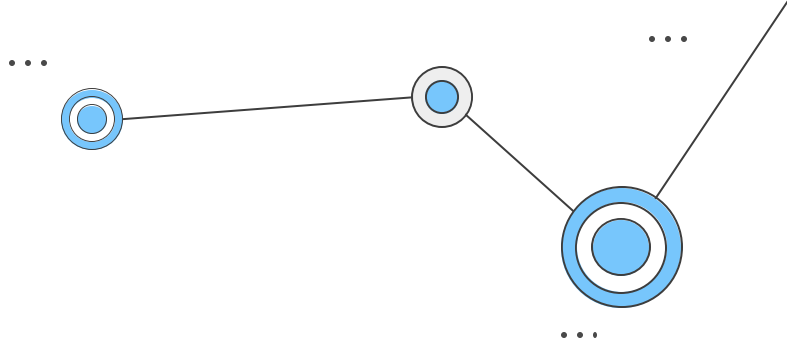




PUC Minas



Projeto de Software

Prof. Dr. João Paulo Aramuni



Unidade 2

Arquitetura de Software

PDS - Manhã /Noite

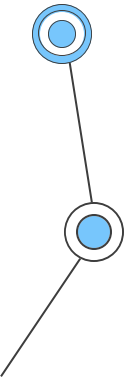
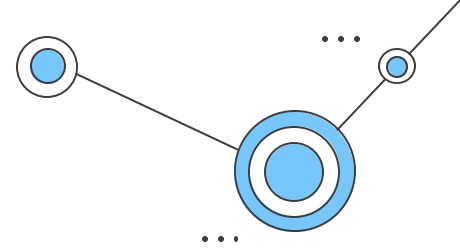


Arquitetura de Software – Sistemas Web

Aplicação Web

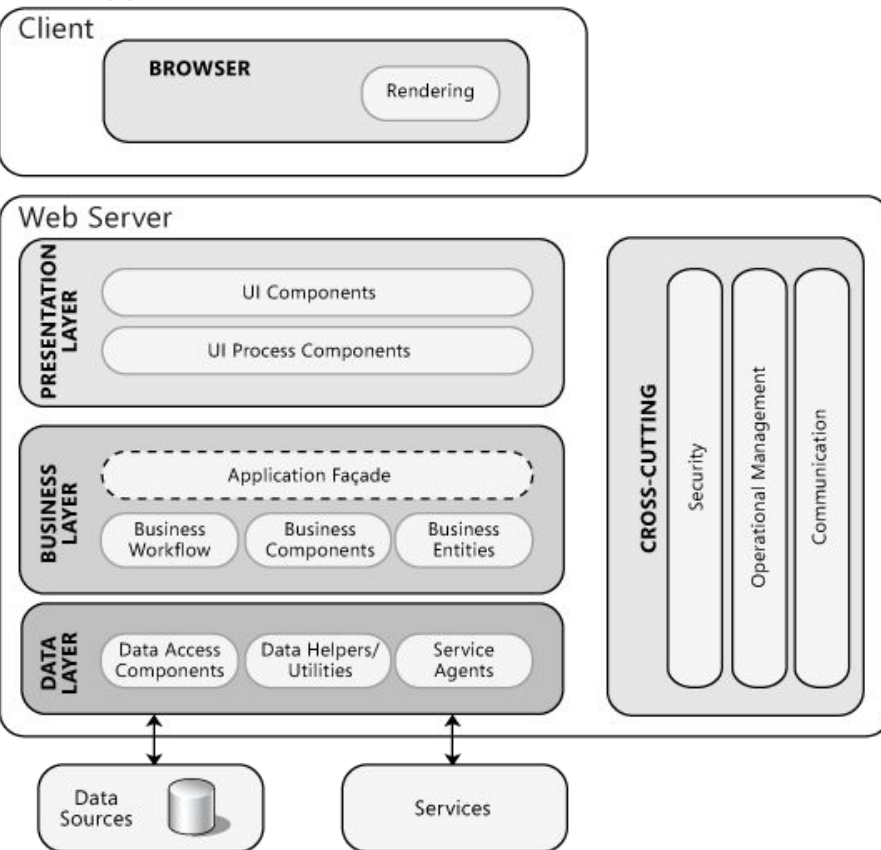
O que é Aplicação Web?

Qualquer aplicação que seja hospedada em um servidor Web e opere sobre o protocolo HTTP.

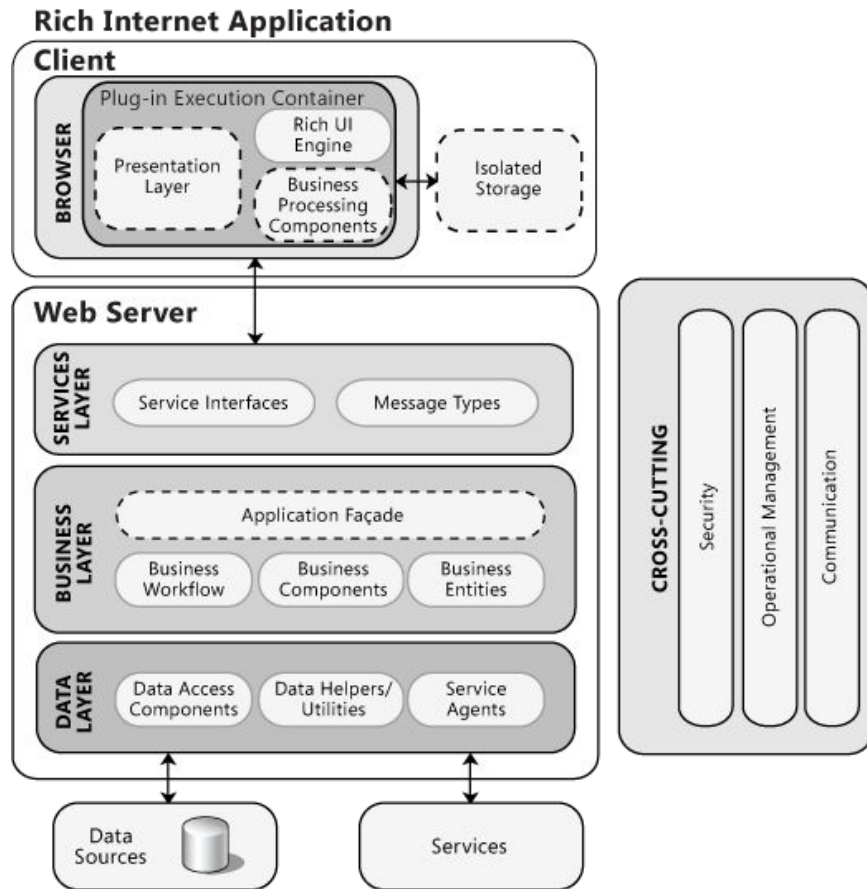


Arquitetura Web Application

Web Application



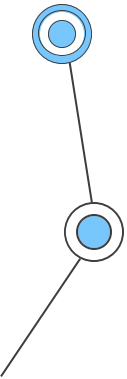
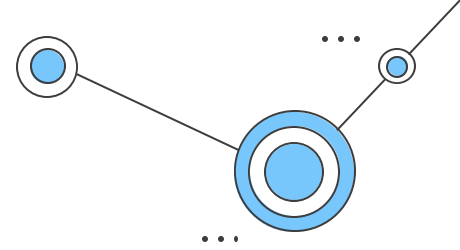
Arquitetura Rich Internet Application



Arquitetura de Software – Sistemas Web

Servidores Web

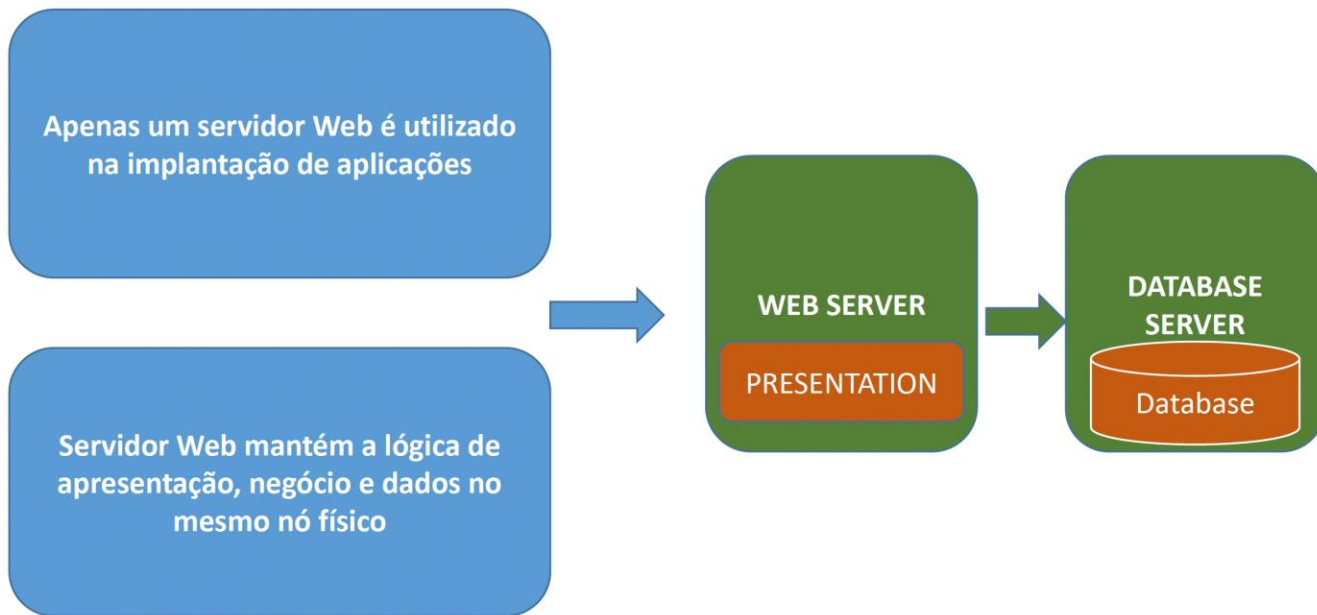
- Software que suporta requisições e respostas sobre protocolo HTTP
- Exemplos: Apache Web Server, Microsoft IIS, JBOSS Web Server.



Arquitetura de Software – Sistemas Web

Distribuição da Aplicação

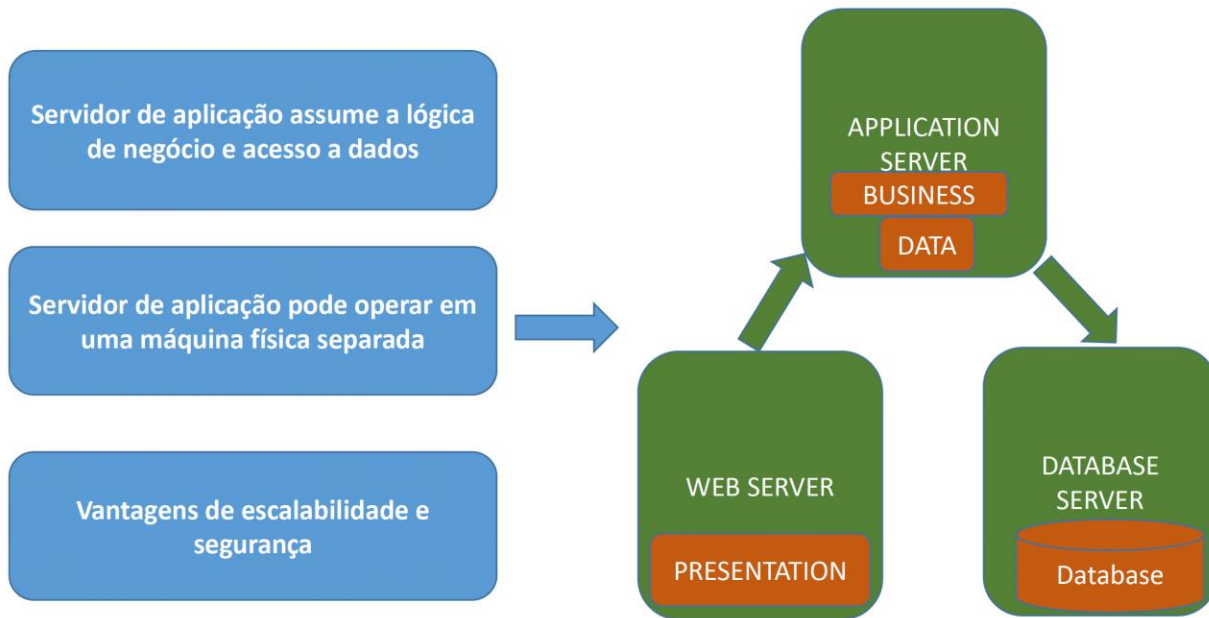
Apenas um servidor Web



Arquitetura de Software – Sistemas Web

Distribuição da Aplicação

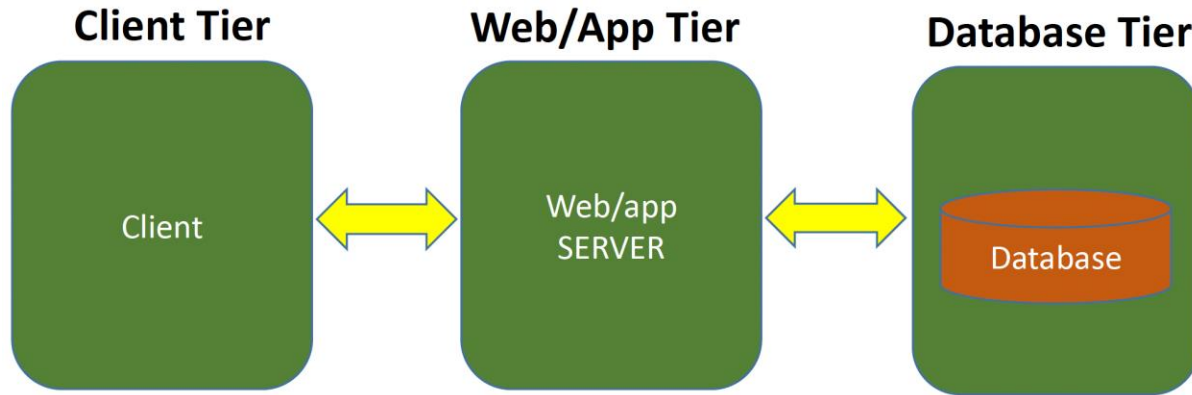
Servidor Web e Aplicação



Arquitetura de Software – Sistemas Web

Distribuição da Aplicação

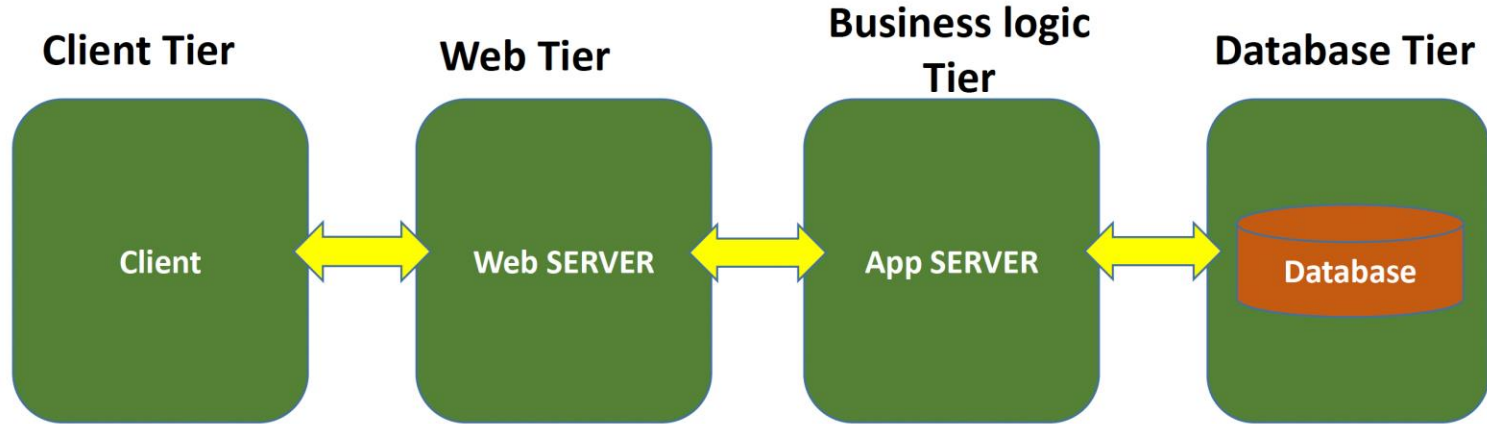
3 camadas



Arquitetura de Software – Sistemas Web

Distribuição da Aplicação

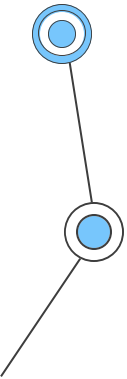
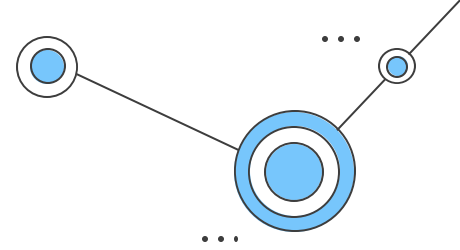
4 camadas



Arquitetura de Software – Sistemas Web

Padrões Arquiteturais

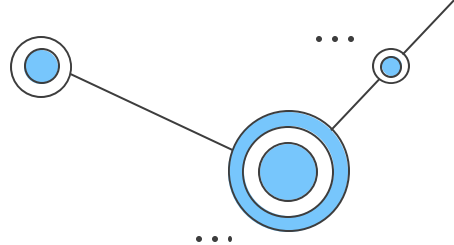
- MVC
- MVP e
- MVVM



Arquitetura de Software – Sistemas Web

O que é Model View (C ou P ou VM)?

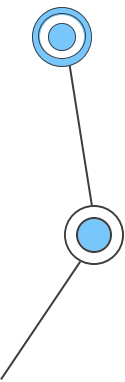
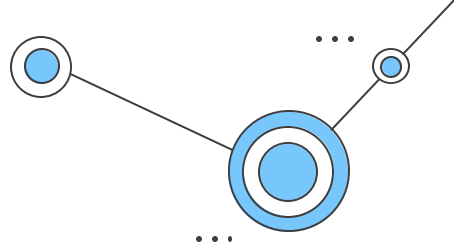
- Padrões que descrevem uma abordagem para desenvolvimento de software
- Os módulos incluem
 - Model – Dados e comportamento
 - View – Camada de apresentação
 - C ou VM ou P – Lógica para “grudar as coisas”
- Esses padrões se baseiam em “Separação de Responsabilidades”
 - Como em muitos outros tipos de frameworks de sistemas



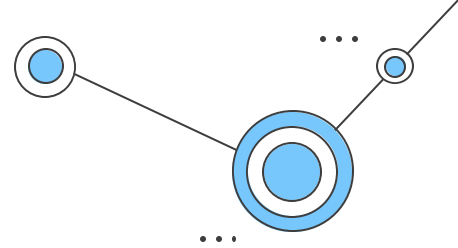
Arquitetura de Software – Sistemas Web

Por que Model View (C ou P ou VM)?

- Os padrões possuem objetivos semelhantes. Contudo, o fazem de modos diferentes.
- Os objetivos dos padrões são aumentar:
 - Modularidade
 - Flexibilidade
 - Testabilidade
 - Manutenibilidade

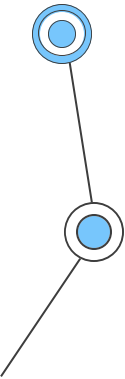


Arquitetura de Software – Sistemas Web



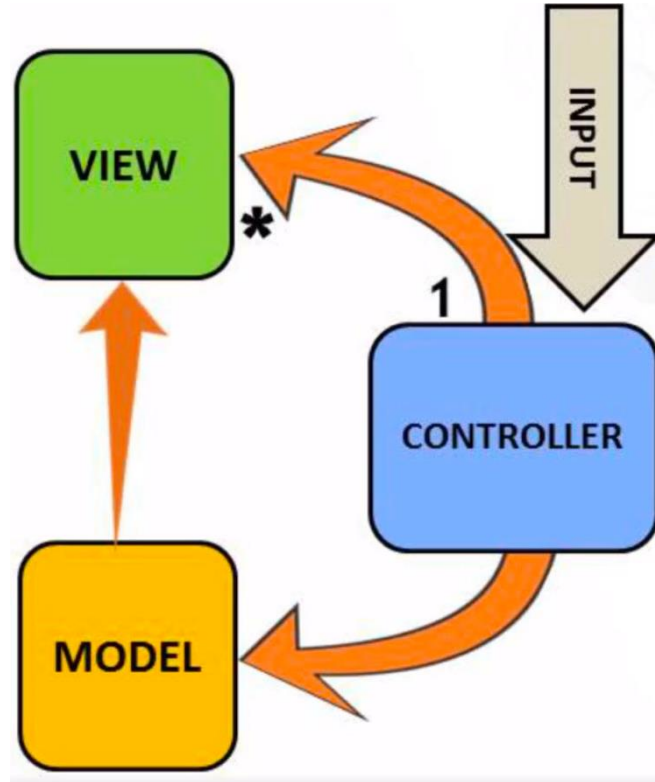
Model View Controller (MVC)

- Descrito inicialmente em 1979 para SmalTalk para Xerox PARC.
- O controller é a peça central que desacopla o Model e o View.
- Fluxo de controle:
 - Evento de interação do usuário.
 - Controller manipula o evento e o converte para uma ação do usuário que o Model possa entender.
 - Model gerencia o comportamento e dados do domínio da aplicação.
 - O View interage com o Controller e o Model para gerar a interface com o usuário.

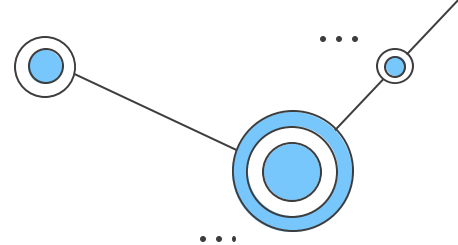


Arquitetura de Software – Sistemas Web

MVC

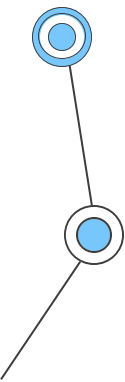


Arquitetura de Software – Sistemas Web



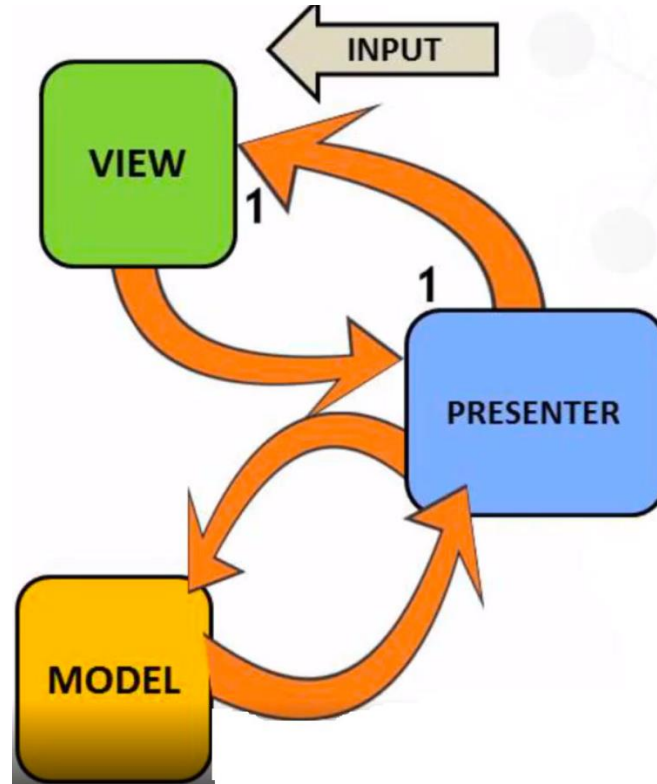
Model View Presenter (MVP)

- MVP originou-se no início dos anos 1990.
- MVP é derivado do MVC.
- Dois tipos de implementação
 - Passive View
 - Supervising Controller
- Presenter assume a funcionalidade do MVC Controller.
- View é responsável por manipular os eventos da IU.
- Model torna-se estritamente um Modelo de Domínio (Domain Model).
- Mais voltado a Interface com Usuário (IU).

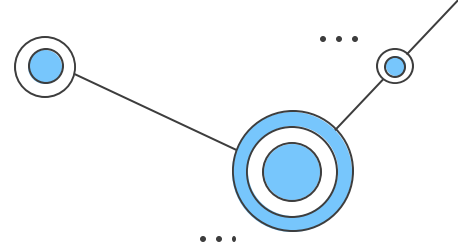


Arquitetura de Software – Sistemas Web

MVP

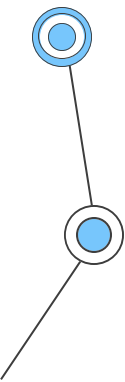


Arquitetura de Software – Sistemas Web



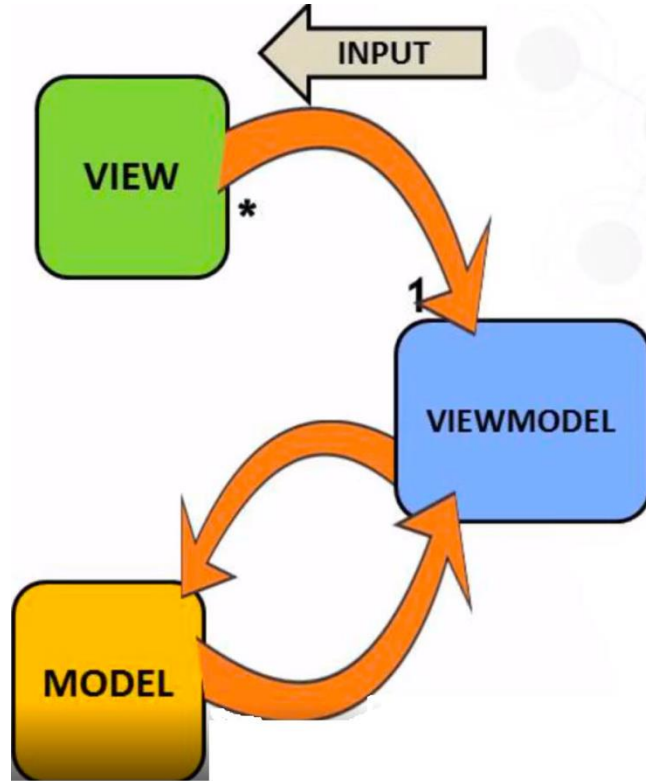
Model View ViewModel (MVVM)

- Especialização do MVP, conhecida como Modelo de Apresentação (Presentation Model)
- Construído especificamente para os ambientes WPF e Silverlight
- Model e View funcionam como no MVC
- ViewModel é um “Model da View”
 - Ele estende o Model com comportamentos (behaviors) que o View possa usar
 - Associação de dados (data binding) entre View e Model
 - Passa comandos entre View e Model



Arquitetura de Software – Sistemas Web

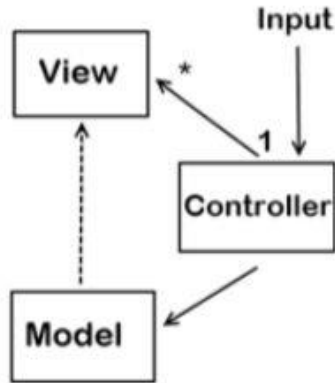
MVVM



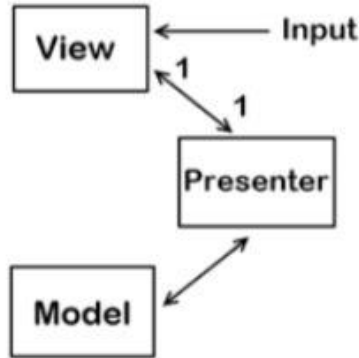
Arquitetura de Software – Sistemas Web

MVC x MVP x MVVM

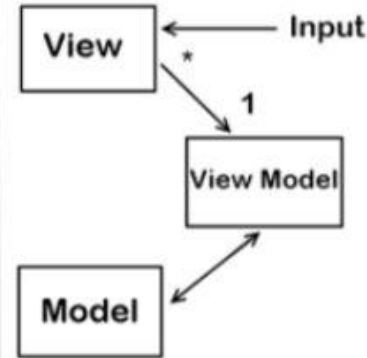
MVC x MVP x MVVM



MVC



MVP



MVVM

Fonte: **MVVM Compared To MVC and MVP** - <http://geekswithblogs.net/dlussier/archive/2009/11/21/136454.aspx>

PUC Minas **Virtual**

Arquitetura de Software – Sistemas Web

Plataforma Aplicações Web

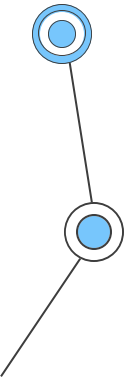
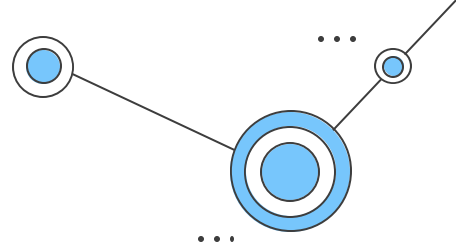
Tecnologias Web mais populares

LAMP

Java EE Web

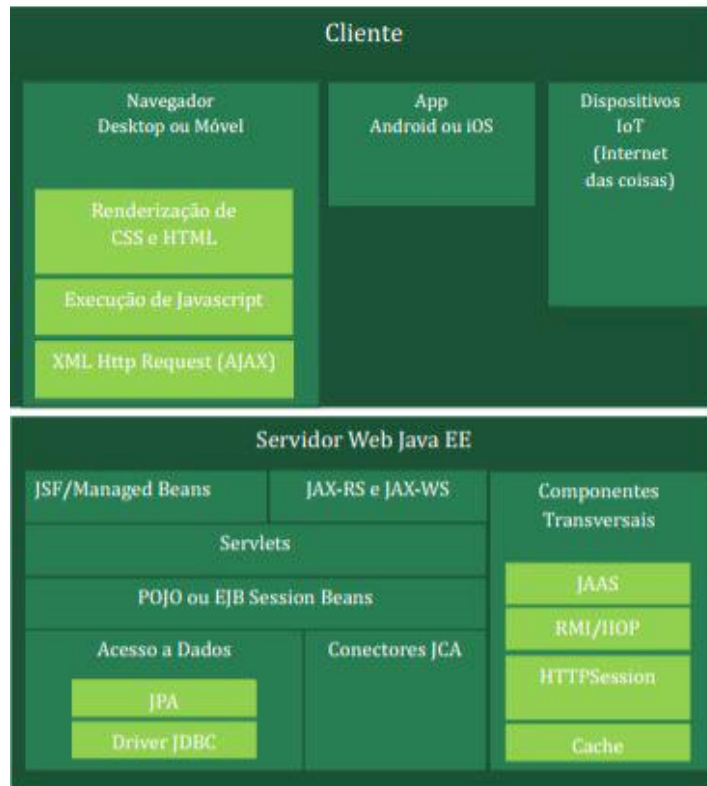
ASP.NET

Centrados em
javascript como
o Node.js



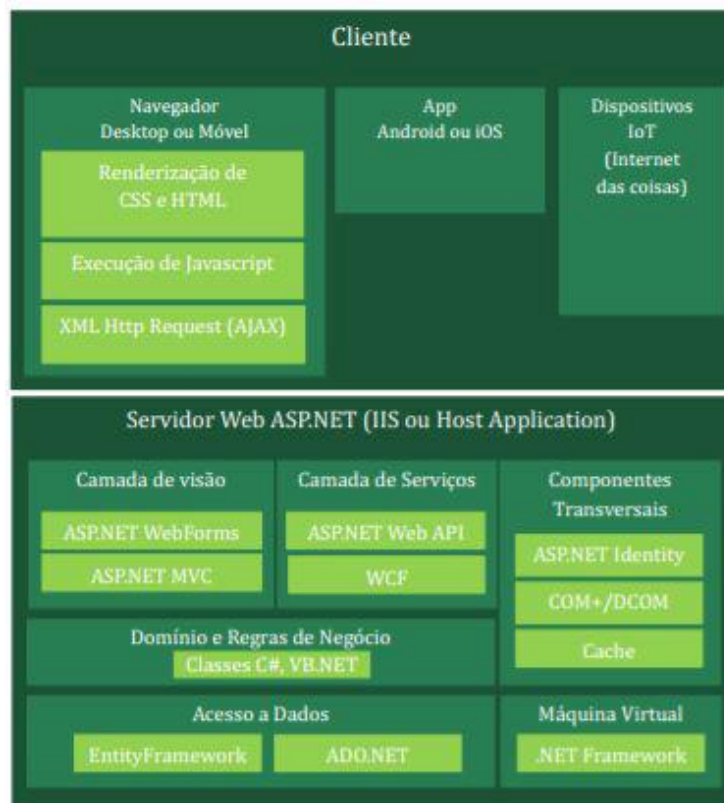
Arquitetura de Software – Sistemas Web

Plataforma Java EE Web



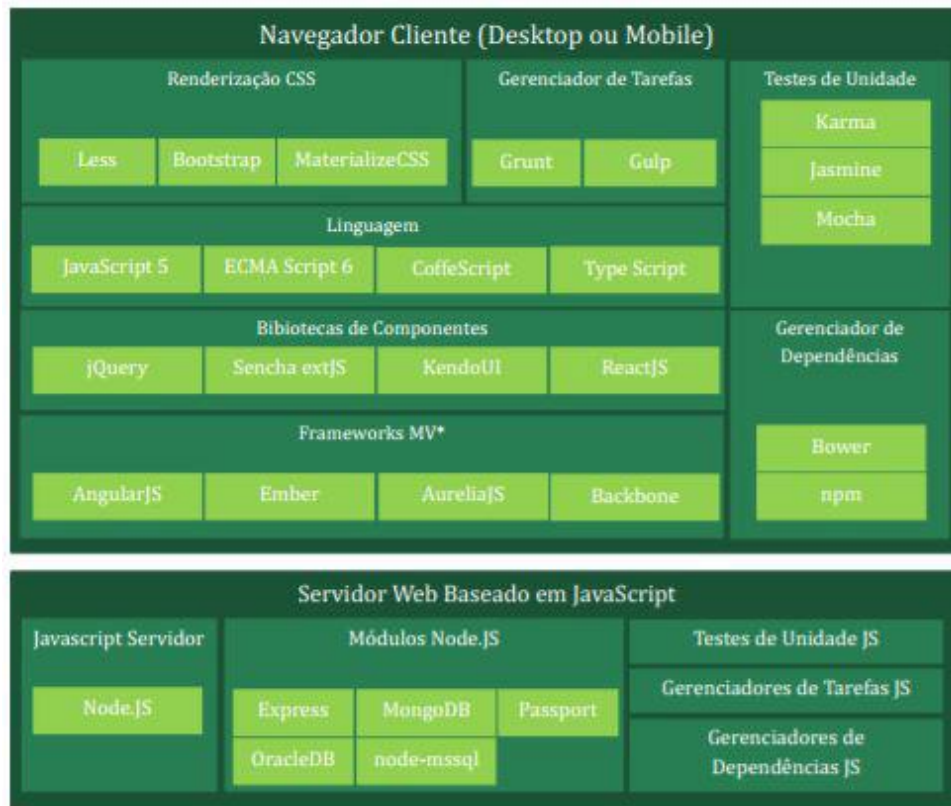
Arquitetura de Software – Sistemas Web

Plataforma Asp.Net



Arquitetura de Software – Sistemas Web

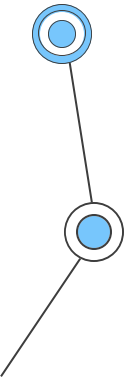
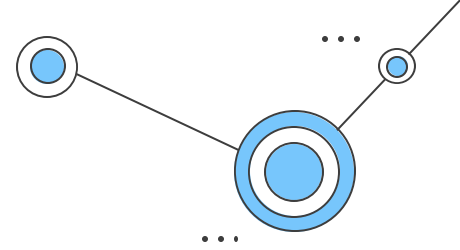
Plataforma Javascript

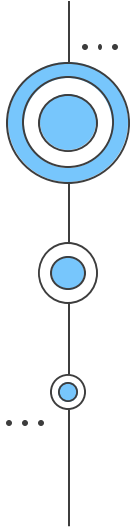


Arquitetura de Software – Sistemas Web

Plataforma LAMP

- Termo genérico que denota o desenvolvimento de aplicações em ambiente Linux, Apache httpD, MySQL e PHP.
- Generalizado para sinalizar o uso de tecnologias rápidas e simples para o desenvolvimento de aplicações Web.
- Termo generalizado para:
 - Linguagens dinâmica: PHP, Python, Ruby ou Groovy.
 - SGBD's abertos: MySQL, PostgreSQL.
 - Linux (SISE, Redhat, outros e Windows.

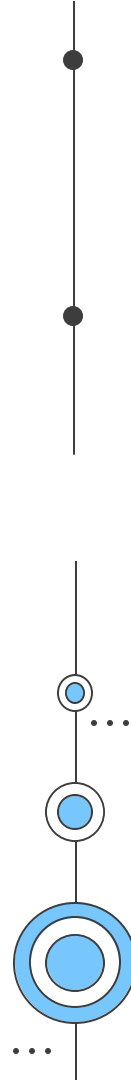


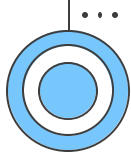


Projeto de Software

Referências básicas:

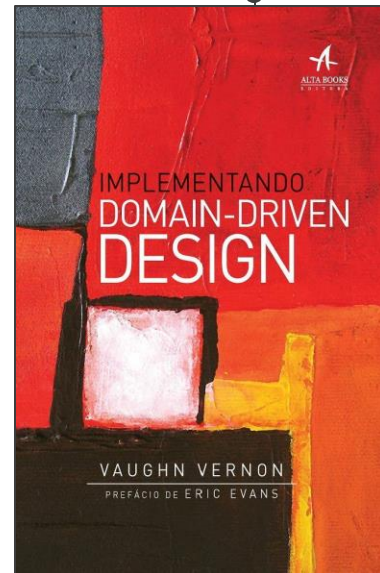
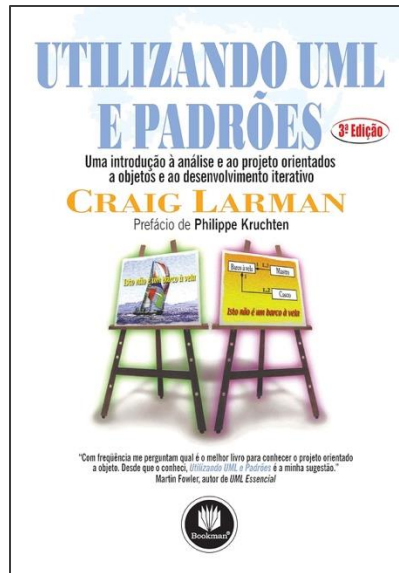
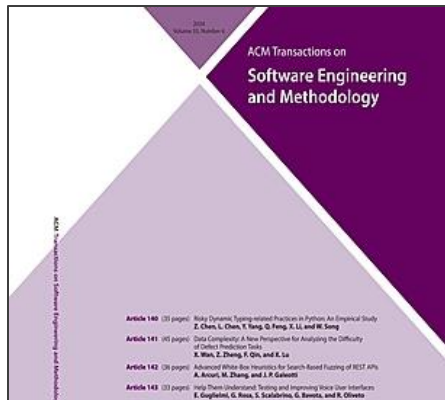
- **ACM TRANSACTIONS ON SOFTWARE ENGINEERING AND METHODOLOGY**. New York, N.Y., USA: Association for Computing Machinery, 1992-. Trimestral. ISSN 1049-331X. Disponível em: <https://dl.acm.org/toc/tosem/1992/1/2>. Acesso em: 19 jul. 2024. (Periódico On-line).
- LARMAN, Craig. **Utilizando UML e padrões**: uma introdução á análise e ao projeto orientados a objetos e desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. E-book. ISBN 9788577800476. (Livro Eletrônico).
- SILVEIRA, Paulo et al. **Introdução à arquitetura e design de software**: uma visão sobre a plataforma Java. Rio de Janeiro, RJ: Elsevier, Campus, 2012. xvi, 257 p. ISBN 9788535250299. (Disponível no Acervo).
- VERNON, Vaughn. **Implementando o Domain-Driven Design**. Rio de Janeiro, RJ: Alta Books, 2016. 628 p. ISBN 9788576089520. (Disponível no Acervo).



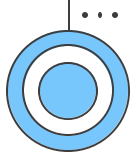


Projeto de Software

Referências básicas:



...

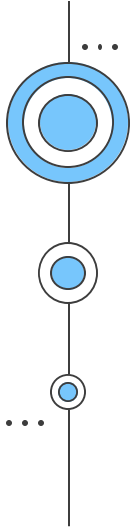


Projeto de Software

Referências complementares:

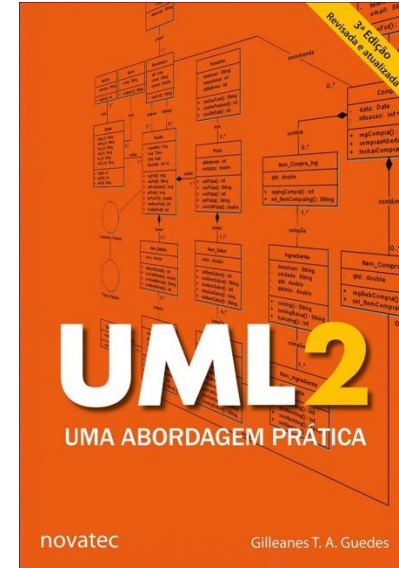
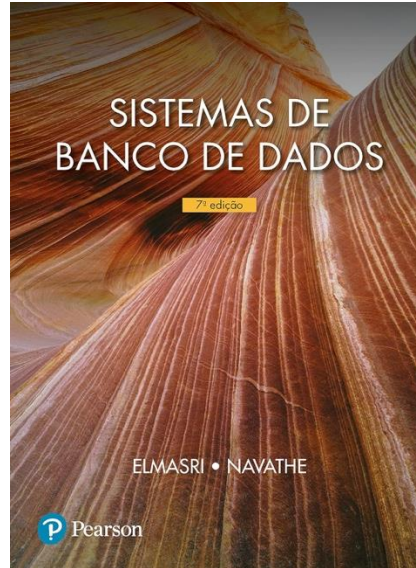
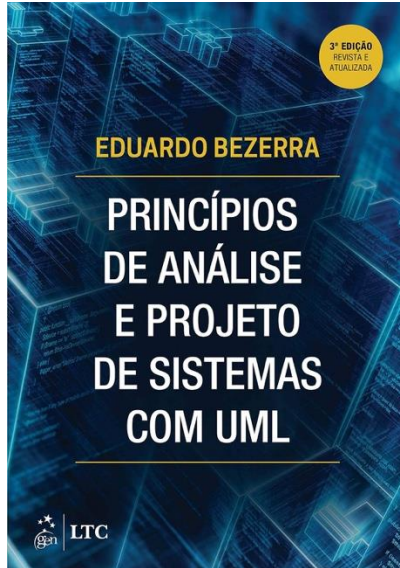
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3. ed. rev. e atual. Rio de Janeiro: Elsevier, 2015. xvii, 398 p. ISBN 9788535226263. (Disponível no Acervo).
- ELMASRI, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**, 7ª ed. Editora Pearson 1152 ISBN 9788543025001. (Livro Eletrônico).
- GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec, c2011. 484 p. ISBN 9788575222812. (Disponível no Acervo).
- **IEEE TRANSACTIONS ON SOFTWARE ENGINEERING**. New York: IEEE Computer Society, 1975-. Mensal,. ISSN 0098-5589. Disponível em: <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32>. Acesso em: 19 jul. 2024. (Periódico On-line).
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil, c2019. xii, 756 p. ISBN 9788543024974. (Disponível no Acervo).
- WAZLAWICK, Raul Sidnei. **Análise e design orientados a objetos para sistemas de informação: modelagem com UML, OCL e IFML**. 3. ed. Rio de Janeiro, RJ: Elsevier, Campus, c2015. 462 p. ISBN 9788535279849. (Disponível no Acervo).



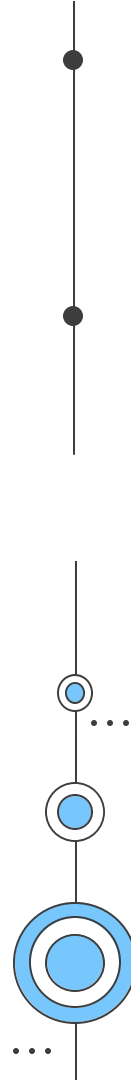


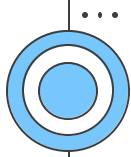
Projeto de Software

Referências complementares:



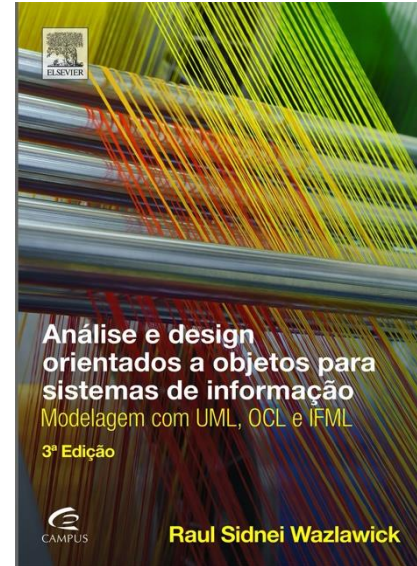
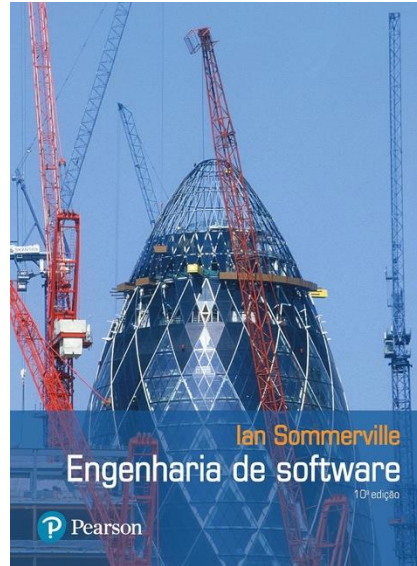
...





Projeto de Software

Referências complementares:



Obrigado!

Dúvidas?

joaopauloaramuni@gmail.com



[GitHub](#)



[LinkedIn](#)



[Lattes](#)

...