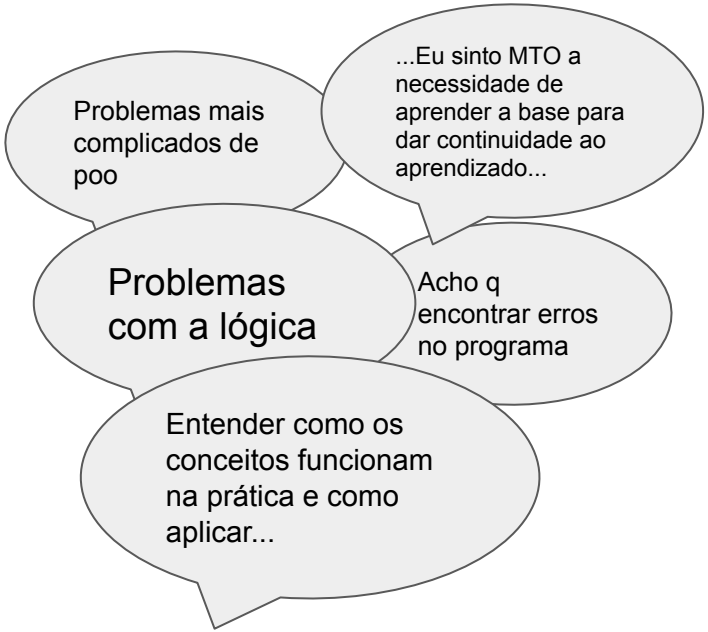
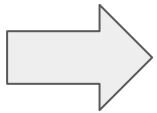
Aula de Programação em C

Jesimon Barreto
jesimonbarreto@gmail.com

Objetivo da apresentação

Foi utilizado os trabalhos práticos de PDS1 e PDS2, além de contato com alguns dos colegas via whatsapp para definir os conteúdos abordados.





Motivação da apresentação

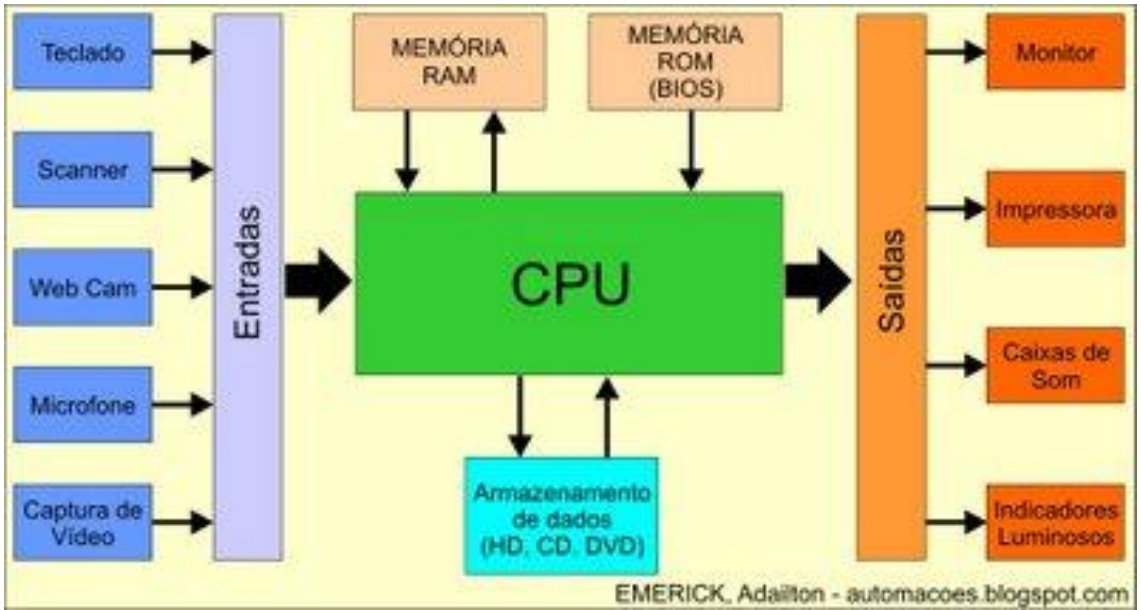
- Conectar os pontos
- Transformar informação em conhecimento
- Passa idéia de algumas estruturas básicas de programação
- Dar embasamento para seguir aprendendo

Motivação

- Criação de Softwares
- Multidisciplinaridade
- Microcontroladores
- Hacker
- Séries:
  - MR Robot
  - Scorpion
  - Person of Interest

Noções de Arquitetura de Computadores

- Computador é uma máquina.
  - Limites de Processamento (CPU. GPU)
  - Limites de Memória Principal (RAM)
  - Limites de Armazenamento (HD, SSD)
- Exemplo de configurações:
  - Processador - Intel Core i5 650 3.20Ghz
  - Memória - Tamanho: 8GB 1333 MHz
  - Armazenamento - HD: 500 GB



Noções de Arquitetura de Computadores

- Capacidade no limite:
  - Quantidade de Programas abertos
  - Computador Lento
  - Tela azul no windows

Noções de Lógica de Programação

- “A maneira de pensar logicamente para estipular sequências de passos para a resolução de um problema, damos o nome de **lógica de programação**; à sequência narrativa desses eventos, damos o nome de **algoritmo**.” [1]
- Exemplos:
  - Planejamento do dia-a-dia.
  - Planejamento de Estudos
  - O que fez para atingir determinado objetivo
- Resposta (Almoço):
  - 1 - Peguei o prato
  - 2 - Coloquei X colheres de arroz
  - 3 - Coloquei X colheres de feijão ...

1 - https://kenzie.com.br/blog/logica-de-programacao/ Acesso em 09/03/2021

Noções de Lógica de Programação

- Dicas:
  - jogos de lógica:
    - https://www.geniol.com.br/logica/
    - https://www.somatematica.com.br/jogos/hanoi/
  - Detalhar objetivos em passos bem definidos:
    - Soma de 2 números:
      1. Inicie
      2. Crie número 1 e número 2
      3. pegue valor número 1
      4. pegue valor número 2
      5. some os dois números
      6. mostre o resultado
      7. Fim

Noções de Programação

Dica: Escrever o passo a passo e depois ir convertendo para código (planejar o programa primeiro, antes de ir codificando)

Noções de Programação [basic_main.c]

```c
//bibliotecas utilizadas no código
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>


//Estrutura da função principal

int main(int argc, char * argv[]){

    //Mostrar todos os parametros de linha de comando
    if(argc>1){
        int i = 0;
        for (i<0; i< argc;i++) printf("%s\n",argv[i])

    }

    //Retorna resultado da execução do main para o SO. 0 significa que nao teve erro.
    return 0;
}
```

Variáveis [ex1_main.c]

- Armazenam dados necessários
- Tipos primitivos:
  1. char - 1 Byte ou seja, 8 bits - caractere
  2. int - 2 Bytes - 32768 + 32767 - inteiros
  3. float - 4 Bytes - casas decimais
  4. void - vazio - nada
  5. double - 8 Bytes - casas decimais (precisão maior)

Variáveis [ex2_main.c]

- Vetores
- Structs
  - Agrupamento de tipos primitivos

INDICE	0	1	2	3	4	5
VALORES	10	7	8	3	9	6

Estruturas de condição [ex3_main.c]

```
if (condição realizada) {

   lista de instruções

}else if (condição realizada) {

  outra série de instruções

}
```

```
switch (variável) {

      case Valor1:

            Lista de instruções;

      break;

      case Valor2 :

            Lista de instruções;

      break;

      default:

       Lista de instruções;

}
```

Estruturas repetição [ex2_main.c]

```
while (condição realizada) {
   lista de instruções ;
}
```

```
for (contador; condição; modificação do contador) {
 lista de instruções ;
}
```

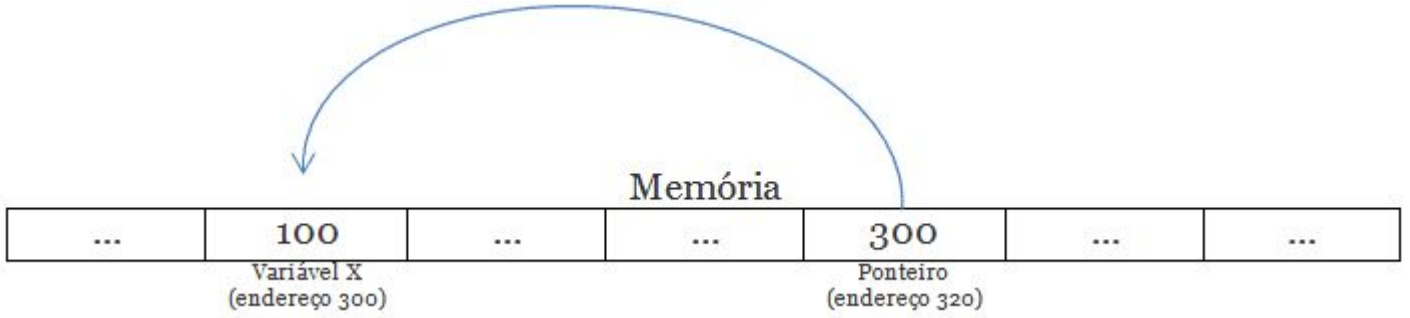
Funções [ex4_main.c]

- Copia as entradas; faz processamento; entrega a saída.

Dica: escrever o objetivo da função; Detalhar os passos e depois ir convertendo para código.

Ponteiro/Referência [ex5_main.c]

- Variável que contém o endereço de outra variável;
- Divisão da memória;
- Para que usar?
  - Alocação dinâmica de memória
  - Manipulação de arrays.
  - Para retornar mais de um valor em uma função.
  - Referência para listas, pilhas, árvores e grafos.

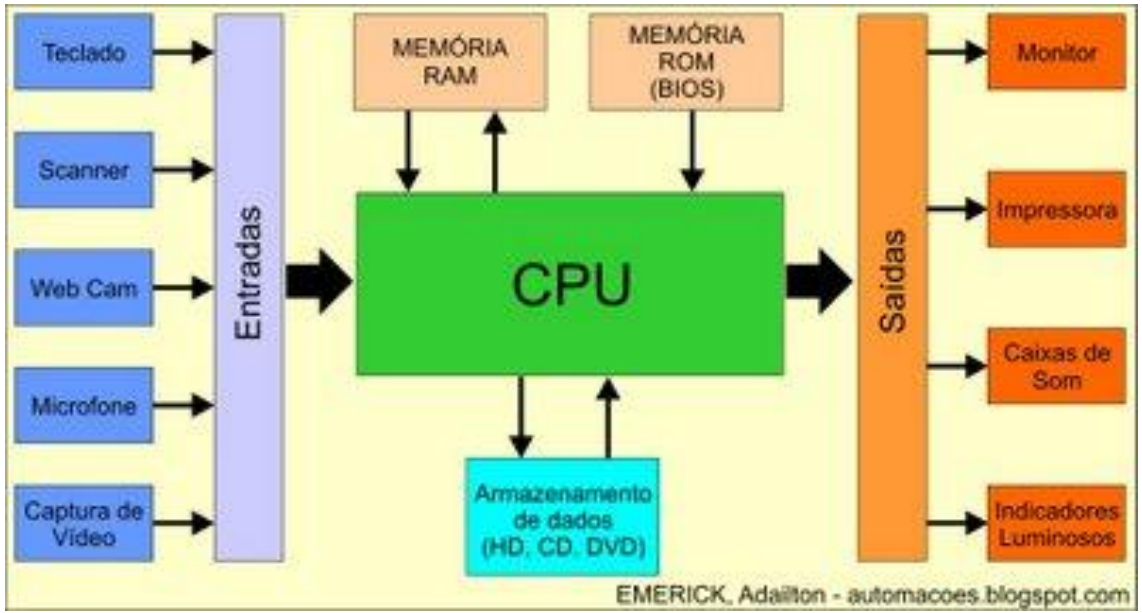


Alocação dinâmica [ex6_main.c]

- Alocação Estática
  - Definido no início da execução;
  - Subutilização ou Sobreutilização;
- Alocação Dinâmica
  - Especificar o tamanho e qual tipo será
  - Liberar o espaço alocado
  - Não perder o ponteiro

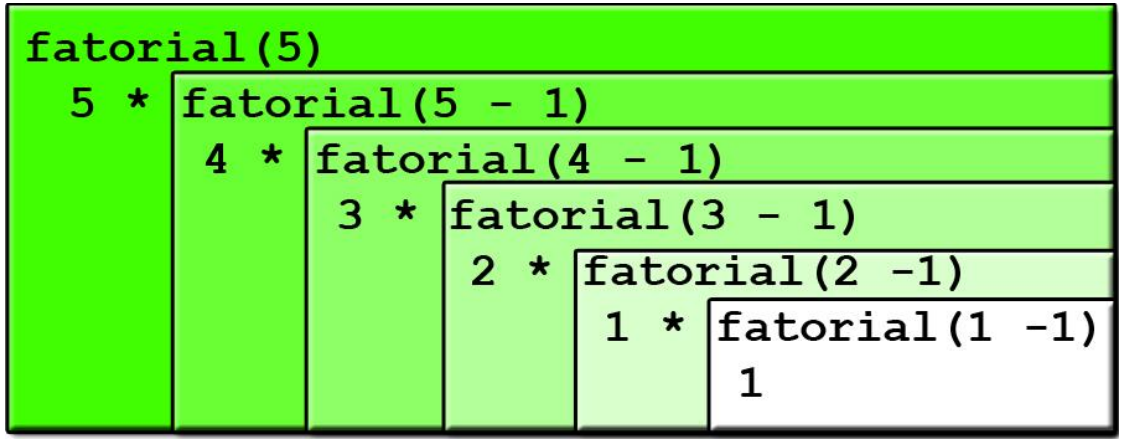
Arquivo [ex7_main.c]

- Levar valores da memória RAM para o HD
- Salvar para futuros processamentos



Recursividade [ex8_main.c]

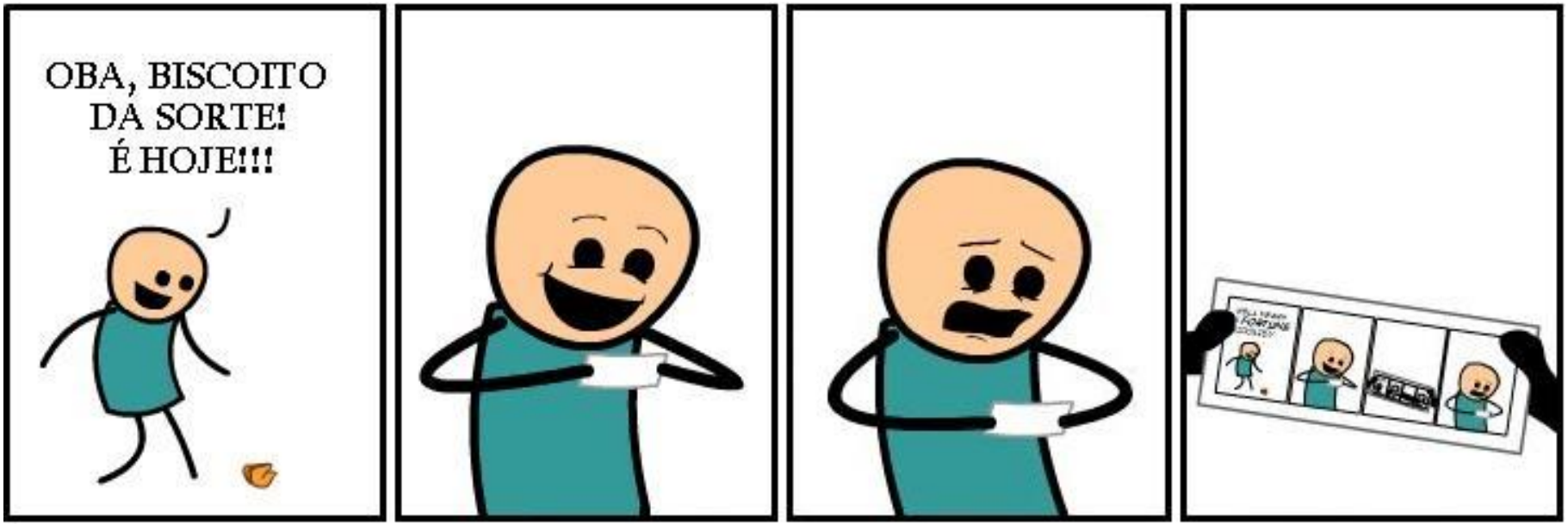
- Uma função chama ela mesmo;
- Funções matemáticas:
  - Fatorial
  - Sequência de Fibonacci



Sequência de Fibonacci

$$F_n = F_{n-1} + F_{n-2}$$
$$F_1 = 1; F_2 = 1$$

Recursividade



Debug (Depurar)

- Escrever no papel as entradas e a posição de memória
  - passo a passo e ir alterando valores das variáveis no papel
- Colocar print de variáveis e comparar com os valores que deveriam ter
- Procurar configurar o depurador da IDE ou ambiente que usarão para programar.

TP pds1

- Apresentar tutorial (https://aprendendoallegro.tk/index.php)
- Explicar formação de frame
- Explicar noções de vídeo

Noções de POO

- Paradigmas de programação
- Linguagens: Java, C++, C#, Python

Programação Orientada a Objetos

- POO
  - Abstração - Pensar nas classes do código (Quais atributos, funções, estrutura).
  - Encapsulamento - Acontece pois os atributos e funções ficam dentro do objeto.
  - Herança - Possibilidade de fazer relações de filiação.
  - Polimorfismo - Execuções diferentes do mesmo método.
- Quantidade de código reduzida;
- Facilidade de criar bibliotecas;
- Representações definidas e fáceis de enxergar;

POO

Problema: Criar um mini-sistema de controle de acesso de pessoas. o Sistema deve cadastrar pessoa, pedir nome e senha e falar se a pessoa tem acesso ou não.

POO

Problema: Criar um mini-sistema de controle de acesso de pessoas. O sistema deve cadastrar pessoa, pedir nome e senha e falar se a pessoa tem acesso ou não.

Classe: Pessoa

Atributos:

- Nome (string)
- senha (int)
- outro? Depende do sistema

Métodos:

- void setNome(string);
- strint getNome();
- void setSenha(int);
- int getSenha();

Classe: Sistema

Atributos:

- Pessoas (vetor)

Métodos:

- void cadastrarPessoa(nome, senha)
- bool acessoPessoa(nome, senha)

Indicações e Referências

- Me salva programação em C:
  - https://www.youtube.com/watch?v=MIzsyY505uo&list=PLf1IowbdbFIDP_aNIU7ZIe5Jc6LyoUmDa
- Linguagem C (site)
  - http://linguagemc.com.br/
- URI - Desafios de programação todos os níveis online
  - https://www.urionlinejudge.com.br/judge/en/login
- Atividades e soluções em C
  - https://oprofessorleandro.files.wordpress.com/2010/03/coletanea-de-exercicios-resolvidos-em-liguagem-c.pdf
- Orientação a objetos
  - https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264
- TP PDS1
  - https://aprendendoallegro.tk/index.php