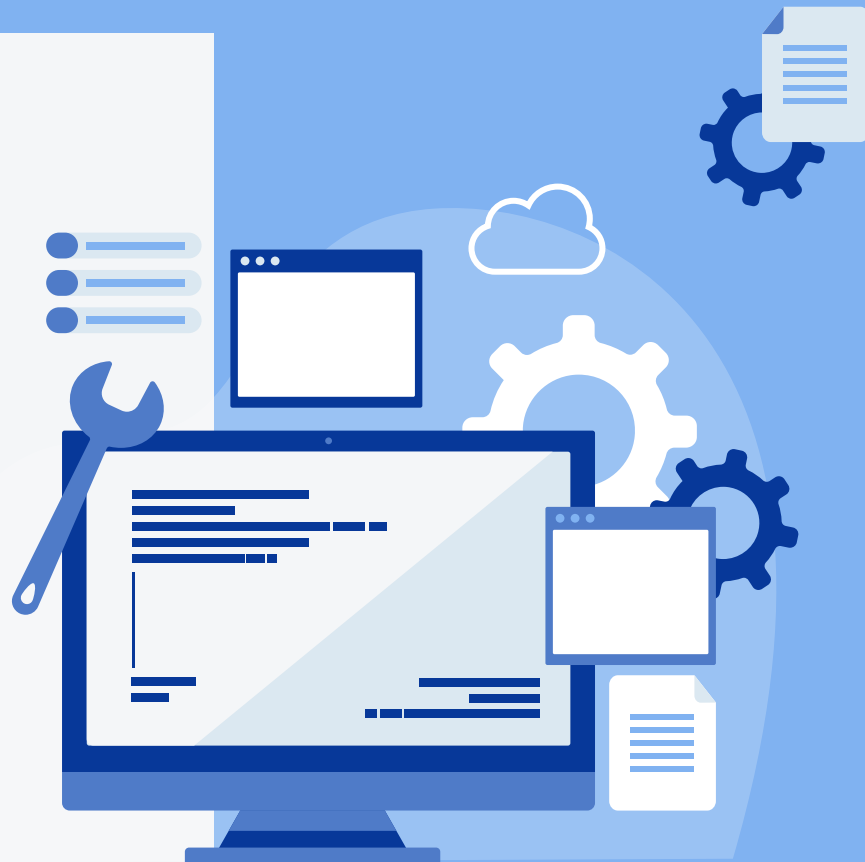


# Estilos Arquiteturais

Parte  
01

Arquitetura de Software  
Joel Santos



# O que é?



- ❑ Um estilo arquitetural na engenharia de software é um conjunto de princípios e padrões de design que define a estrutura e organização de um sistema.
- ❑ Ele descreve como os componentes de um sistema se organizam e interagem entre si, assim como os padrões de comunicação entre eles.

# Estilos Arquiteturais

Pode simplificar o problema de definição de arquiteturas de sistema.



**Um estilo define:**



A arquitetura de um sistema pode aderir a um ou mais estilos arquiteturais.

A maioria dos sistemas de grande porte adere a vários estilos.

Os tipos de elementos que podem aparecer em uma arquitetura.

As regras que regem a sua interconexão.

**Estilos arquiteturais = “modelos arquiteturais”**

# Estilos Arquiteturais

Pode simplificar o  
problema de definição de  
arquiteturas de sistema.

Estilo



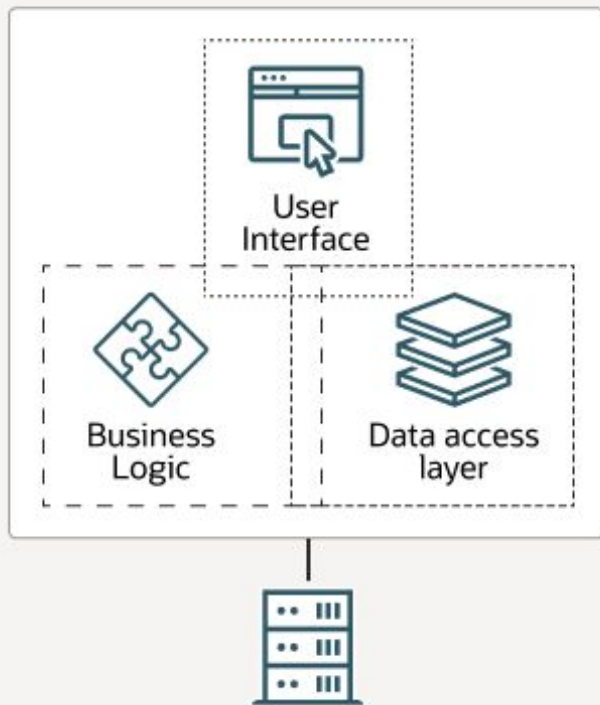
*Um estilo arquitetural define uma família de  
sistemas em termos de um padrão de  
organização estrutural [Shaw 96]*

# Exemplos de Estilos Arquiteturais



- Monolítico
- Arquitetura em Camadas
- Arquitetura de Pipe e Filtros
- Microsserviços
- Arquitetura Cliente-Servidor
- Arquitetura de Repositório
- Arquitetura Orientada a Serviços

# Estilo Monolítico



- ❑ A arquitetura monolítica é um padrão de desenvolvimento de software no qual um aplicativo é criado com uma única base de código, um único sistema de compilação, um único binário executável e vários módulos para negócios ou recursos técnicos.
- ❑ Toda a base de código está contida em um só lugar, ou seja, todas as funcionalidades estão definidas no mesmo bloco.

# **Estilo Monolítico**



**Pontos positivos ???**

**E os negativos ???**

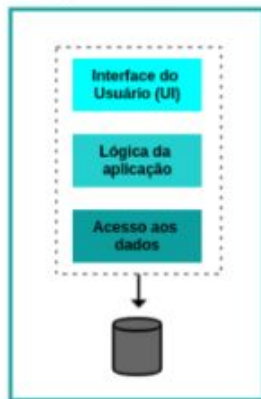
# Estilo Monolítico



## Vantagens

- Desenvolvimento simples
- Implantação
- Testes simples

Arquitetura Monolítica



## Desvantagens

- Difícil de escalar
- Baixa manutenibilidade
- Impacto de falhas

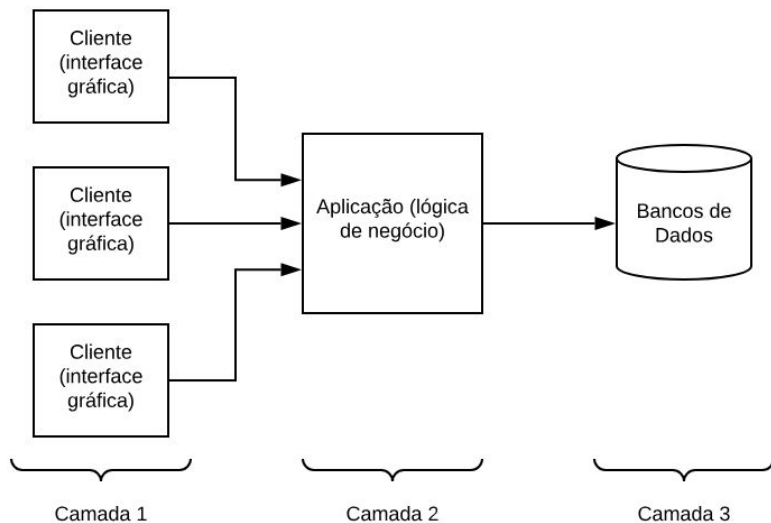


# Estilo em Camadas



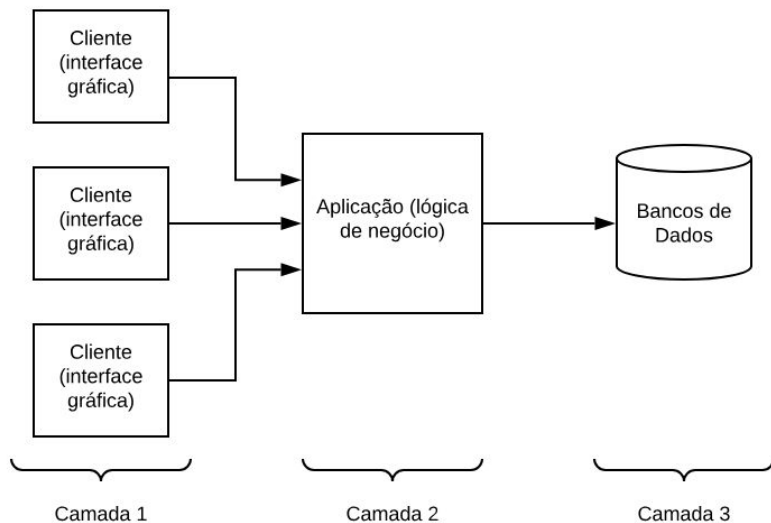
- ❑ Um sistema em camadas é organizado hierarquicamente, cada camada oferecendo serviço a camada acima dela e servindo como cliente da camada inferior.
- ❑ Um exemplo de uma arquitetura em camadas é o modelo OSI (Reference Model for Open Systems Interconnection).

# Estilo em Camadas



- ❑ O sistema é dividido em diferentes níveis ou camadas, onde cada camada tem uma responsabilidade bem definida, como interface do usuário, lógica de negócios, e persistência de dados.
- ❑ Cada camada comunica-se apenas com a camada diretamente adjacente.

# Estilo em Camadas



Uma estruturação em camadas muito utilizada em sistemas de informação, especialmente os baseados na web, consiste em utilizar três camadas:

- (i) uma responsável pela interação entre o sistema e o usuário,
- (ii) uma outra responsável por implementar as regras de negócio da aplicação e
- (iii) uma terceira que lida com o armazenamento dos dados.

Arquiteturas Three-Tier  
(ou “arquiteturas em três camadas”).

# **Estilo em Camadas**



**Pontos positivos ???**

**E os negativos ???**

# Estilo em Camadas



## Vantagens

- Permite projetos baseados em níveis crescentes de abstração
- Permite particionar problemas complexos em uma sequência de passos incrementais.
- Mudanças em uma camada afetam, no máximo, as duas adjacentes.
- Permite que diferentes implementações da mesma camada possam ser usadas desde que mantenham a mesma interface com as camadas adjacentes.

# Estilo em Camadas



## Desvantagens

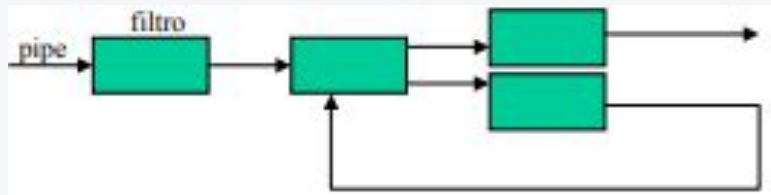
- Nem todos os sistemas são facilmente estruturados em forma de camadas.
- É difícil encontrar os níveis de abstração corretos  
  
(muitas vezes os serviços abrangem diversas camadas).

# Pipe e Filtros



o sistema é organizado como uma sequência de filtros (componentes que processam dados) conectados por pipes (canais de comunicação), onde cada filtro recebe dados, os processa e envia o resultado para o próximo filtro.

Tipicamente divide a tarefa de um sistema em vários passos de processamento sequencial.



# Pipe e Filtros

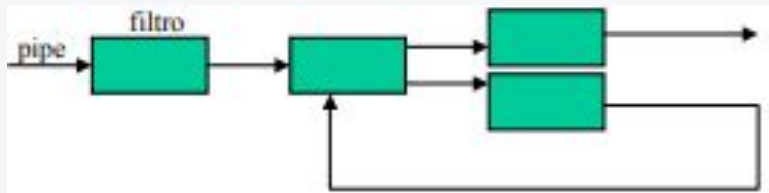


## Componentes:

- São chamados Filtros.
- Tem um conjunto de entradas e um conjunto de saídas.
- Realiza o processamento de um stream de dados.

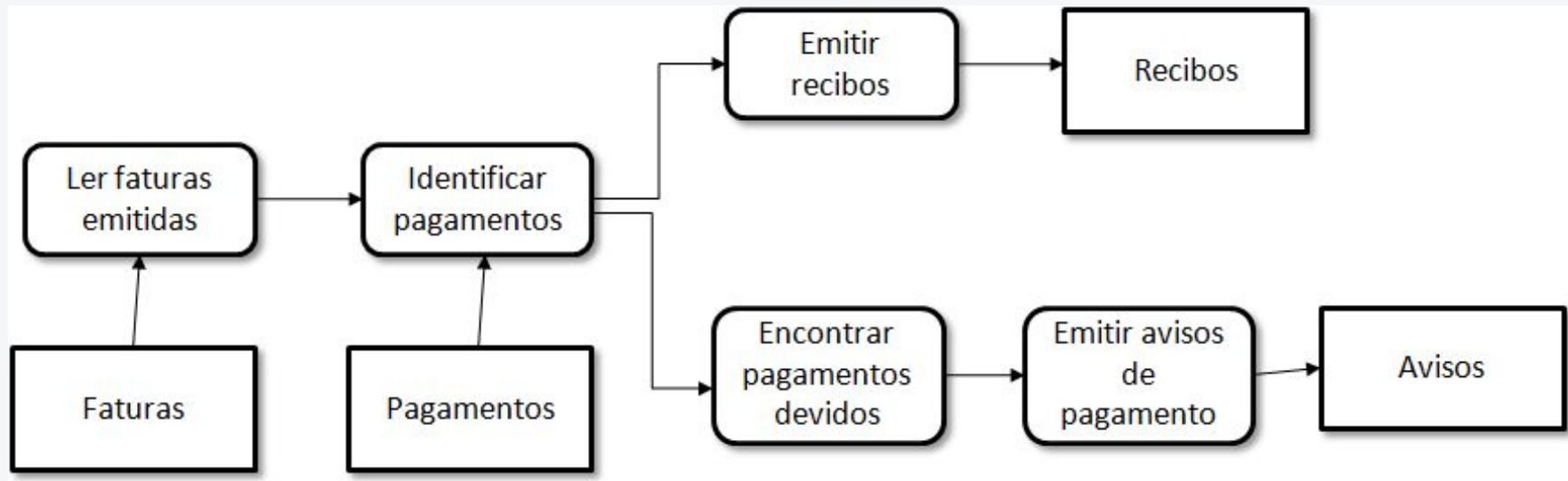
## Conectores:

- São chamados Pipes
- Servem como condutores, transmitindo as saídas de um filtro para as entradas de outro.





# Pipe e Filtros



# Pipe e Filtros



**Pontos positivos ???**

**E os negativos ???**

# Pipe e Filtros



## Vantagens

- Útil para aplicações de processamento de informação que interagem pouco com usuários
- Rápida prototipação
- Apóia reuso de transformações (filtros)
- É fácil adicionar, recombinar, ou trocar, novas transformações (flexibilidade)
- É relativamente simples implementar como sistema concorrente (vários filtros em paralelo) ou seqüencial
- Eficiência em processamento

# Pipe e Filtros



## Desvantagens

- Requer um formato comum para a transferência de dados ao longo do pipeline
- Não é apropriado para aplicações interativas
- Não existe compartilhamento de dados
- Ausência de gerenciamento de erros.

# **Para refletir...**

**Em quais situações podemos  
usar cada estilo apresentado  
na aula ?**

**Dúvidas**

**???**

# Exercício 01 - Arquitetura Monolítica

Crie uma aplicação monolítica utilizando Html e códigos JavaScript, onde todas as funcionalidades estão concentradas em um único arquivo.

- Desenvolva uma calculadora de conversão de temperatura que converte valores de Celsius para Fahrenheit.
- Toda a lógica, interface e armazenamento de resultados deve estar implementada em um único arquivo.

# Exercício 02 – Arquitetura em

Desenvolva uma aplicação para gerenciar um cadastro de usuários. Divida a aplicação em três camadas: apresentação (frontend), lógica de negócios e persistência de dados.

- A interface permitirá ao usuário inserir nome e idade, que serão validados na camada de negócios.
- Caso o usuário tenha mais de 18 anos, os dados serão salvos em uma camada de persistência simulada.

## Requisitos:

- A camada de apresentação será um formulário HTML onde o usuário insere nome e idade.
- A lógica de negócios validará se a idade é maior de 18 anos.
- A persistência de dados armazenará os dados dos usuários em um array ou objeto.



# Exercício 03 – Arquitetura Pipe e

Desenvolva uma aplicação que simule o estilo arquitetural de Pipe e Filtros, onde os dados passam por uma série de transformações (filtros) antes de serem exibidos.

- A aplicação será um processador de texto simples. O usuário poderá inserir um texto, que passará por uma série de filtros (transformações) aplicados em sequência.
- Os filtros aplicados serão: transformar o texto em letras maiúsculas, remover espaços extras e contar o número de palavras no texto.

## Requisitos:

Crie três filtros: (1) converter o texto para letras maiúsculas, (2) remover espaços extras e (3) contar o número de palavras.

A sequência de filtros deve ser aplicada em ordem e o resultado final exibido na tela.