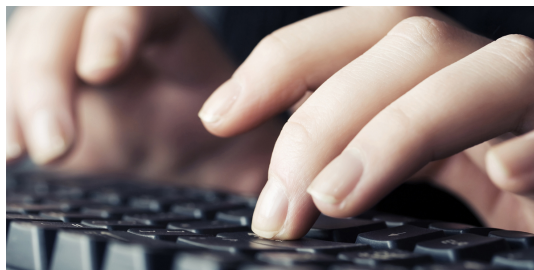


Arquitetura de Sistemas

Aula 8: Especificação de componentes

INTRODUÇÃO

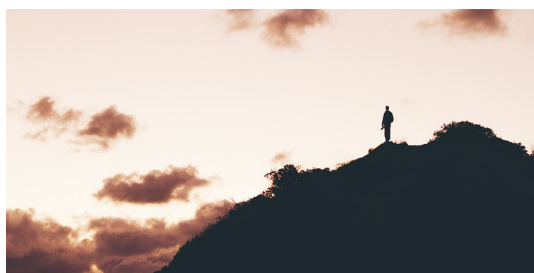


Um dos pontos mais críticos no desenvolvimento de um sistema é o momento em que os desenvolvedores têm que traduzir os requisitos levantados junto às partes interessadas, em especificações que façam sentido para os programadores, já que nem sempre as informações repassadas pelas partes interessadas vão estar de acordo com as normas para desenvolvimento.

Nesta aula, você verá um conjunto de ações necessárias para que a especificação consiga, de forma concreta, representar todas as informações levantadas de maneira consistente para os programadores. Você verá também a apresentação do conceito de especificação como a tradução dos elementos de requisitos para elementos de programação.

Isso significa que apresentaremos aqui uma nova maneira de se lidar com especificação de componentes e a sua relação com os resultados a serem alcançados. Dessa maneira, ficará evidente sua importância e sua aplicação no contexto da arquitetura de sistemas.

OBJETIVOS



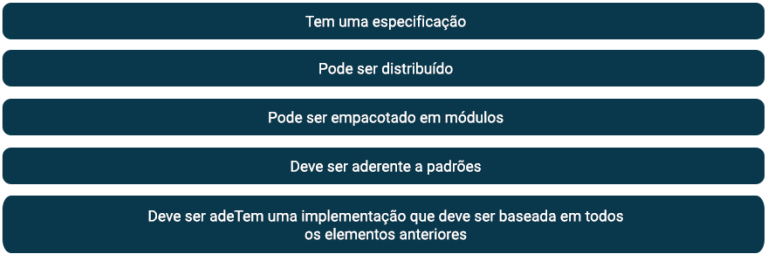
Reconhecer a importância da especificação de componentes para o melhor resultado na arquitetura de sistemas;

Analisar o processo de especificação de componentes, na arquitetura de sistemas, que melhoram o resultado dos projetos;

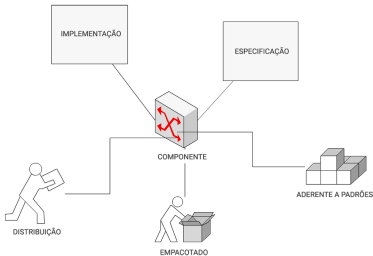
Identificar a relação entre especificação de componentes e o sucesso do projeto.

DEFINIÇÃO DE ESPECIFICAÇÃO DE COMPONENTES

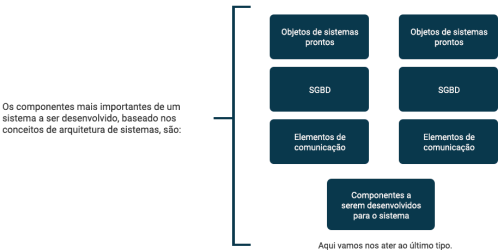
Para entendermos os conceitos de especificação de componentes, é necessário compreender quais os elementos estão relacionados a um componente. Vamos começar conhecendo as características de um componente:



A imagem a seguir representa esse conceito de forma simplificada, para facilitar o entendimento das relações entre a especificação de um componente e seus elementos:



TIPOS DE COMPONENTES

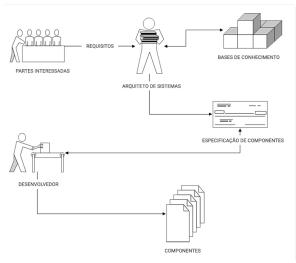


ESPECIFICAÇÃO DE COMPONENTES A SEREM DESENVOLVIDOS

Como vimos anteriormente nesta aula, especificar um componente pode ser definido como o processo de traduzir as necessidades das partes interessadas em uma linguagem que os desenvolvedores entendam e consigam implementar.

Durante esse processo, é necessário que o arquiteto de sistemas complete as informações técnicas necessárias que as partes interessadas não conseguem informar, baseado em seu framework de conhecimentos.

Esse processo pode ser melhor entendido quando observamos a imagem ao lado.

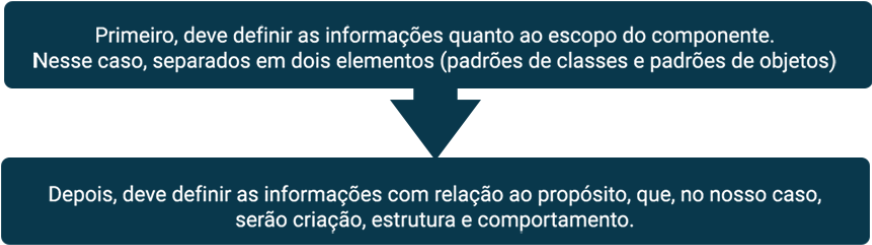


Atenção

Note que todo esse processo tem como objetivo a geração de artefatos de sistemas (componentes) mais assertivos e comprometidos com o resultado final do sistema.

METODOLOGIA/PADRÕES

O arquiteto de sistemas deve se preocupar com os seguintes conceitos quando for especificar um componente.



O esquema ao lado representa bem esse conceito e ajuda no entendimento da especificação de componentes.

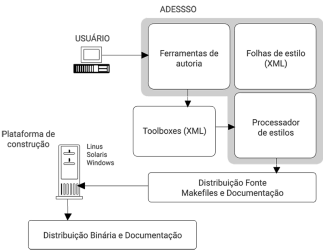
ESCOPO		PROPÓSITO		
		criação	estrutura	comportamento
	classe	Factory Method	Class Adapter	Interpreter
OBJETO		Abstract Method	Object Adapter	Template Method
		Builder	Bridge	Chain of Responsibility
		Prototype	Composite	Command
		Singleton	Decorator	Iterator
			Facade	Mediator
			Flyweight	Observer
			Proxy	State
				Strategy
				Visitor

Fonte: Baseado em padrões de projetos UML.

Atenção

Note que o objetivo aqui não é abordar detalhadamente cada um dos padrões, mas apresentar a você, futuro arquiteto de sistemas, quais são as diretrizes a serem seguidas quando for desenvolver sistemas utilizando o conceito de componentes da arquitetura de sistemas.

EMPACOTAMENTO DE COMPONENTES



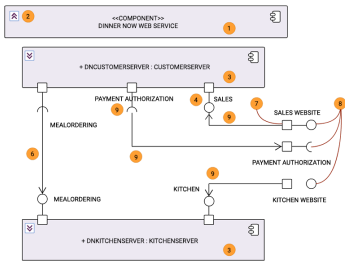
Fonte: www.dca.fee.unicamp.br

Uma das mais úteis ferramentas do desenvolvimento de sistemas por componentes é o **empacotamento**.

Com esse recurso, o arquiteto consegue resolver o sistema, abstraindo conceitos de mais alto nível, sem se preocupar muito com os conteúdos dos objetos, mas, sim, com sua funcionalidade.

Nesse caso especificamente, os componentes são reunidos por funcionalidade em pacotes que auxiliam muito o arquiteto em sua tarefa de resolver os problemas das partes interessadas.

DISTRIBUIÇÃO DE COMPONENTES



Fonte: msdn.microsoft.com

O arquiteto de sistemas deve ter a preocupação de definir e construir middlewares que conectem as diversas camadas de componentes e, consequentemente, os componentes, de maneira a conseguir uma forte acoplagem para a organização.

Os componentes devem ser distribuídos em containers seguindo as políticas estabelecidas pelos middlewares, ou seja, dentro de um mesmo container somente podem existir componentes que atendam a mesma configuração.

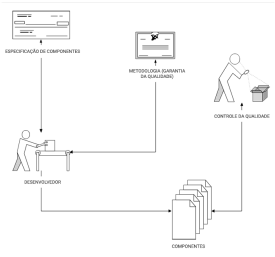
Essa configuração está presente nos descritores de distribuição que, geralmente, são escritos em XML e devem conter informações suficientes para que:

- Os componentes sejam corretamente conectados
- As camadas contenham os níveis corretos de suporte de configuração
- Os níveis de segurança corretos para a aplicação estejam implementados

IMPLEMENTAÇÃO DE COMPONENTES

A implementação de componentes é tarefa dos programadores, mas o arquiteto de sistemas também tem sua parcela de responsabilidade neste contexto, gerando especificações que atendam às necessidades e sigam os padrões definidos pela organização.

Outro conjunto de participantes nesse contexto é a área de garantia e controle da qualidade, que tem a função de prover elementos que garantam a qualidade dos componentes produzidos.



Exercício

No tocante ao desenvolvimento de software orientado ao reuso, embora o estágio inicial de especificação de requisitos e o estágio de validação sejam comparáveis com outros processos, os estágios intermediários em um processo orientado ao reuso são diferentes.

Nesse caso, são processos em estágios intermediários:

- ☐ A) Projeto de interface, análise de componentes, projeto arquitetural e testes de aceitação.
- ☐ B) Análise de componentes, modificação de requisitos, projeto de sistemas com reuso e desenvolvimento e integração.
- ☐ C) Desenvolvimento de protótipo, projeto arquitetural, análise de componentes e plano de teste.
- ☐ D) Especificação de sistema, avaliação de mudanças, análise de componentes e desenvolvimento e integração.

- ☐
- ☐
- ☐
- ☐

Justificativa

Na arquitetura de sistemas, determinado conceito permite que, entre dois elementos de software A e B, seja possível postular alguma mudança de A, que pediria que B fosse mudado (ou, no mínimo, cuidadosamente verificado), a fim de preservar a exatidão global e também postular alguma mudança, que pediria que tanto A como B mudassem juntos para preservar a exatidão global. Isso trata-se do conceito de:

☐

A) Polimorfismo.

☐

B) Congeneridade.

☐

C) Mutabilidade.

☐

D) Polidependência.

☐☐☐☐

Justificativa

Glossário