



# **ESTRUTURA DE DADOS**

## **INTRODUÇÃO ÀS ESTRUTURAS DE DADOS**

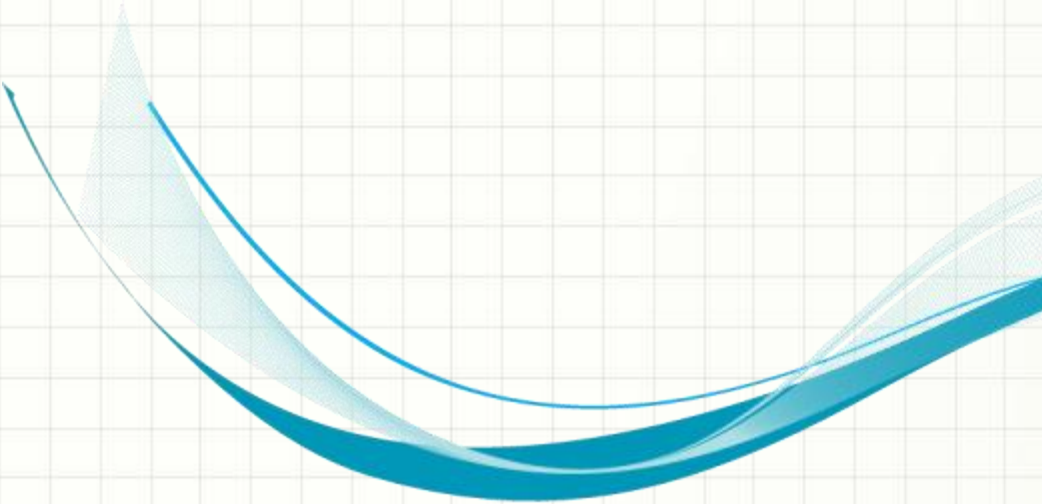
Prof. Dr. Daniel Caetano

2014 - 2

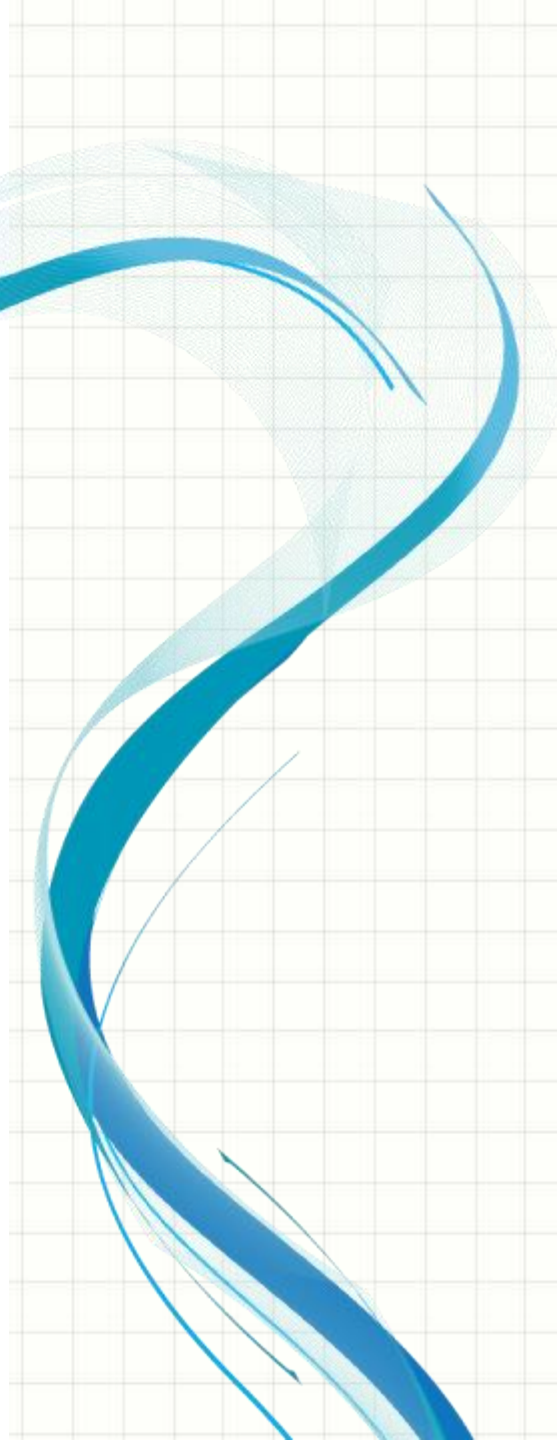
# Objetivos

- Conhecer o professor e o curso
- Compreender o que são estruturas de dados e sua importância
- Relembrar conceitos de algoritmos





# Apresentação



Quem é o  
professor?



Vamos  
começar?

# Quem É Quem – Lista de Presença

---

## Professor

## Informações de Contato

Daniel Caetano

[prof@caetano.eng.br](mailto:prof@caetano.eng.br)

---

Nome Completo	CPF	Matrícula
Fulano	012.345.678-90	201101123456
Beltrano	012.345.678-91	201101123457
Cicrano	012.345.678-92	201101123458



# **PLANO DE ENSINO E DE AULA**

# Plano de Ensino

Disponível no WebAula



1. Entre no **SIA**
2. **CAMPUS VIRTUAL**
3. **MINHAS DISCIPLINAS PRESENCIAIS**
4. Clique no **NOME DA DISCIPLINA**
5. Selecione **PLANO DE ENSINO**



# Plano de Aula

- 28/07 – 0. Apresentação
- 04/08 – 1. Funções
- 11/08 – 2. Vetores e Laços
- 18/08 – Exercícios
- 25/08 – 3. Listas Sequenciais
- 01/09 – 4. Listas: Ordenação
- 08/09 – 5. Listas: Ordenação
- 15/09 – 7. Pilhas
- 22/09 – 7. Pilhas
- 29/09 – **P1**
- 06/10 – 8. Filas
- 13/10 – 9. Filas Circulares
- 20/10 – 10. Estruturas
- 27/10 – 11. Ponteiros
- 03/11 – Exercícios
- 10/11 – 12. Listas Encadeadas
- 17/11 – 13. Pilhas Encadeadas
- 24/11 – 14. Filas Dinâmicas
- 01/12 – **P2**
- 08/12 – Vista de Prova
- 16/12 – **P3**



# **TRABALHOS, DATAS E CRITÉRIO DE APROVAÇÃO**

# Trabalhos, Datas e Aprovação

Trabalho	Valor	C.H.	Entrega
AE1 (Grupo / Individual)	2,0 na AV1	8h	<b>31/08</b> (SIA)
P1 (Individual / Com Consulta*)	8,0 na AV1	2h	29/09 (Aula)
AE2 (Grupo / Individual)	0,5 na AV2	8h	<b>05/10</b> (SIA)
AE3 (Grupo / Individual)	0,5 na AV2	8h	<b>19/10</b> (SIA)
AE4 (Grupo / Individual)	0,5 na AV2	8h	<b>16/11</b> (SIA)
AE5 (Grupo / Individual)	0,5 na AV2	8h	<b>30/11</b> (SIA)
P2 (Individual / Sem Consulta)	8,0 na AV2	2h	01/12 (Aula)
P3 (Individual / Sem Consulta)	8,0 na AV3	2h	15/12 (Aula)

(\*) Consulta nos moldes da folha de referência fornecida no site da disciplina.

# Trabalhos, Datas e Aprovação

- Atenção ao **prazo de entrega** das AE1 a AE5...
- As Atividades Estruturadas serão entregues pelo SIA e serão penalizadas em 1/5 de seu valor por dia de atraso.
- Mesmo que já não valham nota, elas **precisam** ser entregues, pois valem parte significativa da carga-horária!
- As atividades são grandes, não marque bobeira!

# Trabalhos, Datas e Aprovação – AV1

- Prova P1 à caneta, incluindo o ***preenchimento do cabeçalho***, ganha: 0,25 na P1
- Entregando a folha de consulta (***dentro do padrão***) com a prova, ganha: 0,25 na P1
- A nota da AE1 será **somada** à nota da prova P1 para compor a média AV1.

$$\underbrace{AV1}_{0,0 \text{ a } 10,0} = \underbrace{P1}_{0,0 \text{ a } 8,0} + \underbrace{AE1}_{0,0 \text{ a } 2,0}$$

# Trabalhos, Datas e Aprovação – AV2

- A nota das AE2 e AE5 (total de 0 a 2) será somada à nota da P2 para compor a nota AV2.

$$\underbrace{AV2}_{0,0 \text{ a } 10,0} = \underbrace{AE2a5}_{0,0 \text{ a } 2,0} + \underbrace{P2}_{0,0 \text{ a } 8,0}$$

# Trabalhos, Datas e Aprovação – AV3

- A nota da AV3 é composta **apenas** da nota da avaliação P3! Cuidado!

$$\underbrace{AV3}_{0,0 \text{ a } 10,0} = \overbrace{P3}^{0,0 \text{ a } 10,0}$$

# Trabalhos, Datas e Aprovação – Final

**A** = Maior nota entre { **AV1** , **AV2** , **AV3** }

**B** = Segunda maior nota entre { **AV1** , **AV2** , **AV3** }

**Critérios de Aprovação** (TODOS precisam ser atendidos)

1) **A**  $\geq$  4,0

2) **B**  $\geq$  4,0

3) **A** + **B**  $\geq$  12,0

4) Frequência  $\geq$  75%



(Média 6,0!)

(No máximo **4** faltas!)

**ATENÇÃO:** Se você tiver mais que uma nota abaixo de 4,0, ainda que o SIA aponte uma média maior que 6,0, você estará **REPROVADO!**





# **BIBLIOGRAFIA E FONTES DE INFORMAÇÃO**

# Bibliografia

- **Biblioteca Virtual**
  - Estrutura de Dados



- **Material do Curso**

- Estrutura de Dados – Série Livros Didáticos Informática da UFRGS, Volume 18 (1ª Edição, 2009)

- Edelweiss e Galante
- Artmed / Bookman
- ISBN: 9788577803811

# Bibliografia

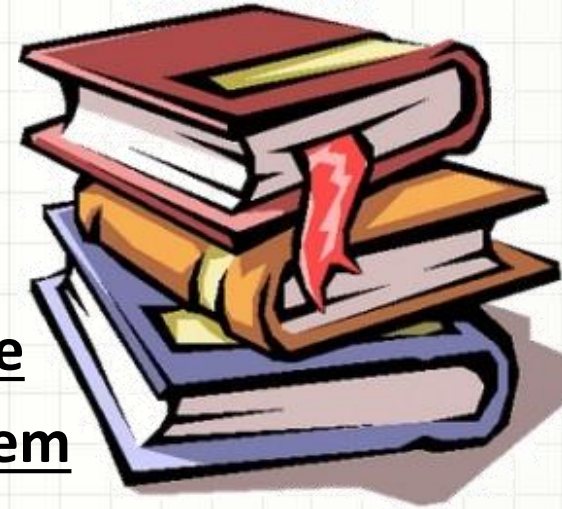
- **Mais Livros!**

- Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++ (1ª Edição, 2011)

- Ascêncio e Araújo
- Editora Pearson Education
- ISBN: 9788576058816 **BIBLIOTECA VIRTUAL!**

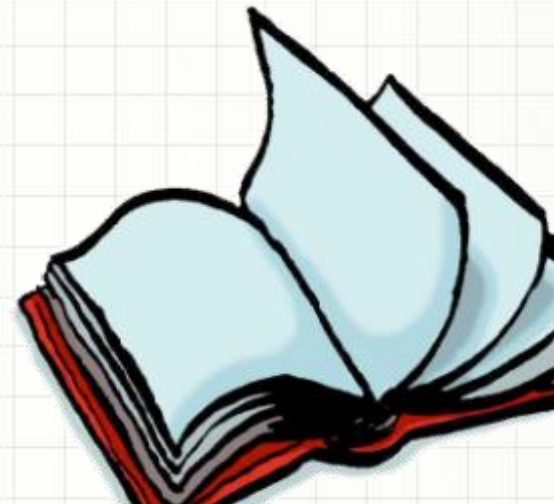
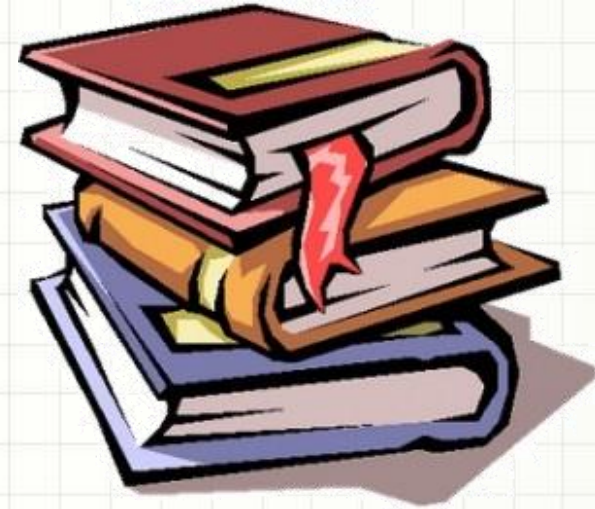
- Lógica de Programação: a construção de algoritmos e estruturas de dados (3ª Edição, 2005)

- Forbellone e Eberspacher
- Editora Pearson
- ISBN: 9788576050247 **BIBLIOTECA VIRTUAL.**



# Bibliografia

- Ótimo apoio!
  - C Completo e Total (3ª Edição, 1996)
    - Herbert Schildt
    - Makron Books / Pearson
    - ISBN: 8534605955



# Material Didático

## Deve Ser Solicitado no SIA



1. Entre no **SIA**
2. **SECRETARIA VIRTUAL**
3. **SOLICITAÇÃO DE MATERIAL**

# Bibliografia

- **Notas de Aula e Apresentações**



<http://www.caetano.eng.br/>

The screenshot shows the website for Prof. Caetano. At the top left is a photo of a man in a white shirt and glasses, looking towards a group of students in a classroom. To the right of the photo, the name "Prof. Caetano" is written in a large, black, cursive font. In the top right corner, the date and time "17/07/2012, 10:55" and the ID "00021224" are displayed. Below the name, there are two small flags: the Brazilian flag and the UK flag. A navigation menu is located at the bottom of the page, with buttons for "Home", "Ensino", "Pesquisa", "Publicações", "Software", and "Pessoal". The "Ensino" button is highlighted with a red circle. Below the navigation menu, a paragraph of text reads: "Nesta seção você encontra acesso ao material didático desenvolvido pelo Prof. Caetano para os cursos já ministrados. O material está dividido por períodos, visto que boa parte do material não está atualizado."



# **FORMAÇÃO DE GRUPOS DE TRABALHO**

# Formação de Grupos

- Por que formar grupos?
- Quantos alunos?
  - No **mínimo** 4 alunos
  - No **máximo** 8 alunos
- Entregar, na **aula que vem**, lista de NOMES de cada aluno, indicando o NOME DA EQUIPE.
- **Atenção:**
  - Elejam UM responsável por subir os dados no SIA, que deve fornecer o e-mail para o professor!



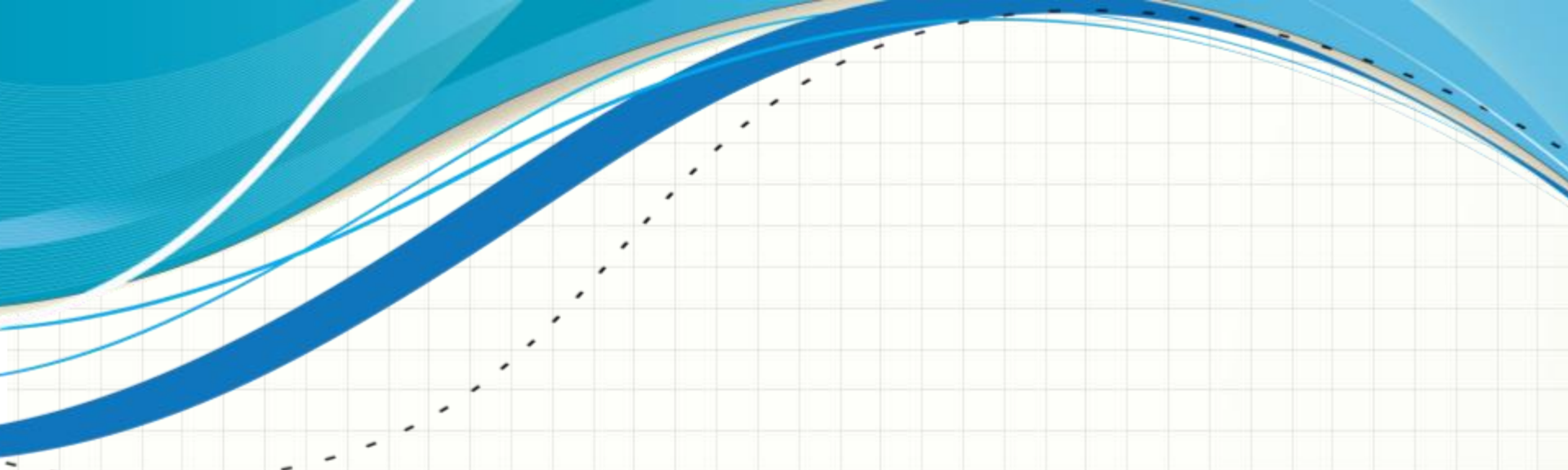


**VOLTANDO À PROGRAMAÇÃO NORMAL:**  
**CONTEXTUALIZAÇÃO**

# Contextualização

- Continuação de **Algoritmos**
  - Qual a melhor forma de implementar?
- Relação Disciplina x Curso
  - Desenvolvimento de Software
  - Análise / Projeto de Software
  - Sistemas Operacionais
  - Banco de Dados
- Empregabilidade?
  - Criatividade e senso crítico
  - Domínio da programação





**COMO ERAM OS  
ALGORITMOS, MESMO?**

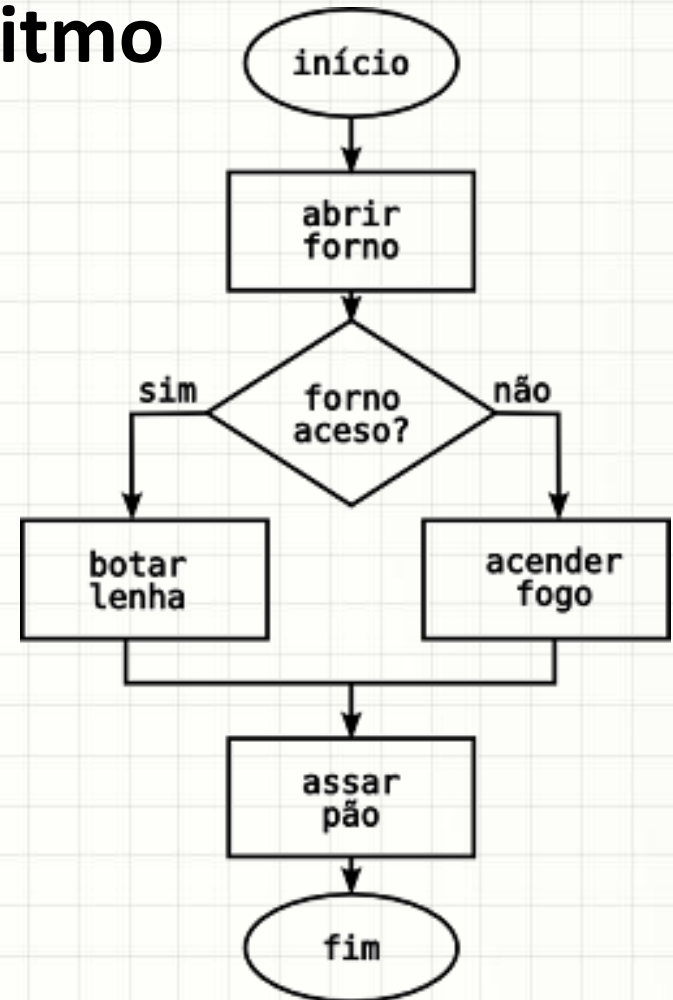
# O que são Algoritmos

- Toda tarefa complexa pode ser subdividida
  - Tarefas menores e mais simples
- Exemplo: fabricar vinho para venda
  - Plantar a uva
  - Colher a uva
  - Amassar a uva
  - Deixar fermentar
  - Engarrafar
  - Distribuir para a venda



# O que são Algoritmos

- O procedimento para realizar uma tarefa complexa chama-se **algoritmo**
- Um algoritmo envolve:
  - **Tarefas/Processos**
  - **Decisões**



# Fluxograma

- Forma gráfica tradicional



Início e fim de algoritmo



Processo (trabalho do computador)



Entrada de dados (leitura)



Saída de dados (impressão)



Tomada de decisão



Sentido do fluxo de dados

# Fluxograma

- Forma gráfica tradicional



Início e fim de algoritmo



Processo (trabalho do computador)



Entrada de dados (leitura)



Saída de dados (impressão)



Tomada de decisão

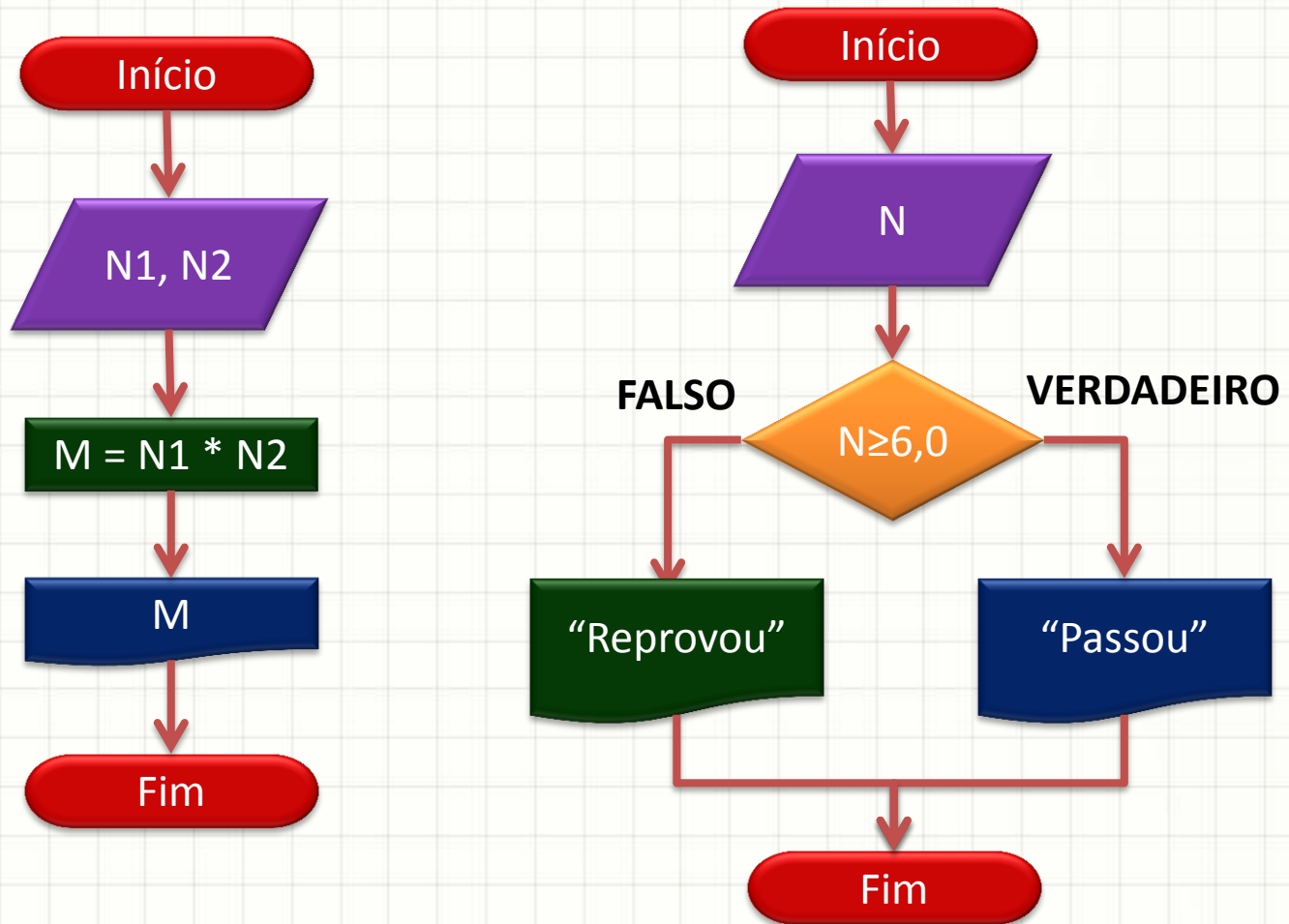


Sentido do fluxo de dados

ALTERNATIVOS

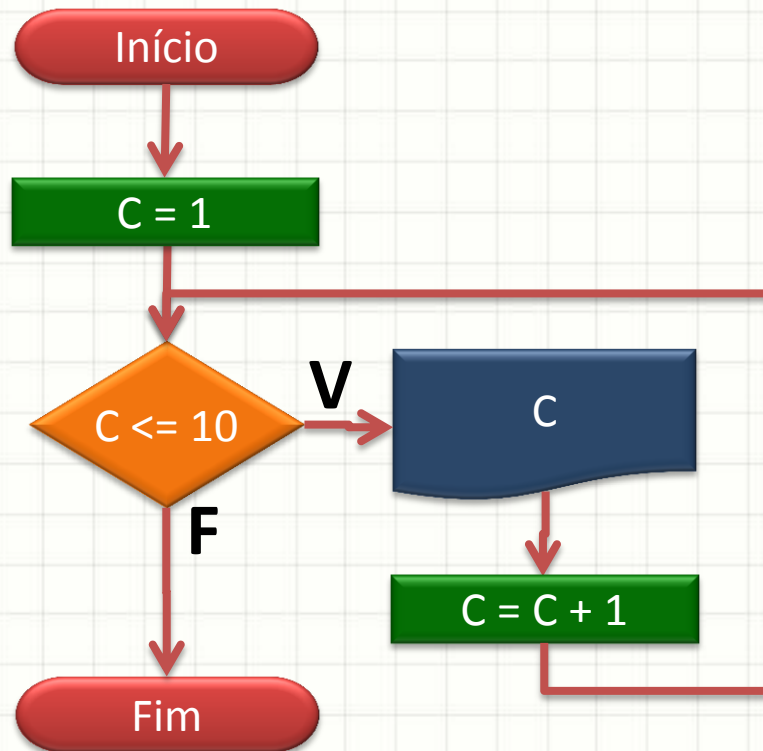


# Exemplos de Fluxograma





# Exemplos de Fluxograma



# Exemplos de Linguagem C++

```
#include <iostream>
using namespace std;
main()
{
    int C;
    C = 0;
    while ( C < 1000 )
    {
        cout << "Sei programar!";
        cout << endl;
        C = C + 1;
    }
}
```

# Exemplos de Linguagem C++

```
#include <iostream>
using namespace std;
main() // Imprime contador
{
    int C;
    C = 0;
    while ( C < 1000 )
    {
        cout << C << " - ";
        cout << "Sei programar!";
        cout << endl;
        C = C + 1;
    }
}
```

# Exemplos de Linguagem C++

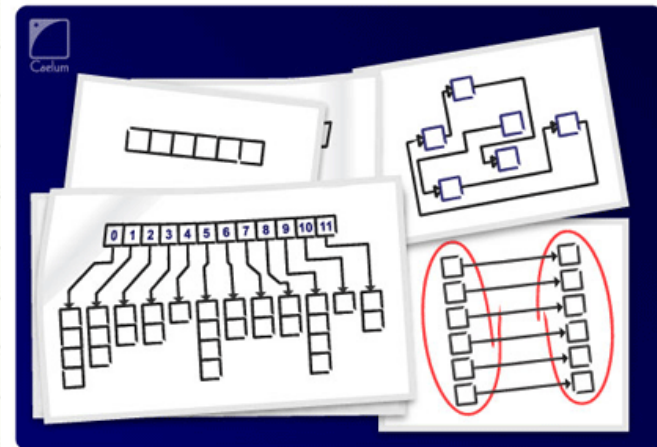
```
#include <iostream>
using namespace std;
main()
{
    int N1, N2, R, C;
    cout << "Digite um No.:";
    cin >> N1;
    cout << "Digite outro No.:";
    cin >> N2;
    C = 0;
    while ( C <= N1 )
    {
        R = C * N2;
        cout << C << "*" << N2 << "=" << R << endl;
        C = C + 1;
    }
}
```



**ESTRUTURA DE DADOS?  
HEIN?!**

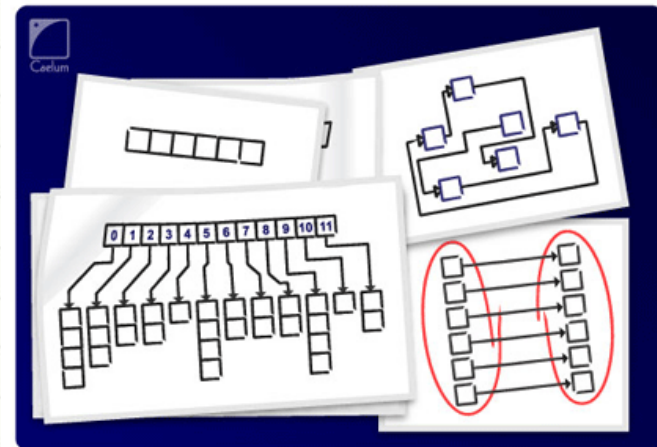
# Estrutura de Dados

- Programa = Algoritmo + Dados
- Resolução de Problema: abstração
- Cadastro de Clientes
  - Quais dados são importantes?
  - Qual o algoritmo usar?



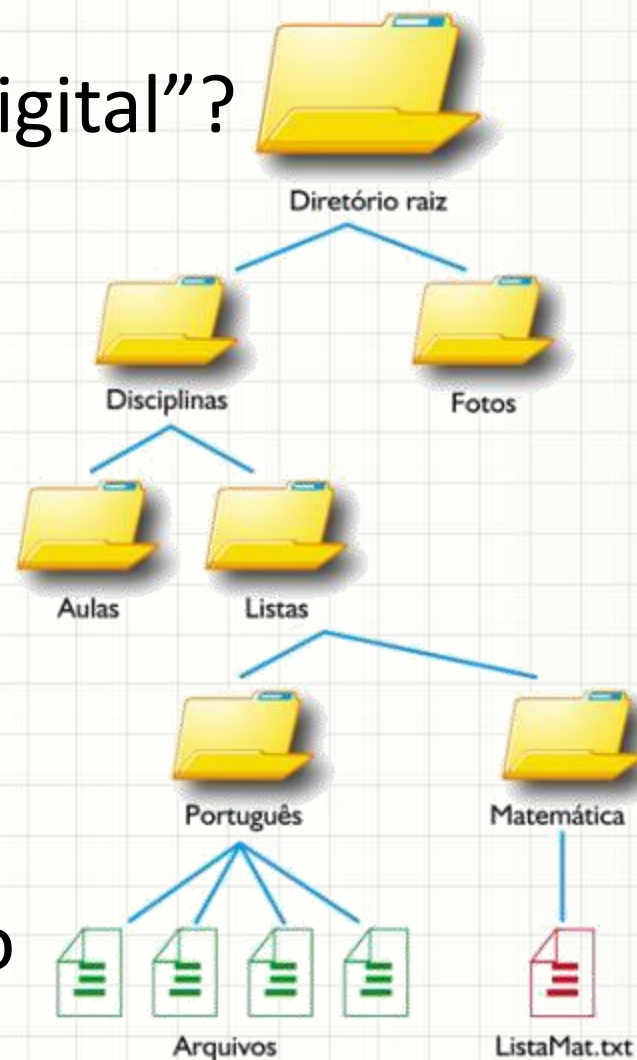
# Estrutura de Dados

- Programa = Algoritmo + Dados
- Resolução de Problema: abstração
- Cadastro de Clientes
  - Quais dados são importantes?
    - A idade do cliente é importante?
    - A cor do cabelo do cliente é importante?
  - Qual o algoritmo usar?
    - Como encontrar um cliente?
    - Como inserir um novo cliente?



# Estrutura de Dados

- O que é um “dado digital”?
- O que diferencia de “lixo digital”?
- Sua **organização**
  - Sabemos como encontrá-los
- E isso permite...
  - Busca
  - Remoção
  - Inserção...
- Organização → Desempenho



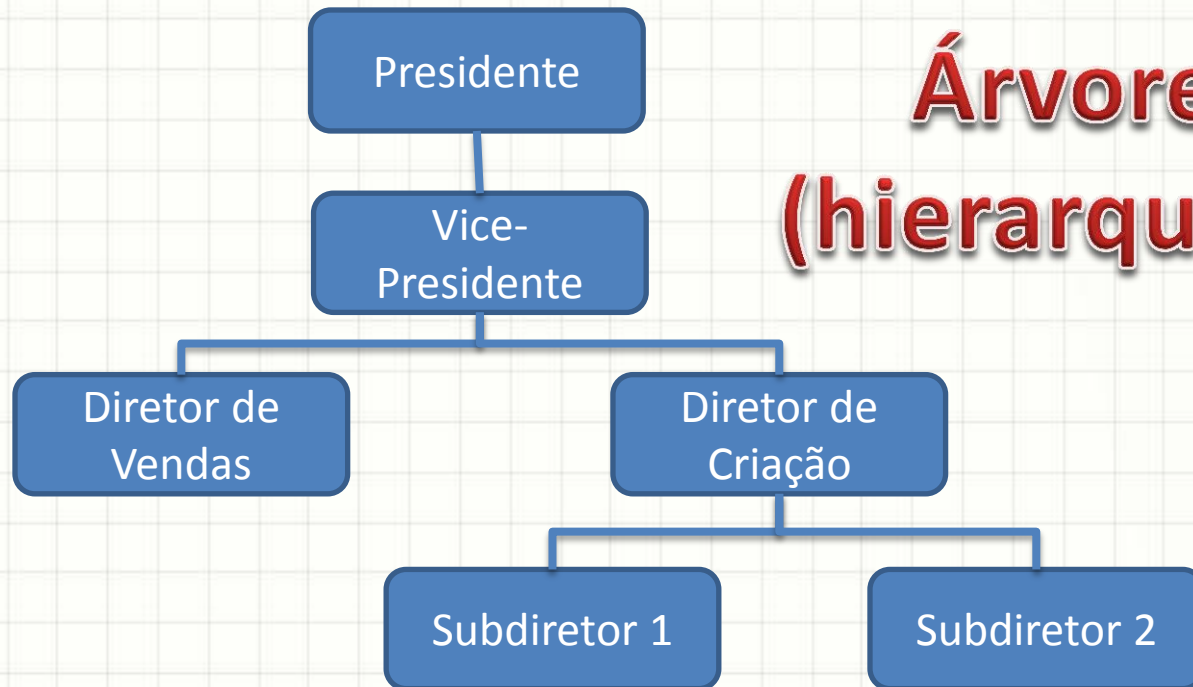




**No DIA-A-DIA**

# Estrutura de Dados no Dia-a-Dia

- Representar a organização de uma empresa
  - 1 presidente, 1 vice-presidente, 1 diretor de vendas e 1 de criação, este último com 2 subdiretores?



**Árvore**  
**(hierarquia!)**

# Estrutura de Dados no Dia-a-Dia

- Como representar a bibliografia do curso?
  - Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++
  - Lógica de Programação: a construção de algoritmos e estruturas de dados
  - Estrutura de Dados – Série Livros Didáticos Informática da UFRGS, Volume 18

**Lista**



# Estrutura de Dados no Dia-a-Dia

- Como o motoboy organiza as pizzas?

**Pilha**



# Estrutura de Dados no Dia-a-Dia

- Como as pessoas esperam no banco?

**Fila**



# Estrutura de Dados no Dia-a-Dia

- Como representar os trajetos possíveis em uma companhia aérea?

## Grafo

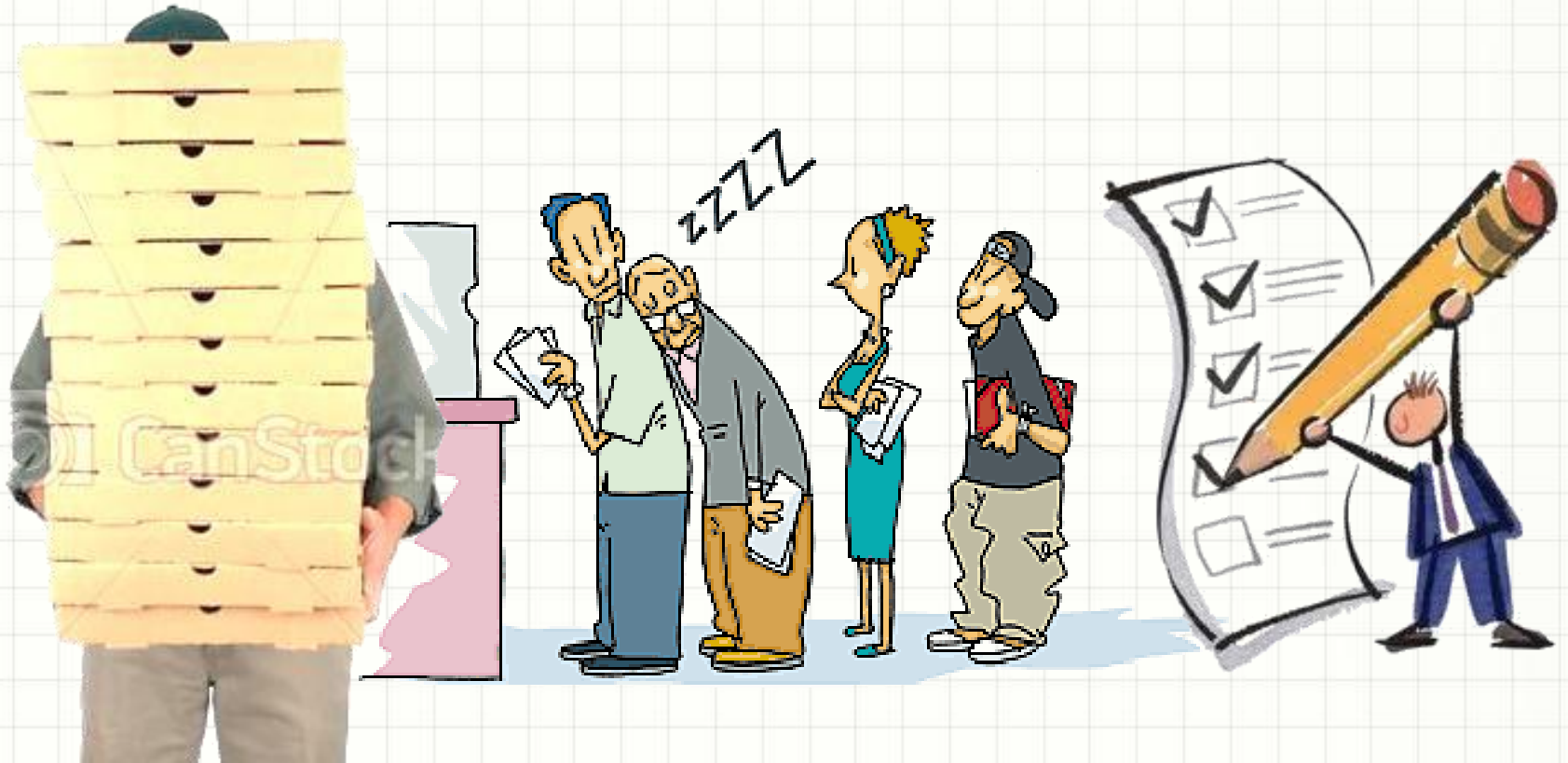




# **TIPOS DE ESTRUTURA DE DADOS**

# Tipos de Estrutura de Dados

- Lineares x Não-lineares
- Lineares





# Tipos de Estrutura de Dados

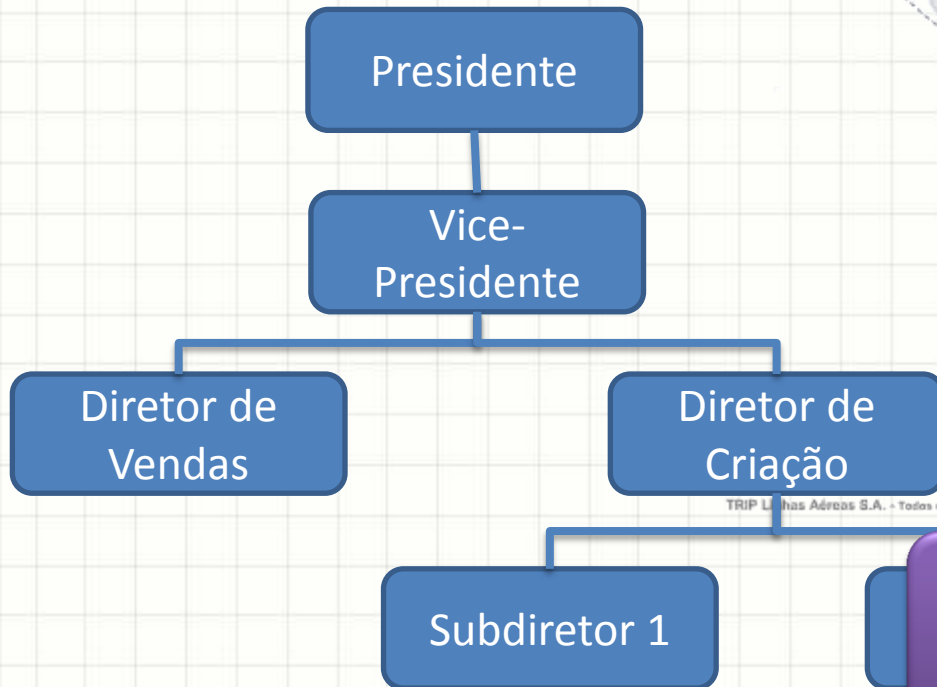
- Lineares x Não-lineares
- Lineares

- 1º. Elemento bem definido
- Último elemento bem definido
- Elementos intermediários: um antecessor e um sucessor



# Tipos de Estrutura de Dados

- Não-lineares

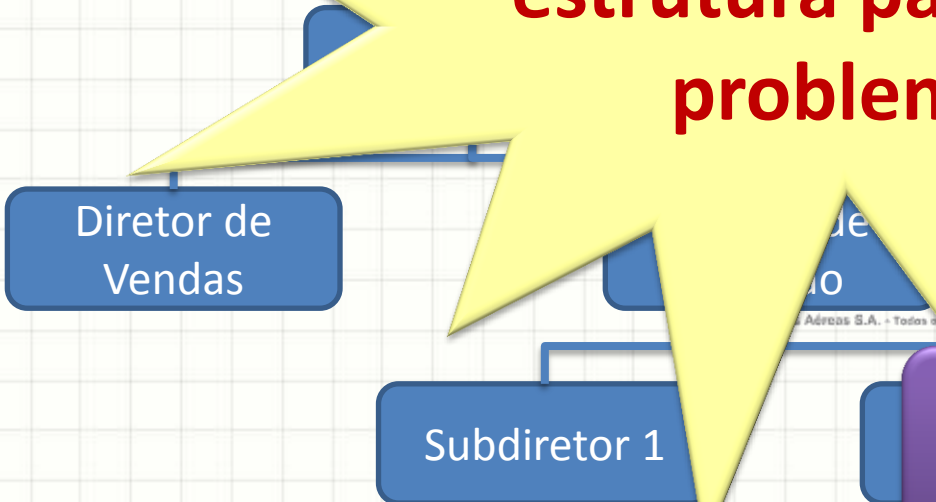


- Árvore: relação hierárquica
- Grafo: relação qualquer

# Tipos de Estrutura de Dados

- Não-lineares

**É fundamental identificar a melhor estrutura para cada problema!**



- Árvore: relação hierárquica
- Grafo: relação qualquer





# **FORMAS DE ARMAZENAMENTO E MANIPULAÇÃO DE ESTRUTURA DE DADOS**

# Armazenamento de Estruturas

- Duas maneiras de armazenar
  - **Sequencial** (ou contígua)
    - Espaço pré-alocado
    - Tamanho pré-definido
  - **Encadeada** (ou ligada)
    - Tamanho inicialmente desconhecido
    - Alocação à medida da necessidade
- Neste curso
  - Estruturas lineares sequenciais e encadeadas

# Armazenamento de Estruturas

- Iniciaremos com as sequenciais...
- Qual o tipo de variável de C/C++ que serve para guardar, sequencialmente na memória, muitos dados iguais?
  - **Vetores**
- Vamos começar com uma lista de números
  - Que operações vocês conseguem imaginar?
  - O que gostaríamos de poder fazer com uma lista?

# Manipulação de Estruturas

- Imagine uma lista de notas
    - Inicialmente vazia
  - Inserir notas
  - Remover notas
  - Buscar notas...
  - Como realizar essas tarefas?
- 
- Existiria muita diferença se tivéssemos uma lista de alunos? Ou uma lista de rendimentos?

# Manipulação de Estruturas

- Inserir, Remover e Buscar serão semelhantes para qualquer lista
- Sempre que precisarmos inserir, o código é o mesmo
- Que tal criarmos um algoritmo chamado **inserir**, por exemplo?
  - Sempre que precisarmos inserir um valor, bastará solicitar que o computador execute o algoritmo **inserir**



# Manipulação de Estruturas

- Esses “algoritmos” com nome recebem o nome de **funções**
- Antes de estudarmos as estruturas em si...
  - Vamos aprender a implementar funções!
  - Vamos nos familiarizar com vetores!





**PERGUNTAS?**



# EXERCÍCIOS DE FIXAÇÃO

# Exercícios de Fixação

1) Qual a melhor estrutura de dados para representar o sistema de pastas e arquivos do sistema operacional?

- a) Pilha
- b) Fila
- c) Árvore
- d) Grafo

# Exercícios de Fixação

2) Os navegadores web armazenam as páginas visitadas de maneira que ao apertar o botão “voltar” a última página visitada seja apresentada, retirando este endereço da estrutura. Considerando só esse aspecto, qual é a melhor estrutura de dados?

- a) Pilha
- b) Fila
- c) Árvore
- d) Lista



# CONCLUSÕES

# Resumo

- Planos de Ensino e Aula
  - Datas de avaliações e critérios de aprovação
  - Fontes de informação
  - O que são dados e estrutura de dados
  - Operações e usos comuns de estruturas
- 

- Funções?
  - O que são?
  - Para que servem?