

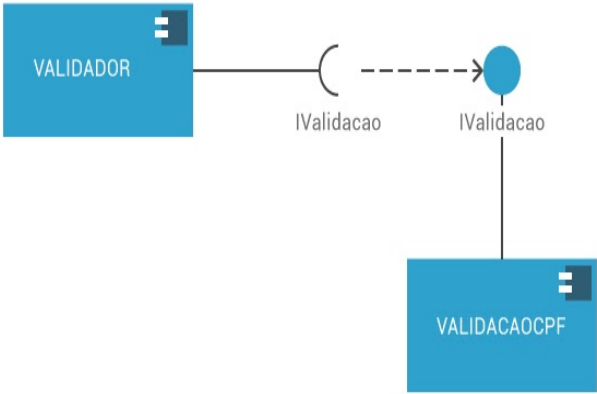
DIAGRAMA DE COMPONENTES	DIAGRAMA DE COMPONENTES
O que é um Diagrama de Componentes ?	Diagrama de Componentes: primeiro para entender o que é um Diagrama de Componentes, temos que entender o que é um "componente", quando falamos de componente dentro das ferramentas UML estamos nos referindo a algum recurso pré-construído que poderá ser reutilizado dentro do nosso código, como por exemplo: a Classe de uma Biblioteca, um sistema que faz ligação com um banco de dados, uma função de outra aplicação e etc. Um Diagrama de Componentes é um diagrama que busca representar a ligação entre o nosso sistema e os componentes que interagirão com ele, para isso ele utiliza ferramentas UML bem específicas para expôr esse tipo de relação.
Como os Componentes também podem ser chamados ?	Vamos encontrar por aí várias nomenclaturas para os componentes, que podem ser: <ul style="list-style-type: none"> - Bibliotecas - Frameworks - Softwares de Prateleira - Desenvolvimento Orientado a Objetos - Desenvolvimento Orientado a Serviços - Desenvolvimento Orientado a Aspectos - Design Patterns
Quais podem ser a vantagens e desvantagens da utilização de Componentes numa aplicação?	Vantagens: Economia no desenvolvimento de código. Confiabilidade no sistema, por estarmos usando um sistema que já conhecemos. Risco de falhas reduzido, por estarmos usando um sistema que já testamos antes. Padronização do serviço, entre outras muitas vantagens; Desvantagens: Falta de algumas funções que deveriam ser aplicadas no código atual, exigindo que o desenvolvedores quebrem a cabeça para gerar implementações. Em caso de implementação de bibliotecas desconhecidas pela equipe, pode haver uma grande aversão quanto a implementação de uma biblioteca;
Por que a reutilização de Componentes é algo desejável para uma equipe de desenvolvimento?	Por que pode tornar a equipe muito mais eficiente , é verdade que o desenvolvimento de bibliotecas é algo complexo e exige muito tempo, esforço e dedicação de toda a equipe, afinal é preciso desenvolver aplicações pensando na possibilidade de reutilizá-las em vários tipos de softwares. Mas, depois de desenvolvida, ela vai economizar muito tempo, trazer confiabilidade ao código e padronizar o serviço de toda a equipe, facilitando a criação de equipes de alto desempenho.
Qual a importância das Interfaces na criação de uma aplicação que utilize Componentes?	As Interfaces têm uma alta relevância em softwares que utilizem Componentes, afinal, serão as Interfaces que irão fazer a ligação entre os métodos ou funções que serão integrados á aplicação. Por isso, sempre que desenvolvemos um Diagrama de Componentes, somos obrigados a demonstrar a ligação por através de Interfaces Requerentes e Providas.
Como um Diagrama de Componentes pode ser representado por através das ferramentas UML ?	Para representar um Diagrama de Componentes usamos as seguintes ferramentas: <ul style="list-style-type: none"> - Componentes: Representados por um retângulo com o nome do componente e nas bordas do retângulo teremos 2 retângulos menores, como se fossem as abas de um caderno; - Ligação de Interfaces Requerida e Provida: Podem ser representadas da maneira comum como estamos acostumados nas ferramentas UML (um traço com círculo completo para Providas e um traço com meio círculo para requeridas) ou podemos representá-las com o uso das setas tracejadas, onde, a seta tracejada com a flecha sem preenchimento representa Interfaces Providas (também pode ser identificadas como "Realização") e seta com a flecha em vazio (sem fechamento triangular) representa Interfaces Requeridas (também chamadas de "Uso"); - Interfaces: Podem ser escritas encima dos relacionamentos de Interface Requerida e Provida se usarmos o modelo UML padrão (Lembrando de colocar o "I" maiúsculo na frente do nome da interface), ou, pode ser representada dentro de um Retângulo se usarmos a representação das setas tracejadas no relacionamento;
Ilustre a representação de um Diagrama de Componentes comum usando ferramentas UML: (Perceba que estamos fazendo a ligação de Componente entre um sistema validador e um sistema que já gera a validação)	 <pre> graph LR VALIDADOR[VALIDADOR] -- IValidacao --> VALIDACAO((IValidacao)) VALIDACAO -- IValidacao --> VALIDACAOCPF[VALIDACAOCPF] </pre>
- reservada para a questão acima -	
- reservada para a questão acima -	
- reservada para a questão acima -	

DIAGRAMA DE COMPONENTES

Ilustre a representação de um Diagrama de Componentes usando setas tracejadas:

(Perceba que estamos fazendo a ligação entre um sistema de cadastro com um sistema que faz a ligação com bancos de dados em MySQL e Oracle)

- reservada para a questão acima -

DIAGRAMA DE COMPONENTES

