

FIAP GRADUAÇÃO

# ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Computational Thinking

PROF. EDUARDO GONDO

# Agenda

- ▶ Conteúdo do curso
- ▶ Metodologia
- ▶ Competências
- ▶ Habilidades
- ▶ Algoritmos — Introdução

## Currículo

Mestre em Ciência da Computação - IME USP 2002

Área de pesquisa: algoritmos de aproximação, teoria dos grafos e otimização combinatória

Professor desde 2002 dos cursos de Ciência da Computação, Sistemas de Informação e Tecnologia e Desenvolvimento de Sistemas

Estou na FIAP desde 2008 e já ministrei as seguintes disciplinas: algoritmos, estrutura de dados, desenvolvimento web, componentes distribuídos entre outras.

Trabalho com análise e desenvolvimento web, mobile e integração de sistemas.

## Conteúdo do curso

- ▶ Algoritmo - introdução e estrutura
- ▶ Entrada e saída
- ▶ Variáveis, operadores aritméticos e de comparação
- ▶ Comandos condicionais e operadores booleanos
- ▶ Comandos de repetição
- ▶ String
- ▶ Listas
- ▶ Dicionários
- ▶ Funções e escopo de variáveis
- ▶ Tratamento de exceções
- ▶ Arquivos
- ▶ Matrizes
- ▶ Conexão com banco de dados
- ▶ API

## Bibliografia Básica

- ▶ PUGA, Sandra; GOMES, RISSETTI Gerson. Lógica de Programação e Estrutura de Dados com Aplicações em Java. 2ª ed. São Paulo: Pearson Education do Brasil, 2013.
- ▶ SZWARCFITER, J.L.; Markenzon, L. Estruturas de Dados e seus Algoritmos. 3ª ed. São Paulo: LTC, 2015.
- ▶ ASCENCIO, A.F.G.; CAMPOS, E.A.V. Fundamentos da Programação de Computadores. 3ª ed. São Paulo: Pearson Education do Brasil, 2013. <sup>1</sup>

---

<sup>1</sup>acervo online

## Bibliografia Complementar

- ▶ Python Programming For Beginners: Learn The Basics Of Python Programming (Python Crash Course, Programming for Dummies) (English Edition). Kindle
- ▶ Python: 3 Manuscripts in 1 book: - Python Programming For Beginners - Python Programming For Intermediates - Python Programming for Advanced (English Edition). Kindle
- ▶ LEME, E. S. (organizador). Programação de Computadores. São Paulo: Pearson Education do Brasil, 2014. \*
- ▶ GUEDES, S. (organizador). Lógica de Programação Algorítmica. São Paulo: Pearson Education do Brasil, 2014. \*
- ▶ FORBELLONE, A.L.V; EBERSPACHER, H.F. Lógica de Programação: a construção de algoritmos e estruturas de dados. 3ª ed. São Paulo: Pearson Education do Brasil, 2005. \*

(\*) *acervo online*

## Eventos importantes em cada semestre

- ▶ 3 Checkpoints
- ▶ Challenge
- ▶ Global Solution



# Introdução

Em qual grande área do conhecimento podemos enquadrar o curso de Análise e Desenvolvimento de Sistemas?

Se você respondeu **Ciências Exatas**, está absolutamente certo! Mas o desenvolvimento de sistemas serve a todas as áreas do conhecimento! Constantemente somos responsáveis por entender o funcionamento de todos os tipos de negócio relacionados também as Ciências Humanas e Biológicas!

# Computational Thinking

E o que aprenderemos nesta disciplina?

# Computational Thinking

E o que aprenderemos nesta disciplina?

- ▶ Resolver PROBLEMAS
- ▶ dos mais variados tipos, sempre envolvendo informações ou dados!
- ▶ as informações podem ser números, texto ou uma combinação delas.
- ▶ no início resolveremos problemas envolvendo números (matemática básica)
- ▶ depois trabalharemos com coleções de números e textos (string)
- ▶ As soluções desses PROBLEMAS são chamados de **Algoritmos!**

# | Algoritmos

- ▶ E o que é Algoritmo?

# Algoritmos

## ▶ E o que é Algoritmo?

Algoritmo: é uma sequência de instruções que devem ser executadas com o objetivo de resolver o problema em questão!

# Algoritmos

- ▶ E o que é Algoritmo?

Algoritmo: é uma sequência de instruções que devem ser executadas com o objetivo de resolver o problema em questão!

- ▶ Mas não estudaremos todo tipo de algoritmos, estamos interessados nos algoritmos que são executados pelo computador ou algum dispositivo compatível!

# Algoritmos

- ▶ E o que é Algoritmo?

Algoritmo: é uma sequência de instruções que devem ser executadas com o objetivo de resolver o problema em questão!

- ▶ Mas não estudaremos todo tipo de algoritmos, estamos interessados nos algoritmos que são executados pelo computador ou algum dispositivo compatível!
- ▶ Ah, então preciso escrever o algoritmo de forma que o computador entenda, não é Professor???

# Algoritmos

- ▶ E o que é Algoritmo?

Algoritmo: é uma sequência de instruções que devem ser executadas com o objetivo de resolver o problema em questão!

- ▶ Mas não estudaremos todo tipo de algoritmos, estamos interessados nos algoritmos que são executados pelo computador ou algum dispositivo compatível!
- ▶ Ah, então preciso escrever o algoritmo de forma que o computador entenda, não é Professor???
- ▶ Sim, caros alunos! Para isso precisaremos de algo chamado **Linguagem de Programação!**



# Linguagem de Programação

Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software) de computador.

Veja um pequeno exemplo de um algoritmo escrito na linguagem de programação Java:

```
1 public class Exemplo {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Ola Mundo!");  
5     }  
6 }
```

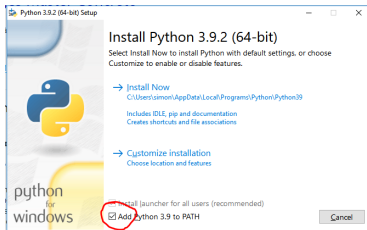
Damos o nome de **código-fonte** o algoritmo escrito em uma linguagem de programação. Usaremos Python para escrever nossos algoritmos.

## Python - características

- ▶ lançada em 1991 na Holanda por Guido van Rossum
- ▶ seu nome (*Python*) teve a sua origem no grupo humorístico britânico Monty Python
- ▶ Open Source e compatível com a GPL (General Public License)
- ▶ Imperativa, orientada a objetos e funcional (ou seja, abrange todos os principais paradigmas da programação)
- ▶ usada nas mais diversas plataformas: desktop, web, mobile, dispositivos embarcados, análise de dados, etc
- ▶ parte integrante dos sistemas operacionais Linux

## Python - instalação

- ▶ usaremos a versão 3.x do interpretador do Python
- ▶ faça o download do python em <https://www.python.org/downloads/>
- ▶ na hora da instalação, marque a opção de adicionar o Python no Path como mostrado na figura abaixo:



## Python - instalação

- ▶ depois de finalizada a instalação
- ▶ abra um Command Prompt (terminal) e digite python no windows ou python3 no linux
- ▶ deve aparecer algo do tipo<sup>2</sup>:

```
$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
```

<sup>2</sup>caso não apareça, é provável que você tenha marcado a opção que adiciona o Python no PATH do computador

# Python - Ola Mundo

Depois digite nessa tela:

```
1 print("Ola Mundo")
```

- ▶ você acabou de escrever seu primeiro programa em Python!!!!
- ▶ Python é uma linguagem simples, não necessita de cabeçalhos, nem separadores de linhas ou símbolos de aberturas de blocos como em muitas outras linguagens de programação
- ▶ basta digitar as instruções diretamente no interpretador e elas são executadas
- ▶ porém, o modo mais comum é criarmos um arquivo texto contendo nosso código-fonte
- ▶ salvamos com a extensão <nome>.py e executamos o comando `python <nome>.py`
- ▶ Python é **case sensitive**, ou seja, há diferenciação entre maiúsculas e minúsculas

## Editor de texto

- ▶ para escrevermos nossos algoritmos em Python precisaremos um editor de texto puro
- ▶ nas aulas usarei o Visual Studio Code (vscode) para escrever os algoritmos
- ▶ você pode fazer o download em <https://code.visualstudio.com/download>
- ▶ devemos instalar o plugin Python no Visual Studio Code para ajudar no desenvolvimento
- ▶ mas sinta-se livre para usar o que for mais conveniente: atom, vim, notepad++, sublime, etc
- ▶ existem também as IDE - Integrated development environment: pyCharm, eclipse, etc
- ▶ também é possível desenvolver na nuvem, existem algumas ferramentas disponíveis: Google Colab, Replit, etc

## Metodologia de ensino

- ▶ explicação dos conceitos de desenvolvimento de algoritmos através de problemas
- ▶ resolução desses problemas utilizando a linguagem de programação Python
- ▶ vamos ter alguns problemas envolvendo matemática mas também apresentaremos jogos, aplicações em imagens entre outros
- ▶ fixação dos conceitos através de lista de exercícios ou exercícios de fixação
- ▶ discussão dos problemas, exercícios e soluções apresentadas

# | Problemas que iremos explorar durante o ano

Alguns problemas são fundamentais no aprendizado da programação:

- ▶ média
- ▶ contagem
- ▶ máximo e mínimos
- ▶ aplicação de cálculos matemáticos (porcentagem por exemplo)
- ▶ trabalhar com conjuntos de dados
- ▶ fazer buscas
- ▶ implementar e aplicar as estruturas de dados do tipo pilha e fila

Ao longo do ano, apresentaremos a solução desses problemas e mais alguns outros



## Dicas para esta disciplina

- ▶ Assistir e participar das aulas: perguntando e fazendo as tarefas
- ▶ Trazer lápis/caneta e papel para as aulas e anotar tudo que for colocado no quadro-branco
- ▶ Fazer os exercícios propostos primeiro no papel e depois passar para o computador
- ▶ Perguntar sempre, por menor que seja sua dúvida!

## I Recapitulando alguns conceitos

**algoritmo** sequência de passos para resolver um problema

**linguagem de programação** um conjunto de instruções que são usadas para escrever um algoritmo

**código-fonte** algoritmo escrito em uma linguagem de programação

**python** linguagem de programação que será usada para desenvolver nossos algoritmos

# Enquete

1. Nome e idade?
2. Já fez algum curso superior (mesmo que não tenha concluído)? Qual?
3. Por quê escolheu Tecnologia da Informação (TI)?
4. Dentro de TI, já sabe no que você gostaria de trabalhar?
5. Quais são seus hobbies?

# I Copyleft

Copyleft © Prof. Eduardo Gondo Todos direitos liberados. Reprodução ou divulgação total ou parcial deste documento é liberada.