

# DESENVOLVIMENTO WEB

## Estrutura de Aplicações Web

# ROTEIRO

- Entendendo as aplicações Web
- Camadas de uma aplicação Web
- Um pouco sobre Cookies
- Arquitetura MVC
- Aplicação Web e Computação em Nuvem
- Quebra-Cabeça de uma aplicação Web

# ENTENDENDO AS APLICAÇÕES WEB

- **Recapitulando:** Uma aplicação Web é um software implantado/instalado em um servidor Web, sendo projetada para receber requisições, processá-las, armazenar (se for o caso) e responder a essas requisições
- As respostas podem vir de apenas um servidor ou de vários servidores Web, especialmente por conta da enorme demanda de requisições para determinadas aplicações acessadas pelos usuários.

# ENTENDENDO AS APLICAÇÕES WEB

- Variações de aplicações Web:
  - **Baseada no navegador** – Aquelas que possuem códigos que parcialmente serão processados no servidor Web, mas são executadas dentro do navegador. Elas têm capacidade de acessar dados do servidor remoto ou também localmente
  - **Baseada no cliente** – Semelhante à anterior, mas ao invés de ser executada dentro do navegador, são executadas como aplicações próprias, e exigem que sejam instaladas do lado do cliente/usuário
  - **Aplicações Móveis** - Mais comumente utilizada, em razão de os usuários terem aparelhos celulares. O app é executado em um SO móvel (Android, iOS)

# CAMADAS DE UMA APLICAÇÃO WEB

- Embora de forma mais simples, uma aplicação Web consiste de apenas um servidor e seus clientes/usuários. Podemos vislumbrar uma aplicação Web baseada em 3 camadas:
  - **Apresentação** – Apresenta informações ao usuário
  - **Lógica** – Transforma, consulta, edita e manipula informações de e para os formulários nos quais deve ser apresentada ou armazenada
  - **Dados** – Retêm dados para a aplicação de forma geral

# CAMADAS DE UMA APLICAÇÃO WEB

- As camadas descritas anteriormente dependem das tecnologias utilizadas do lado do servidor e do cliente
- No entanto, o padrão de comunicação deve ser feito via protocolo HTTP
- Servidores Web podem ser diferentes
  - Apache
  - Nginx
  - IIS (Microsoft)
- Um importante aspecto das aplicações Web é a utilização de Cookies
  - Os **cookies** são uma forma de armazenar informações que o protocolo HTTP não consegue armazenar

# UM POUCO SOBRE COOKIES

- O protocolo HTTP não armazena informações das sessões de uma conexão para a próxima
- Cada conexão HTTP é tratada de forma independente como uma ligação separada
- Uma aplicação em execução que usa o HTTP precisa manter o controle das informações de alguma forma, e é por isso que os cookies são utilizados.
- Os cookies são importantes e necessários para o funcionamento de milhares de aplicações Web no mundo

# UM POUCO SOBRE COOKIES

- Embora importantes, os cookies podem ser usados por usuários mal-intencionados, que os utilizam para comprometer a segurança geral da aplicação Web
- Um cookie pode ser um token de autenticação, ou uma espécie de mecanismo para armazenamento de dados
- Por isso que invasores desejam muito o acesso aos cookies
  - Ataques como o XSS e Sniffing são comuns para roubar informações de sessões de cookies que residem na máquina do usuário



# UM POUCO SOBRE COOKIES

- Há um lado bom em relação ao uso de cookies
- Eles nos ajudam, por exemplo, em processos de compras em lojas online, em que vamos colocando tudo o que desejamos em um carrinho de compras e só finalizamos uma conexão com o servidor quando clicamos definitivamente no botão **comprar**

# ARQUITETURA MVC

- Model View Controller (MVC)
  - Um padrão de projeto de software ou padrão de arquitetura de software formulado na década de 1970
  - Separação de conceitos em 3 camadas interconectadas
    - Apresentação dos dados e interação com Controle usuários (front-end) são separados dos métodos que interagem com o back-end (banco de dados e outros)

# ARQUITETURA MVC

- Tornou-se popular para aplicações Web, muito embora, originalmente, tenha sido desenvolvida para construir aplicações com interface gráfica para os usuários
- Serve também para aplicações móveis e para aplicações desktop
- Linguagens que utilizam MVC
  - PHP, Java, C#, Ruby, JavaScript e outros frameworks

# ARQUITETURA MVC

- Os componentes do MVC
  - **Model** (camada de modelo ou lógica da aplicação) – Ponte entre a View e Controller. É a parte lógica da aplicação e gerencia o comportamento dos dados por meio de regras de negócios, lógica e funções
  - **View** (camada de apresentação ou visualização) – Saída de representação dos dados (tabela, diagrama, etc). É como os dados solicitados do modelo são exibidos. Aqui podemos ter várias visões de um mesmo dado (usar uma tabela, usar um gráfico, etc). Ela permite interações com o usuário
  - **Controller** (camada de controle ou controlador) – Faz a mediação entre o modelo e a View, ou seja entre a entrada e a saída dos dados

# ARQUITETURA MVC

- Interação entre os componentes
  - Controlador – envia comandos para modelo para atualizar o seu estado (editar um documento, por exemplo)
  - Modelo – armazena dados e notifica suas visões e controladores associados
  - Visão – gera uma representação dos dados presentes no modelo solicitado e faz a exibição dos dados (html ou xml)

# ARQUITETURA MVC

- Utilização em aplicações Web
  - Muitos frameworks de desenvolvimento de aplicações Web foram desenvolvidos com base neste modelo
- Atualmente, os frameworks Web MVC utilizando uma abordagem thin-client quase colocaram o model, view e controller tudo no servidor
- Exemplo: Cliente envia requisições ao controlador e então recebe uma página Web completa atualizada da View. Veja que o modelo está praticamente todo no servidor Web neste caso

# ARQUITETURA MVC

- Utilização em aplicações Web
- Exemplo: Em uma aplicação Web em que a visão é um documento HTML gerado pela aplicação, o controlador recebe uma entrada GET ou POST após um estímulo do usuário (um clique em um botão do form) e processa o documento. Isso invoca objetos do domínio para tratar a lógica de negócio, e em seguida invoca uma visão para apresentar a saída ao usuário.

# ARQUITETURA MVC

- **Vantagens**

- A aplicação torna-se escalável
- Facilita o reuso de códigos
- Facilita a manutenção da aplicação
- Como a arquitetura é modular, permite que desenvolvedores e designers desenvolvam em paralelo
- Partes da aplicação podem ser alteradas sem alterar outras

- **Desvantagens**

- À medida que cresce o projeto, cresce o número de arquivos e pastas
- Requer tempo para modelar o sistema



# APLICAÇÃO WEB E COMPUTAÇÃO EM NUVEM

- Uma nuvem computacional não armazena somente fotos, vídeos e áudios
- Muitas aplicações que você utiliza, como Redes Sociais, Bancos Online, Lojas de Departamentos Online estão também em um ambiente computacional na nuvem
- As aplicações Web se integram aos provedores de serviços como AWS, Azure e Google Cloud, o que permite maior flexibilidade e acesso em relação ao uso das aplicações fora do ambiente de Cloud
- A nuvem também contempla muitas partes de aplicações que consumimos por meio de smartphones e tablets.

# QUEBRA-CABEÇA UMA APLICAÇÃO WEB

- Há vários componentes em uma aplicação Web, em que cada um possui uma função específica
  - **Login/Autenticação** – Usado para ter acesso ao sistema/aplicação
    - Ela pode ser: anônima, básica, com certificados digitais
  - **Servidores Web** – Hardware + Software para hospedar a aplicação Web
  - **Permissões** – Com base na autenticação e autorização, é determinado o nível de acesso que o usuário tem em relação aos recursos no servidor
  - **Conteúdo da Aplicação** – Informação com a qual o cliente/usuário vai interagir

# QUEBRA-CABEÇA UMA APLICAÇÃO WEB

- Continua...
  - **Páginas de acesso aos dados** – As que são anexadas a uma biblioteca que fornece acesso aos dados/
  - **Armazenamento de dados** – Onde são armazenadas as informações da aplicação Web. O armazenamento é feito em bases de dados remotas e protegidas
  - **Lógica** – Interage com o usuário e fornece mecanismos para obter as informações a serem obtidas do banco de dados
  - **Logout** – Utilizada pelos usuários para se desconectar da aplicação
    - Isso pode ocorrer sem a interação direta do usuário, depois de um período de inatividade do usuário com a aplicação (a sessão é fechada automaticamente)

# REFERÊNCIAS

<https://www.devmedia.com.br/aplicacao-web-usando-arquitetura-mvc/24375>

<https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>

<https://pt.wikipedia.org/wiki/MVC>

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

<https://www.redhat.com/pt-br/topics/cloud-native-apps/what-are-cloud-applications>

# **DESENVOLVIMENTO WEB**

## **Estrutura de Aplicações Web**