CÂMPUS

Bagé

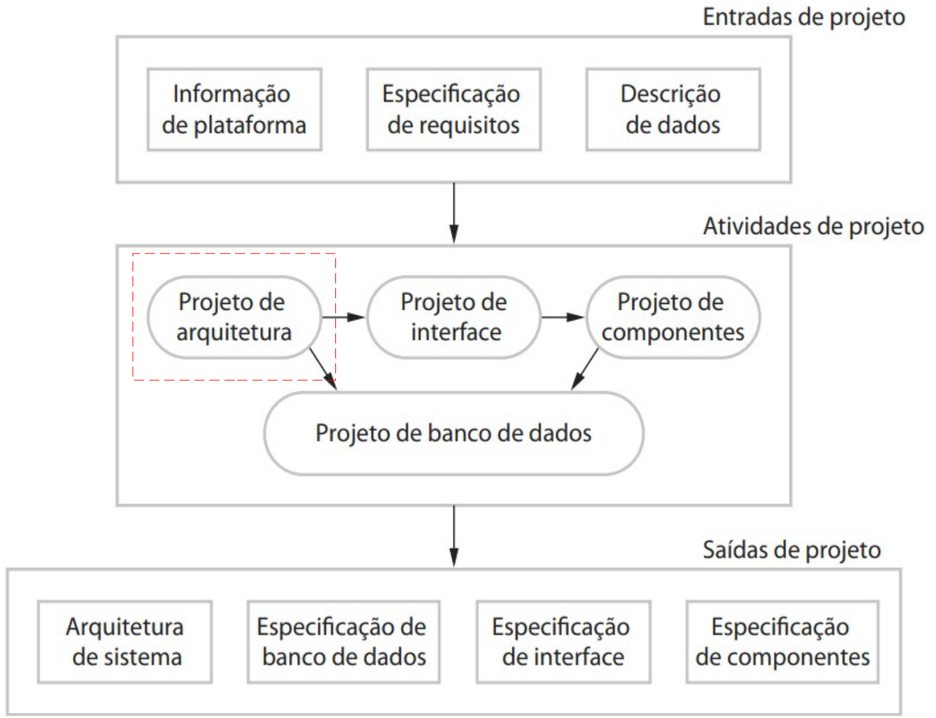
ARQUITETURA DE SOFTWARE

RODRIGO R SILVA

Definição de Projeto de Arquitetura

- Ao se desenvolver um software, temos algumas etapas básicas: análise de requisitos, modelagem de software, implementação, e testes

- A arquitetura de software é um dos **processos** dentro etapa de Modelagem/Projeto de Software:



Definição de Projeto de Arquitetura

- No projeto de arquitetura você **projeta uma organização geral do sistema** (em blocos que se comunicam) de forma a atender aos requisitos funcionais e não funcionais do sistema

- **Exemplo:**
  Arquitetura de software de um sistema operacional

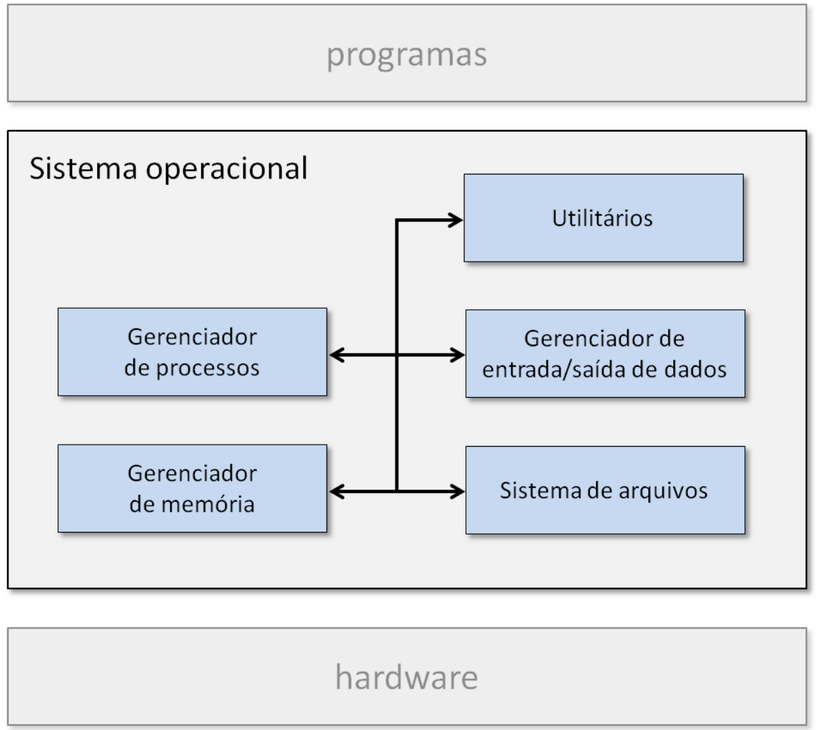


Diagrama de projeto conceitual da arquitetura

Definição de Projeto de Arquitetura

- O projeto de arquitetura depende:
  - do **tipo de sistema** a ser desenvolvido (ex.: um editor de texto tem arquitetura diferente de uma rede social)
    - geralmente, sistemas de um mesmo domínio (ex.: os antivírus) seguem uma mesma arquitetura – e este padrão pode ser reusado para o desenvolvimento de novos sistemas de antivírus
  - dos **requisitos do sistema**
    - veremos que um mesmo sistema pode ser construído com base em diferentes arquiteturas;
    - qual delas usar? escolhe-se a arquitetura que atende aos requisitos funcionais e não funcionais
  - da **formação** e **experiência** do arquiteto de sistemas

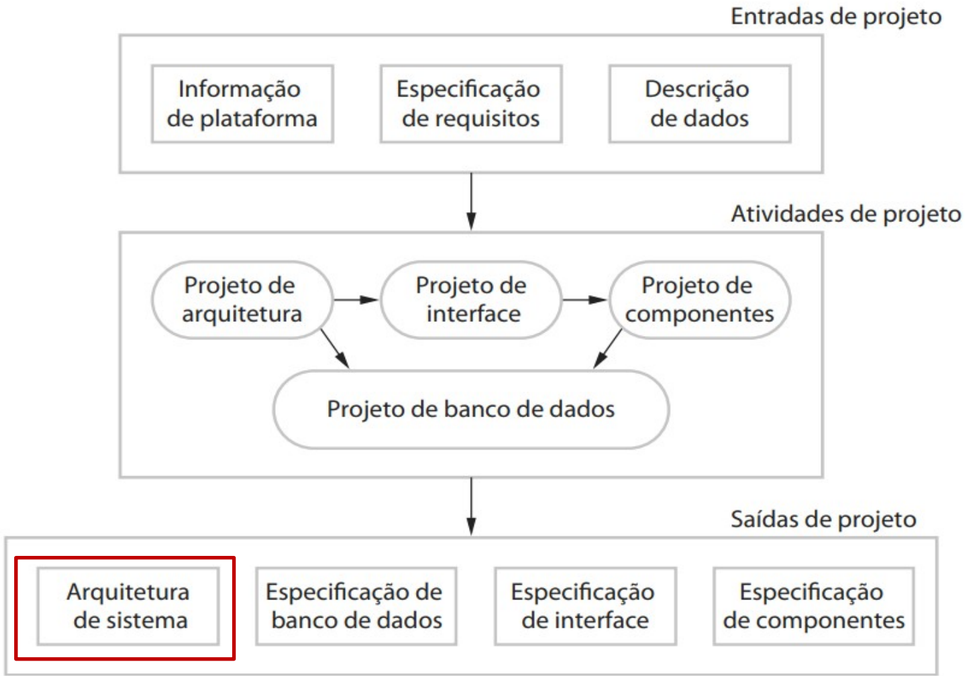
Objetivo

- O projeto de arquitetura foca:

- na compreensão de como um sistema deve ser organizado, e
- com a estrutura geral desse sistema

Objetivo

- É um **modelo conceitual (um diagrama de blocos e setas)** que mostra os componentes **estruturais** do sistema (que precisam ser desenvolvidos ou adquiridos) e o relacionamento de **comunicação** entre eles



Como Modelar

- Geralmente são usados **diagramas de blocos simples**, com
  - **blocos** contendo o nome de cada componente de software, e
  - **setas** indicado a passagem de dados ou sinais de controle entre os componentes do sistema
  - **blocos compostos** = um componente tem sub componentes

- **Vantagem**:
  - stackholders fora da área de informática (ex.: cliente, futuros usuários do software, analistas de negócio do cliente) conseguem entender facilmente a estrutura de alto nível do sistema

Como Modelar

- Há também **diagramas específicos** para modelagem de arquitetura de sistemas, por exemplo:
  - definem diferentes tipos de setas, para indicar diferentes tipos de comunicação entre blocos
    (ex.: setas para processamento paralelo e processamento condicional)
  - definem diferentes tipos de componentes
    (ex.: um bloco com um ícone para componente externo ao sistema, um outro para componente composto, etc.)
- Estes, geralmente, são usados quando o **cliente** ou **órgão regulador externo** (ex.: Banco Central) exige que seja usada uma ferramenta de modelagem específica para **documentação formal** da arquitetura do sistema

Padrões de Arquitetura

- Um **padrão de arquitetura** é um modelo de arquitetura que já foi testado em vários sistemas, e, assim, pode ser reusado no projeto de novos sistemas

- Existem diferentes padrões de arquitetura: cliente-servidor, MVC, sistema em camadas, filtros e dutos, repositório, etc.

- Cada padrão tem documentado:
  - seu nome
  - suas vantagens
  - suas desvantagens ou pontos fracos
  - quando deve ser usado
  - exemplos de uso

Veremos isso em detalhe nas próximas aulas

Padrões de Arquitetura

- **Exemplo de padrão de arquitetura:** Cliente/Servidor (VALENTE, 2022)

Cliente/Servidor é uma arquitetura muito usada na implementação de serviços básicos de rede.

Clientes e servidores são os dois únicos módulos desse tipo de arquite-tura e eles se comunicam por meio de uma rede de computadores.

Os clientes solicitam serviços ao módulo servidor e aguardam o processamento.

Padrões de Arquitetura

- **Exemplo de padrão de arquitetura:** Cliente/Servidor (VALENTE, 2022)
  - Arquiteturas cliente/servidor são usadas para implementar serviços como os seguintes:
    - **serviço de impressão**, que possibilita que clientes imprimam em uma impressora remota, que não está fisicamente conectada à máquina deles
    - **serviço de arquivos**, que possibilita que clientes acessem o sistema de arquivos (isto é, o disco) de uma máquina servidora
  - Arquiteturas cliente/servidor são usadas para implementar serviços como os seguintes:
    - **serviço de bancos de dados**, que permite que clientes acessem um banco de dados instalado em uma outra máquina
    - **serviço Web**, que permite que clientes (no caso, navegadores) acessem recursos (no caso, páginas HTML) armazenadas e providas por um servidor Web

Padrões de Arquitetura

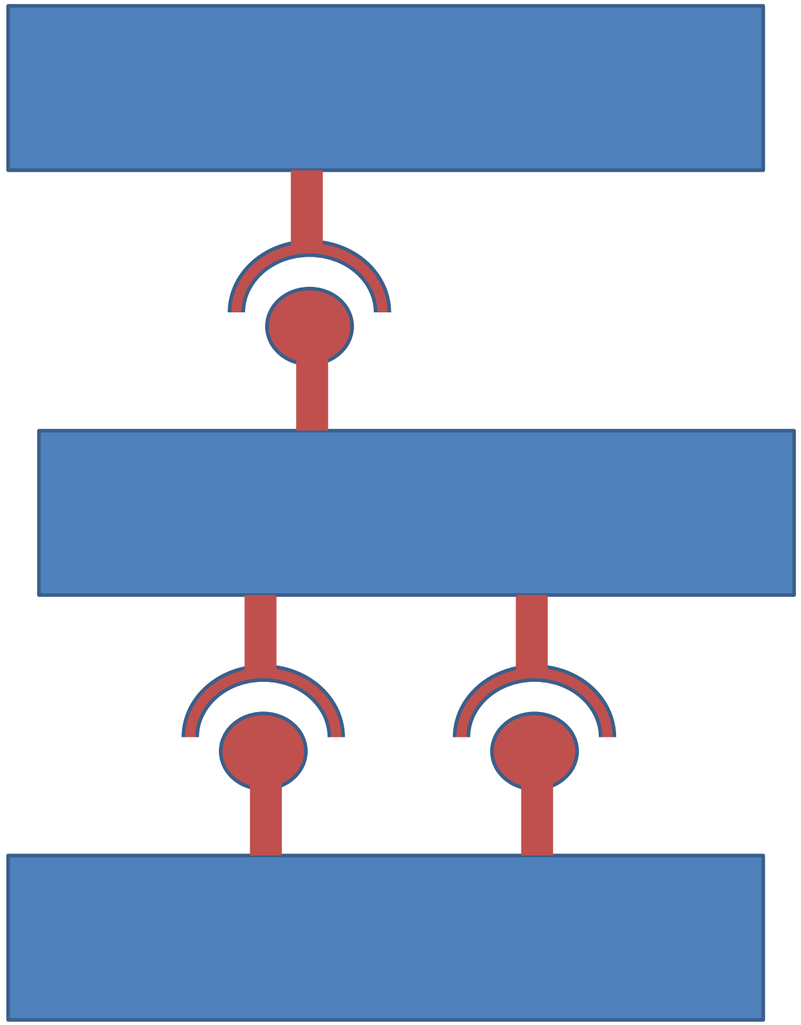
Nome	Cliente-servidor
Descrição	Em uma arquitetura cliente-servidor, a funcionalidade do sistema está organizada em serviços — cada serviço é prestado por um servidor. Os clientes são os usuários desses serviços e acessam os servidores para fazer uso deles.
Exemplo	A Figura 6.7 é um exemplo de uma biblioteca de filmes e vídeos/DVDs, organizados como um sistema cliente-servidor.
Quando é usado	É usado quando os dados em um banco de dados compartilhado precisam ser acessados a partir de uma série de locais. Como os servidores podem ser replicados, também pode ser usado quando a carga em um sistema é variável.
Vantagens	A principal vantagem desse modelo é que os servidores podem ser distribuídos através de uma rede. A funcionalidade geral (por exemplo, um serviço de impressão) pode estar disponível para todos os clientes e não precisa ser implementada por todos os serviços.
Desvantagens	Cada serviço é um ponto único de falha suscetível a ataques de negação de serviço ou de falha do servidor. O desempenho, bem como o sistema, pode ser imprevisível, pois depende da rede. Pode haver problemas de gerenciamento se os servidores forem propriedade de diferentes organizações.

Padrões de Arquitetura

- **Exemplo de padrão de arquitetura:** Camadas (Sommerville, 2018)

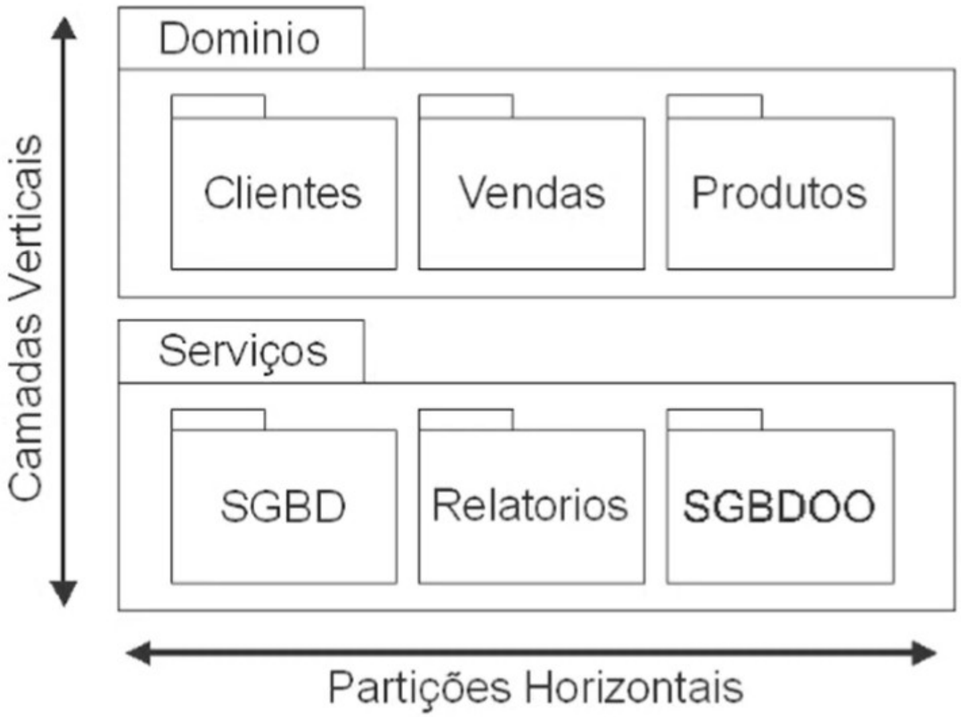
- As funcionalidades e serviços do sistema são organizados em camadas
- Cada camada depende apenas dos recursos e serviços oferecidos pela camada abaixo dela
- Uma camada pode ser substituída, refatorada, ou comprada e instalada sem causas efeitos colaterais nas outras camadas

Padrões de Arquitetura



- Cada camada depende apenas dos recursos e serviços oferecidos pela camada abaixo dela
- Cada camada é uma caixa-preta: uma não sabe detalhes de implementação da outra
- Só se conhece a interface (API) da camada

Padrões de Arquitetura



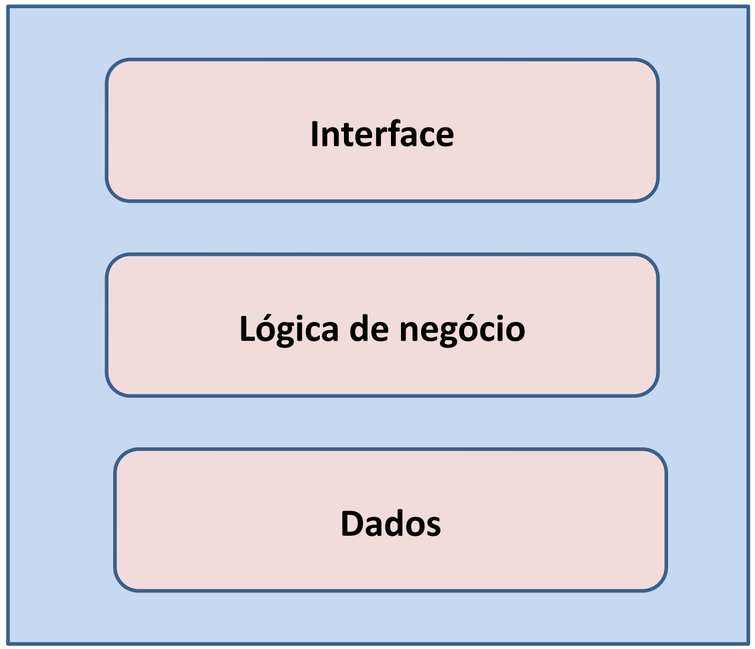
- Cada camada pode ter partições lógicas (ex.: diagrama de pacotes da UML)
- Veremos alguns destes diagramas UML mais adiante nesta disciplina

Padrões de Arquitetura

Nome	Arquitetura em camadas
Descrição	Organiza o sistema em camadas com a funcionalidade relacionada associada a cada camada. Uma camada fornece serviços à camada acima dela; assim, os níveis mais baixos de camadas representam os principais serviços suscetíveis de serem usados em todo o sistema. Veja a Figura 6.4.
Exemplo	Um modelo em camadas de um sistema para compartilhar documentos com direitos autorais, em bibliotecas diferentes, como mostrado na Figura 6.5.
Quando é usado	É usado na construção de novos recursos em cima de sistemas existentes; quando o desenvolvimento está espalhado por várias equipes, com a responsabilidade de cada equipe em uma camada de funcionalidade; quando há um requisito de proteção multinível.
Vantagens	Desde que a interface seja mantida, permite a substituição de camadas inteiras. Recursos redundantes (por exemplo, autenticação) podem ser fornecidos em cada camada para aumentar a confiança do sistema.
Desvantagens	Na prática, costuma ser difícil proporcionar uma clara separação entre as camadas, e uma camada de alto nível pode ter de interagir diretamente com camadas de baixo nível, em vez de através da camada imediatamente abaixo dela. O desempenho pode ser um problema por causa dos múltiplos níveis de interpretação de uma solicitação de serviço, uma vez que são processados em cada camada.

Padrões de Arquitetura

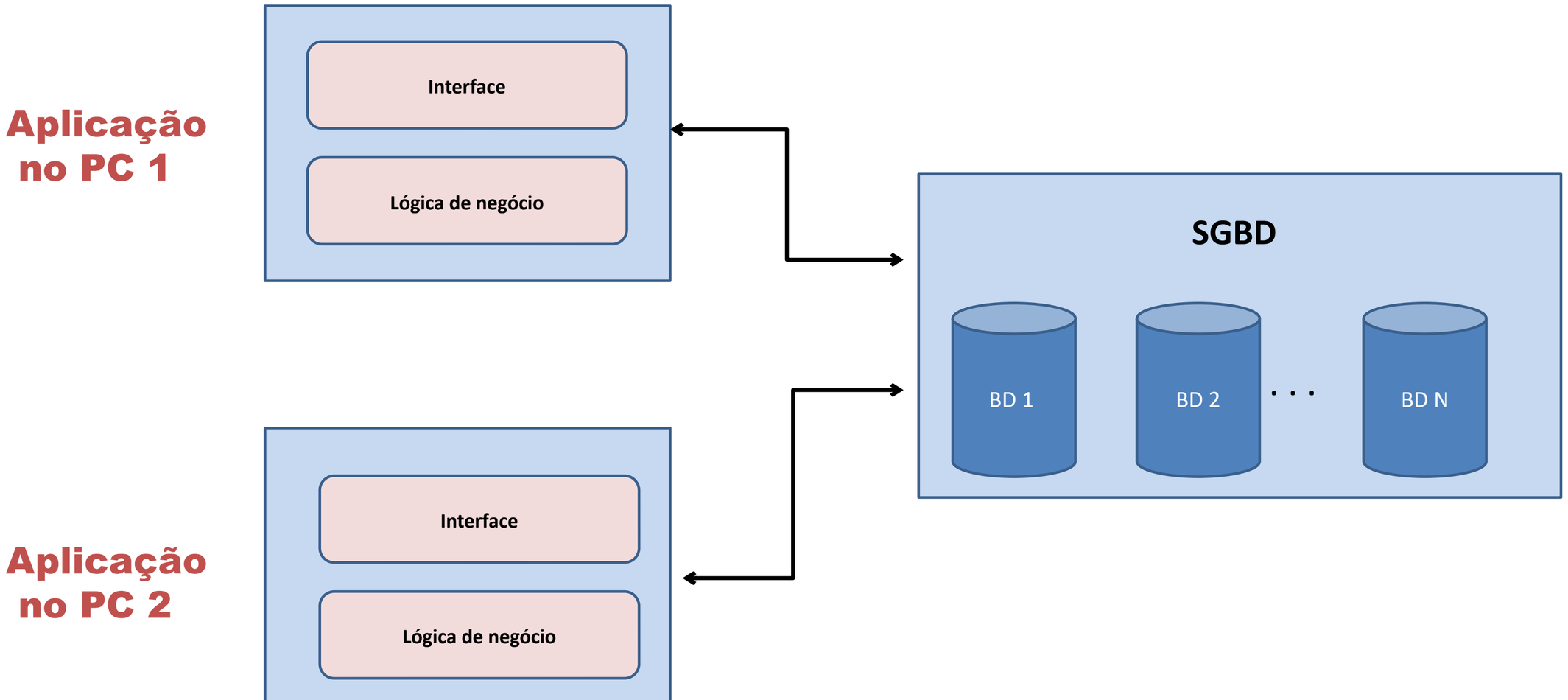
- **Arquitetura Monolítica**
  - uma aplicação que segue esta arquitetura funciona sem integração com outras aplicações, e executa no mesmo computador



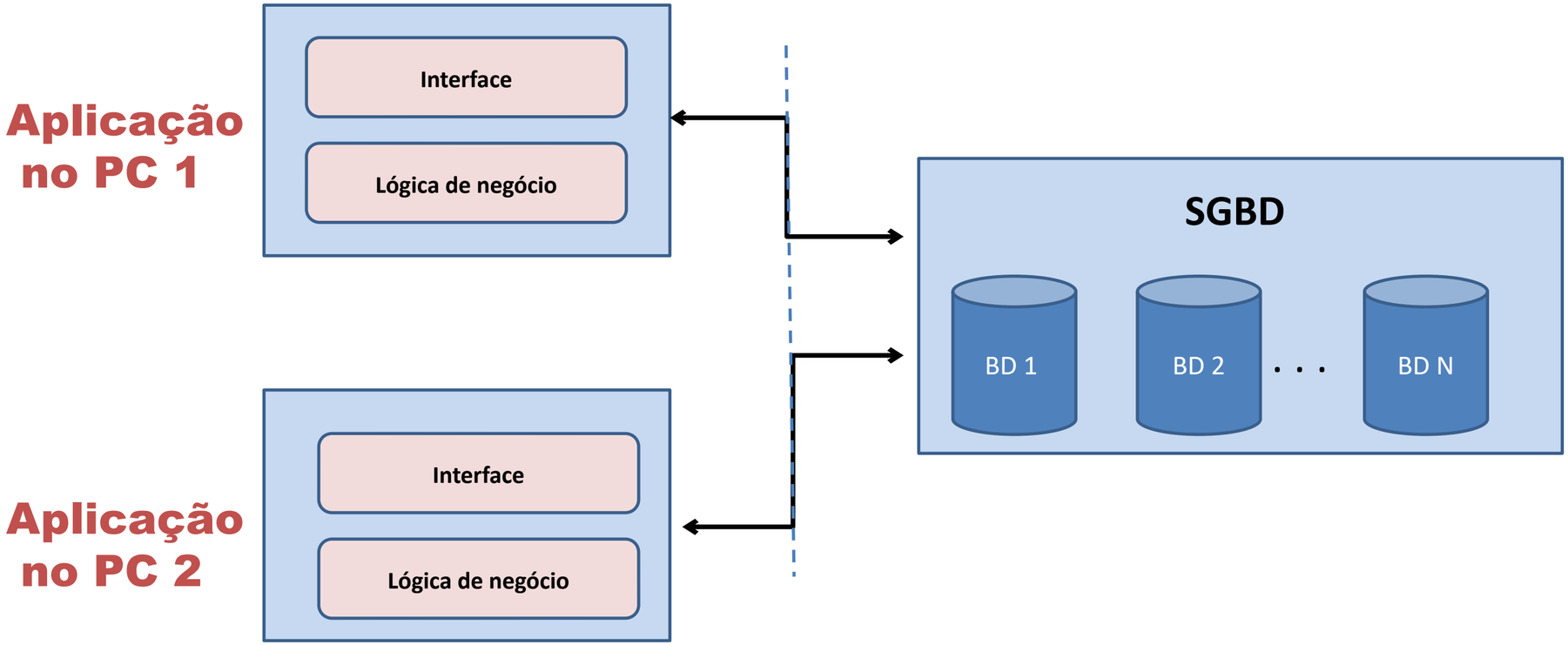
Geralmente tem código-fonte único, que contém interface de usuário/API, lógica de negócios e armazenamento de dados em arquivos locais

Padrões de Arquitetura

- **Arquitetura em 2 Camadas**
  - em um sistema que segue esta arquitetura, a aplicação se conecta diretamente a um SGBD. (usa, também, a arquitetura cliente/servidor)



Padrões de Arquitetura

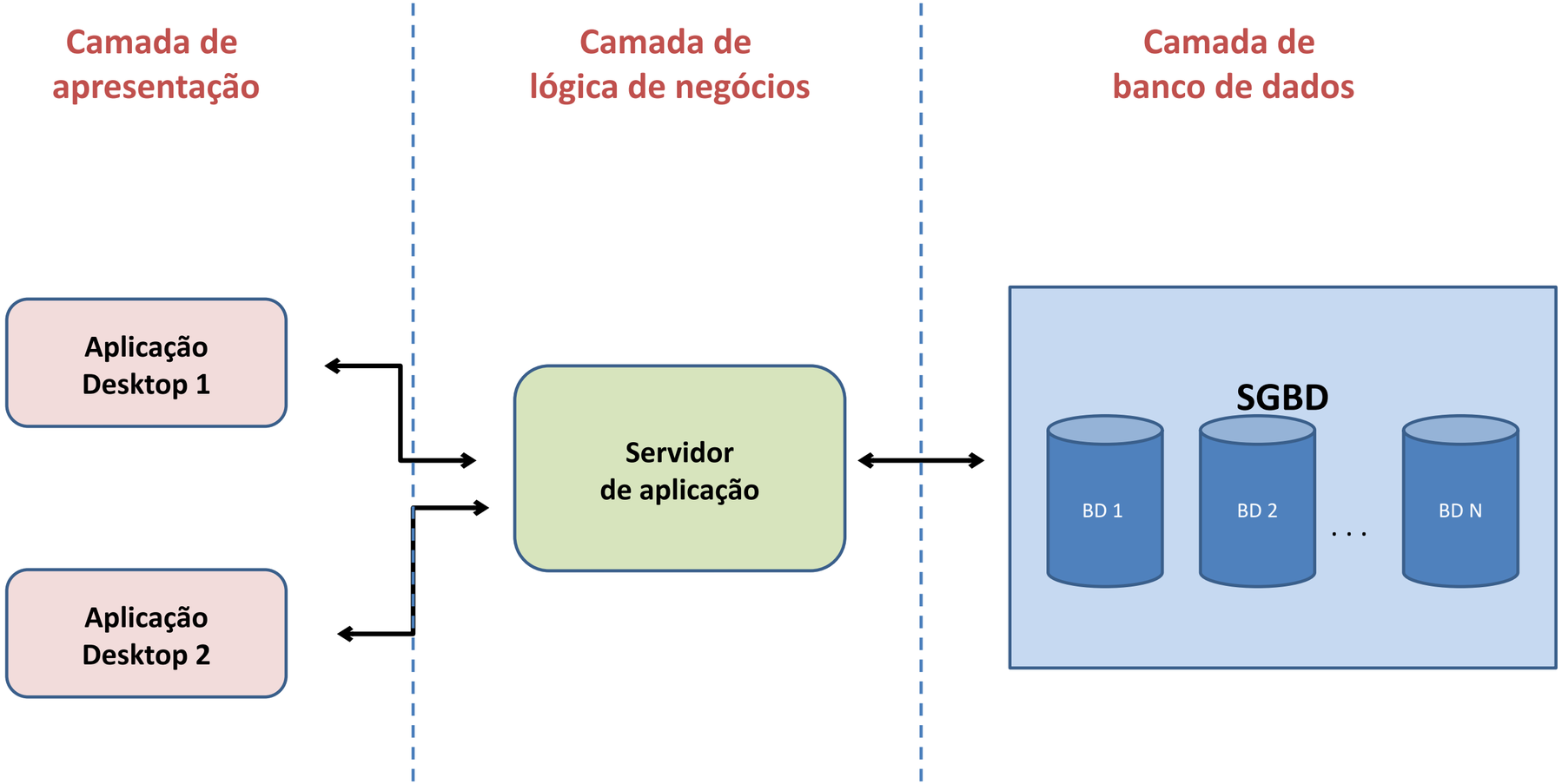


A camada de aplicação fica responsável pela lógica de negócios e interface de usuário

O SGBD é a camada que fica responsável só pelo armazenamento de dados

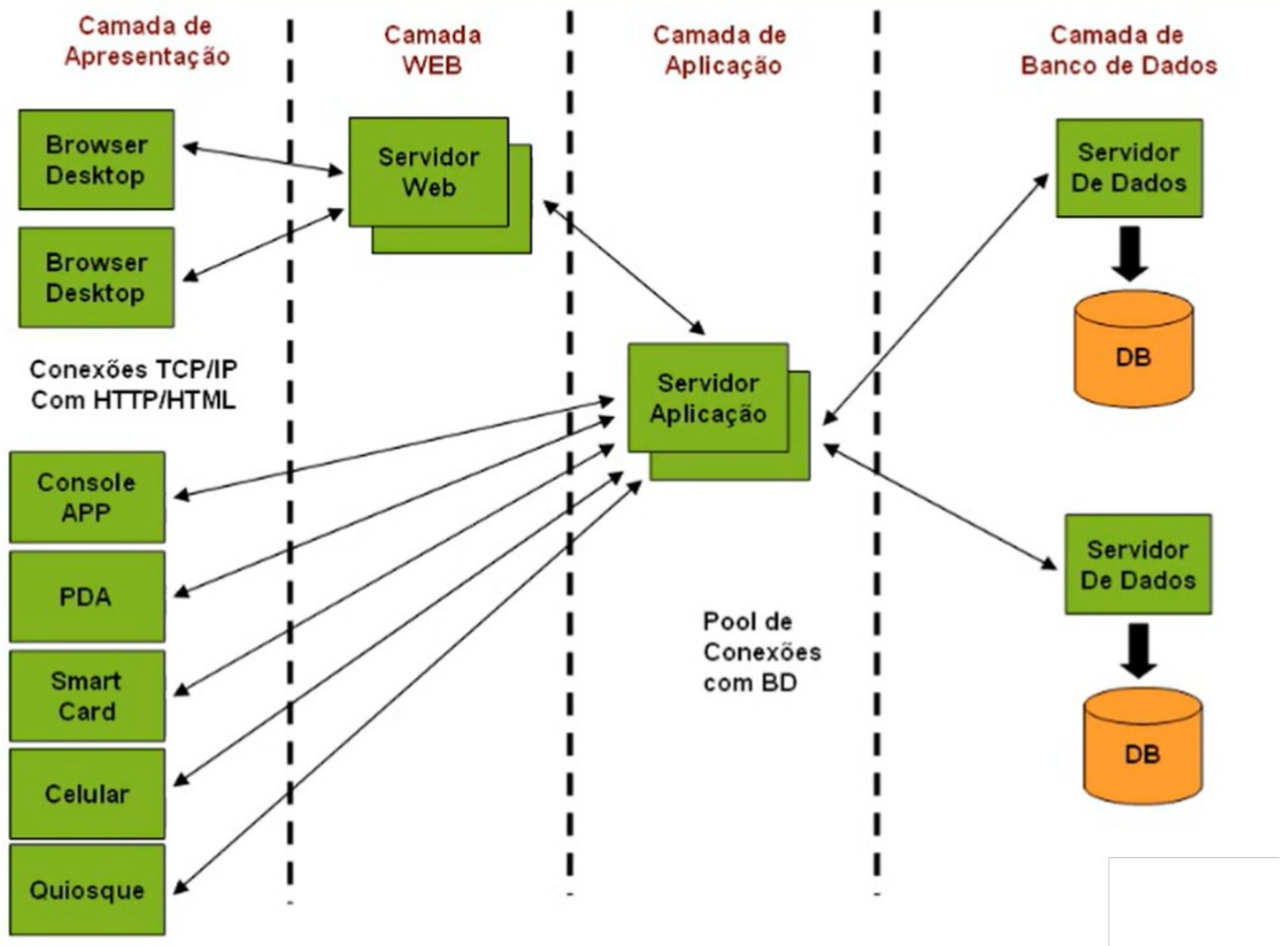
Padrões de Arquitetura

- **Arquitetura em 3 Camadas**
  - em um sistema que segue esta arquitetura, há 3 subsistemas conectados: um fica responsável pela interface de usuário, outro pela lógica de negócios do sistema e outro pelo armazenamento de dados (usa, também, a arquitetura cliente/servidor)

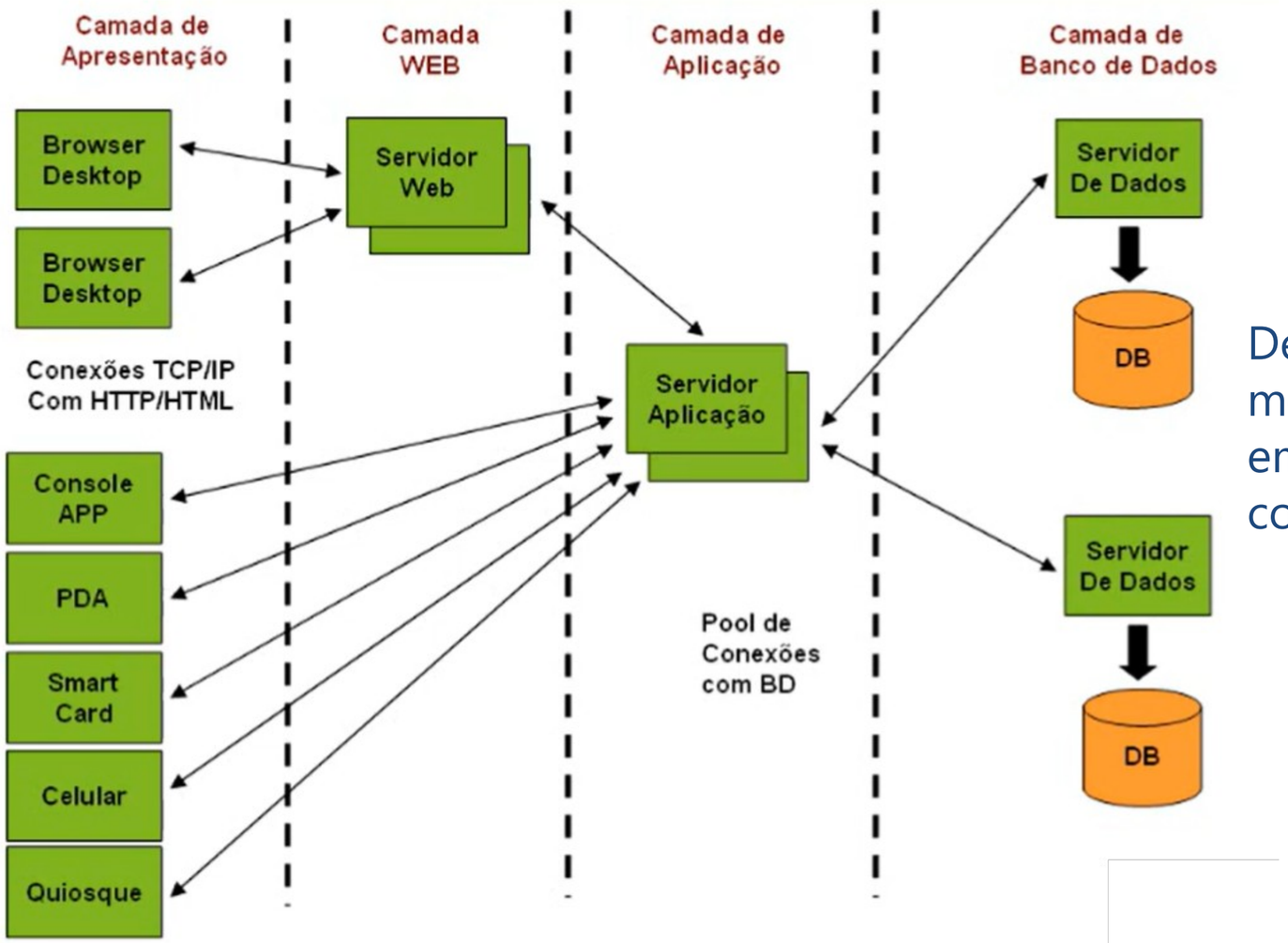


Padrões de Arquitetura

- **Arquitetura em Várias Camadas**
  - cada camada tem uma responsabilidade e fornece serviços para outras camadas mais altas



Padrões de Arquitetura



Dependendo do sistema, pode não seguir o modelo clássico de Camadas de “camadas empilhadas”, no qual uma camada só se comunica com a de cima

Conclusão

- Definição de projeto de arquitetura.
- Exemplos de projetos de arquitetura de software.

Bibliografia básica

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br.

MORAIS, Izabelly Soares de (org.). **Engenharia de Software**. São Paulo: Pearson, 2017. E-book. Disponível em: https://plataforma.bvirtual.com.br.

GALLOTTI, Giocondo Marino Antonio (org.). **Arquitetura de Software**. São Paulo: Pearson, 2016. E-book. Disponível em: https://plataforma.bvirtual.com.br.

Bibliografia complementar

GIRIDHAR, Chetan; KINOSHITA, Lúcia Ayako. **Aprendendo padrões de projeto em Python**: Tire proveito da eficácia dos padrões de projeto (design patterns) em Python para resolver problemas do mundo real em arquitetura e design de software. São Paulo, SP: Novatec, 165 p. ISBN 9788575225233.

FREEMAN, Eric; FREEMAN, Elisabeth. **Use a cabeça:** padrões e projetos. 2.ed. Rio de Janeiro, RJ: Alta Books, 2009. 478 p. ISBN 9788576081746.