Introdução à Programação e Pensamento Computacional

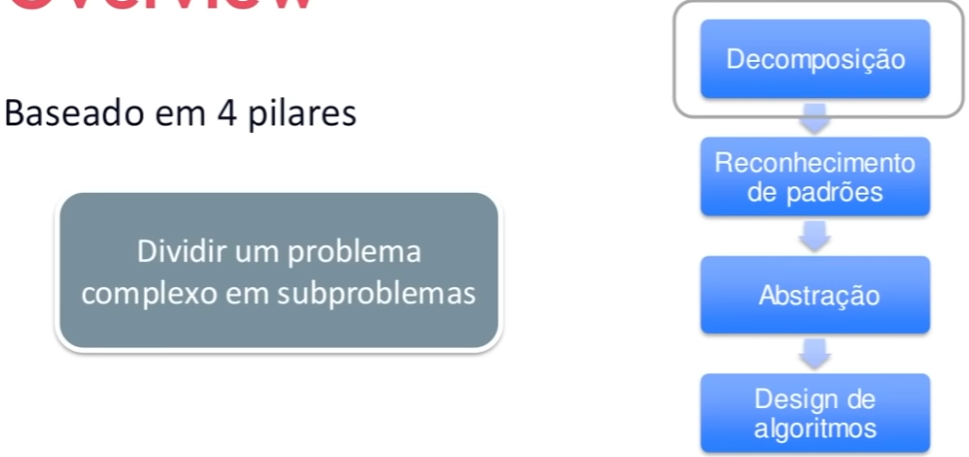
# Aula 1 – Apresentação

Será capaz de entender o que significa pensar computacionalmente. Pensamento aplicável à qualquer área do conhecimento

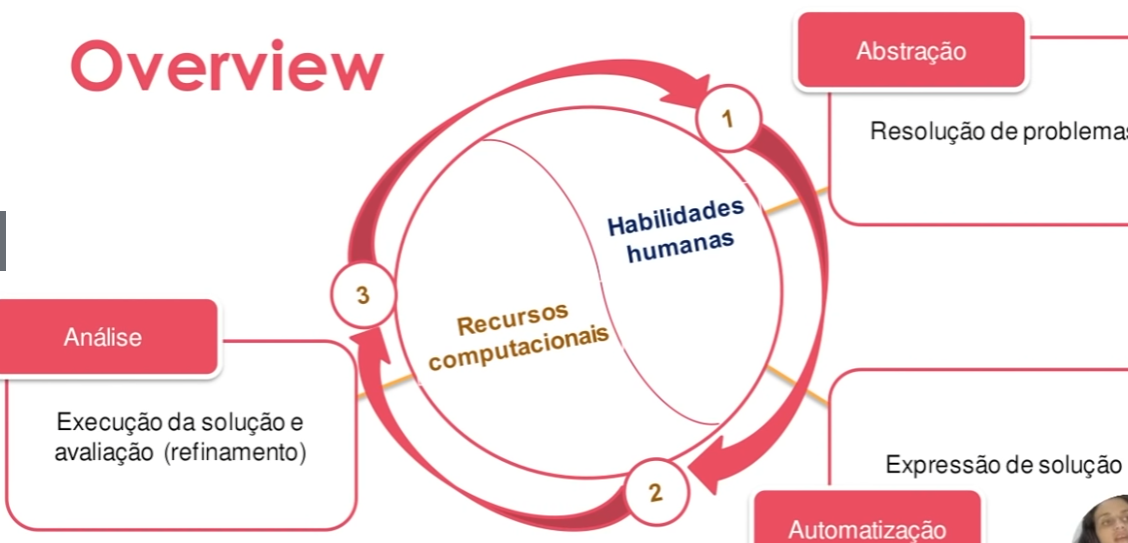
Dessa forma, será capaz de resolver qualquer problema de uma maneira mais objetiva e eficiente

# Aula 2 – Introdução ao Pensamento Computacional

Processo de pensamento envolvido na expressão de soluções em passos computacionais ou algoritmos que podem ser implementados no computador



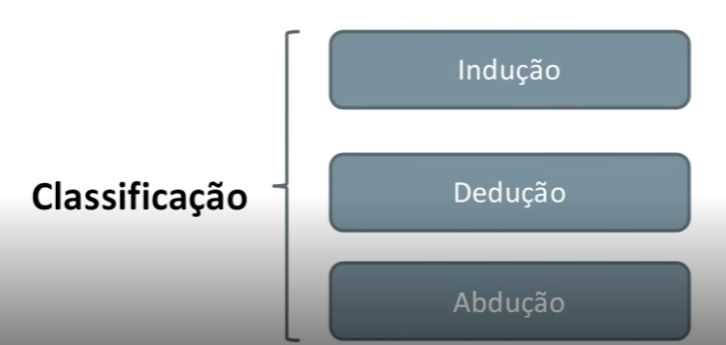


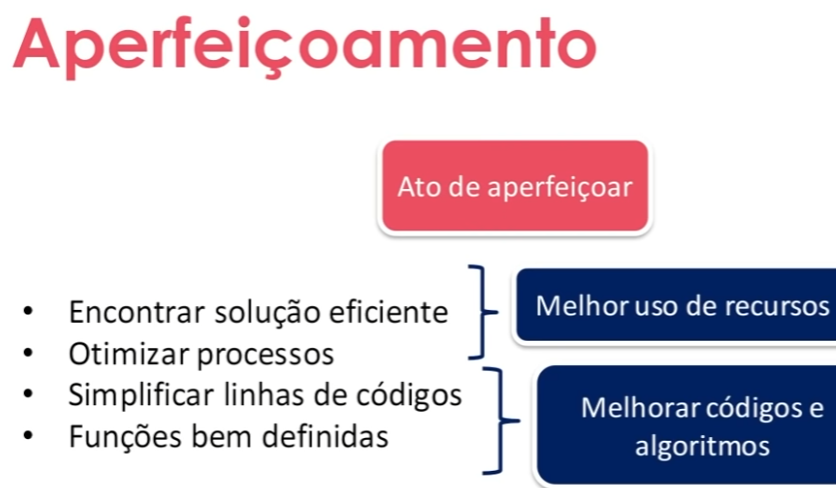


# Aula 3 – Habilidades Complementares

É necessário ter Raciocínio Lógico e Aperfeiçoamento

Raciocínio lógico é uma forma de pensamento estruturado, ou raciocínio, que permite encontrar a conclusão ou determinar a resolução de um problema, é uma habilidade que só consegue ser aperfeiçoada na prática:

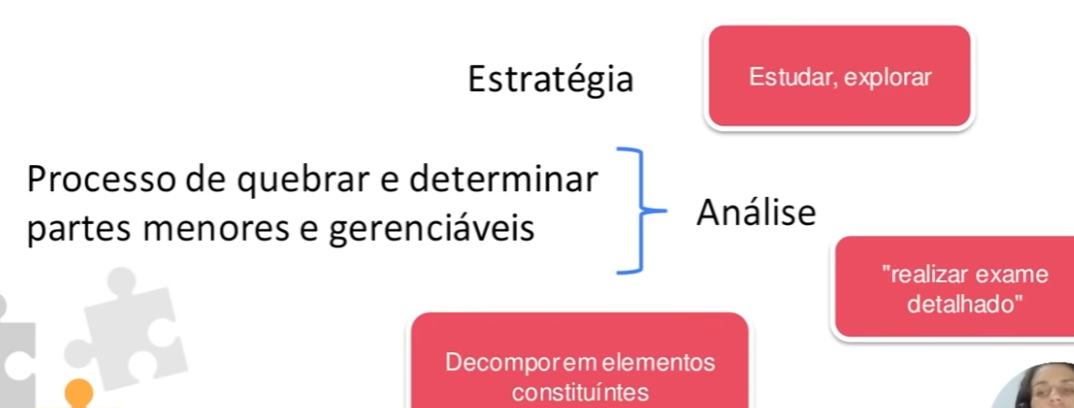


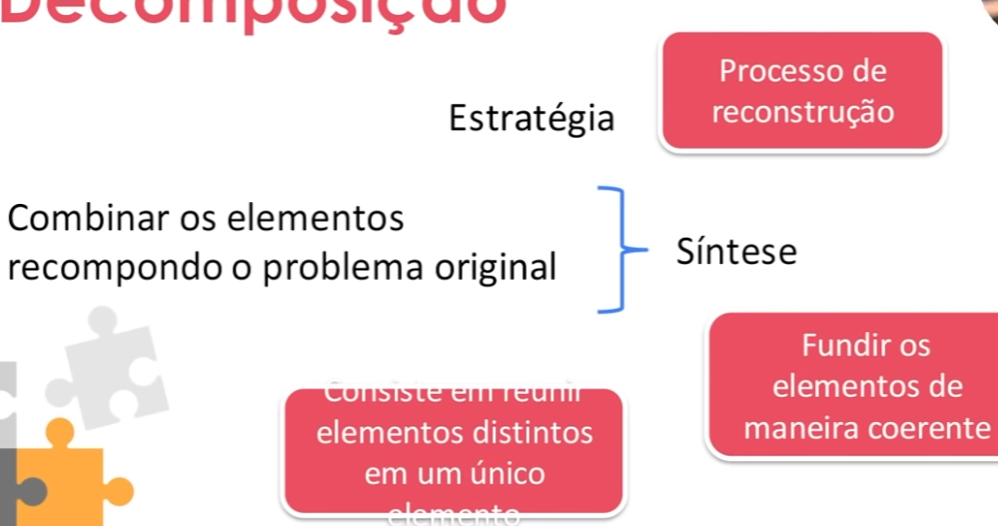


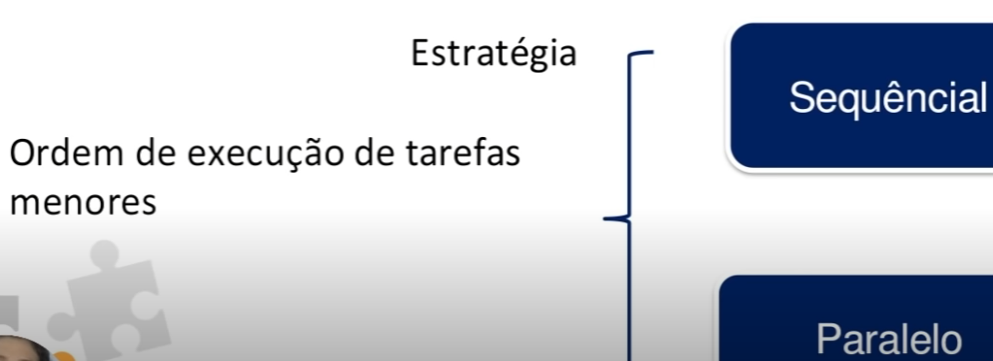
# Aula 4 – Pilares: Decomposição

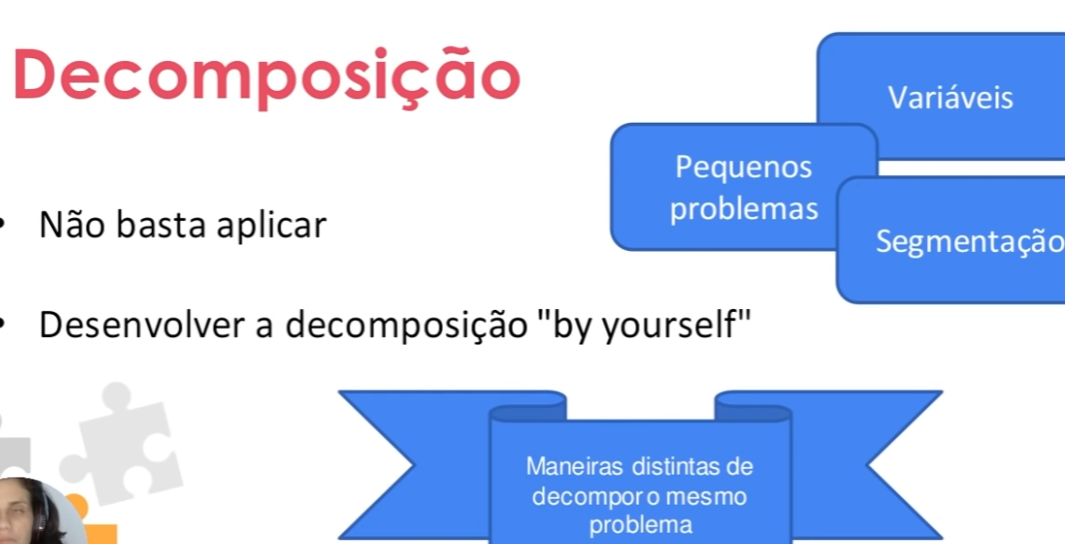
“If you can’t solve a problem, then there is an easier problem that you can solve: find it”

Primeiro passo da resolução de problemas dentro do conceito de pensamento computacional. Dado um problema complexo, devemos quebra-lo em problemas menores. Portanto, problemas mais fáceis e gerenciáveis









Como decompor? Primeiro identifica ou coleta os dados, depois agrega os dados e por fim entrega a funcionalidade

# Aula 5 – Pilares: Padrões



Padrões

**Como computador reconhece?**

Por comparação, simulando esse comportamento, representando os atributos, aprendendo (conceito associado ao objeto), armazenando os dados e tendo com isso regras de decisão

A detecção de padrão vem da extração de características e classificação de dados

## Aplicações

* Classificação de dados
* Reconhecimento de imagem
* Reconhecimento de fala
* Análise de cenas
* Classificação de documentos

# Aula 6 – Pilares: Abstração

Abstração = Observar um ou mais elementos avaliando as características e propriedades em separado, é um processo intelectual de isolamento de um objeto da realidade

Generalizar = Tornar-se geral, mais amplo, extensão, na lógica, é a operação intelectual que consiste em reunir numa classe geral, um conjunto de seres ou fenômenos similares

## Como classificar os dados?

* Características
* Pontos Essenciais
* Generalizar x Detalhar

# Aula 7 – Pilares: Algoritmos

* Energia
* Trabalhador
* Eficiência
* Rapidez
* Não opera sozinho

O computador recebe, manipula e armazena dados

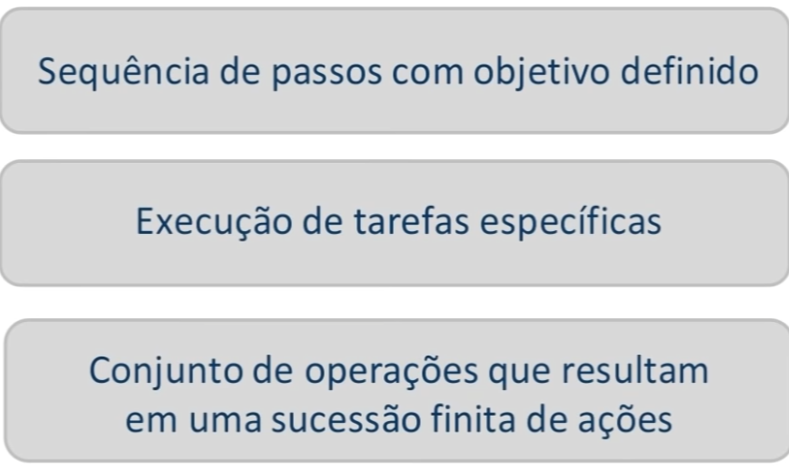
## Algoritmos

Processos de resolução de problemas “step by step” utilizando instruções, precisam ser entendidas por um humano e por uma máquina



## Desenvolvimento do Programa

* Análise
  + Estudo e definição dos dados de entrada e saída
* Algoritmo
  + Descreve o problema por meio de ferramentas narrativas, fluxograma e códigos
* Codificação
  + O algoritmo é codificado de acordo com a linguagem de programação escolhida



## Como construir um algoritmo?

* Compreensão do problema
* Definição dados de entrada
* Definir processamento
* Definir dados de saída
* Utilizar um método de construção
* Teste e diagnóstico

## Construção de algoritmos

* Narrativa

Utilização da linguagem natural, possui diversas interpretações possíveis, sem conceitos novos

* Fluxograma

Utilização de símbolos pré-definidos, precisa de conhecimento prévio da estrutura e símbolos, precisa ter um simples entendimento

* Pseudocódigo

Portugol, possui passos a serem seguidos, e tem regras definidas

# Aula 8 – Estudo de caso conceitual: Perdido

# Aula 9 – Estudo de caso conceitual: Soma de um intervalo

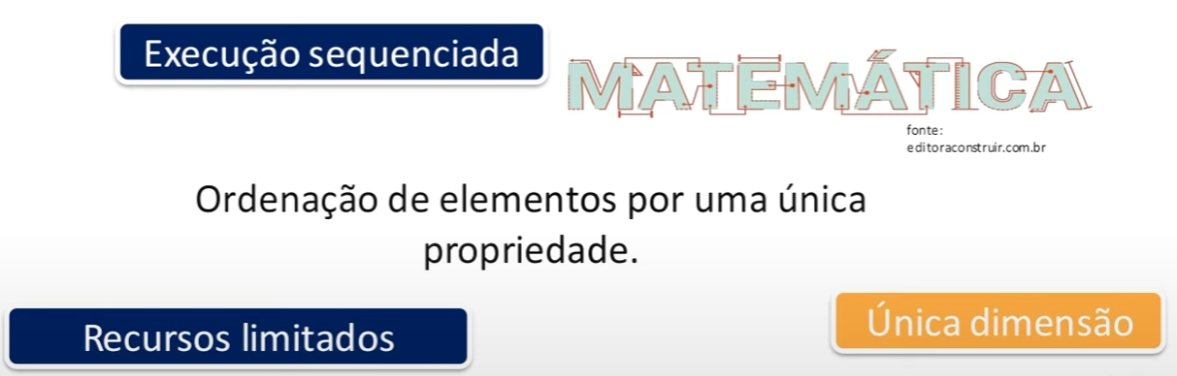
# Aula 10 – Estudo de caso conceitual: Adivinhe o número

# Aula 11 – O que é lógica computacional?

Lógica vem para solucionar um problema

# Aula 12 – Técnicas de lógica de programação

## Técnica Linear

* Modelo tradicional
* Não tem vínculo
  + Estrutura hierárquica
  + Programação de Computadores

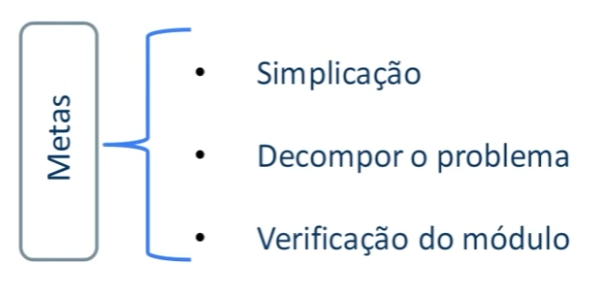
## Técnica Estruturada

Organização, disposição e ordem dos elementos essenciais que compõem um corpo (concreto ou abstrato)



## Técnica Modular

Controlada por um conjunto de regras, feito por partes independentes



# Aula 13 – Conceitos de algoritmos

## Tipologias e Variáveis

A função do computador é processar as informações que trazemos a ele, muitas dessas informações são dados ou instruções recebidas por nós, os dados são tratados e processados

### Variável

Regras

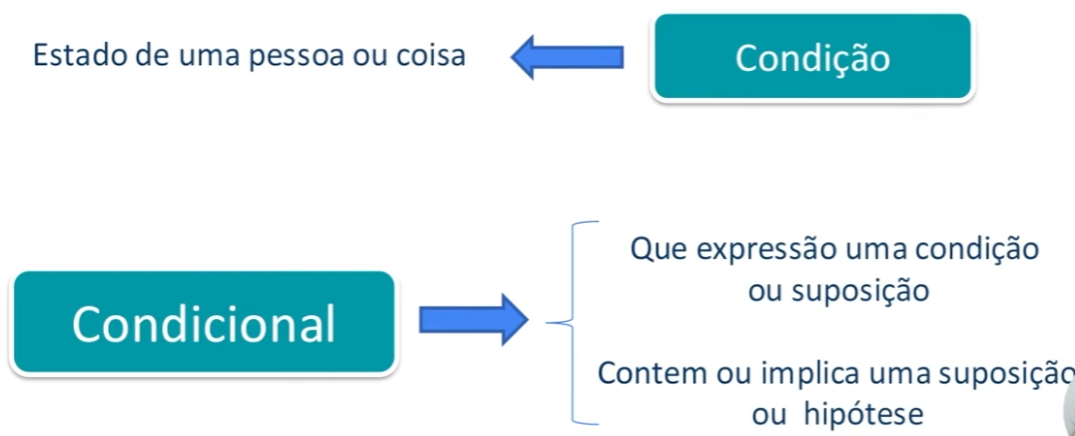
* Atribuição de um ou mais caracteres
* Primeira letra – não número
* Sem espaços em branco
* Vedado
  + Utilização de palavras reservadas
* Caracteres e números

# Aula 14 – Instruções Primitivas



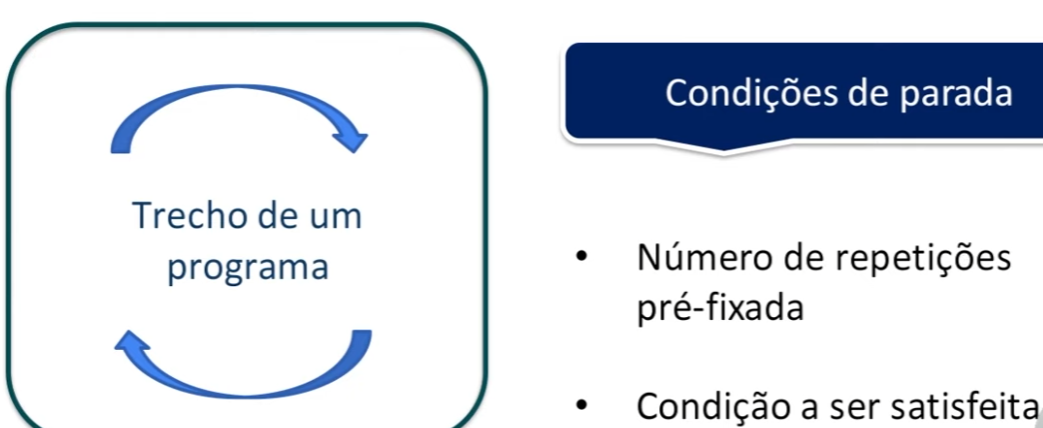
Instrução executa uma determinada programação que tem por finalidade comanda um computador a tratar dados

# Aula 15 – Estruturas Condicionais



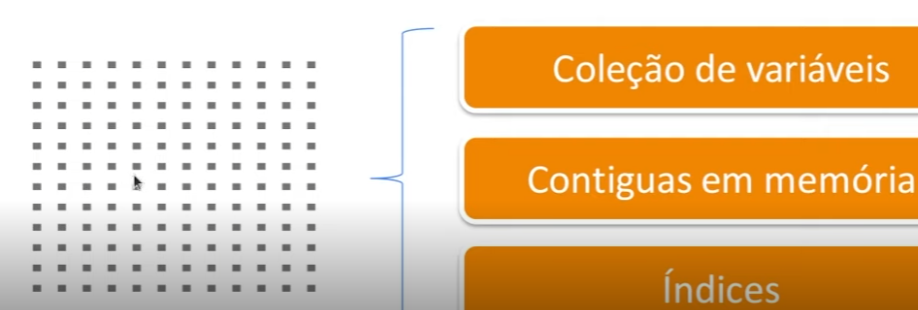
# Aula 16 – Estruturas de repetição





O motivo do uso é redução de linhas, menor chance de cometer erros e é mais otimizado

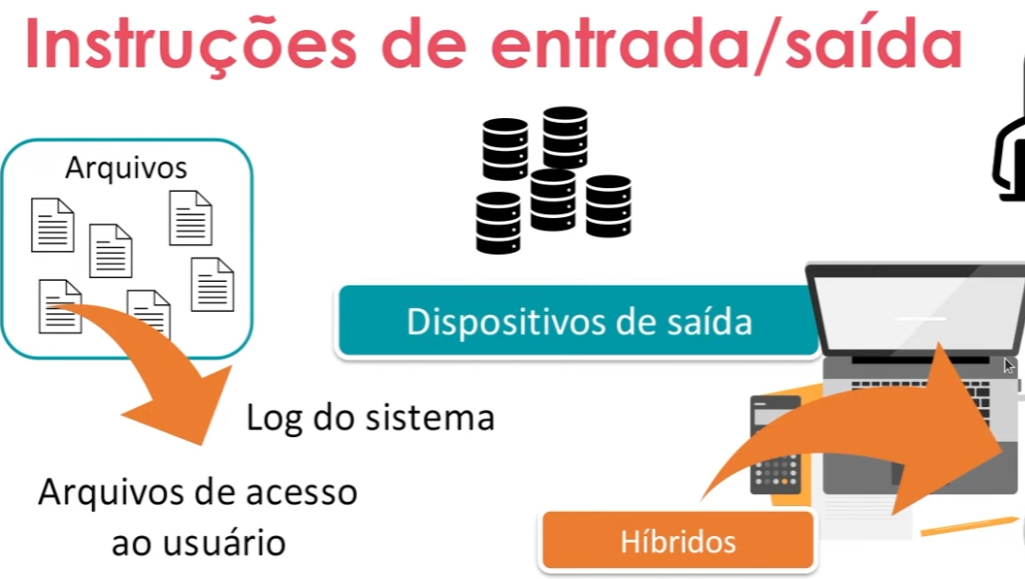
# Aula 17 – Vetores e Matrizes



# Aula 18 – O que são funções?

As funções, ou sub-rotinas são blocos de instruções que realizam tarefas específicas

# Aula 19 – Instruções de entrada e saída

Dados que são inseridos no computador e processados pelo mesmo, gerando um resultado, consiste na inserção via alguma interface física ou virtual, onde a máquina pode imprimir esses dados em algum lugar

Existe dois tipos de saída, programada e por interrupção, a programada pode ser condicional ou não



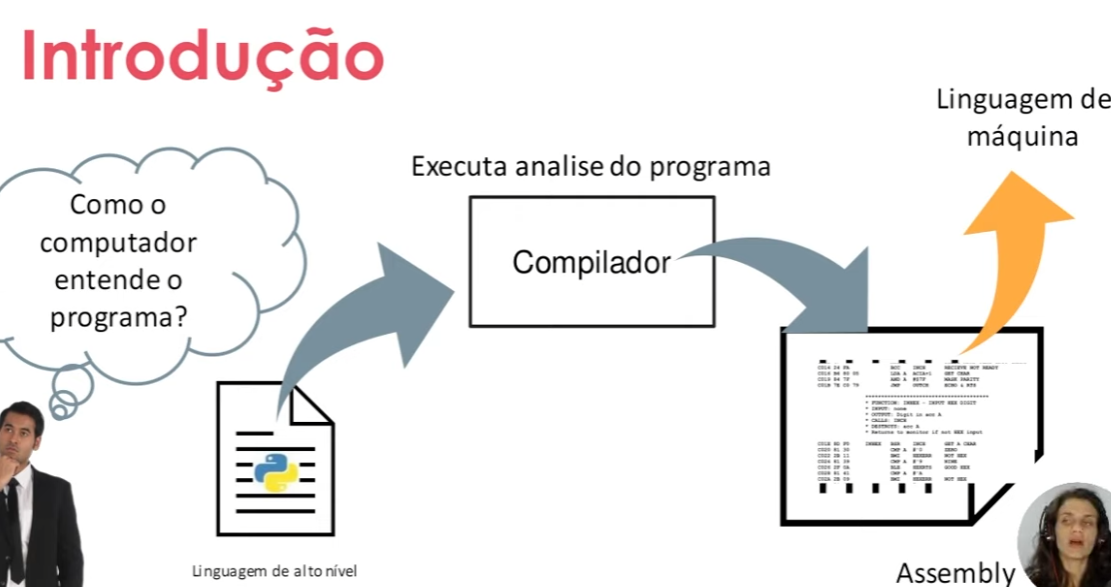
# Aula 20 – Linguagens da programação – Introdução

Em 1949 surgiu a primeira linguagem de programação (na verdade sendo a de máquina) o Assembly

Problemas computacionais possui instruções passo a passo que são mais facilmente resolvíveis em ambiente computacional

Linguagem de programação é um método padronizado composto por um conjunto de regras sintáticas e semânticas de implementação de um **código fonte**

# Aula 21 – Como o computador entende o Programa?



Na tradução há geração e execução de um programa objeto (execução mais rápida), já na interpretação o programa fonte é executado diretamente (maior flexibilidade)

# Aula 22 – Características de um Programa



## Legibilidade

* Facilidade de leitura
* Compreensão
* Ortogonalidade
* Definição adequada das estruturas

## Redigibilidade

* Pode conflitar com legibilidade
* Ortogonalidade
* Simplicidade da escrita
* Suporte à abstração
* Reuso do código
* Expressividade

## Confiabilidade

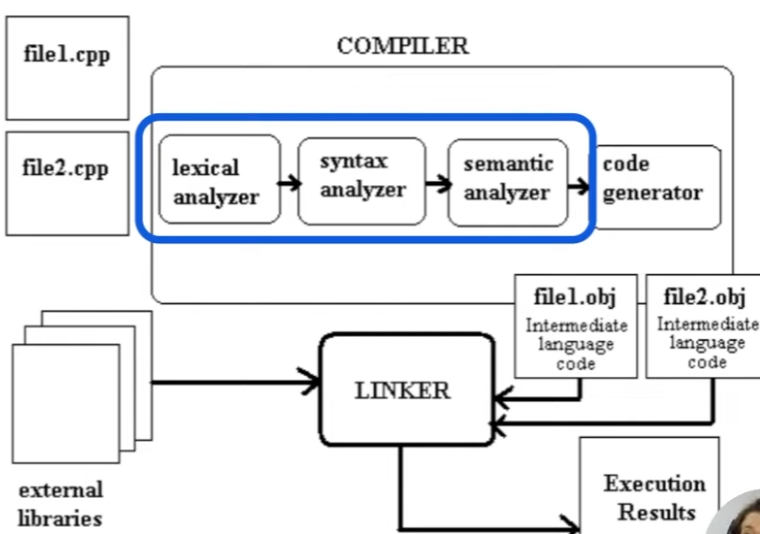
Possui:

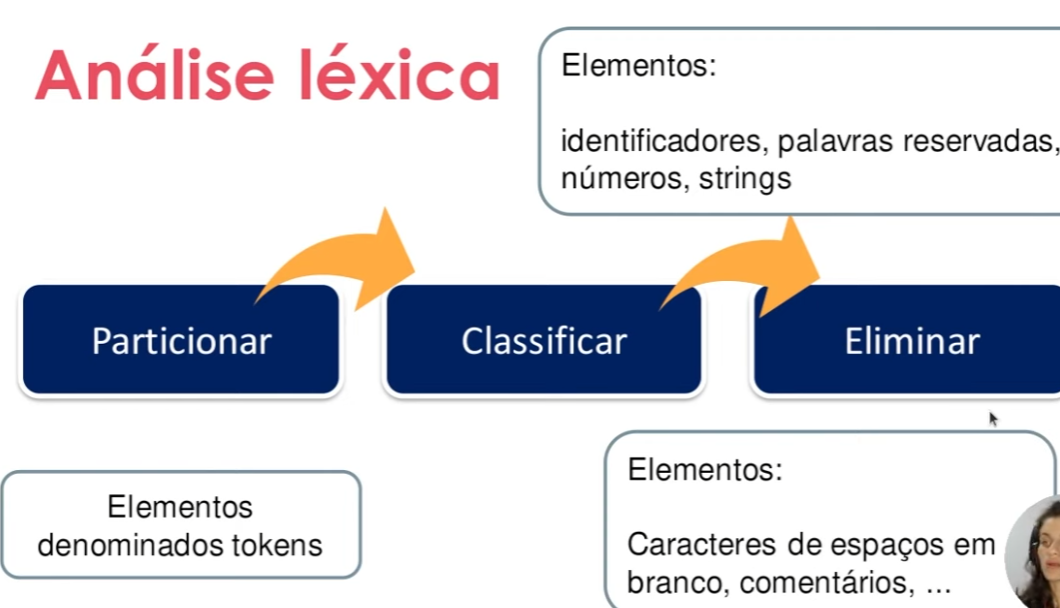
* Verificação de tipos
* Trata exceções
* Uso de ponteiros
* Compatibilidade entre compiladores

## Custo

* Treinamento
* Codificação
* Compilação
* Execução
* Infra-estrutura

# Aula 23 – Análises de código





## Análise Sintática

Sintaxe: Componente do sistema linguístico que interliga os constituintes da sentença, atribuindo-lhe uma estrutura

Padrão – gramática

* Depende da linguagem de programação utilizada

## Análise Semântica

Semântica: É o estudo do significado. Incide sobre a relação entre significantes, como: palavras, frases, sinais e símbolos

Erro de semântica:

* Não faz o que é esperado

# Aula 24 – Paradigmas de programação

## Paradigma Estruturado

Conceitos:

* Sequência
* Decisão
* Iteração

Utilização:

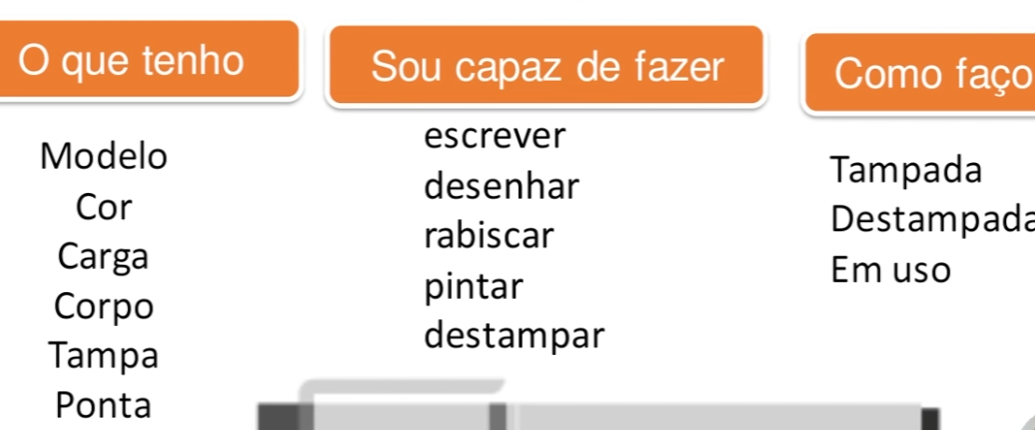
* Problemas simples e diretos
* Aprender programação

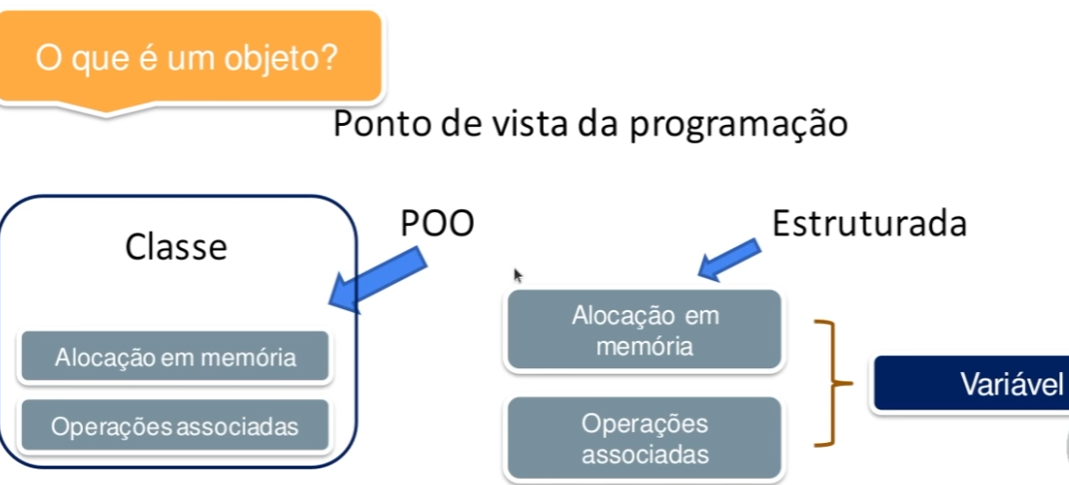
POO ainda não compreendida por muitos

## Orientação à Objeto

Paradigma de programação baseado na utilização de objetos e suas interações

Análogo ao mundo real

Um objeto é descrito por características específicas, comportamentos e estado



Herança: classe filha herda características da classe mãe

# Aula 25 – Primeiro contato com programação

## Algoritmos em portugol

Já fiz em outro word/notepad