



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE COMPUTAÇÃO

# Arquitetura de Software

Eduardo Kinder Almentero  
[ekalmentero@gmail.com](mailto:ekalmentero@gmail.com)

# O que é arquitetura?

- A arquitetura é **estrutura** do software considerada no seu **nível de abstração mais alto**, que engloba os seus **elementos**, as **propriedades** destes elementos **visíveis externamente** e o **relacionamento** entre eles<sup>1</sup>.
- Dois tipos de estruturas são do interesse do arquiteto de software: **dinâmicas** e **estáticas**.
  - As estruturas **estáticas** de um software definem seus **elementos internos** e sua **organização em tempo de projeto**.
  - As estruturas **dinâmicas** definem **elementos** e suas **interações em tempo de execução**.

<sup>1</sup>Rozanski, Nick, and Eóin Woods. Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, 2011.

# Importância da Arquitetura de Software

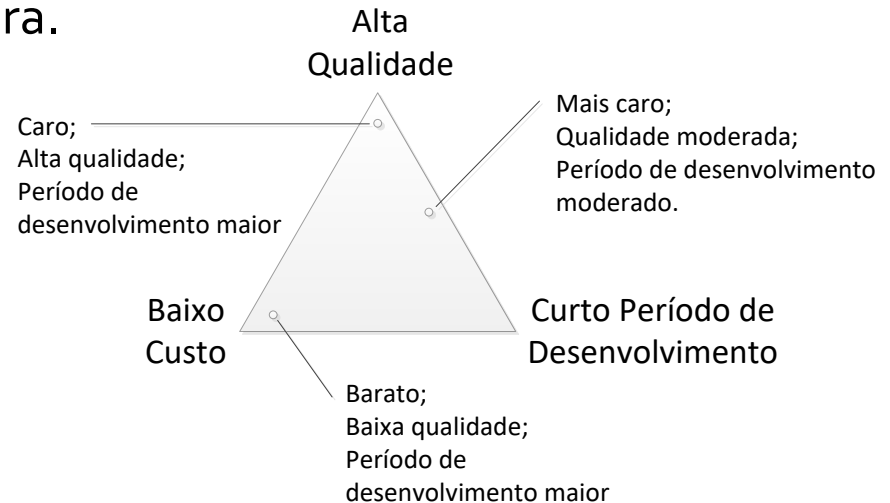
- Todo sistema de computador, grande ou pequeno, é **composto de partes** integradas, que se **comunicam umas com as outras** e com o **exterior**.
- Podemos ter apenas **uma parte, um conjunto pequeno** ou **centenas** de partes.
- A integração entre as partes pode ser **trivial, muito complicada, ou intermediária**.
- Desta forma, podemos tomar como princípio que **todo sistema de computador tem uma arquitetura, ainda que ela não seja conhecida ou documentada**.

# Conceitos

- Um **elemento arquitetural** é uma parte fundamental, a partir da qual um sistema pode ser considerado para ser construído.
- A **natureza** do elemento arquitetural dependerá do **tipo de sistema** e do **contexto** que seus elementos são considerados.
  - Bibliotecas de programação, subsistemas, produtos de software reutilizáveis (Ex.SGBDs) ou aplicações inteiras podem compor elementos arquiteturais em um sistema de informação.
- O **elemento arquitetural** deve possuir os seguintes elementos:
  - **Conjunto** bem definido de **responsabilidades**;
  - **Fronteira bem definida**;
  - **Conjunto bem definido de interfaces**, que definem os “serviços” que o elemento disponibiliza para os outros elementos arquiteturais.

# Conceitos

- Um **interesse** sobre uma arquitetura é um requisito, um objetivo, uma intenção ou uma aspiração que um stakeholder possui para esta arquitetura.
- Muitos **interesses** serão **comuns** entre os stakeholders, mas alguns serão **distintos** e podem até ser **conflitantes**.
- Resolver os conflitos de forma que **todos os stakeholders fiquem satisfeitos** é uma tarefa desafiadora.
- Uma **descrição arquitetural** é um conjunto de produtos que documentam a arquitetura de forma que os stakeholders possam entender e **demonstrar que a arquitetura atende suas necessidades**.



# Visões da arquitetura

- Ao iniciar a **complexa tarefa de projetar a arquitetura** do software, teremos que **responder algumas questões** difíceis, tais como:
  - Quais são os principais elementos funcionais da arquitetura?
  - Como estes elementos irão interagir uns com os outros e com o mundo externo?
  - Como as informações serão gerenciadas, armazenadas e apresentadas?
  - Quais elementos físicos de hardware e software serão requeridos para suportar os elementos funcionais?
  - Quais serão as características operacionais e capacidades oferecidas?
  - Quais ambientes de teste, suporte e treinamento serão fornecidos?

# Visões da arquitetura

- Uma tentação comum, que deve ser evitada a todo custo, é tentar responder todas as questões em um **único modelo, muito complexo.**
- Uma **visão da arquitetura** é uma forma de **retratar** os aspectos ou elementos da arquitetura que são relevantes de acordo com os **interesses de um grupo específico** de stakeholders.
- Uma **visão** é uma representação de um ou mais **aspectos estruturais** de uma arquitetura, que ilustra como a **ela aborda um ou mais interesses** de um ou mais de seus stakeholders

# Visões da arquitetura

- Questões que devem ser consideradas ao determinar uma visão da arquitetura:
  - Que **grupos de stakeholders a visão pretende atender?**
    - Pode ser muito focada em uma classe de stakeholder ou até mesmo em um único, ou pode atender a um grupo grande, com diferentes interesses e conhecimento.
  - Qual o **conhecimento técnico** dos stakeholders?
    - Usuários, por exemplo, normalmente sabem pouco sobre detalhes de hardware e software.
  - Quais **interesses** dos stakeholders a view pretender atender?
  - **Quanto os stakeholders precisam saber** sobre este aspecto da arquitetura?



# Catálogos de pontos de vista

- Seria muito difícil se toda vez que fossemos criar uma arquitetura precisássemos **definir as diferentes visões e o conteúdo de cada uma**.
- A fim de reduzir esta dificuldade, a IEEE criou um **padrão** que define, de forma genérica, **quatro pontos de vista essenciais**:
  - Lógico;
  - Processos;
  - Físico;
  - Desenvolvimento.
- Baseado nas visões genéricas, Rozanski<sup>1</sup> definiu um catálogo com **6 pontos de vista**.

<sup>1</sup>Rozanski, Nick, and Eóin Woods. *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley, 2011.

# Catálogo de Pontos de Vista

- **Funcional**

- Descreve os **elementos funcionais** do sistema, suas **responsabilidades, interfaces e interações** primárias.
- Uma visão funcional é a **principal** da maioria das descrições arquiteturais e é muitas vezes a primeira parte da descrição que interessados tentam ler.
- Ela conduz a forma de outras estruturas do sistema, tais como a estrutura de informação, concorrência, implantação e assim por diante.

- **Informação**

- Descreve a forma como a **arquitetura armazena, manipula, gerencia e distribui** informação.

- **Concorrência**

- Descreve a **estrutura de concorrência** do sistema e mapeia os elementos funcionais para unidades de concorrência, com o propósito de identificar claramente as **partes do sistema que podem executar concorrentemente** e como isto é **coordenado e controlado**.

# Catálogo de Pontos de Vista

- **Desenvolvimento**

- Descreve a arquitetura que **suporta o processo de desenvolvimento**.
- Comunica os aspectos da arquitetura importantes para os stakeholders envolvidos **em desenvolvimento, teste, manutenção e evolução**.

- **Implantação**

- Descreve o ambiente **onde o software será implantado**.

- **Operação**

- Descreve como o **sistema será operado, administrado e suportado** quando estiver operando em seu ambiente de produção.

# Benefícios de Visões e Pontos de vista

- **Separação de conceitos;**
- **Comunicação com grupos distintos de stakeholders;**
- **Gerência da complexidade;**
- **Aumento da qualidade do software.**
  - Visão voltada para a equipe de desenvolvedores.

# Problemas de Visões e Pontos de vista

- **Inconsistência;**
  - Tarefa de **checagem** praticamente manual.
- **Seleção do conjunto incorreto de visões;**
  - Não é fácil **identificar o conjunto correto**.
  - Muitos fatores influenciam.
- **Fragmentação;**
  - **Muitos modelos independentes**.
  - Difícil de entender como um todo.

# Princípios

- Para que o processo de definição da arquitetura **tenha êxito** em sua execução, os seguintes **princípios** devem ser respeitados:
  - Deve ser **guiado pelos interesses dos *stakeholders***.
    - Estes interesses são **os principais, mas não os únicos** que devem ser **considerados**.
    - O processo deve **equilibrar os interesses** envolvidos no processo de criação do software de **maneira eficiente**, pois podem ocorrer **conflitos** ou **implicações incompatíveis**.
  - Deve **encorajar a comunicação das decisões arquiteturais, princípios e a própria solução aos *stakeholders***.
  - Deve **assegurar**, de forma **contínua**, que as **decisões e princípios sejam respeitados** durante todo o **ciclo de vida do processo** de desenvolvimento do software, até sua implantação.
  - Deve ser **estruturado**, isto é, deve ser **organizado através de etapas encadeadas**, com objetivos claros e entradas e saídas bem definidas.
    - Normalmente, as **saídas** de uma etapa servem como **entrada** para a **etapa seguinte**.

# Princípios

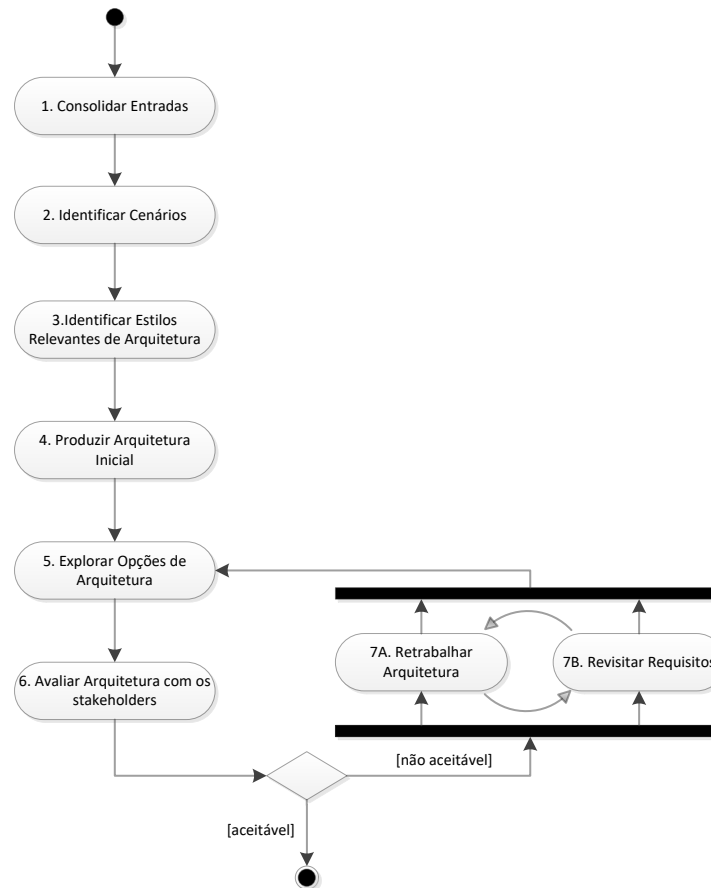
- Princípios (cont.)
  - Deve **ser pragmático**, isto é, **questões do mundo real**, como orçamento, tempo disponível, nível de certeza dos requisitos e habilidades técnicas dos envolvidos, **devem ser considerados** de forma realista.
  - Deve **ser flexível**, de forma que possa ser **adaptado para circunstâncias específicas**.
    - Esta característica é, normalmente, conhecida como abordagem de **framework** ou **toolkit**, seguindo a ideia de que as **partes necessárias são aproveitadas** e o resto é ignorado.
  - Em seus níveis mais altos, deve ser **independente de tecnologia**, isto é, o processo não deve impor o uso de uma tecnologia ou estilo de desenvolvimento específico.
    - Também não deve indicar uma modelagem ou estilo de documentação particular.
  - Deve ser **integrado ao ciclo de vida** escolhido para o desenvolvimento do software.
  - Deve **alinhar-se com boas práticas da engenharia de software e padrões para gerenciamento da qualidade** (como o CMMI-DEV), de forma que possa ser facilmente integrado com as abordagens existentes

# Processo para definição da arquitetura

- A maior dificuldade do arquiteto de software é lidar com as **incertezas e mudanças** que **surgem** quando os **stakeholders** são **reunidos**.
- Apesar do trabalho de construção da arquitetura ser **baseado** em um **escopo definido em comum acordo** ele está **sujeito a mudanças**, pois as **implicações** da inclusão ou exclusão de determinadas características **emergem**, à medida que os **stakeholders** **entendem melhor o que estão pedindo**.
  - Além disto, **requisitos funcionais e não funcionais** também estão **sujeitos a evolução**.
- Portanto, o **processo** de definição da arquitetura deve ser **iterativo**.
  - As etapas principais devem ser repetidas até que uma versão final do documento de arquitetura seja produzida.
- Como a **arquitetura evolui no decorrer do processo** de desenvolvimento de software, é preciso **retornar ao ciclo de atividades** relacionado a definição da arquitetura **durante todo o projeto**.



# Processo para definição da arquitetura



# Etapas do Processo

- Etapa 1 – Consolidar Entradas

<b>Objetivo</b>	Entender, validar e refinar as entradas iniciais.
<b>Entradas</b>	Artefatos não estruturados do processo (escopo e definição do contexto, interesses dos stakeholders)
<b>Saídas</b>	Entradas consolidadas, com as maiores inconsistências resolvidas e as questões em aberto respondidas. Identificação de áreas que requerem uma exploração futura.
<b>Atividades</b>	Resolver as inconsistências entre as entradas recebidas, elaborar de respostas as perguntas em aberto e aprofundar o conhecimento, quando necessário, a fim de produzir uma base sólida
<b>Comentários</b>	É raro receber artefatos de entrada do processo que sejam consistentes, precisos e aprovados por todos os stakeholders. Durante esta etapa, é necessário documentar a informação disponível, preencher as lacunas, resolver inconsistência e obter consenso dos stakeholders chave.

# Etapas do Processo

- Etapa 2 – Identificar Cenários

<b>Objetivo</b>	Identificar um conjunto de cenários que ilustram os requisitos mais importantes do sistema.
<b>Entradas</b>	Entradas consolidadas.
<b>Saídas</b>	Cenários de arquitetura
<b>Atividades</b>	Produzir um conjunto de cenários que caracterizem os atributos mais importantes necessários para a arquitetura. Estes cenários poderão ser utilizados para avaliar se a arquitetura proposta atende aos requisitos funcionais e não funcionais.
<b>Comentários</b>	Um cenário é uma a descrição de uma situação possível, que permite a avaliação da eficácia da arquitetura. Os cenários podem ser identificados por comportamentos de requisitos funcionais ("Como o sistema faz X?") e por propriedades desejadas de qualidade ("Como o sistema lida com a carga Y").

# Etapas do Processo

- Etapa 3 – Identificar Estilos Relevantes de Arquitetura

<b>Objetivo</b>	Identificar um ou mais estilos arquiteturais comprovados, que possam ser utilizados como base para a organização geral do software.
<b>Entradas</b>	Entradas consolidadas; cenários arquiteturais.
<b>Saídas</b>	Estilos arquiteturais que devem ser considerados como base para as estruturas principais da arquitetura.
<b>Atividades</b>	Revisão de catálogos existentes de estilos arquiteturais; considerar as organizações de software com as quais o arquiteto tenha trabalhado bem anteriormente. Identificar aquelas que parecem ser relevantes para a arquitetura atual, conforme o conhecimento sobre ela vai aumentando.
<b>Comentários</b>	Utilizar um estilo arquitetural é uma forma de reutilizar o conhecimento sobre arquitetura que já demonstrou ser eficaz em situações anteriores. Isto pode auxiliar a construção de uma organização adequada para o software, sem que ela precise ser feita do zero. Isto reduz os riscos envolvendo novas ideias, que ainda não foram comprovadas.

# Etapas do Processo

- Etapa 4 – Produzir a arquitetura inicial

---

<b>Objetivo</b>	Criar uma primeira versão da arquitetura para o software, que reflita os interesses principais e possa atuar como uma base, permitindo uma avaliação aprofundada da arquitetura e um refinamento futuro.
<b>Entradas</b>	Entradas consolidadas; estilos arquiteturais relevantes; pontos de vista; perspectivas.
<b>Saídas</b>	Esboço das diferentes visões da arquitetura.
<b>Atividades</b>	Produzir um conjunto inicial de visões da arquitetura, com o intuito de definir as ideias iniciais sobre a arquitetura, utilizando como guia os pontos de vista e perspectivas e qualquer estilo arquitetural relevante.
<b>Comentários</b>	Embora possam conter lacunas, inconsistências ou erros, o esboço inicial das visões consolida um ponto inicial para que o trabalho de detalhamento seja executado mais adiante.

---

# Etapas do Processo

- Etapa 5 – Explorar opções de arquitetura

<b>Objetivo</b>	Explorar as diversas possibilidades de arquitetura para o software e tomar decisões arquiteturais importantes, com o intuito de escolher, dentre as possibilidades, as que mais se adéquam a realidade do software.
<b>Entradas</b>	Entradas consolidadas; esboço inicial das visões; cenários arquiteturais; pontos de vista; perspectivas.
<b>Saídas</b>	Visões mais precisas ou detalhadas para algumas partes da arquitetura.
<b>Atividades</b>	Aplicar os cenários ao esboço dos modelos, a fim de demonstrar que eles são viáveis, que atendem aos requisitos e que não há problemas escondidos. Tomar todas as áreas de risco, preocupação e incerteza que são reveladas e explorar ainda mais as exigências, problemas e questões. Onde houver mais de uma solução possível, avaliar os pontos fortes e fracos de cada uma, com o propósito de selecionar a mais adequada.
<b>Comentários</b>	O objetivo desta etapa é preencher as lacunas, remover inconsistências nos modelos e fornecer mais detalhes onde é necessário.

# Etapas do Processo

- Etapa 6 – Avaliar a arquitetura com os stakeholders

---

<b>Objetivo</b>	Analisar a arquitetura com os stakeholders chave, capturar qualquer problema e deficiência e obter a aprovação da arquitetura pelos stakeholders.
<b>Entradas</b>	Entradas consolidadas; visões arquiteturais e perspectivas.
<b>Saídas</b>	Comentários de revisão da arquitetura.
<b>Atividades</b>	Aplicar os Avaliar a arquitetura com um conjunto de stakeholders representativos (de cada grupo). Capturar e concordar com qualquer melhoria ou comentários sobre os modelos.
<b>Comentários</b>	Embora cada grupo de stakeholders tenha interesses distintos, o objetivo geral é confirmar que os interesses de todos sejam atendidos e que a arquitetura tenha uma boa qualidade. O trabalho para obter o consenso entre os stakeholders pode ser difícil se seus interesses forem conflitantes.

---

# Etapas do Processo

- Etapa 7A – Retrabalhar arquitetura

---

<b>Objetivo</b>	Abordar qualquer questionamento que tenha surgido durante a fase de avaliação.
<b>Entradas</b>	Visões arquiteturais; comentários de revisão da arquitetura; estilos relevantes de arquitetura; pontos de vista; perspectivas.
<b>Saídas</b>	Visões arquiteturais refeitas; áreas marcadas para posterior investigação (opcional).
<b>Atividades</b>	Tomar os resultados da avaliação arquitetural e abordá-los a fim de produzir uma arquitetura mais adequada aos seus objetivos. Esta etapa envolve, normalmente, análise funcional, uso de pontos de vista e perspectivas e prototipação.
<b>Comentários</b>	Esta etapa é feita de forma concorrente e, muitas vezes, colaborativa com a etapa 7B. O resultado destas duas etapas é utilizado como entrada na etapa 5.

---



# Etapas do Processo

- Etapa 7B – Revisitar requisitos

---

<b>Objetivo</b>	Considerar qualquer mudança nos requisitos do software que podem impactar na avaliação da arquitetura.
<b>Entradas</b>	Visões arquiteturais; comentários de revisão da arquitetura.
<b>Saídas</b>	Requisitos revisados (se houver).
<b>Atividades</b>	O trabalho realizado até este ponto revela inadequações ou inconsistências em requisitos, ou ainda, requisitos que são inviáveis ou muito caros para implementar. Neste caso, pode ser necessário revisitar estes requisitos com os stakeholders e obter sua aprovação sobre as revisões necessárias.
<b>Comentários</b>	Esta etapa é feita de forma concorrente e, muitas vezes, colaborativa com a etapa 7A. O resultado destas duas etapas é utilizado como entrada na etapa 5.

---

# Como determinar o fim do processo

- Em um **cenário ideal**, o processo de definição da arquitetura **continuará** até que a **arquitetura estivesse perfeita** e fosse **perfeitamente documentada** através do documento de arquitetura.
- Entretanto, como nesta fase é feita uma **análise de alto nível** do software que será construído, é **difícil**, na prática, **confiar que a arquitetura produzida é a correta** para os problemas que pretende resolver.

# Como determinar o fim do processo

- O **melhor critério** para determinar quando a arquitetura está terminada, e é possível iniciar a etapa de construção, é quando não há **comentários significativos** resultantes da etapa de avaliação da arquitetura, isto é, a arquitetura não necessita de nenhuma mudança.
- Em outras palavras, o **processo de construção da arquitetura** está **completo** quando **todos os stakeholders concordam** que o documento de arquitetura aborda de forma adequada seus interesses.

# Estilos e Padrões Arquiteturais

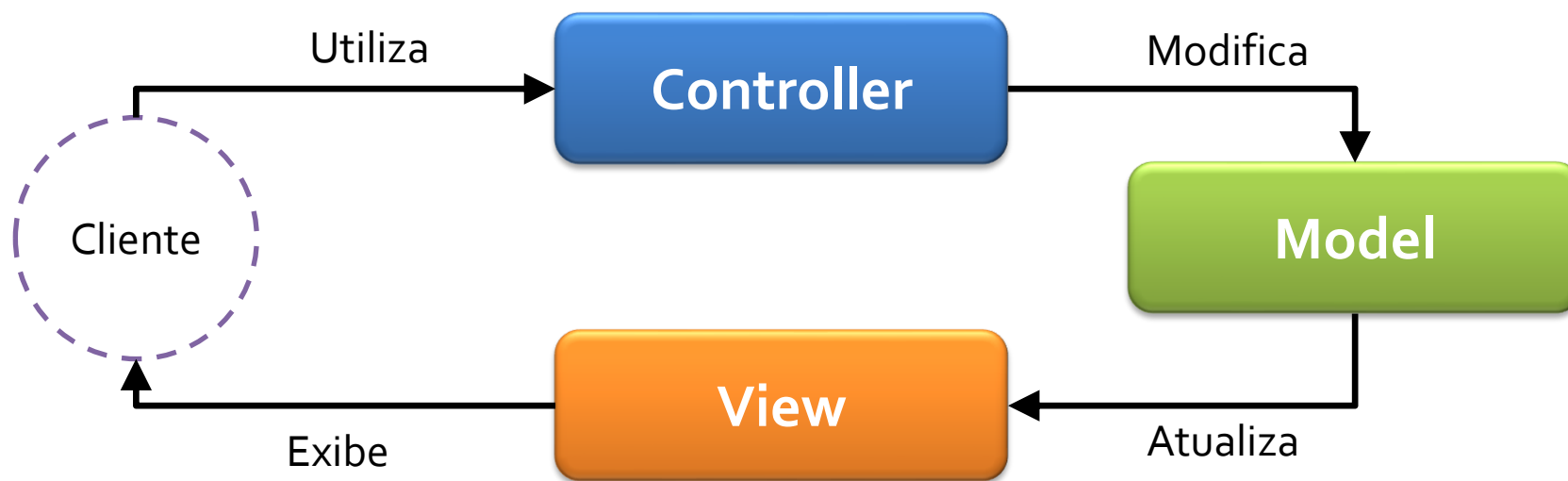
- Três camadas (*three-tier*)



# Estilos e Padrões Arquiteturais

- Três camadas (*three-tier*)
  - Apresentação
    - Permite a exposição **de conteúdo produzido/armazenado** pelo sistema para **elementos externos**.
  - Lógica
    - Determina o **fluxo do sistema** e contém as **regras de negócio**, isto é, detém o **controle das funcionalidades**.
  - Dados
    - Inclui **componentes para persistência de dados** (servidores de banco de dados, compartilhamento de arquivos, etc.)
    - Também inclui **componentes para exposição dos dados**.
    - Normalmente, a camada de dados oferece uma **API** para o sistema, expondo **métodos para gerenciar os dados** armazenados **sem criar dependência**.

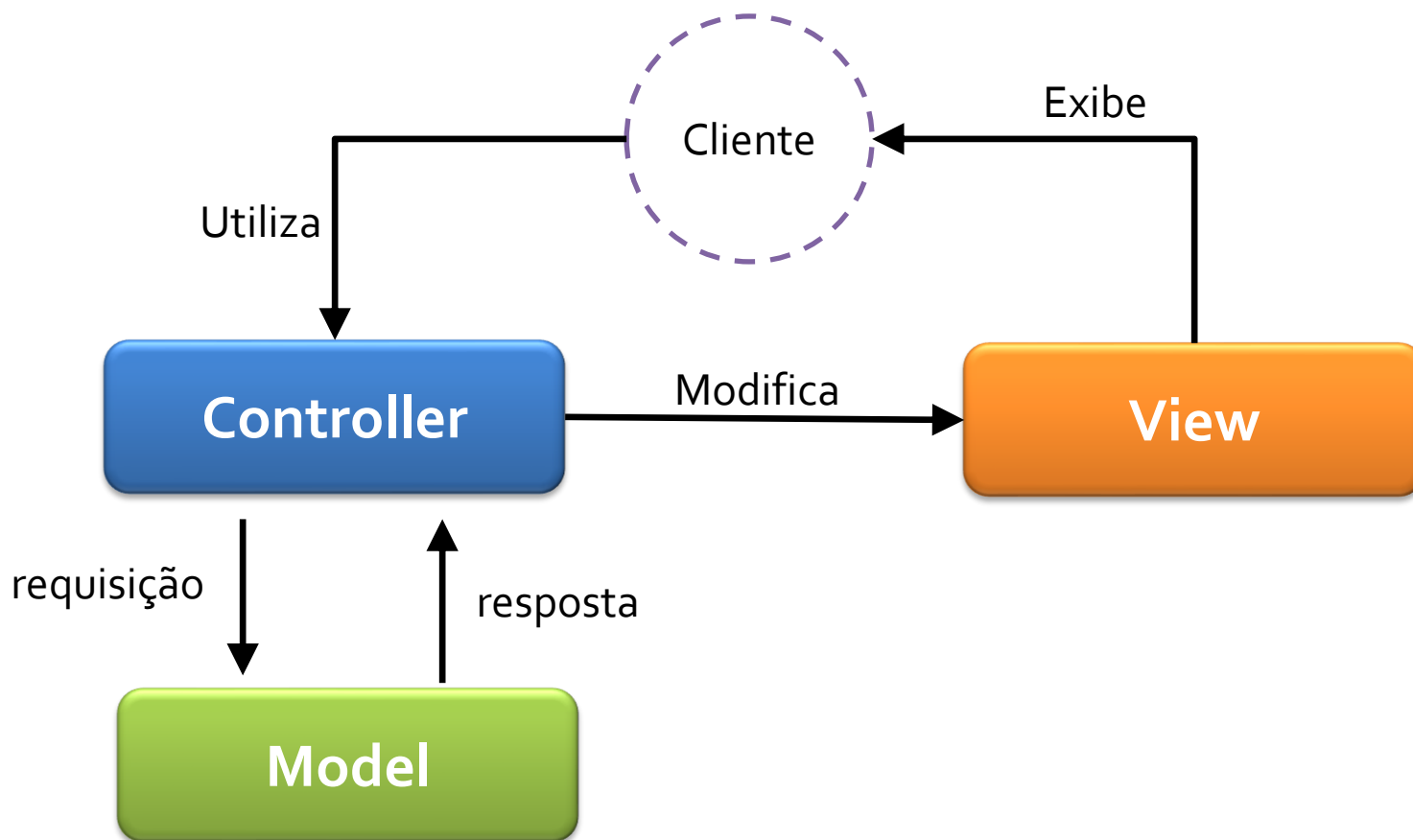
# Padrão MVC (*Model View Controller*)



# Padrão MVC (*Model View Controller*)

- A proposta do padrão é dividir um sistema em **três partes interconectadas, separando a apresentação da informação de sua representação.**
- Camadas
  - Visão
    - Contém a interface gráfica para interação com os usuários.
  - Controle
    - Controla o fluxo do sistema e contém regras simples.
  - Modelo
    - Contém o conhecimento sobre a representação dos dados e as regras de negócio.

# Variação do MVC





# Referências

- Rozanski, Nick, and Eóin Woods. Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, 2011.



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE COMPUTAÇÃO

Perguntas?