

Aula #07

Interpretação e Projeto de Programas

O que é um projeto *top-down* e *bottom-up*? Quais são boas estratégias de leitura e interpretação de enunciados? Como aplicar uma metodologia *top-down*?

Projeto de Programas

Projetar um programa não é uma tarefa trivial!

A maioria dos sistemas contempla enunciados complexos e requisitos robustos. Não é fácil identificar e traduzir um **problema do mundo real**, em linguagem natural, para uma **solução do mundo virtual**, em linguagem de programação.

Porém, existem algumas estratégias que podem nos ajudar!

Projeto de Programas

Naturalmente, existem disciplinas da **linha de engenharia de software** que tratam especificamente sobre projeto de desenvolvimento de software.

O nosso objetivo aqui não é estudar em detalhes e formalmente as estratégias de projeto; mas sim ter uma visão ampla das mesmas, verificando como colocar alguns de seus conceitos em prática imediatamente.

O nosso objetivo é mais prático do que teórico!

Projeto de Programas

De qualquer forma, precisamos de um pouco de teoria antes da prática.

Existem duas abordagens de projeto de programas muito difundidas:

Top-Down

Bottom-Up

Essas abordagens, de forma geral, indicam como vamos pensar e organizar um dado problema para o processo de desenvolvimento de uma solução.

Vamos ver as principais características de cada uma delas!

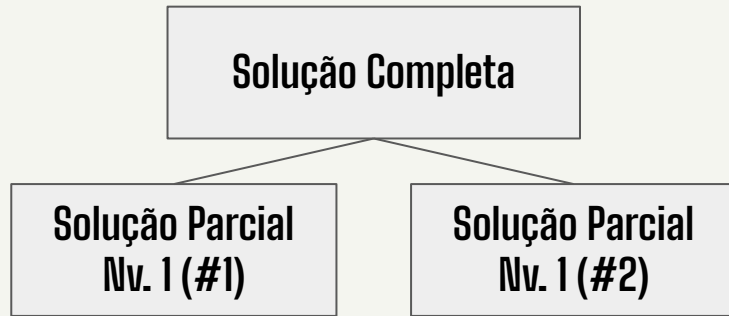
A Abordagem *Top-Down*

Solução Completa

Resolver um problema através da abordagem *top-down* consiste em entender o mesmo, inicialmente, como um **monólito**.

Ou seja, eu parto da **ideia geral do programa**, do problema, em contexto amplo, que é seu objetivo resolver.

A Abordagem *Top-Down*

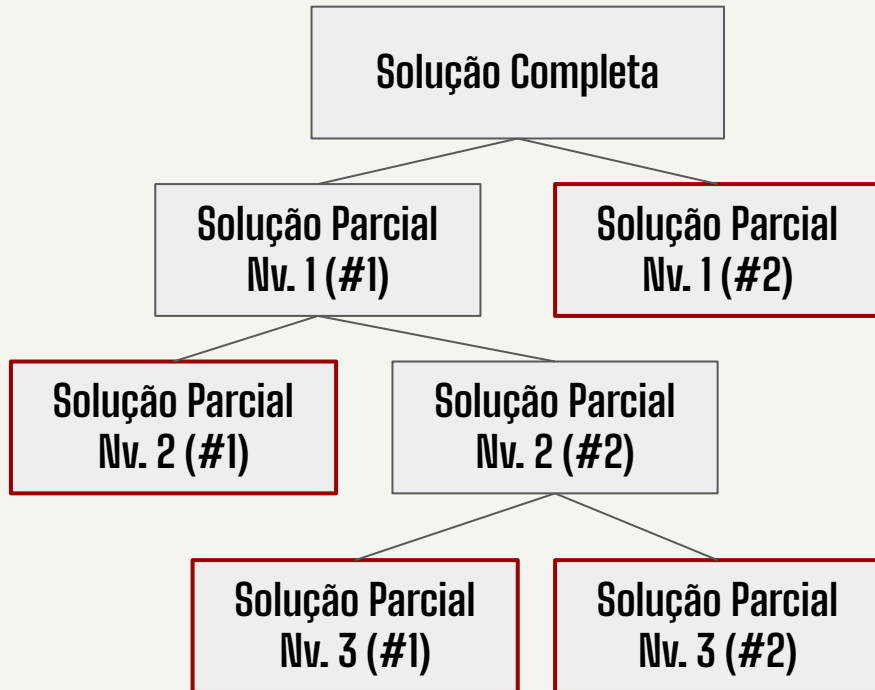


Juntas, as duas soluções parciais geram a solução completa

A cada iteração de análise, verifica-se se os problemas e soluções de mais alto nível sendo considerados podem ser quebrados em problemas mais simples.

Geralmente, para simplificar, busca-se fazer **cisões binárias** nos problemas/soluções!

A Abordagem *Top-Down*



O processo se repete quantas vezes forem necessárias até que um conjunto de problemas e soluções atômicas seja encontrado.

Desse ponto em diante, desenvolvem-se as soluções para os problemas de mais baixo nível, as **unido e adaptando** para gerar soluções de mais alto nível.

A Abordagem *Top-Down*

Um exemplo simples...

Uma loja requisita um **programa de controle de vendas**. Você deve identificar os produtos comprados pelo seu ID e valor. Em uma compra, cada produto tem uma certa quantidade e a multiplicação da quantidade pelo valor somada para todos os produtos resulta no valor da compra. **Dados os produtos, seu valor e quantidade, apresente a nota de compra com o valor total que efetiva a venda.**

A Abordagem *Top-Down*



A Abordagem *Top-Down*

A abordagem *top-down* é **especialmente adequada quando temos uma visão geral do problema e da solução do mesmo**, mas temos liberdade de determinar os passos intermediários (cumprindo premissas definidas) para alcançar os objetivos finais.

Devido a isso, a abordagem *top-down* é **preferida quando o paradigma de programação escolhido é o procedural**, pois se adequa muito bem às funções e rotinas criadas através do mesmo.

A Abordagem *Bottom-Up*

Tratar um problema via uma abordagem *bottom-up* consiste em ter uma visão clara e detalhada das **partes individuais** que formam um programa complexo.

Solução Parcial
Nv. 1 (#2)

Solução Parcial
Nv. 2 (#1)

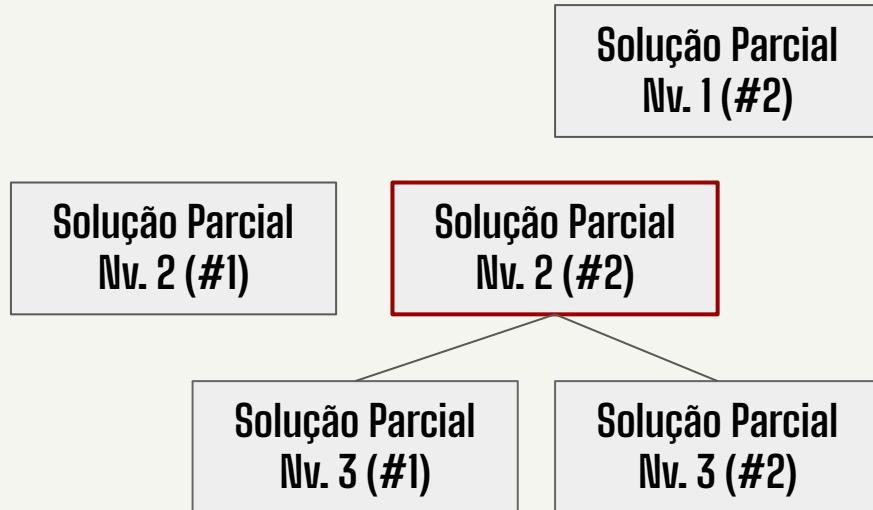
Solução Parcial
Nv. 3 (#1)

Solução Parcial
Nv. 3 (#2)

Assim, eu parto dos **elementos que compõem um programa**, e, ao torná-los coesos entre si, chego a uma solução completa.

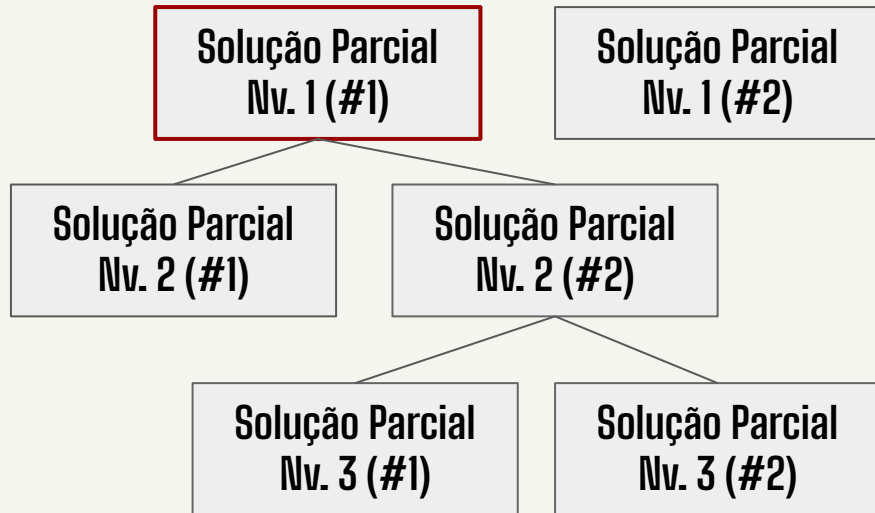
A Abordagem *Bottom-Up*

Assim, a cada iteração de projeto, verifica-se **o que pode ser criado a partir do que foi desenvolvido** nas iterações anteriores.



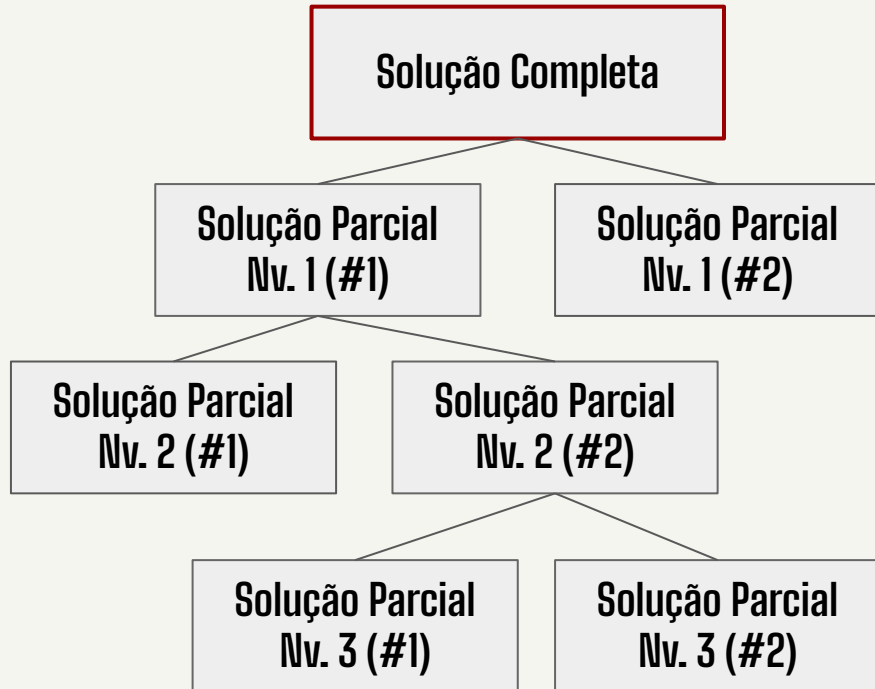
Note que, **nem sempre tudo o que foi desenvolvido nas iterações anteriores será imediatamente utilizado!**

A Abordagem *Bottom-Up*



Sendo assim, na **próxima iteração de projeto/desenvolvimento** teríamos...

A Abordagem *Bottom-Up*



E finalmente, **na última iteração de desenvolvimento/projeto...**

A Abordagem *Bottom-Up*

Um exemplo simples...

Uma loja requisita um programa de controle de vendas. Você deve identificar os produtos comprados pelo seu **ID e valor**. Em uma compra, cada produto tem uma certa **quantidade** e a multiplicação da quantidade pelo valor somada para todos os produtos resulta no valor da compra. Dados os produtos, seu valor e quantidade, apresente a nota de compra com o **valor total** que efetiva a venda.

A Abordagem *Bottom-Up*



Comparação de Abordagens

<i>Top-Down</i>	<i>Bottom-Up</i>
O projeto parte do problema completo, e iterativamente o divide em partes menores.	O projeto parte da definição de operações e objetos atômicos, compostos para solucionar problemas maiores.
Muito utilizado para programas escritos com linguagens estruturadas.	Muito utilizado para programas escritos em linguagens orientadas a objetos.
Partes individuais de cada iteração de projeto/desenvolvimento podem ser individualmente analisadas.	Muito associado a conceitos de herança e encapsulamento, tornando as partes do programa muito mais interdependentes.
Tipicamente, os enunciados tem menos detalhes, focam nas operações gerais.	Tipicamente, os enunciados tem muitos detalhes, focam em muitas características e requisitos de cada operação.

Colocando em Prática

Como trabalhamos com a linguagem C e um paradigma de programação estruturado, **a abordagem de projeto e desenvolvimento *top-down* é uma excelente alternativa.**

Vamos considerar o enunciado de um trabalho antigo da disciplina (provido a seguir) e vamos discutir possíveis projetos para eles.

Colocando em Prática

Verificador de Ortografia

O verificador de ortografia recebe um texto de entrada e verifica se suas palavras estão em um dicionário. A saída deve reproduzir fielmente o texto de entrada (inclusive espaços e quebras de linha), mas colocando entre colchetes as palavras não encontradas no dicionário.

Consideram-se como palavras as sequências contíguas de letras (A-Z, a-z) com ou sem acentos e as cedilhas; os demais caracteres (números, espaços e outros símbolos) não fazem parte de palavras.

Exercício #07

Faça um projeto simplificado (rascunho de projeto) a partir da abordagem *top-down* considerando o enunciado da primeira avaliação da disciplina (A0):

MÁQUINAS DE PELÚCIAS

Por fim, analise se o seu projeto representa o que foi entregue para a avaliação.

Obrigado!

Vinícius Fülber Garcia
inf.ufpr.br/vinicius
vinicius@inf.ufpr.br