

Aula 1

Professores:

Dante Corbucci Filho

Leandro A. F. Fernandes

Conteúdo:

- Introdução
- Ambiente de Desenvolvimento
- Correção de Exercícios

Objetivos

- Capacitar o aluno no uso de uma linguagem de programação procedural (neste caso Python) para:

- Implementar
- Executar
- Testar

as diferentes soluções concebidas para resolver um problema

Objetivos

- Capacitar o aluno no uso de uma linguagem de programação procedural (neste caso Python) para:

- Implementar
- Executar
- Testar

as diferentes soluções concebidas para resolver um problema

- Ao final da disciplina o aluno deverá estar apto a:

- Implementar
- Testar
- Analisar
- Documentar

programas de computador em uma linguagem imperativa

Ementa

- (1) Introdução
- (2) O Ambiente de Desenvolvimento de Programas
- (3) Variáveis, Tipos e Comandos Básicos
- (4) Estruturas de Controle: Sequência, Seleção e Repetição
- (5) Subprogramação: Funções, Passagem de Parâmetros e Recursividade
- (6) Representação de Dados na forma de:
 - (6.1) Listas
 - (6.2) Vetores e Matrizes
 - (6.3) Strings e Tuplas
 - (6.4) Arquivos
 - (6.5) Conjuntos
 - (6.6) Dicionários
- (7) Noções de Complexidade de Algoritmo

Livros Texto

- Summerfield, M.
Programação em Python 3
Uma Introdução Completa à Linguagem Python
Editora Alta Books
(2013)
- Barry, P. & Griffiths, D.
Use a Cabeça! Programação.
Editora Alta Books
(2010)
- Barry, P.
Use a Cabeça! Python.
Editora Alta Books
(2012)

Algoritmo e Estrutura de Dados [Guim 94]

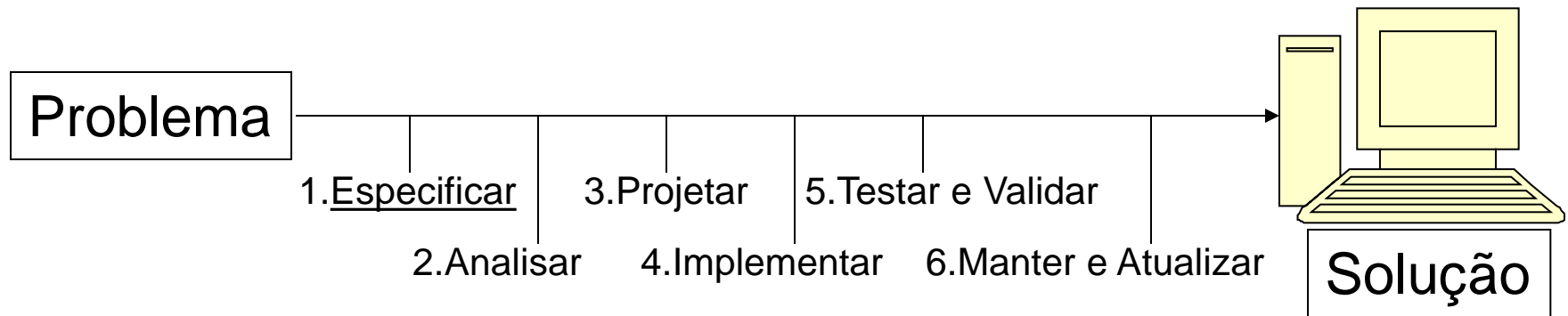
- Estrutura de dados é um modo particular de armazenamento e organização de dados em um computador de modo que possam ser usados eficientemente.
- Algoritmo é a descrição de um padrão de comportamento, especificado em termos de um conjunto bem definido e finito de ações primitivas que podem ser executadas.

Programa = Algoritmo + Estruturas de Dados

Programação Estruturada [Guim 94]

- Metodologia de projeto de programas que visa:
 1. Facilitar o desenvolvimento dos programas
 2. Facilitar a leitura (entendimento) dos programas
 3. Permitir a validação *a priori* dos programas
 4. Facilitar a manutenção e modificação dos programas

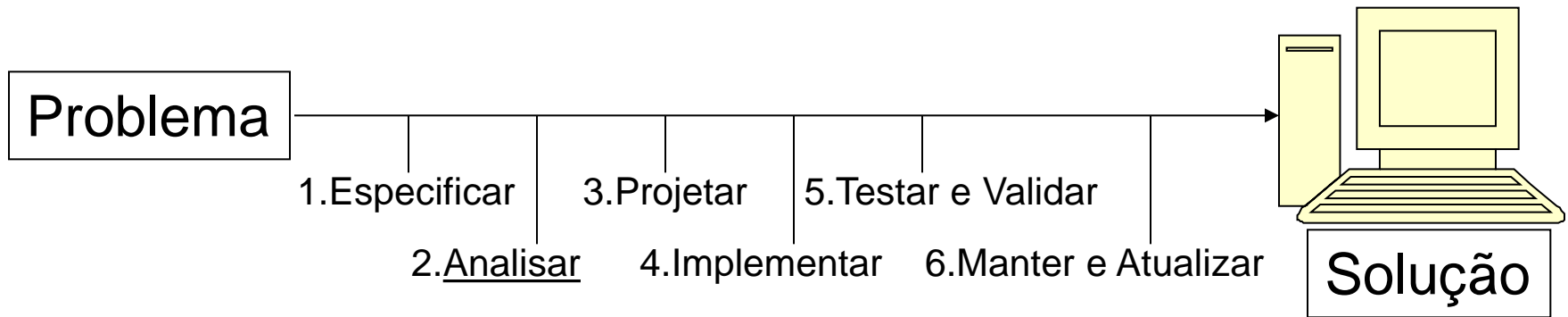
Ciclo de Vida do Software [Koff 94]



1. Especificar os Requisitos do Problema

- Preparar uma especificação completa e não ambígua.

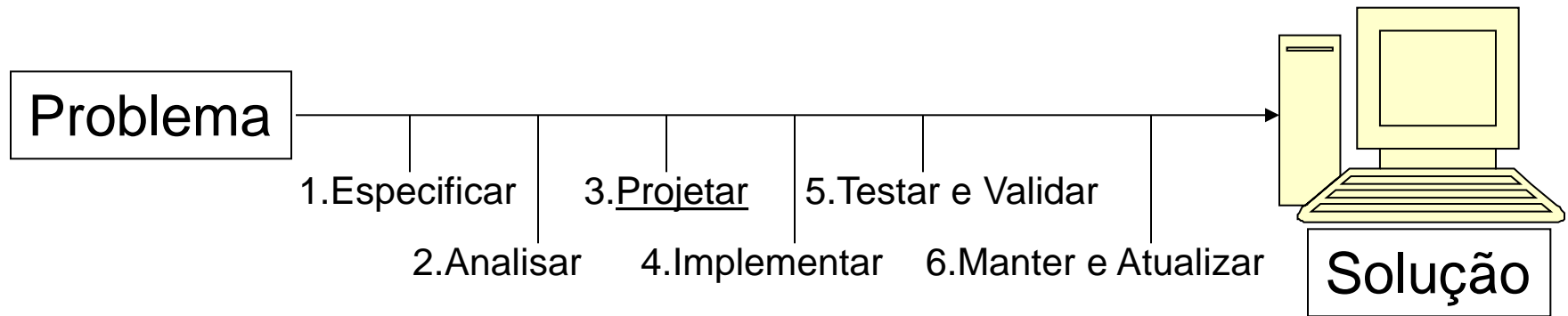
Ciclo de Vida do Software [Koff 94]



2. Analisar o Problema

- Entender o problema,
- Avaliar soluções alternativas,
- Escolher a solução mais adequada.

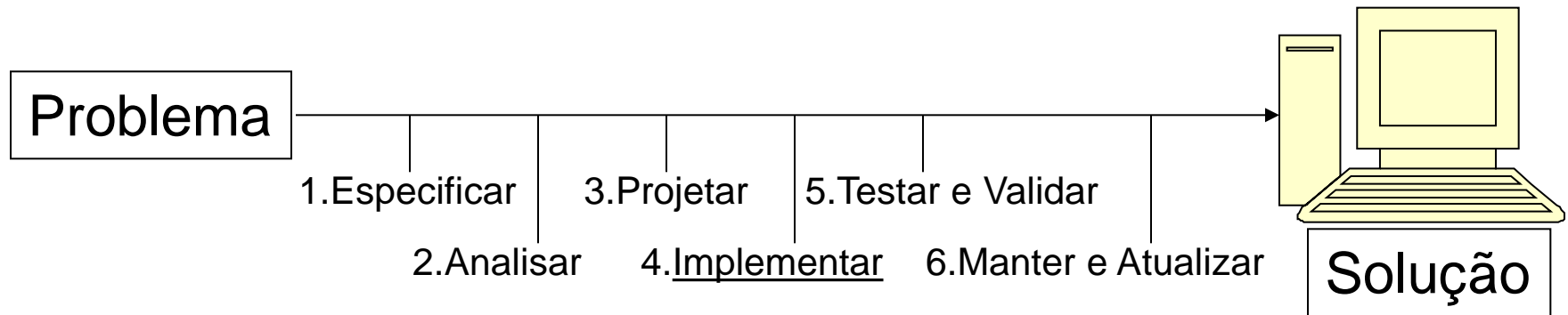
Ciclo de Vida do Software [Koff 94]



3. Projetar o Programa para Solucionar o Problema

- Fazer projeto de cima-para-baixo (*top-down*) do sistema,
- Para cada módulo, identificar as principais estruturas de dados e subprogramas associados,
- Desenvolver algoritmos e estruturas de dados dos subprogramas.

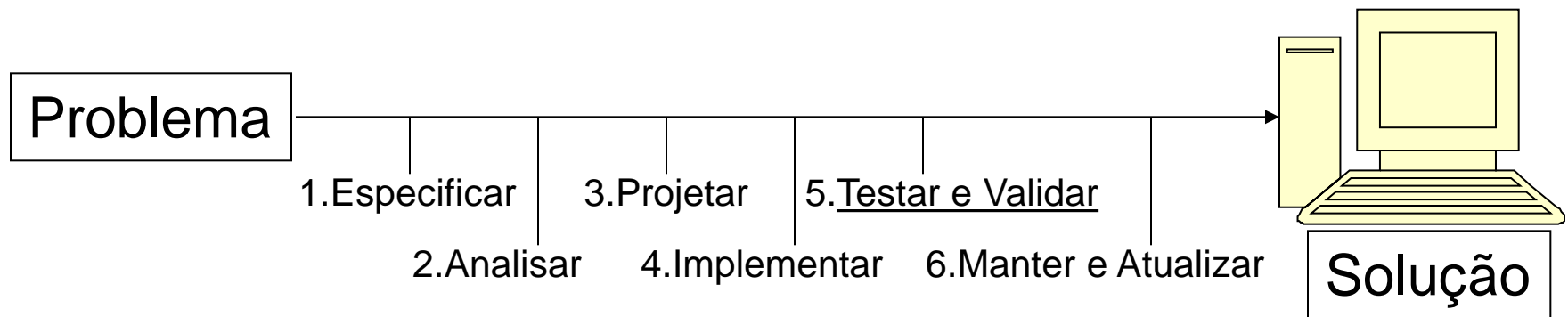
Ciclo de Vida do Software [Koff 94]



4. Implementar o Projeto

- Codificar a solução,
- Corrigir erros de codificação.

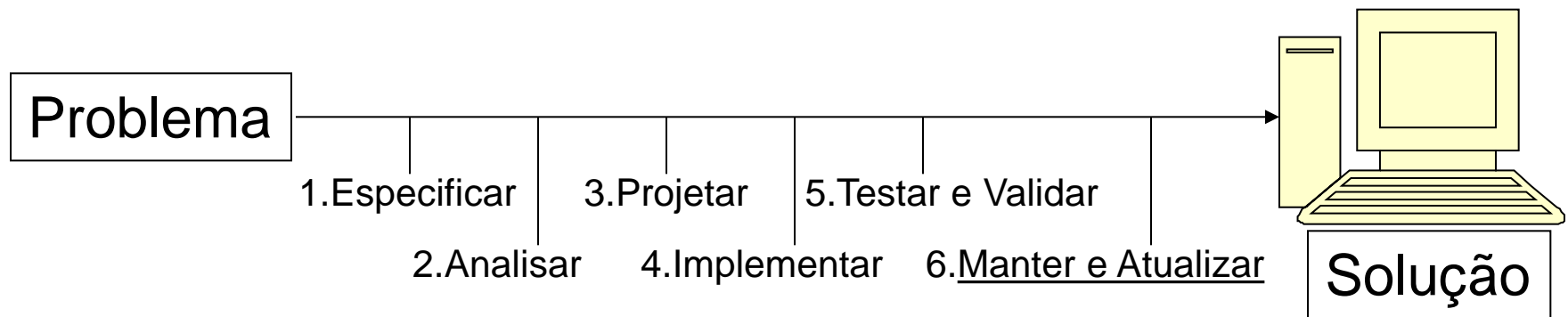
Ciclo de Vida do Software [Koff 94]



5. Testar e Validar o Programa

- Testar o código e validá-lo, se correto.

Ciclo de Vida do Software [Koff 94]



6. Manter e Atualizar o Programa

- Executar o sistema,
- Avaliar seu desempenho,
- Remover novos erros identificados, assim que detectados,
- Realizar modificações de forma a manter o sistema atualizado,
- Validar as modificações.

A Linguagem Python

- Python foi criado idealizado na década de 80 e sua implementação começou em 1989, por Guido van Rossum
 - 2000: Python 2
 - 2008: Python 3
- Características da linguagem:
 - Multiparadigma (estruturado, orientado a objetos, funcional e orientado a aspectos)
 - Multiplataforma (Windows, Linux, iOS, etc.)
 - Interpretada
 - Tipagem dinâmica
 - Gerência de memória automatizada (coletor de lixo)



Guido van Rossum,
o criador do Python

O Ambiente de Desenvolvimento

- Ambiente de desenvolvimento de software Python
 - Usaremos Python 3
- O ambiente de desenvolvimento Python inclui:
 - Um amplo conjunto de Interfaces de Programação de Aplicações (APIs, do inglês “*Application Programming Interfaces*”)
 - Ferramentas de compilação e depuração de código
- Python 3 não é um Ambiente Integrado de Desenvolvimento (IDE, do inglês “*Integrated Development Environment*”)
 - A linguagem Python por si só não oferece editores de código ou ambiente de programação
 - Usaremos PyCharm como IDE (editor + ambiente de programação)

Instalação do Python 3

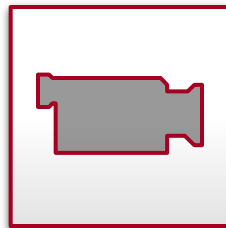
(<https://www.python.org>)

e

do PyCharm

(<https://www.jetbrains.com/pycharm>)

Clique no botão para assistir ao tutorial:

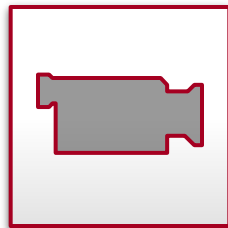


Correção de Exercícios

- Faça o Chinês, ou Teste de Mesa
 - Validação do programa “em papel”
 - Represente graficamente as variáveis utilizadas e acompanhe a atualização de seus valores em função da execução passo-a-passo do algoritmo implementado
 - Você desempenha a tarefa do compilador e do executor do programa!!!
- Juiz Remoto (“*Online Judge*”):
 - Sistema web que compila e executa um código fonte submetido a julgamento. Ele testa a saída gerada pelo programa submetido contra um banco de testes, com entradas e saídas previamente elaboradas.
 - Avalia se, para toda entrada conhecida pelo juiz, o programa submetido produz saída idêntica à esperada
 - Usaremos o URI Online Judge

Cadastro no URI Online Judge (<http://urionlinejudge.com.br>) **e** **Submissão de Um Exemplo**

Clique no botão para assistir ao tutorial:



Referências Autorais

- Estas aulas foram preparadas a partir de uma primeira versão, em Pascal, da Disciplina Fundamentos de Programação, do Curso de Tecnologia em Sistemas de Computação do Cederj.
- Autores da versão Pascal:
Prof. Alexandre Plastino de Carvalho e
Prof. Dante Corbucci Filho
- A versão Pascal foi produzida baseada no livro:
Título: **Pascal e Técnicas de Programação**
Autores: Eber Assis Schmitz e
Antônio Aníbal de Souza Teles
Editora: LTC

Aula 1

Professores:

Dante Corbucci Filho

Leandro A. F. Fernandes

Conteúdo:

- Introdução
- Ambiente de Desenvolvimento
- Correção de Exercícios