

Engenharia de software

João Caldeira

Professor Convidado

O email. joaocarlos.caldeira@my.istec.pt

Mob. +351 917769544

14 de dezembro de 2022



Design de software

14 de dezembro de 2022



Design de software

Projeto arquitetônico

- **Definição**

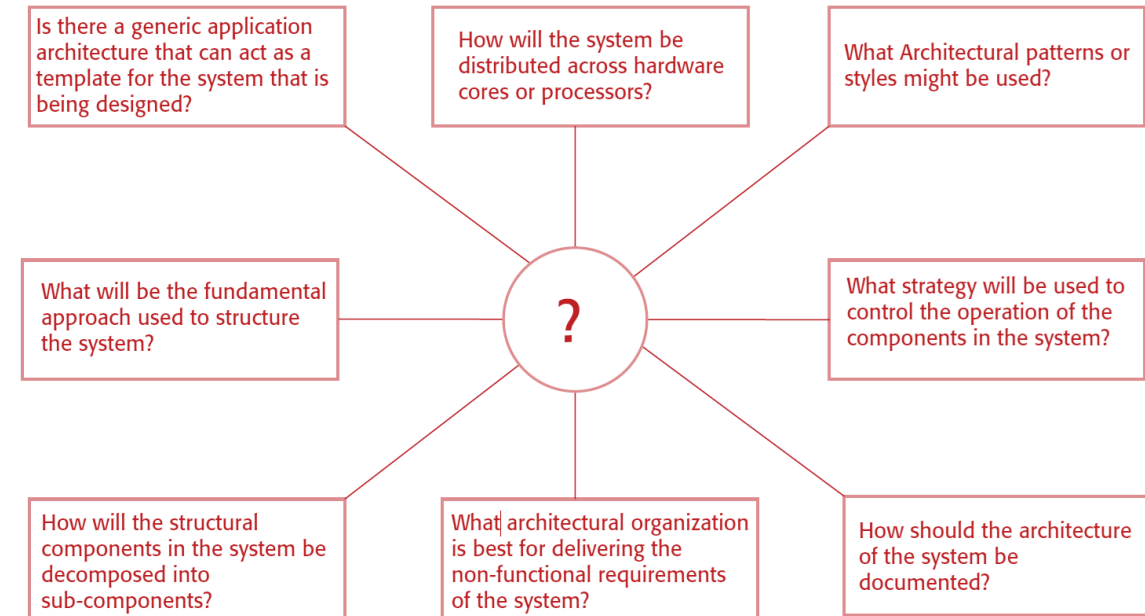
1. O projeto arquitetônico está preocupado em entender como um sistema de software deve ser organizado e projetar a estrutura geral desse sistema.
2. É o elo crítico entre o projeto e a engenharia de requisitos, pois identifica os principais componentes estruturais de um sistema e as relações entre eles
3. A saída do processo de projeto arquitetônico é um modelo arquitetônico que descreve como o sistema é organizado como um conjunto de componentes comunicantes
4. Em processos ágeis, é geralmente aceito que um estágio inicial de um processo de desenvolvimento ágil deve se concentrar em projetar uma arquitetura de sistema geral

Design de software

Decisões de projeto arquitetônico

- **Preocupações**

1. **Desempenho.** A arquitetura deve ser projetada para localizar operações críticas dentro de um pequeno número de componentes
2. **Segurança.** Uma estrutura em camadas para a arquitetura deve ser usada, com os ativos mais críticos protegidos nas camadas mais internas
3. **Segurança.** A arquitetura deve ser projetada de forma que as operações relacionadas à segurança sejam co-localizadas em um único ou em um pequeno número de componentes
4. **Disponibilidade.** A arquitetura deve ser projetada para incluir componentes redundantes para que seja possível substituir e atualizar componentes sem parar o sistema

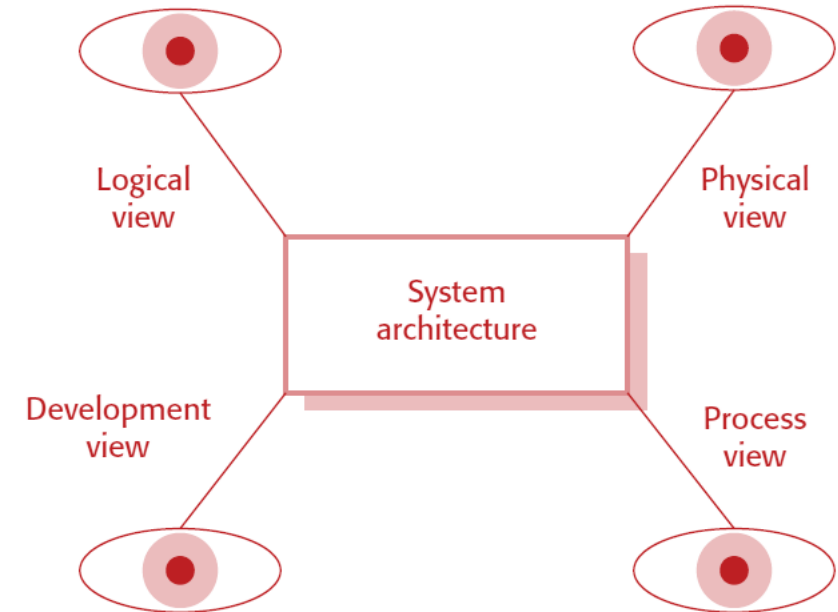


Design de software

Vistas arquitetônicas

- **Opções**

- 1. Visualização Lógica.** Mostra as principais abstrações no sistema como objetos ou classes de objetos. **Útil para relacionar os requisitos do sistema a entidades nesta visão lógica**
- 2. Visualização do Processo.** Mostra como, em tempo de execução, o sistema é composto por processos interativos. **Útil para fazer julgamentos sobre características não funcionais do sistema, como desempenho e disponibilidade**



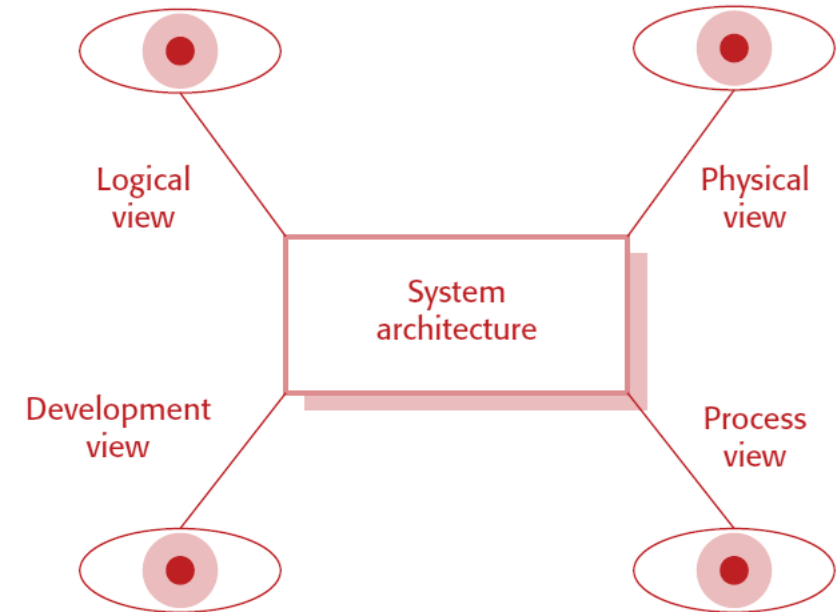
Design de software

Vistas arquitetônicas

- **Opções**

3.Visualização de Desenvolvimento.Mostra como o software é decomposto para desenvolvimento; ou seja, mostra a divisão do software em componentes que são implementados por um único desenvolvedor ou equipe de desenvolvimento.**Útil para gerentes de software e programadores**

4.Visão Física.Mostra o hardware do sistema e como os componentes de software são distribuídos pelos processadores do sistema.**Útil para engenheiros de sistemas que planejam uma implantação de sistema**



Design de software

Padrões de arquitetura

- A ideia de padrões como forma de apresentar, compartilhar e reutilizar conhecimento sobre sistemas de software tem sido adotada em várias áreas da engenharia de software.
 - **Padrão Arquitetônico.** Uma descrição estilizada e abstrata de boas práticas, que foi experimentada e testada em diferentes sistemas e ambientes
 - Deve descrever uma organização de sistema que foi bem-sucedida em sistemas anteriores
 - Deve incluir informações sobre quando é e quando não é apropriado usar esse padrão e detalhes sobre os pontos fortes e fracos do padrão

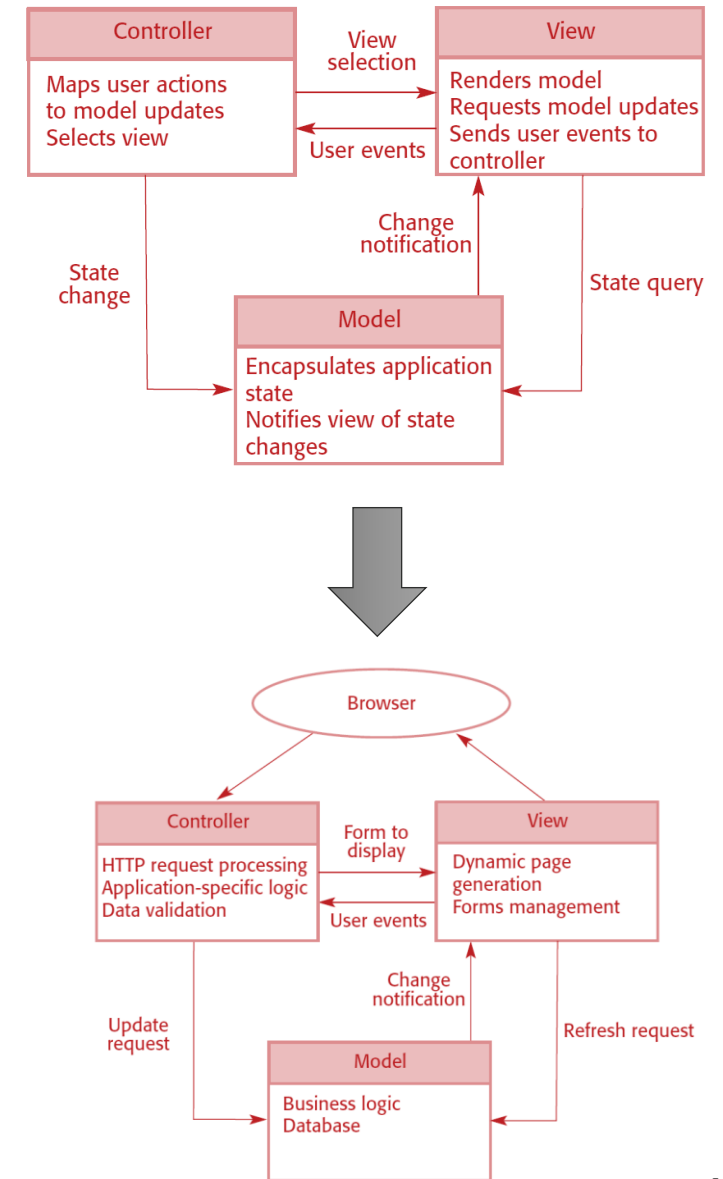
Name	MVC (Model-View-Controller)
Description	Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model. See Figure 6.5.
Example	Figure 6.6 shows the architecture of a web-based application system organized using the MVC pattern.
When used	Used when there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown.
Advantages	Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways, with changes made in one representation shown in all of them.
Disadvantages	May involve additional code and code complexity when the data model and interactions are simple.

Design de software

Padrões de arquitetura

Padrão Model-View-Controller (MVC)

- Esse padrão é a base do gerenciamento de interação em muitos sistemas baseados na Web e é suportado pela maioria dos frameworks de linguagem
- Separa os elementos de um sistema, permitindo que eles mudem de forma independente.
 - Por exemplo: adicionar uma nova visualização ou alterar uma visualização existente pode ser feito sem nenhuma alteração nos dados subjacentes no modelo



Design de software

Padrões de arquitetura

Padrão de arquitetura em camadas

- As noções de separação e independência são fundamentais para o projeto arquitetônico porque permitem que as mudanças sejam localizadas
- A funcionalidade do sistema é organizada em camadas separadas, e cada camada conta apenas com as facilidades e serviços oferecidos pela camada imediatamente abaixo dela

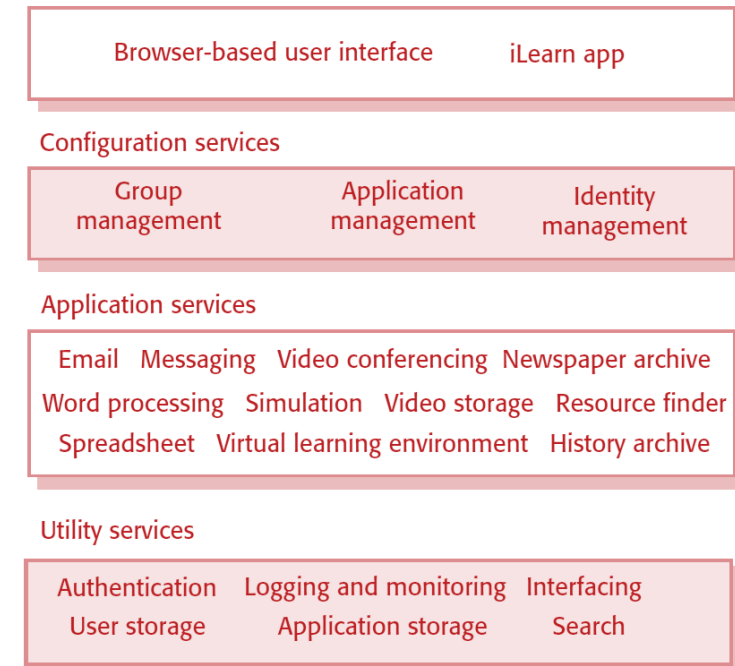
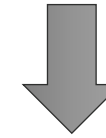
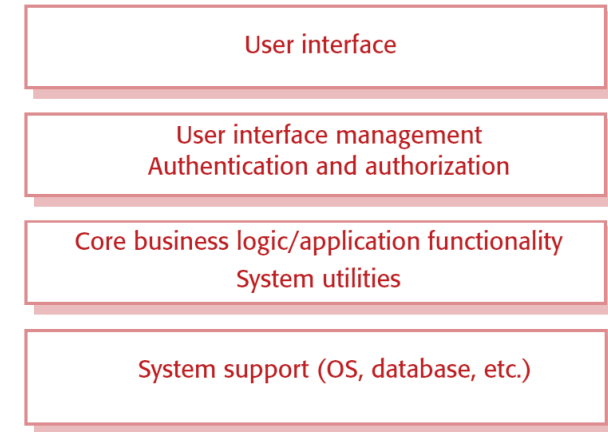
Name	Layered architecture
Description	Organizes the system into layers, with related functionality associated with each layer. A layer provides services to the layer above it, so the lowest level layers represent core services that are likely to be used throughout the system. See Figure 6.8.
Example	A layered model of a digital learning system to support learning of all subjects in schools (Figure 6.9).
When used	Used when building new facilities on top of existing systems; when the development is spread across several teams with each team responsibility for a layer of functionality; when there is a requirement for multilevel security.
Advantages	Allows replacement of entire layers as long as the interface is maintained. Redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system.
Disadvantages	In practice, providing a clean separation between layers is often difficult, and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.

Design de software

Padrões de arquitetura

Padrão de arquitetura em camadas

- Esta abordagem em camadas suporta o desenvolvimento incremental de sistemas
- À medida que uma camada é desenvolvida, alguns dos serviços fornecidos por essa camada podem ser disponibilizados aos usuários
- A arquitetura também é mutável e portátil
- Se sua interface não for alterada, uma nova camada com funcionalidade estendida pode substituir uma camada existente



Design de software

Padrões de arquitetura

Padrão de Arquitetura Cliente-Servidor

- Organizado como um conjunto de serviços e servidores associados e clientes que acessam/usam os serviços

1. Um conjunto de servidores. Que oferecem serviços a outros componentes.

- **Por exemplo** Inclui servidores de impressão que oferecem serviços de impressão, servidores de arquivos que oferecem serviços de gerenciamento de arquivos e um servidor de compilação que oferece serviços de compilação de linguagem de programação. **Os servidores são componentes de software e vários servidores podem ser executados no mesmo computador.**

2. Um conjunto de clientes. Que chamam os serviços oferecidos pelos servidores. Normalmente haverá várias instâncias de um programa cliente executando simultaneamente em diferentes computadores

Name	Client-server
Description	In a client-server architecture, the system is presented as a set of services, with each service delivered by a separate server. Clients are users of these services and access servers to make use of them.
Example	Figure 6.13 is an example of a film and video/DVD library organized as a client-server system.
When used	Used when data in a shared database has to be accessed from a range of locations. Because servers can be replicated, may also be used when the load on a system is variable.
Advantages	The principal advantage of this model is that servers can be distributed across a network. General functionality (e.g., a printing service) can be available to all clients and does not need to be implemented by all services.
Disadvantages	Each service is a single point of failure and so is susceptible to denial-of-service attacks or server failure. Performance may be unpredictable because it depends on the network as well as the system. Management problems may arise if servers are owned by different organizations.

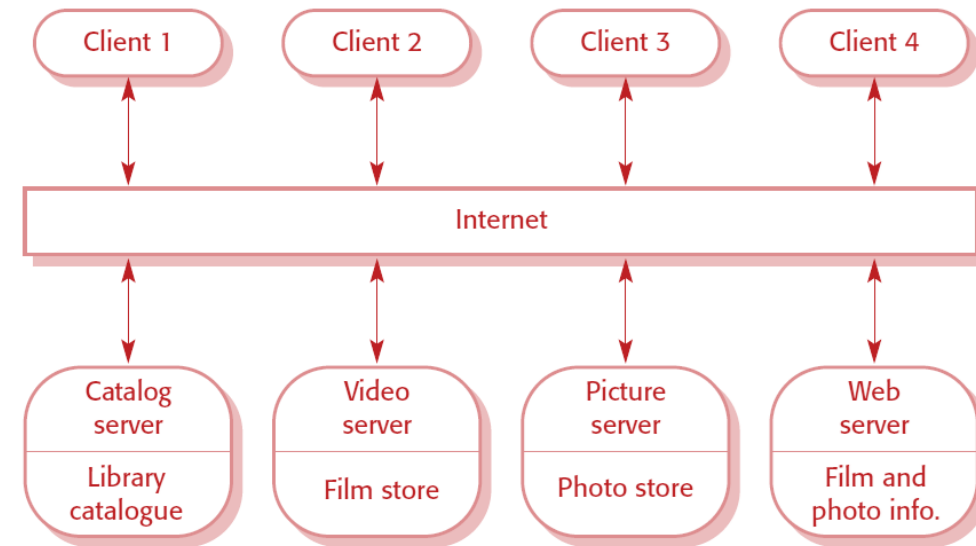
3. Uma rede que permite aos clientes aceder a estes serviços. **Sistemas cliente-servidor** são geralmente implementados como sistemas distribuídos, conectados usando protocolos da Internet

Design de software

Padrões de arquitetura

Padrão de Arquitetura Cliente-Servidor

- A vantagem mais importante do modelo cliente-servidor é que ele é distribuído arquitetura
- O uso efetivo pode ser feito de sistemas em rede com muitos processadores distribuídos
- É fácil adicionar um novo servidor e integrá-lo ao restante do sistema ou atualizar servidores de forma transparente sem afetar outras partes do sistema

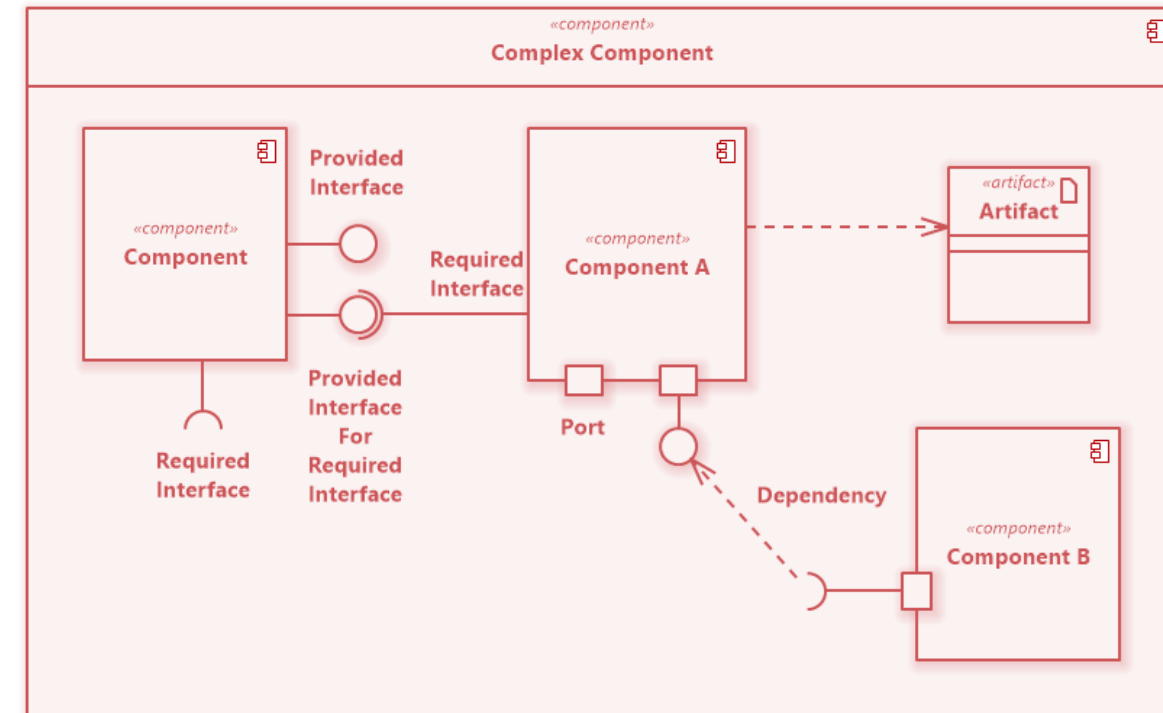
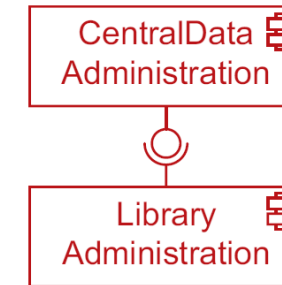


Design de software

Modelos Estruturais

- **Componentes (Diagramas)**

1. Um componente é uma unidade executável independente que fornece serviços a outros componentes ou usa os serviços de outros componentes
2. Ao especificar um componente, você pode modelar duas visualizações explicitamente:
 - **A visão externa.** Que representa a especificação do componente
 - **A visão interna.** Que define a implementação do componente



Design de software

Modelos Estruturais

- **Implantação (diagramas)**

1. Representa a topologia de hardware usada e o sistema de tempo de execução atribuído
2. O hardware engloba unidades de processamento na forma de nós, bem como relações de comunicação entre os nós
3. Um sistema de tempo de execução contém artefatos que são implantados nos nós

