

---

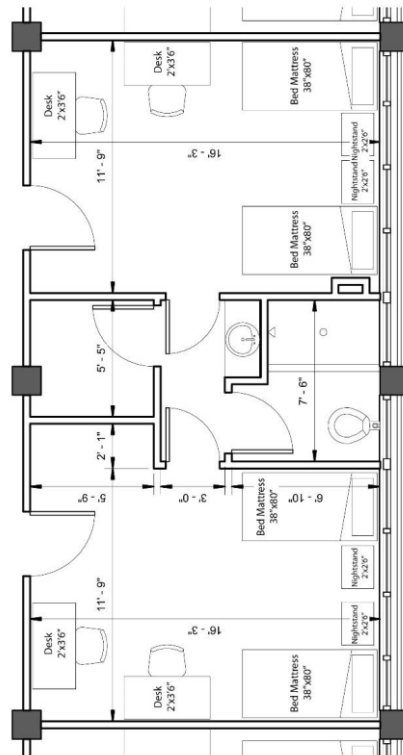
# **Desenvolvimento de Aplicações Web**

Arquitetura de software para a Web

Prof. Márcio Barros

# Mas o que é arquitetura?

- ❑ É o processo pelo qual uma obra de construção civil é planejada, projetada e construída
- ❑ A arquitetura define **abordagens padronizadas** para tratar certos problemas recorrentes e indica como as responsabilidades são divididas entre os participantes



# Estilos arquitetônicos

Diferentes **estilos arquitetônicos** foram criados por diferentes sociedades para atacar os seus problemas específicos.



Catedral de Notre Dame, Paris, França



Praça de San Marco – Veneza, Itália



Santorini, Grécia



Pirâmides de Giza, Egito



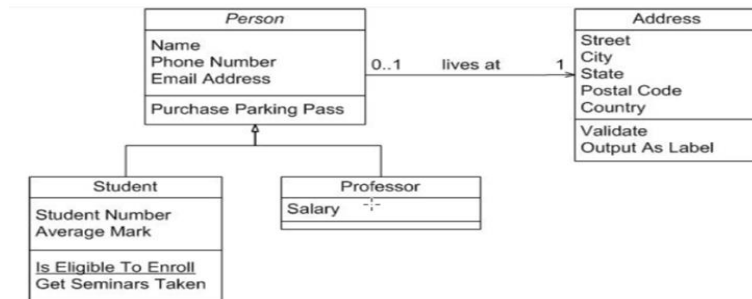
Arranha-céus – Nova York, EUA



Hagia Sofia – Istambul, Turquia

# Mas como se define um estilo Web?

- Algumas perguntas orientam a arquitetura
  - Como a aplicação Web deve ser implementada?
  - Que componentes de software participam da arquitetura?
  - Quais são as responsabilidades destes componentes?
  - Que componente roda em que parte da aplicação?
  - Qual é a melhor maneira de organizar esta aplicação?



# Arquitetura de software

## Lado Cliente

HTML: linguagem para definir a organização visual da aplicação e os dados apresentados

CSS (*cascading style-sheets*): padrão de formatação dos componentes HTML

Script de lado cliente: permite que a aplicação execute parte da sua lógica no lado cliente.



## Lado Servidor

Bancos de dados: armazenamento dos dados manipulados pelo sistema

Modelos: representação em memória dos dados armazenados no banco

Serviços: oferecem acesso a recursos da plataforma computacional (e-mail, documentos, relatórios, etc).

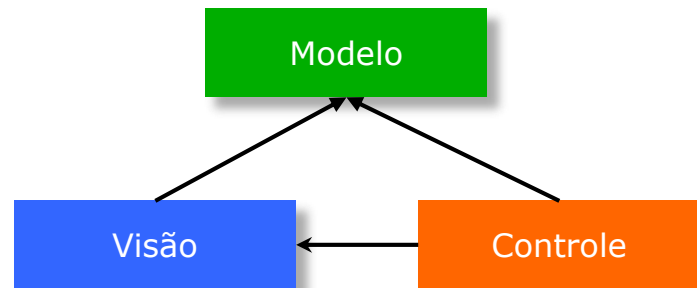
Controladores: recebem as requisições da Internet e executam os serviços



# Arquitetura de software

---

- ❑ Os dois lados se baseiam na **arquitetura MVC**, que divide a aplicação em três tipos de classes: modelo, visão e controle
  - **Modelo**: classes que representam os elementos do domínio do problema tratado pela aplicação
  - **Visão**: classes que apresentam os elementos para o usuário, recebendo informações e eventos
  - **Controle**: classes que recebem os eventos e informações e realizam mudanças nas classes de modelo

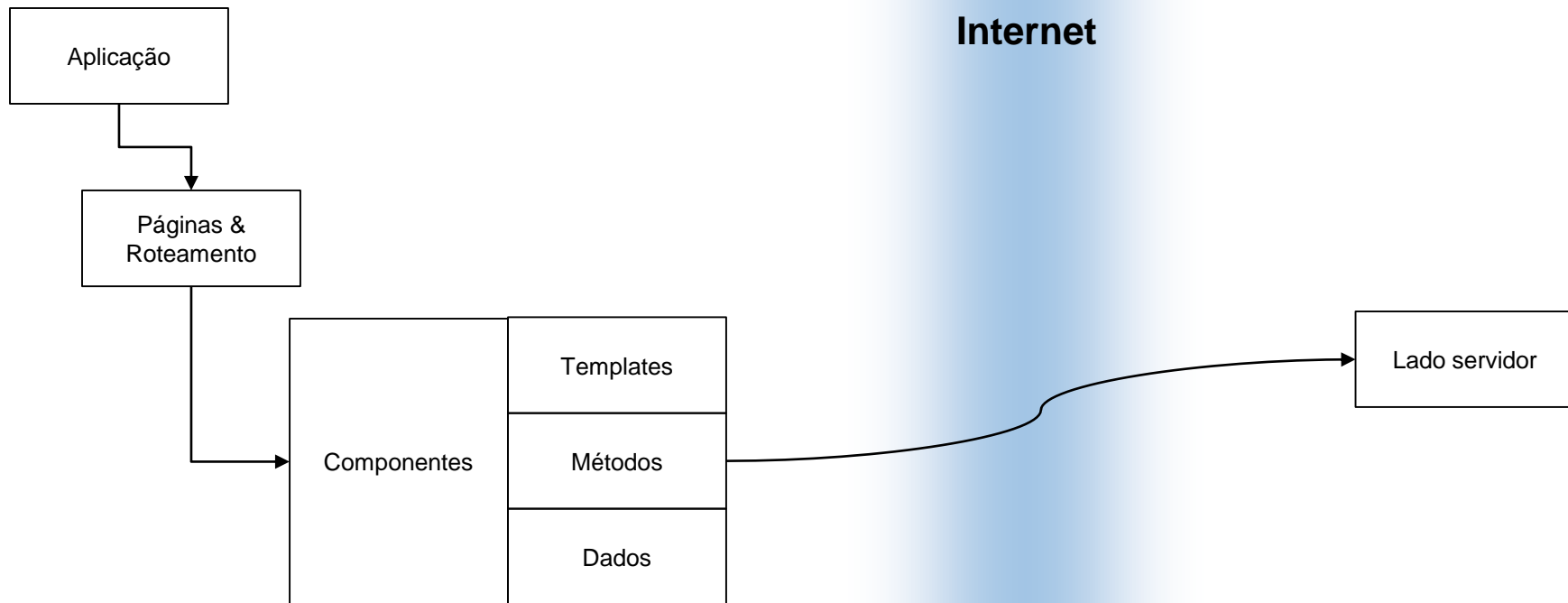


# Lado cliente

---

- ❑ O lado cliente utiliza um *framework* que envia requisições para o lado servidor, recebe o resultado e apresenta para o usuário
  - Uma aplicação é composta por diversas **páginas** (uma ou mais páginas), cada qual sendo formada por componentes
  - Os componentes seguem o modelo Model-View-Controller para separar os dados, a forma de apresentar e a forma de adquirir estes dados
  - Controladores são implementados em **métodos** dos componentes ou em componentes separados
  - A formatação dos dados é representada em **templates**, que são aplicados sobre marcações no código HTML das páginas

# Arquitetura do lado cliente





# Lado servidor

---

- ❑ O lado servidor não possui uma representação visual
  - O lado servidor apenas recebe as requisições, executa seu processamento e retorna os resultados para o lado cliente
  - Dizemos que o lado servidor oferece uma API (*Application Programming Interface*) para o lado cliente
  - O lado cliente recebe os dados brutos (sem formatação) e resolve como eles serão apresentados para o usuário
  - Este modelo foi possível devido à evolução dos navegadores, a adoção do HTML5 e melhorias na linguagem JavaScript
  - O padrão de API REST se baseia nos métodos do protocolo HTTP para oferecer os serviços para o lado cliente

# Arquitetura do lado servidor

