



FUNDAMENTOS DE DESIGN DE SISTEMAS

Prof. Esp. Winston Sen Lun Fung

Conversa Inicial



2
15



CONVERSA INCIAL

- O desenvolvimento de um sistema ou aplicativo há uma estrutura de definições de tecnologias, desempenho, escalabilidade, interoperabilidade, compatibilidade e performance definidas por uma Arquitetura de Software.

Arquitetura de Software



4
15



Introdução

- A complexidade dos sistemas estão tornando-se maiores e o seu desenvolvimento e manutenção desafiadora buscando maneiras de conceber um sistema de software estruturado considerando as necessidades do cliente e as tecnologias associadas.

Introdução

- O sucesso do desenvolvimento de um sistema depende de formas de documentar todos os seus aspectos de forma bem definida que possa ser entendida pelos interessados no sistema. Isso facilita a comunicação entre a equipe de desenvolvimento permitindo que aplicação atenda a padrões necessários para funcionar de forma assertiva.

O que é um padrão na Arquitetura de Software?

- Um padrão arquitetural tem o objetivo atender uma visão orientada aos negócios.
- O modelo ajuda a tomada de decisões do projeto de software, como e quais serão suas utilidades, linguagens, tecnologias no relacionamento de cada subsistema para a construção de soluções para a aplicação.

O Surgimento da Arquitetura de Software

- No fim dos anos 60, cientistas começaram a discutir formas de estruturar um sistema antes do seu desenvolvimento.
- Apenas nos anos 90 foi escrito o livro ***“Software Architecture: Perspectives on a Emerging Discipline”*** por Mary Shaw e David Garlan, da Carnegie Mellon University.

O Surgimento da Arquitetura de Software

- Devido a importância do tema para o desenvolvimento a norma ISO/IEC/IEEE 42010:2011 foi estabelecida com o objetivo de criar, analisar e descrever arquiteturas de software.

O Profissional

- **O Profissional em Arquitetura de Software deve oferecer soluções para a corporações, por meio do desenvolvimento de sistemas de tecnologia da informação que melhorem o desempenho das empresas.**
- **O profissional deve saber mais do que programar. É necessário compreender os negócios das corporações , colaboratividade, tecnologias, serviços, normas, legislações, *cloud computing* e muito mais.**

O Profissional

- **As habilidades desejadas do Arquiteto de Software:**
 - **Conhecimento do domínio e tecnologias relevantes;**
 - **Conhecimento de questões técnicas para o desenvolvimento de sistemas;**
 - **Conhecimento de técnicas de levantamento de requisitos, e de métodos de modelagem e desenvolvimento de sistemas;**
 - **Conhecimento das estratégias de negócios das empresas;**
 - **Conhecimento de processos, estratégias e produtos das empresas concorrentes.**
 - **Conhecimento para analisar, compreender, propor e criar formas de inovadoras de negócios e soluções computacionais.**

Estilo Arquitetural

Entendendo Estilo Arquitetural

- O estilo arquitetural considera o sistema por completo, permitindo determinar como o sistema está organizado, caracterizando os requisitos, componentes e suas interações.

Entendendo Estilo Arquitetural

- O estilo arquitetural serve para caracterizar a arquitetura de software de um sistema, possibilitando a:
- Identificação de componentes – o arquiteto identifica quais os principais elementos tem funcionalidades bem definidas como, um componente de cadastro de informações de usuários e um componente de autenticação de usuários em uma aplicação web.

Entendendo Estilo Arquitetural

- **Identificação de mecanismo de interação** – a comunicação entre objetos por meio de troca de mensagens constitui uma forma através da qual os componentes de software interage entre si.
- **Identificação de propriedades** – o arquiteto pode analisar as propriedades oferecidas por cada estilo baseado na organização dos componentes ne nos mecanismo de interação.

Por que o Estilo Arquitetural é importante?

- Quanto maiores e mais complexos os sistemas é necessário maiores níveis de abstração. Servindo de apoio à compreensão do projeto e comunicação entre os participantes e interessados no projeto;

Vantagens do Estilo Arquitetural

- **Suporte a atributos de qualidade (ou requisitos não funcionais);**
- **Diferenciação entre arquiteturas;**
- **Menos esforço para entender o projeto;**
- **Reuso de arquitetura e conhecimento em novos projetos;**
- **Suporte ao planejamento e a gerencia da manutenção e integridade da solução.**

Documentação



18
15



Por que documentar a Arquitetura de Software?

- ***“Software architecture documentation speak for the architect, today, tomorrow and 20 Years from now”*** – Software Engineering Institute – Carnegie Mellon University

Por que documentar a Arquitetura de Software?

- Diversas vezes, arquiteturas de software são criada e não são documentadas (e consequentemente comunicadas) de forma efetiva, ou seja, desenvolvedores e outros interessados no sistema não tem acesso a uma representação adequada da arquitetura.

Por que documentar a Arquitetura de Software?

- Define as **atividades** que serão realizadas;
- É o **primeiro artefato** a agregar artefatos de qualidade;
- É o **melhor artefato** nas primeiras fases do desenvolvimento;
- **Elemento chave** para posterior manutenção do sistema.

Princípios de uma Boa Documentação

- **Princípio 1: Documentar sob o ponto de vista de quem irá utilizar a documentação.**

- **Princípio 2: Evitar ambiguidade.**

- **Princípio 3: Usar uma organização padrão para o documento a ser criado, isto é, um modelo ou *template*.**

Princípios de uma Boa Documentação

- **Princípio 4: Evitar repetições desnecessárias.**
Entretanto redundância é aceitável ou desejável se:
 - Uma informação está definida em um local X da documentação e uma elaboração ou refinamento da mesma informação aparece em um local Y. Neste caso, repetir a informação (ou parte dela) no local Y é comum.
 - Duas informações são mapeadas uma para outra. É difícil fazer isso sem repetir parte delas.
 - A repetição é conveniente por motivos práticos, quando trocar de página constantemente para entender o texto fica inviável.

Princípios de uma Boa Documentação

- **Princípio 5: Documentar as razões para as decisões tomadas.**
 - As mais importantes incluem aquelas que resultaria, de uma decisão longa, ou que a mudança seria onerosa, ou aquelas que são cruciais para atingir os requisitos chave.
 - Deve-se documentar tanto a razão para as decisões tomadas quanto alternativas rejeitadas que sejam importantes.

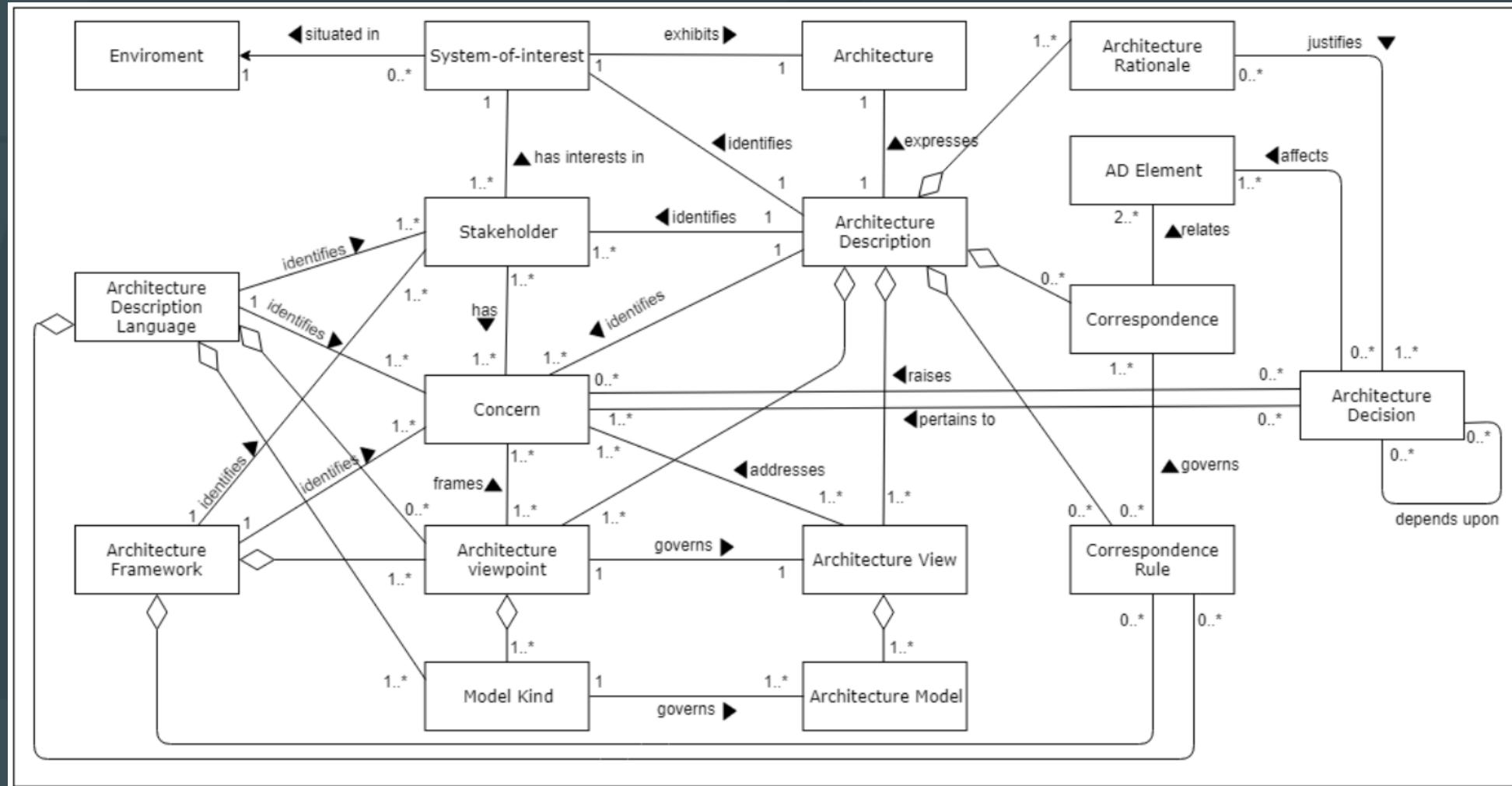
Princípios de uma Boa Documentação

- **Princípio 6: Manter a documentação atualizada.**
- **Princípio 7: Revisar a documentação criada.**
- **Princípio 8: Tornar a documentação acessível a todos os participantes relacionados ao projeto.**

ISO/IEC/IEEE 42010:2011

- A norma oferece o modelo conceitual para a documentação de uma arquitetura de software.

ISO/IEC/IEEE 42010:2011



Fonte: Adaptado da ISO/IEC/IEEE 42010:2011

As Views da ISO/IEC/IEEE 42010:2011



Os Diferentes Modelos Arquiteturais

Modelos Arquiteturais

- Não existe uma única forma de visualizar e desenvolver um sistema.
- Atualmente, há alguns padrões que são mais usados para a criação da solução do software, que também são conhecidos como modelos de arquitetura.

Arquitetura em camadas (Layered pattern)

- **Organiza um sistema de conjunto em camadas que podem ser desconstruídas em diferentes serviços, trazendo um modelo incremental de desenvolvimento. Os casos mais comuns para o uso desse padrão são em software de e-commerce e desktop.**

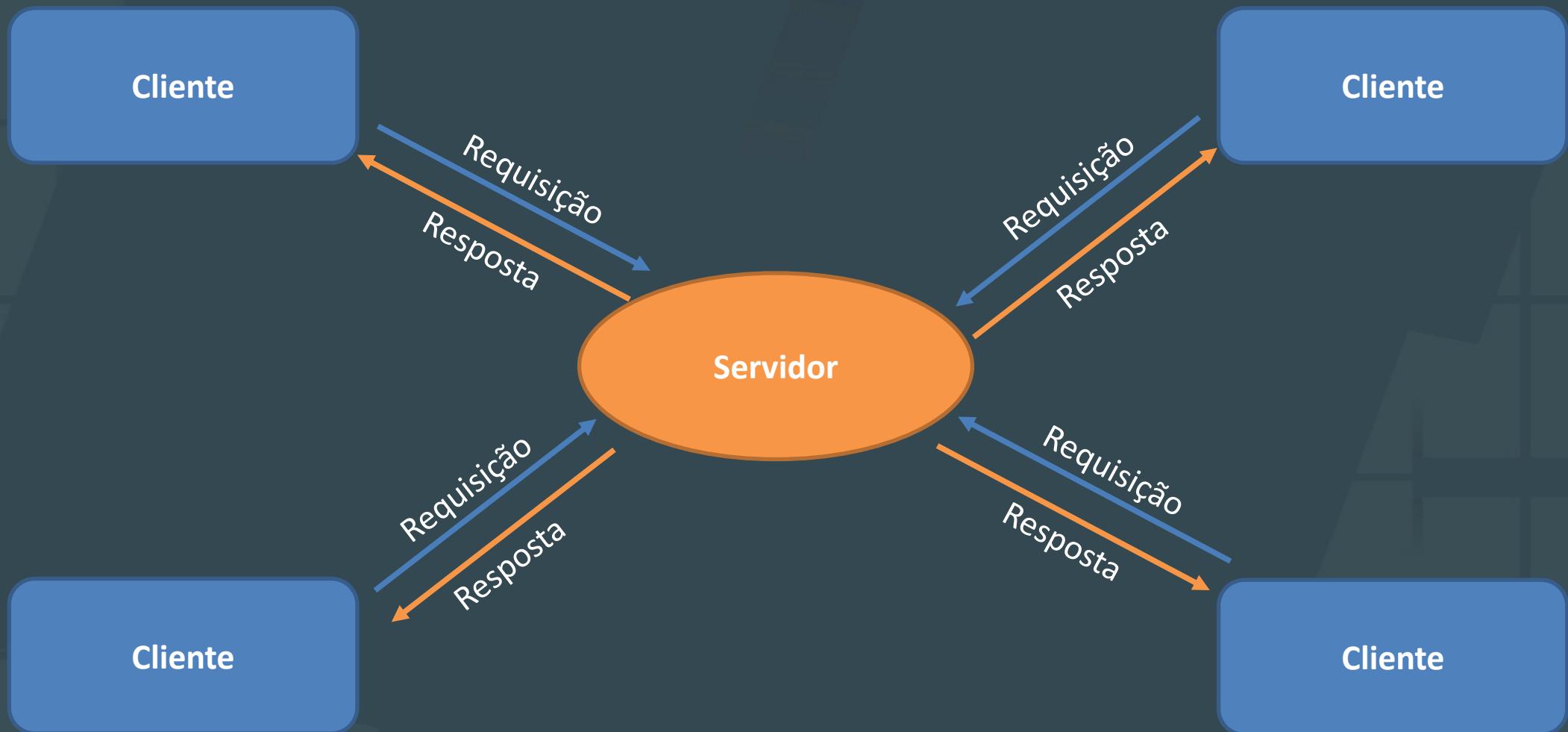
Exemplo de Arquitetura em camadas



Arquitetura cliente-servidor (Client-server pattern)

- **Estilo organizado em serviços combinando dados do cliente e do servidor. Para isso, é primordial que o cliente disponibilize uma rede de acesso às informações. Este cenário é um dos mais conhecidos na rotina das pessoas, já que podem ser aplicativos bancários e e-mail.**

Arquitetura cliente-servidor

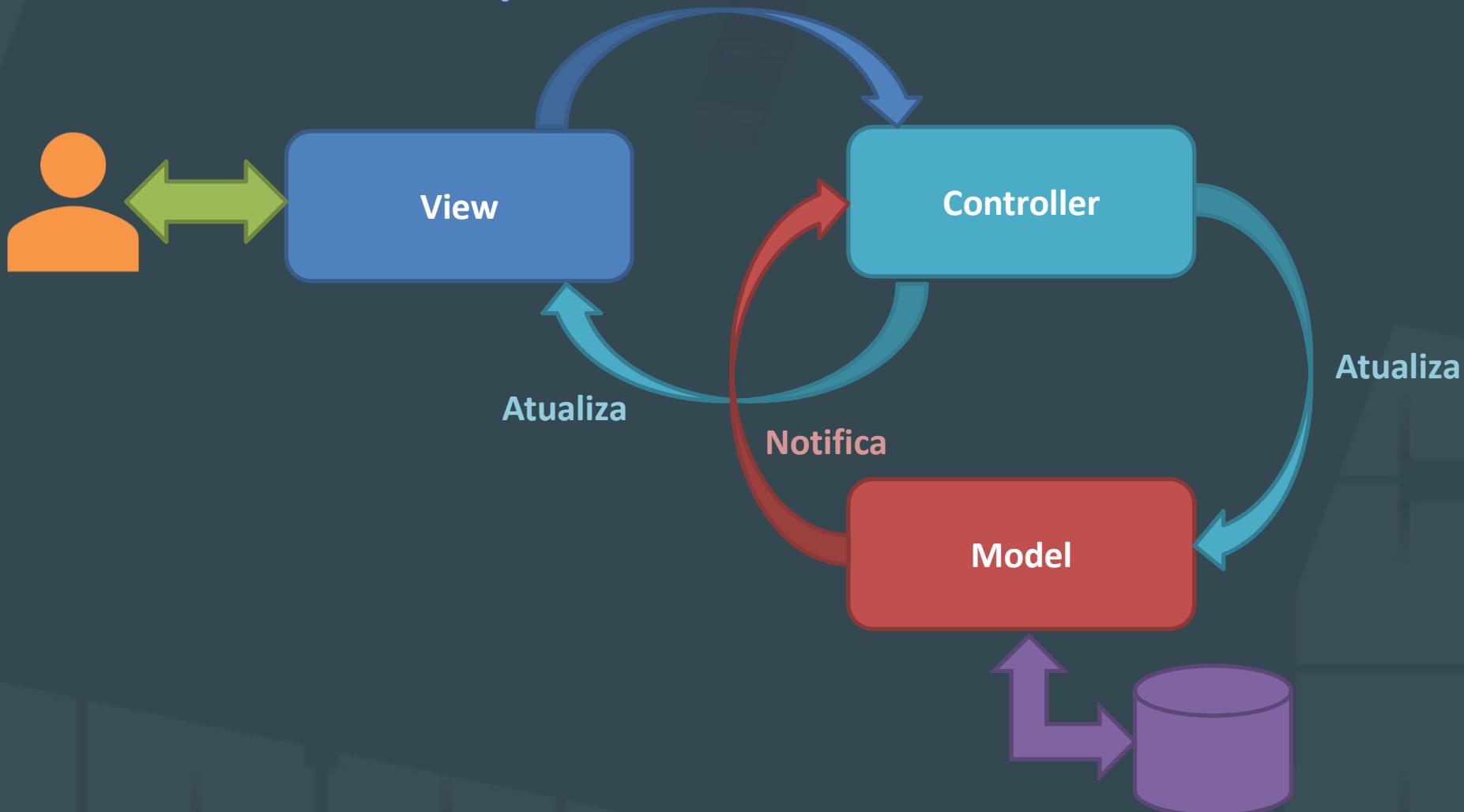


Arquitetura MVC (Model-view-controller pattern)

- **Distribuído em três camadas (Modelo, Visão e Controle), este padrão é um dos mais comuns para o online, porque traz um modelo interativo de sistema.**

Arquitetura MVC

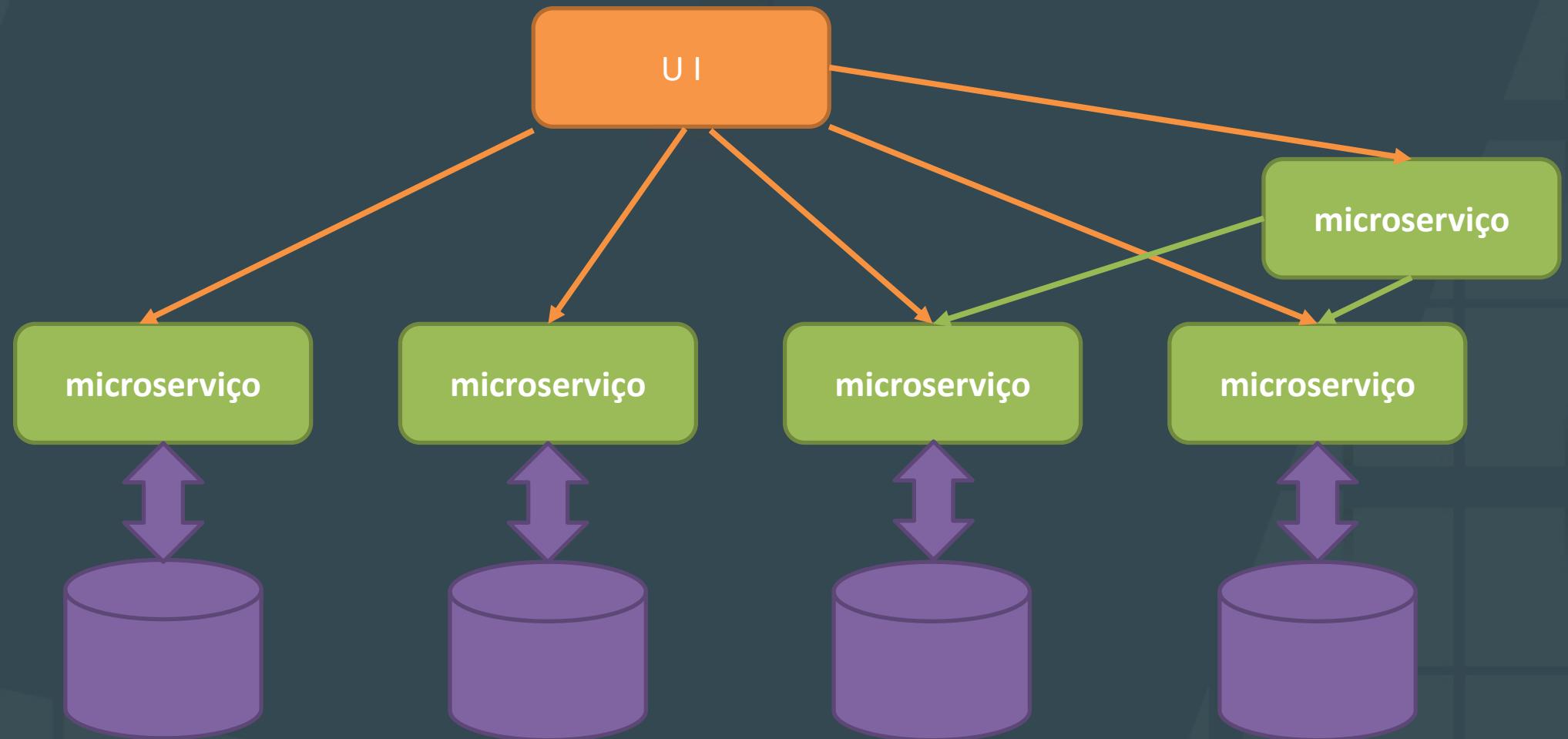
Ações do usuário



Arquitetura de microserviços (Microservices pattern)

- Este padrão utiliza múltiplos serviços e componentes para desenvolver uma estrutura modular favorecida. Hoje, é um dos modelos preferidos dos desenvolvedores e arquitetos de software por possibilitar a escalabilidade e independência dos módulos – que até podem utilizar diferentes linguagens e programações.

Arquitetura de microserviços



O Futuro da Arquitetura de Software

O Futuro da Arquitetura de Software

- O futuro do desenvolvimento de sistemas sofre grande influência das novas redes de comunicações com novos serviços providos através da internet e novos modelos utilizando a computação em nuvem e inteligência artificial.

Computação em nuvem (cloud computing)

- Com a facilidade de conexão a internet em diversos dispositivos os sistemas estão saindo do interior das empresas e migrando para hospedagem, serviços e microserviços disponíveis na nuvem.

Inteligência artificial

- **Com as novas necessidades dos negócios e serviços disponibilizados na nuvem permitem desenvolver sistemas com inteligência artificial.**

Microserviços

- A praticidade dos microserviços unidos a facilidade do processamento na nuvem e a reusabilidade permitindo o desenvolvimento de novos sistemas com maior produtividade.

Finalizando

- O desenvolvimento de sistemas com uma arquitetura documentada e disponibilizada para os participantes do projeto aliado as tecnologias acabam diminuindo os riscos associados ao sistema.

X

Fechar