um padrão, você poderá “abstrair” (ignorar) os detalhes que são responsáveis pelas diferenças e usar a estrutura geral para encontrar uma solução que seja válida para mais de um problema.

Etapla 4) Algoritmo — Quando sua solução estiver completa, você poderá escrevê-la de um modo que ela possa ser processada passo a passo, para que seja fácil atingir os resultados.

ATIVIDADE: CRIAR UM MONSTRO

Cada grupo receberá um catálogo de monstros. Cada monstro precisa ser classificado de acordo com as informações atribuídas a ele no catálogo. Primeiro, os alunos analisarão os monstros para ver o que todos eles têm em comum, então, eles vão abstrair as diferenças para criar uma lista de instruções que podem ser usadas por todos para recriar um dos monstros do catálogo... sem saber antes da hora certa qual deles está sendo feito!

Conseguir que alguém desenhe esses monstros sem vê-los pode ser muito desafiador. E se você analisar a tarefa toda, isso pode parecer até mesmo impossível. Por isso esse é o projeto perfeito para o pensamento computacional.

1) Decomposição — O que precisa ser feito para criar o monstro?

Dependendo do monstro, pode até parecer que ele já está em decomposição, mas agora precisamos “decompor” essa tarefa! Vamos dividi-la em tarefas menores, as quais parecerão mais fáceis individualmente. É melhor fazer isso com a classe toda ao mesmo tempo, para que todos acompanhem a mesma página.

Exemplo:

- Classificar os monstros pelo formato do rosto
- Procurar semelhanças nos monstros
- Fazer uma lista de características a serem identificadas
- Usar as características identificadas para criar um novo monstro
- Descrever seu novo monstro para seus colegas de classe seguindo um passo a passo, e deixá-los tentar reconstruí-lo.

2) Padrões — O que esses monstros têm em comum?

O que é que todos os monstros têm? Quais características são semelhantes entre os monstros de determinado grupo?

Exemplo:

- Todos os monstros têm cabeças
- O Zumbi Ranzinza tem uma boca Ranzinza

3) Abstração — O que é diferente? Vamos retirar o que é diferente.

Um monstro pode ter olhos Doidão, enquanto outro tem olhos Duende, mas ambos têm olhos. Isso significa que poderíamos dizer “Esse monstro tem olhos _____” e, mais tarde, preencher a lacuna de acordo com o monstro que estiver sendo desenhado.

Exemplo:

Crie uma lista com todas as características diferentes que os monstros têm, mas com os detalhes retirados da frase.

- O monstro tem uma cabeça _____.
- O monstro tem olhos _____.
- O monstro tem um nariz _____.
- O monstro tem orelhas _____.
- O monstro tem uma boca _____.

4) Algoritmo — Como você pode reunir todas essas informações para criar uma série de instruções que podem ser seguidas por seus colegas de classe?

Agora, cada grupo deverá organizar suas etapas em uma lista que será usada por todos os outros grupos para recriar um monstro.

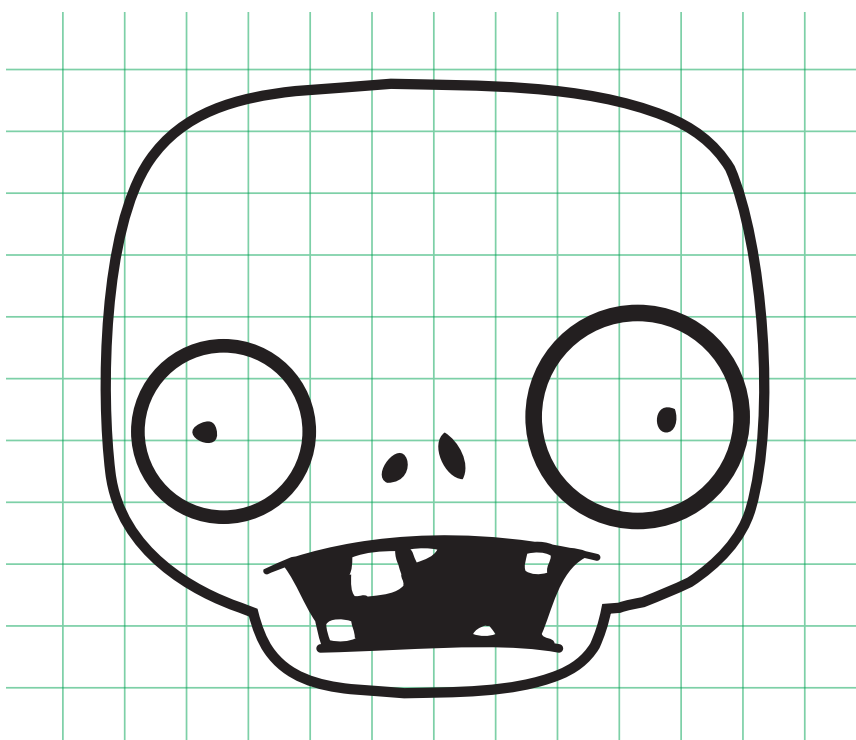
Exemplo:

Crie uma lista que possa preencher para seu artista.

- Desenhe uma cabeça *Doidão*.
- Desenhe olhos *Ranjinza*.
- Desenhe um nariz *Doidão*.
- Desenhe orelhas *Ocultus*.*
- Desenhe uma boca *Duende*.

**Ocultus signi fica que o monstro não tem a característica em questão.*

Os alunos devem testar seu algoritmo para verificar se ele realmente desenha a imagem correta. Em seguida, deixe os alunos trocarem de algoritmos com outro grupo e peça-os que desenhem o monstro com base no algoritmo recebido (sem ver a imagem original). Eles desenharam o monstro correto? Os alunos podem brincar com isso diversas vezes, pedindo para os colegas recriarem monstros que já existem ou descrevendo novos monstros.



AJUSTES:

Pré até 2º ano: Tente fazer esse exercício com a classe toda ao mesmo tempo, em vez de dividi-la em grupos. Usar transparências com as características impressas (olhos, orelhas, nariz) deixará tudo mais prático.

3º-5º ano: Auxilie os alunos no processo de decompor o problema e abstrair os detalhes. Talvez seja necessário fazer um exercício de exemplo com eles.

6º-8º ano: Permita que os alunos se esforcem por um tempo antes de oferecer sugestões. Deixe que “falhem” algumas vezes para que essas falhas forneçam o *feedback* de que eles precisam para que o exercício dê certo. Os alunos dessa faixa etária provavelmente gostarão ainda mais da atividade se desenharem as características em vez de sobrepor as transparências.

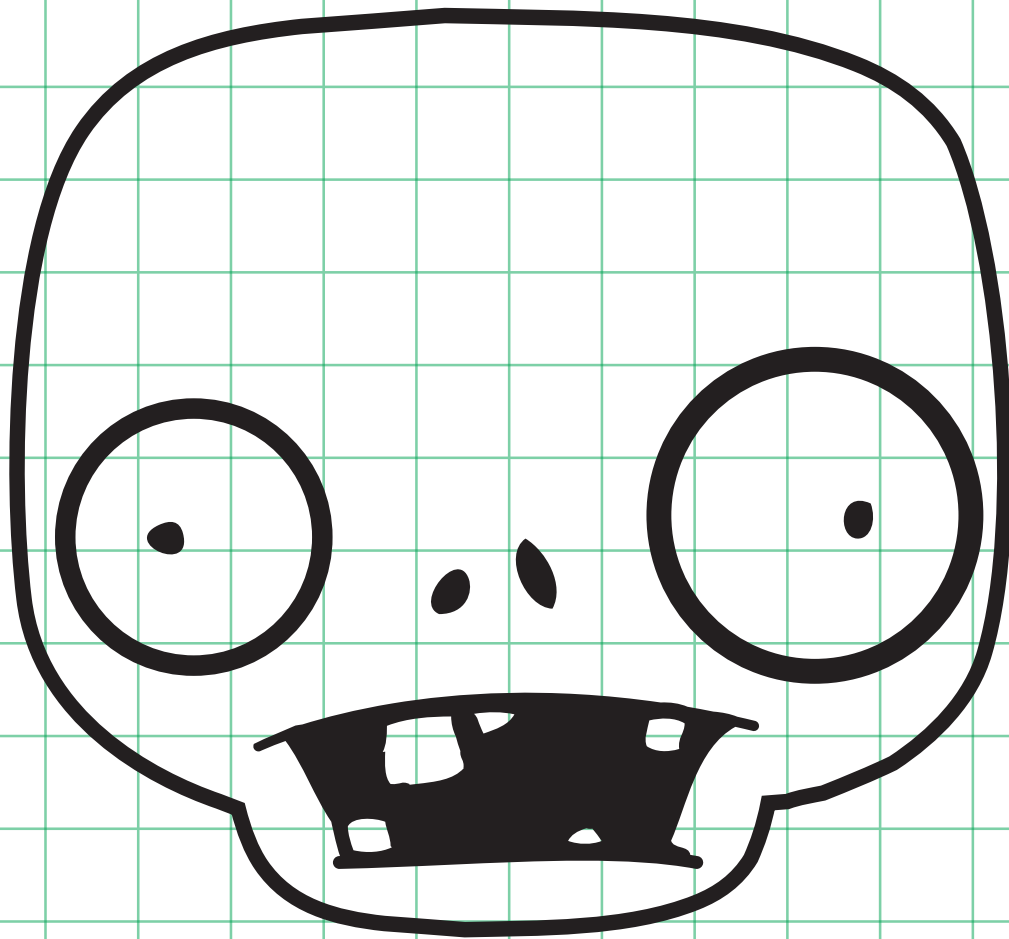
ETAPAS:

- 1) Identificar um problema.
- 2) Dividir o problema em partes menores.
- 3) Abstrair os detalhes para identificar um padrão que pareça válido para diversas partes.
- 4) Organizar cada um dos monstros na forma de lista de instruções para desenhar/criar.
- 5) Testar soluções para garantir que elas funcionam.
- 6) Trocar as instruções para que outros grupos possam juntar todas as partes em uma só obra de arte completa.

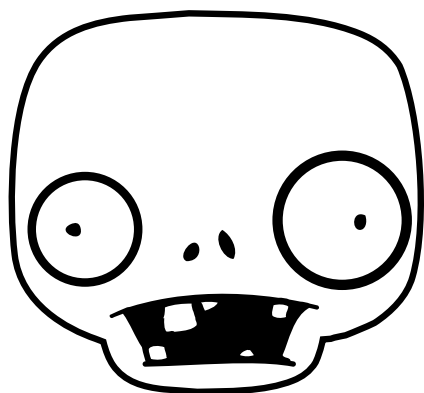
REGRAS:

- 1) Se o monstro não tiver alguma das características, a categoria dessa característica será “Ocultus”.
- 2) A categoria do formato da cabeça determina a primeira metade da classificação do monstro.
- 3) A segunda metade da classificação é determinada pela categoria da maior parte das características.
 - Se houver a mesma quantidade de características para diversas famílias, a classificação deverá ser feita com base nos olhos.

Catálogo de monstrros

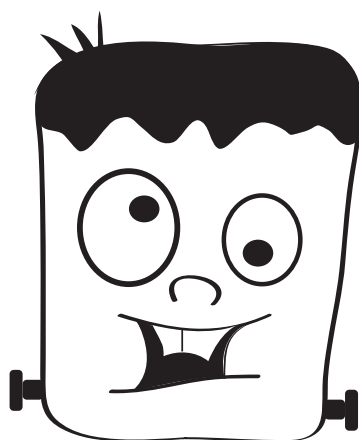


Monstros classificados



Zumbi Ranzinza

Este é o **Zumbi Ranzinza**, chamado assim por causa da classificação de sua cabeça, “Zumbi”, e pela predominância de características “Ranzinza”.



Frank Doidão

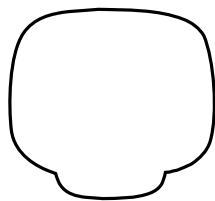
Este é o **Frank Doidão**. Observe o formato “Frank” de sua cabeça e a maioria das características “Doidão”.



Duende Feliz

Este é um monstro da família **Duende Feliz**. Sua cabeça tem um distinto formato “Feliz”, enquanto suas características faciais fazem parte da categoria “Duende”.

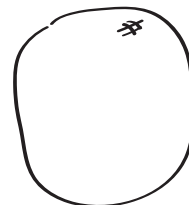
Cabeças



Zumbi



Frank



Feliz

Olhos



Ranzinza



Doidão



Duende

Orelhas

Ranzinza



Doidão



Duende



Nariz



Ranzinza



Doidão



Duende

Boca



Ranzinza



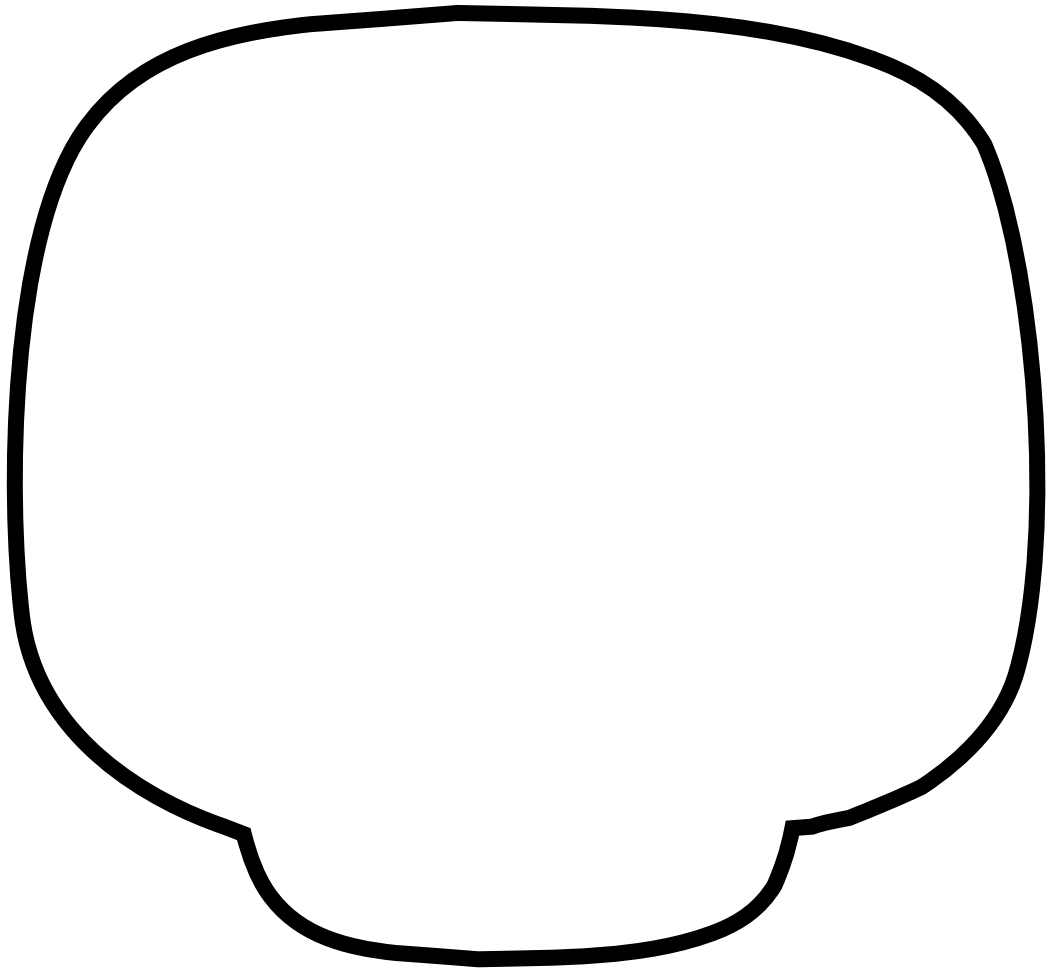
Doidão



Duende

3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



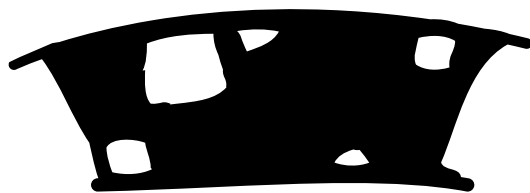
3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação

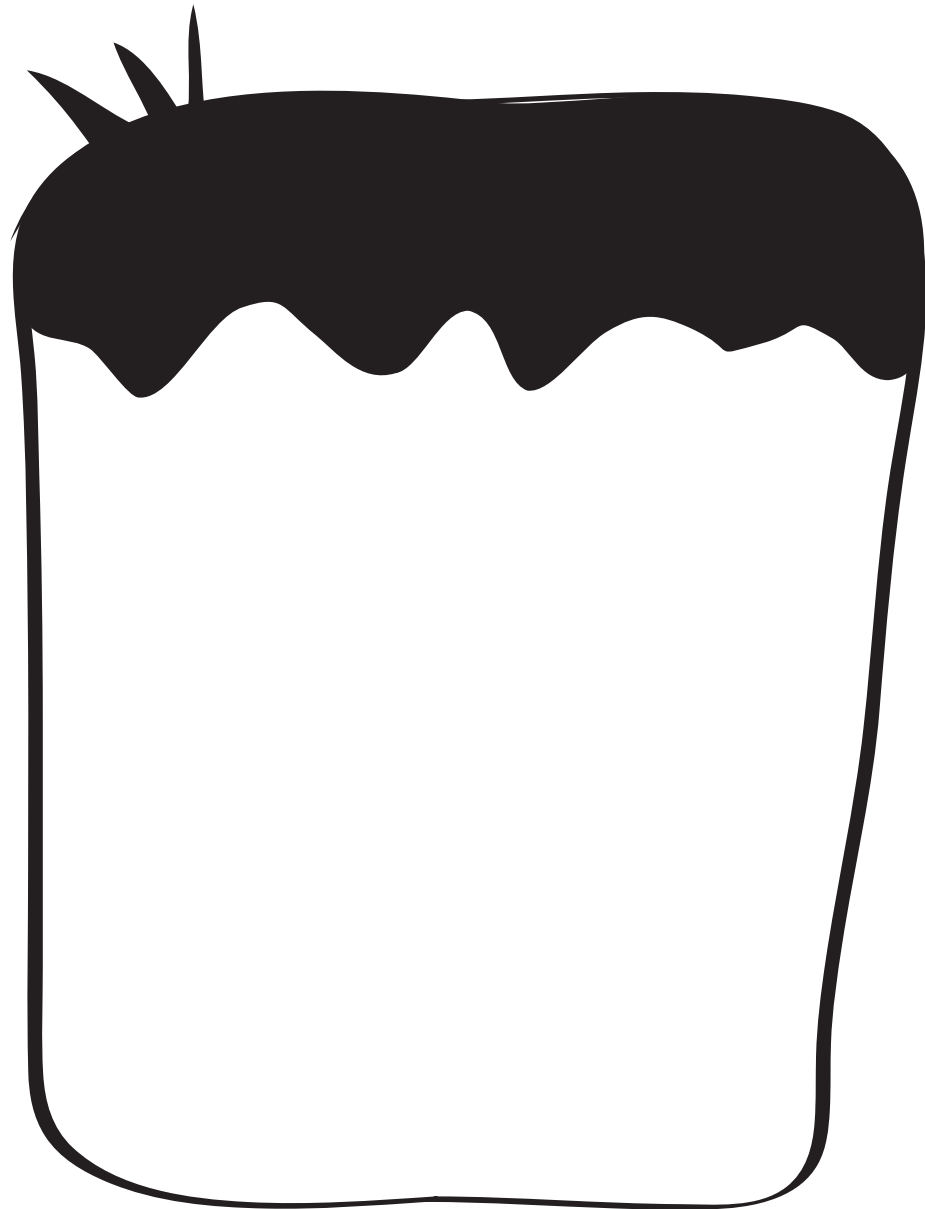


3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



Alinhe com base nesta marcação

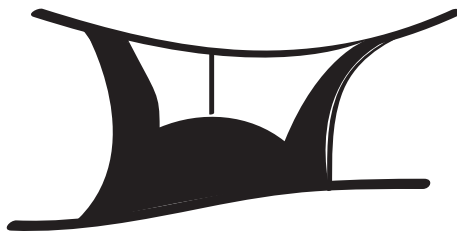


3 AULA 3: PENSAMENTO COMPUTACIONAL

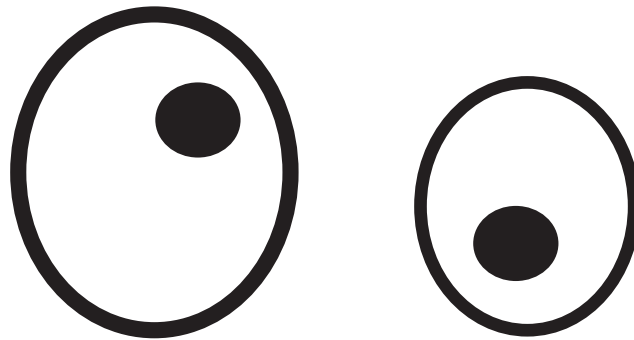
Alinhe com base nesta marcação



Alinhe com base nesta marcação

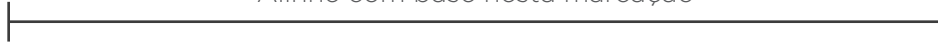


Alinhe com base nesta marcação



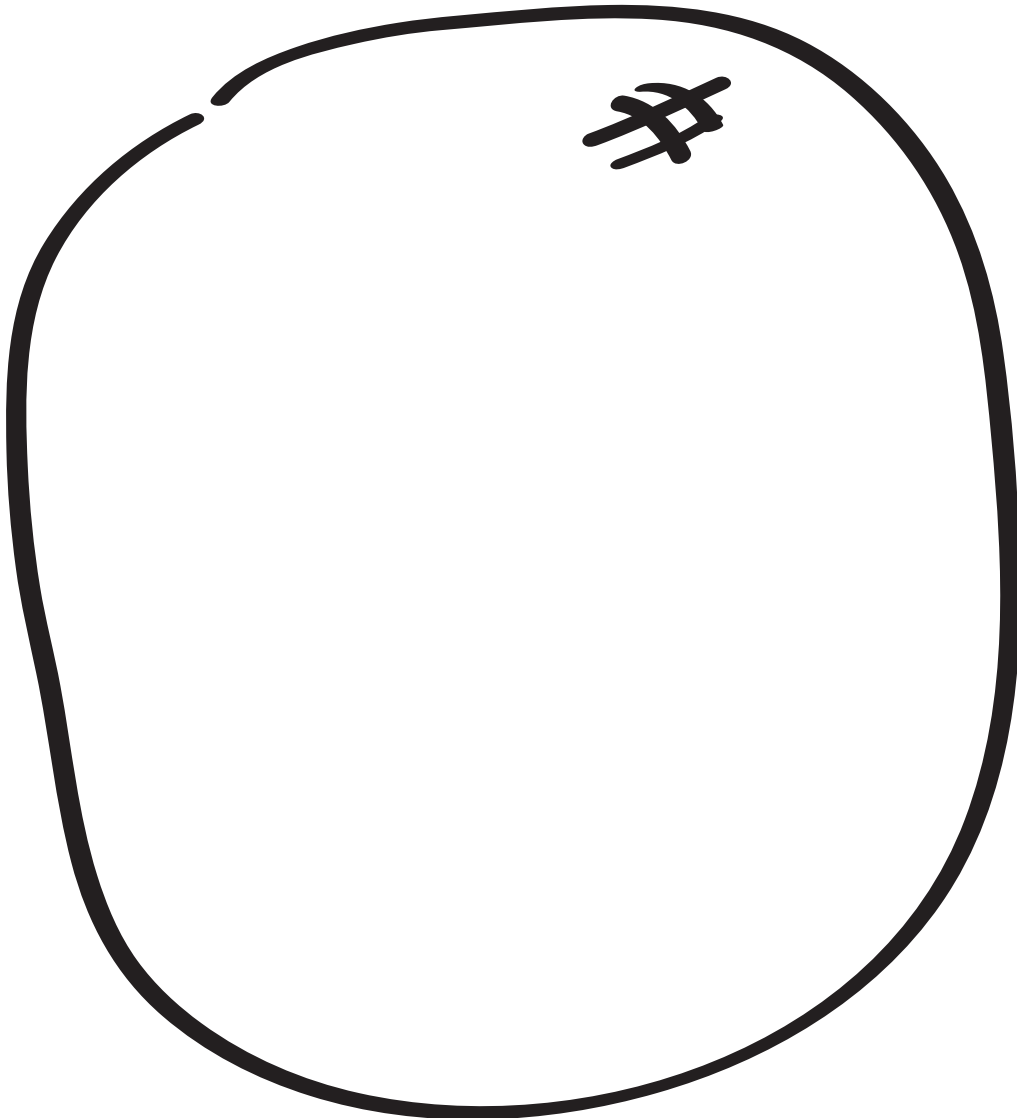
3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



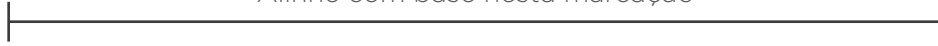
3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação



Alinhe com base nesta marcação



3 AULA 3: PENSAMENTO COMPUTACIONAL

Alinhe com base nesta marcação

