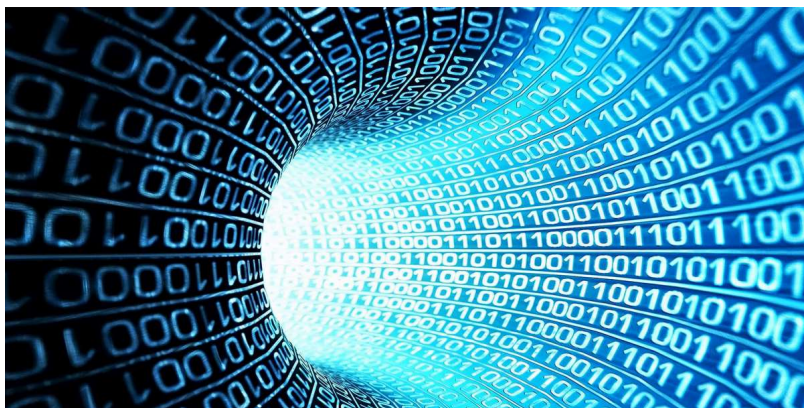


# Arquitetura de Sistemas

## Aula 1 - Componentes de Sistemas

### INTRODUÇÃO

---



O conceito principal do uso de componentes está relacionado ao ditado “dividir para conquistar”.

Isso significa gerenciar a complexidade, quebrando um grande problema em pedaços menores a serem desenvolvidos. Na sequência, integra-se esses pequenos módulos simples para resolver problemas complexos.

Componentes seguem essa definição para sua construção, com a diferença que estão relacionados a objetos de negócio, dados, interfaces que combinadas resolvem os problemas apresentados.

Quando resolvemos criar um sistema de informação, utilizando a abordagem de componentes, será necessário primeiro mapear esses componentes em modelos conceituais, depois construí-los separadamente e, somente ao fim integrá-los para a solução do problema.

Bons estudos!

## OBJETIVOS

---



Identificar os fundamentos e objetivos de um componente;

Analisar um exemplo de componente e definir o que não é um componente;

Distinguir arquitetura de sistemas e componentes;

Definir especificação de componentes, implementação de componentes e níveis de modelo.

## FUNDAMENTOS DE COMPONENTES

Componentes, no contexto da arquitetura de sistemas, são **unidades de software estruturados** de acordo com alguns princípios específicos.

Os princípios fundamentais que regem os componentes estão relacionados ao conceito de objetos, conforme descrito a seguir:



Fonte da Imagem:

### União de dados e funções

Um objeto de software consiste em dados que podem assumir valores e funções que tratam esses dados. Os **dados** e **funções** devem ter uma ligação natural entre eles, formando o conceito de classe.



Fonte da Imagem:

### Encapsulamento

Conceito de esconder de quem vai usar a classe os detalhes de sua funcionalidade e de dados, deixando amostra

somente como **acionar** e o **resultado a ser alcançado** pelo acionamento.

Não importa para quem vai usar tal componente, como as coisas acontecem dentro dele, e, sim, se o resultado esperado foi alcançado.

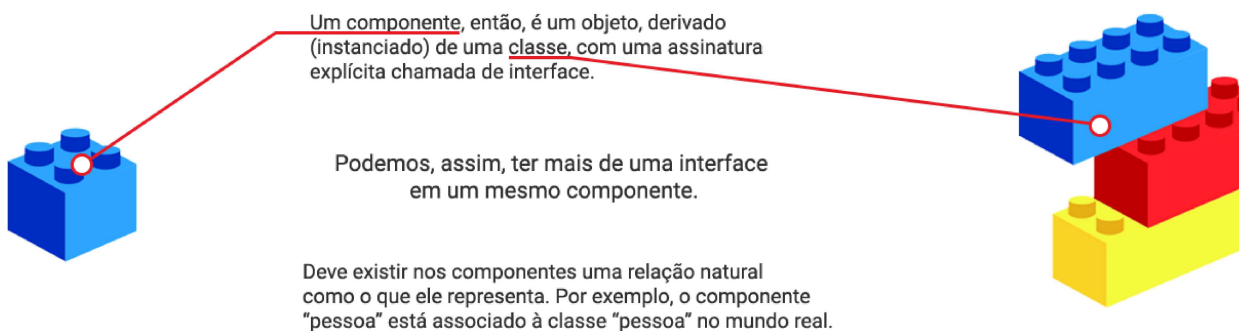


Fonte da Imagem:

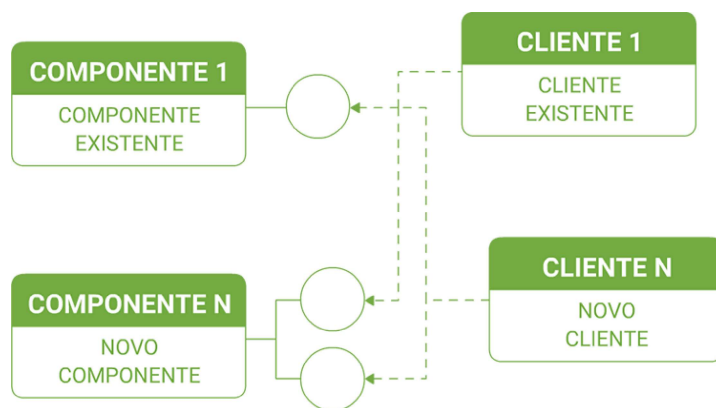
## Identidade

Cada componente encapsulado tem uma **identidade única de dados e funções** e pode assumir estados pré-determinados.

## O que é um componente afinal?



## ARQUITETURA DE SISTEMAS BASEADA EM COMPONENTES



A arquitetura de sistemas baseada em componentes, então, está relacionada à identificação das interfaces possíveis disponíveis e como elas resolvem os problemas que se apresentam.

Esse conceito facilita muito o desenvolvimento e reduz o nível de mudança nos sistemas gerados.



Fonte da Imagem:

## Objetivos de componentes

A partir de um objeto simples estruturado (componente), é necessário combinar funcionalidade e dados para resolver os problemas em sistemas computacionais.

O grande desafio na abordagem por componentes na arquitetura de sistemas está em conseguirmos identificar claramente as interfaces e como elas resolvem os problemas que se apresentam. Isso está relacionado às dependências entre componentes.

Pode até ser uma surpresa para alguns que, quando definem seus objetos, estão pensando somente na reutilização e não em como suas interfaces se relacionam em contextos variados.

### Atenção

Quando pensamos em componentes a partir de suas interfaces, criamos um número maior de possibilidades de uso e reduzimos significativamente o número de mudanças futuras.

## CARACTERÍSTICAS DE COMPONENTES

Existem vários conceitos que podem definir os objetivos de um componente, todos com real validade, mas podemos afirmar que um componente contém as características necessárias para resolver problemas de forma natural e relacionado ao mundo real.

Vamos ver algumas delas:

- Ser capaz de montar uma aplicação, ligando componentes que devem atender a padrões pré-definidos e estarem agrupados por categoria, dentro do conceito de "caixa de componentes". Esses componentes devem se encaixar a outros (através de suas interfaces), para resolver os problemas;
- O uso dado aos componentes, então, pode variar, de acordo com a necessidade e experiência de cada arquiteto de sistemas;
- Quando você, arquiteto de sistemas, for utilizar os componentes para criar suas aplicações, é necessário identificar bem as interfaces e como estas se ligam a outros componentes;
- Para que isso seja possível é necessário que as interfaces estejam bem especificadas e claras;
- Se as interfaces dos componentes estiverem bem definidas, será possível trocar componentes sem afetar os resultados da aplicação e sem a necessidade de mudanças de códigos;
- Caso não estejam bem definidas, essas facilidades e funcionalidades podem ser perdidas;
- As características de funcionalidade de um determinado componente devem estar claras na definição das interfaces (Por exemplo, um componente que atualiza endereço através de CEP, somente poderá ser utilizado em uma aplicação que queira funcionar nos EUA se também tratar ZIP CODE, que é o equivalente ao CEP para este país).

Sobre o segundo item, como mostra a imagem para sistemas de automação, componentes eletrônicos têm as mesmas características de componentes de sistemas:



Fonte: arduino.cc

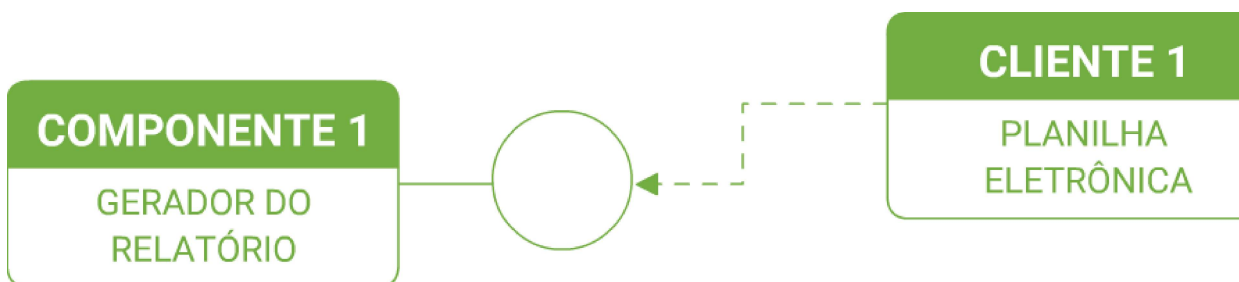
## EXEMPLO DE USO DE COMPONENTES

Imagine que nosso sistema deve gerar um relatório em uma planilha eletrônica para que o usuário tenha como trabalhar com as informações e montar seus relatórios e gráficos personalizados.

É bem razoável que não esteja no escopo do desenvolvimento do sistema a construção de um sistema de planilha eletrônica. É muito mais simples e eficaz utilizarmos o Excel ou a Planilha do OpenOffice para este fim.

Para que isso ocorra, precisamos mapear as interfaces de uso dessas planilhas e que elas sejam repassadas aos desenvolvedores dos novos componentes de maneira que se liguem ao componente “Planilha Eletrônica que se des utilizar”.

Graficamente, teríamos o seguinte:





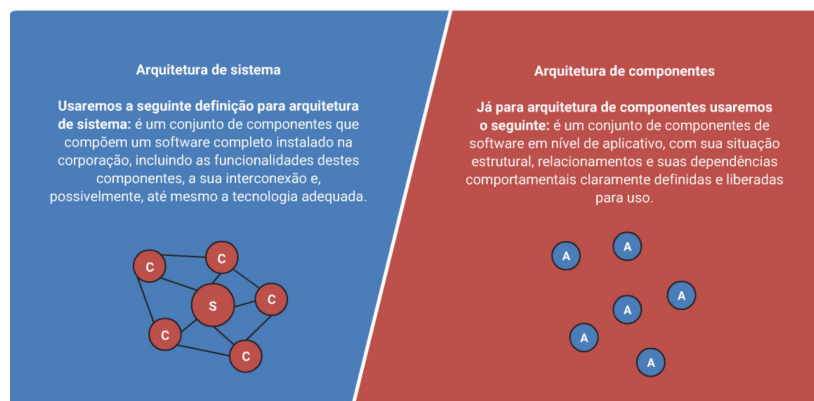
Fonte:

# O que não é um componente?

O fato de termos uma classe ou uma função escrita em uma linguagem de programação, compilada e instalada em nosso ambiente, por si só não atribui a esse código a funcionalidade de um componente de software.

É necessário que tenhamos uma interface bem definida e que padrões de interação sejam suportados pelo componente. Somente neste caso ele estará apto a ser utilizado e classificado como componente.

## DEFINIÇÃO DE ARQUITETURA DE SISTEMAS E COMPONENTES



## ARQUITETURAS DE SISTEMAS

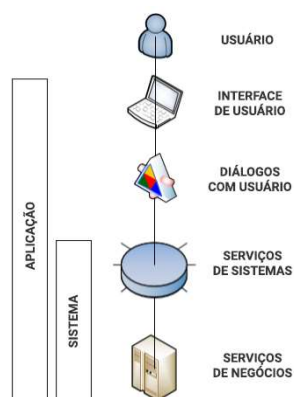
Podemos ter N camadas de arquiteturas distribuídas, ligando bancos de dados corporativos, pacotes de automação sistemas em funcionamento na corporação, interagindo especificamente através dos aplicativos de processos de negócios software com as interfaces de usuário baseadas na web.





### Atenção

Essa é uma típica arquitetura do sistema para os tipos de sistemas que podemos explorar nesta disciplina. Compreender esta questão relacionada à arquitetura do sistema é importante porque nos diz a forma geral do sistema final e explica como usaremos várias tecnologias para montar o sistema que precisamos.



Fonte da Imagem:

## Camadas da arquitetura de sistemas

Queremos usar componentes para fins diferentes e para resolver preocupações diferentes.

Nossa abordagem global é identificar as camadas diferentes nas quais os componentes podem ser utilizados.

Isso é útil porque nos permite raciocinar sobre a finalidade de cada unidade de software que usaremos em nosso sistema.

## ARQUITETURA DE COMPONENTES

Componentes podem ser encontrados em qualquer uma das camadas da Arquitetura de sistemas, conforme imagem anterior. Porém, o que nos interessa nesta aula são os componentes presentes nas camadas de sistemas e de negócio.

Como vimos, a arquitetura de componentes é um conjunto de elementos de software no nível de aplicativo que contém comportamentos e dependências em suas relações estruturais.

Essa é uma definição independente do nível de tecnologia em que será implantado. A arquitetura de componentes pode ser usada em uma única aplicação ou para um contexto mais amplo, como um conjunto de aplicações que servem uma área de processo de negócio em particular.

Dentro desse contexto, os componentes nos permitem entender como, dependendo do nível de integração (forte, moderada ou fraca), nosso sistema vai reagir às modificações e/ou à substituição de componentes.



Fonte da Imagem:

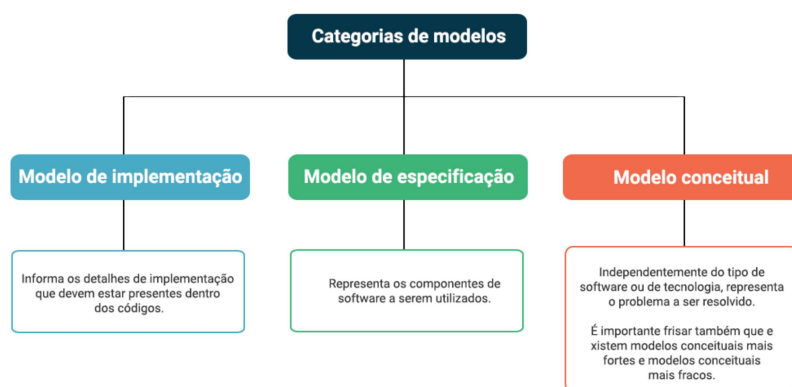
## Relação entre especificação de componentes e interfaces

A especificação componente define a assinatura do componente e, consequentemente, a forma como será construído, utilizado e testado.

Essa especificação define as formas de uso e delimita sua fronteira de acesso. Já a **interface** define a relação com outros componentes, informando o que esperar quando se conectar a este componente.

## NÍVEIS DE MODELO

Um modelo de algo apresenta uma perspectiva ou visão resumida do que é e do que aquilo permite. Deve ficar claro que sempre em um modelo algumas coisas serão enfatizadas e outras serão excluídas.



## Glossário