

LÓGICA DE PROGRAMAÇÃO

Prof: Patrícia Lima Felismino
Bacharel em Ciência da Computação - UNICSUL

O QUE É LÓGICA DE PROGRAMA ÇÃO?

É o conjunto de regras e conceitos que permite aos programadores expressarem a solução de um problema de forma estruturada, para que um computador possa executá-la. É a base para desenvolver qualquer tipo de programa ou desenvolver qualquer tipo de sistema, independente da linguagem de programação utilizada.

É ATRAVÉS DA LÓGICA DE PROGRAMAÇÃO QUE OS ALGORITMOS SÃO CRIADOS.

- 1 . nós visualizamos um problema do mundo real que pode ser resolvido com o uso da tecnologia.
- 2. depois nós identificamos os possíveis padrões que existem nesse "problema", e passamos isso para um algoritmo (criamos o algoritmo depois de analisar os padrões encontrados e abstrairmos todas as etapas que existem nele).
- 3.agora, já sabendo com o que estamos lidando e como podemos resolver, vamos criar a solução usando de ferramentas tecnológicas (algoritmos, código...)
- 4.vamos experimentar descobrir os padrões do desafio passado na aula anterior e após isso, vamos criar o algoritmo que nos permita visualizar mais claramente com o que estamos lidando, para termos condições de

VAMOS SEGUIR UMA ORDEM NESSE PROCESSO:

- Identificar o problema
- Reconhecer os padrões que existem nele
- Organizar esses padrões numa ordem que chegue o mais próximo possível de organização focando numa ordem bem lógica.
- Por etapas após abstrair cada um dos padrões nós vamos sequenciar essa ordem visualmente. Vamos realmente escrever essa ordem.

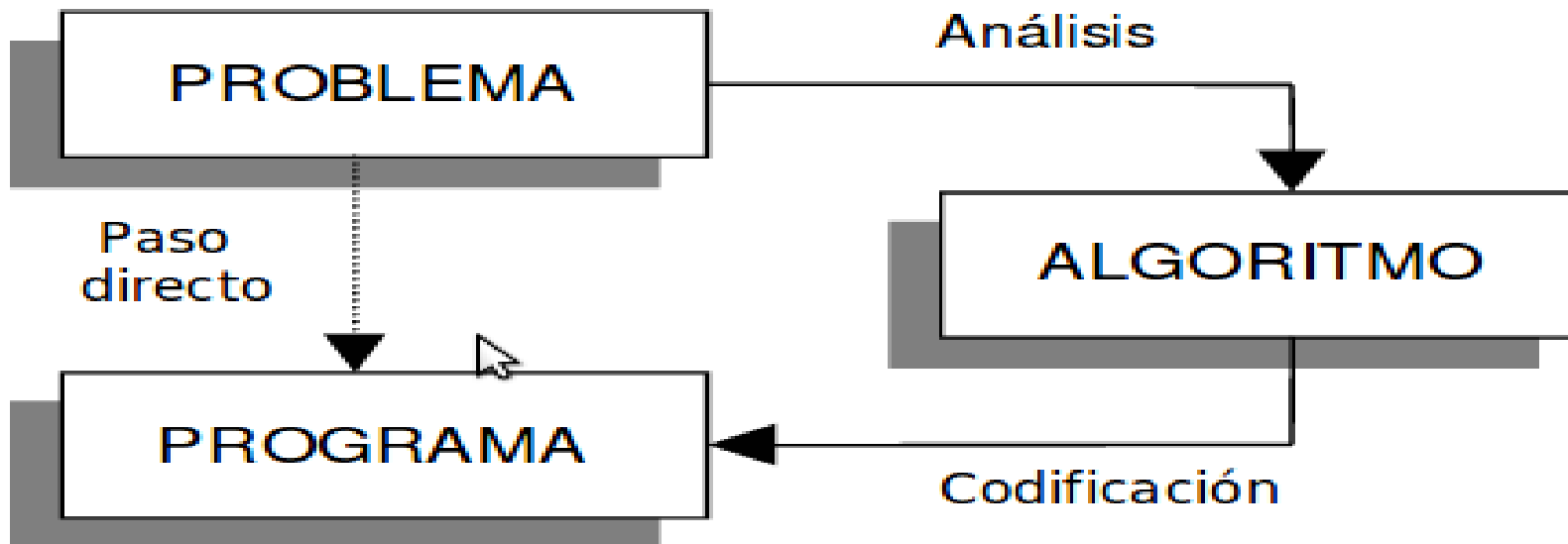
OBS! Sempre vamos ser responsáveis por localizar padrões, eles nos guiam para as resoluções de problemas. Quando sabemos com o que estamos lidando fica possível criar soluções e automatizações. Este é o melhor uso de lógica no dia a dia de um desenvolvedor.

ENTENDENDO ALGUNS PONTOS

- Em linhas gerais programas de Computador são algoritmos executados pelo computador.
- O computador é uma máquina que automatiza e necessita executar algoritmos para dar algum retorno das solicitações feitas. No sentido figurado, "o computador se alimenta puramente de lógica para conseguir pensar e executar seus processos internos".
- Ele apenas "se alimenta" de algoritmos computacionais, por isso nós convertemos através da lógica de programação, os problemas do mundo real para algoritmos computacionais.
- Todo algoritmo tem começo, meio e fim. Ou seja, são finitos.



VISÃO GERAL DE UM ALGORITMO



ALGUNS TERMOS TÉCNICOS IMPORTANTES NESSE MOMENTO

- **Compilador** - é um programa que traduz o código-fonte na linguagem de alto nível para uma de baixo nível.
- **Depurador** - é um software de computador usado para testar outros programas e fazer a depuração (encontrar os erros e defeitos seja na escrita ou versão de uma ferramenta).
- **Linguagem de programação** - é um sistema formal que permite aos desenvolvedores escrever instruções que um computador pode entender e executar após o processo de compilação.
- **Extensões** - Complementos ou plugins que adicionam funcionalidades extras.
- **Plugins** - Componente de software que adiciona funcionalidade específica a um software existente ou navegador (servem para estender uma funcionalidade já existente)
- **Navegador ou Browser** - software que permite ao usuário navegar na internet.
- **Bibliotecas ou Libs** - é uma coleção de subprogramas usados no desenvolvimento de softwares.
- **Terminal (CMD)** - prompt de comando, ferramenta que nos permite interagir com o sistema através de comandos.

VAMOS CONFIGURAR NOSSO AMBIENTE DE DESENVOLVIMENTO PYTHON.

- Para podermos ver lógica de programação com python precisamos antes ter um ambiente propício para isso e vamos criar ele nesta aula.
- Vamos instalar uma IDE (Ambiente de Desenvolvimento Integrado). Um software que reúne diversas ferramentas para o desenvolvimento de software em um único lugar, facilitando o trabalho dos desenvolvedores.
- **OBS!** Uma IDE contém um compilador ou interpretador, um depurador e outras ferramentas úteis para o desenvolvimento de software. Em resumo, uma IDE tem tudo que precisamos em um só lugar e ainda aceita que baixemos novas ferramentas nela caso seja necessário.

CONFIGURANDO NOSSO AMBIENTE DE DESENVOLVIMENTO (INSTALANDO AS FERRAMENTAS BASE)

- VSCode - IDE
- Python na versão mais recente (python.org) e marque a caixa de texto que adiciona a variável Path.
- Verifique a versão instalada do python e também se ele realmente foi instalado com o comando **python --version**
- Extensões do VSCode para suportar o Python - Com o comando **Ctrl+shift+x** ou clicando no ícone de extensões do VSCode.
- Pesquise na aba das extensões por python e instale a versão oficial da microsoft.
- Selecionando o interpretador python - Pressione **Ctrl+shift+P** para abrir a paleta de comandos e digite **Python: Select Interpreter** (esse comando é para o caso de na instalação do python não tiver colocado a variável PATH).
- Vamos testar criando uma pasta -> um novo arquivo com a extensão **.py** (**vamos ver as formas de se abrir um projeto no vscode**)
- Vamos nos livrar da "maldição com nosso primeiro Hello World! (`Print("Hello, World!");`)
- **Obs!** Falando sobre espaços, indentação, comentários e no impacto que podem ou não causar - clean code / boas práticas.

COMANDOS NO TERMINAL DO VSCODE

- OBS! Estes comandos podem ser usados em outros terminais além do terminal do vscode.
- OBS! O terminal faz distinção de letras maiúsculas e minúsculas e também de espaços, isso afeta de forma direta os comandos e proporcionalmente a execução deles.
- **mkdir** - criar pasta
- **ls** - listas arquivos que estão dentro de uma pasta
- **clear** ou **cls** ambos usados no terminal do vscode e no terminal Shell para limpar a tela..
- **Cd** - colocamos um caminho / URL
- **Cd ..** - volta uma pasta
- **Code .** Abre o arquivo ou pasta no vscode
- OBS! Vamos testar o terminal do vscode e outro.
- OBS! O cmd não reconhece o **clear** mas reconhece o **cls**, já o cmd do vscode reconhece os dois comandos
- **ctrl+"** abre o terminal no vscode
- **CTRL+X / CTRL+C / CTRL+V**

OBRIGADA!