Capítulo 6

Projeto de arquitetura

Os tópicos abordados

engenharia de SOFTWARE

- Decisões de projeto de arquitetura
- Visões de arquitetura
- Padrões de arquitetura
- Arquiteturas de aplicações

Arquiteturas de software

engenharia de SOFTWARE

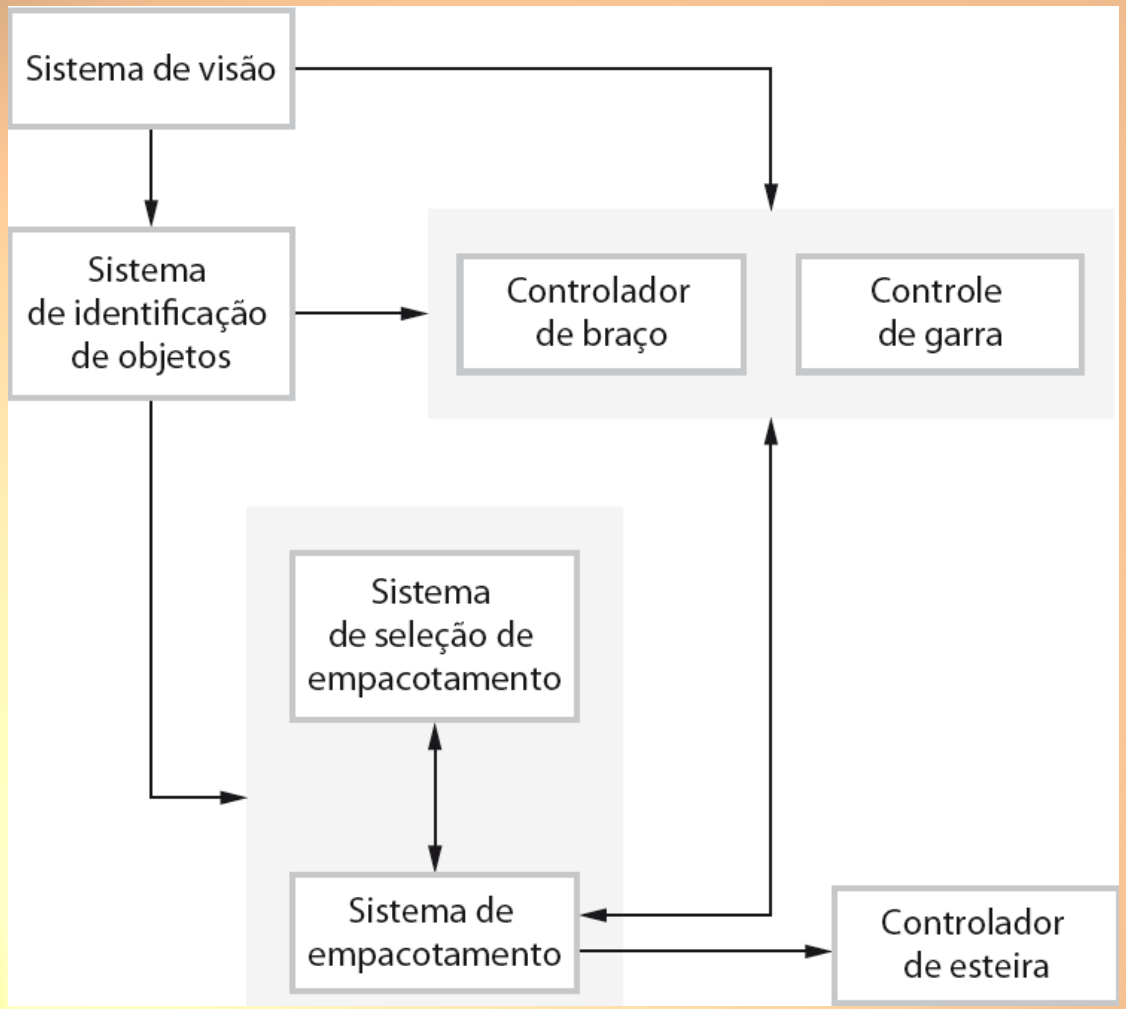
- O processo de projeto para identificar os subsistemas que compõem um sistema e o framework para controle e comunicação do subsistema é o projeto de arquitetura.

- A saída desse processo de projeto é uma descrição da arquitetura do software.

Projeto de Arquitetura

- Uma fase inicial do processo de concepção do sistema.
- Representa a ligação entre a especificação e os processos de projeto.
- Muitas vezes realizadas em paralelo com algumas atividades de especificação.
- Trata-se de identificar os principais componentes do sistema e suas comunicações.

A arquitetura de um sistema de controle robotizado de empacotamento



Abstração sobre a arquitetura

- Arquitetura em pequena escala está preocupada com a arquitetura dos programas individuais.
- Nesse nível, estamos preocupados com a maneira como um programa individual é decomposto em componentes.
- Arquitetura em grande escala preocupa-se com a arquitetura de sistemas corporativos complexos que incluem outros sistemas, programas e componentes do programa.
- Esses sistemas empresariais estão distribuídos em diferentes computadores, que podem ser possuídos e geridos por diferentes empresas.

Vantagens da arquitetura explícita

engenharia de SOFTWARE

- Comunicação de *stakeholders*
  - ✓ A arquitetura pode ser usada como um foco de discussão pelos *stakeholders* do sistema.
- Análise de sistemas
  - ✓ Significa que a análise a respeito da possibilidade do sistema atender a sua requisitos não-funcionais é possível.
- Reuso em larga escala
  - ✓ A arquitetura pode ser reusável em uma variedade de sistemas.
  - ✓ Podem ser desenvolvidas arquiteturas de linhas de produtos.

Representações de arquiteturas

- Diagramas de blocos informais simples mostrando as entidades e os relacionamentos são o método mais usado para documentar as arquiteturas de software.

- Mas esses têm sido criticados pela falta de semântica, e por não mostrarem os tipos de relacionamentos entre as entidades, nem as propriedades visíveis das entidades na arquitetura.

- Depende do uso dos modelos de arquitetura.

- Os requisitos para a semântica do modelo dependem de como os modelos são usados.

Diagramas de caixa e linha

engenharia de SOFTWARE

- Muito abstrato – não mostram a natureza dos relacionamentos dos componente nem as propriedades externamente visíveis dos subsistemas.

- No entanto, é útil para a comunicação com os *stakeholders* e para o planejamento do projeto.

Uso de modelos de arquitetura

- Como forma de facilitar a discussão sobre o projeto do sistema

  - ✓ Uma visão de alto nível da arquitetura de um sistema é útil para a comunicação com os *stakeholders* do sistema e planejamento do projeto, pois essa não é cheio de detalhes. Os *stakeholders* podem se relacionar e entender uma visão abstrata do sistema. E então, discutir o sistema como um todo, sem a possibilidade de serem confundidos pelos detalhes.

- Como uma forma de documentar uma arquitetura projetada

  - ✓ O objetivo aqui é produzir um modelo de sistema completo que mostre os diferentes componentes em um sistema, suas interfaces e suas conexões.

Decisões de projeto de arquitetura

engenharia de SOFTWARE

- O projeto de arquitetura é um processo criativo, assim, o processo difere de acordo com o tipo de sistema que está sendo desenvolvido.

- No entanto, uma série de decisões comuns abrangem todos os processos de projeto e essas decisões afetam as características não-funcionais do sistema.

Decisões de projeto de arquitetura

engenharia de SOFTWARE

- Existe uma arquitetura genérica de aplicação que possa ser usada?
- Como o sistema será distribuído?
- Quais estilos de arquitetura são apropriados?
- Que abordagem será usada para estruturar o sistema?
- Como o sistema pode ser decomposto em módulos?
- Qual estratégia de controle deve ser usada?
- Como o projeto de arquitetura será avaliado?
- Como a arquitetura deve ser documentada?

Reuso de arquitetura

- Muitas vezes os sistemas no mesmo domínio têm arquiteturas similares que refletem os conceitos do domínio.

- Linhas de produtos de aplicações são construídas em torno de uma arquitetura central com variantes que satisfaçam os requisitos particulares do cliente.

- A arquitetura de um sistema pode ser projetada em torno de um ou mais padrões ou "estilos " de arquitetura.

  - ✓ Essas capturam a essência de uma arquitetura e podem ser instanciadas de diferentes maneiras.

Características de arquitetura e de sistema

- Desempenho
  - ✓ Localize operações críticas e minimize as comunicações. Use componentes de alta granularidade ao invés de baixa granularidade.
- Proteção
  - ✓ Nas camadas internas, use uma arquitetura em camadas com ativos críticos.
- Segurança
  - ✓ Localize atributos de segurança crítica em um pequeno número de subsistemas.
- Disponibilidade
  - ✓ Incluem componentes redundantes e mecanismos de tolerância a defeitos.
- Manutenibilidade
  - ✓ Use componentes autocontidos, de baixa granularidade.

Visões de arquitetura

- Que pontos de vista ou perspectivas são úteis ao fazer o projeto e documentar a arquitetura de um sistema?

- Quais notações devem ser usadas para descrever os modelos de arquitetura?

  - ✓ Cada modelo de arquitetura mostra apenas um ponto de vista ou perspectiva do sistema.

  - ✓ Pode mostrar como um sistema é decomposto em módulos, como os processos interagem em tempo de execução ou as diferentes formas em que os componentes do sistema são distribuídos através de uma rede. Para ambos, projeto e documentação, você geralmente precisa apresentar múltiplas visões da arquitetura do software.

Modelo de visão 4 + 1 de arquitetura de software

- Uma visão lógica, que mostra as principais abstrações no sistema como objetos ou classes de objetos.
- Uma visão de processo, que mostra como, em tempo de execução, o sistema é composto por processos de interação.
- Uma visão de desenvolvimento, que mostra como o software é decomposto para o desenvolvimento.
- Uma visão física, que mostra o hardware do sistema e como os componentes do software são distribuídos entre os processadores do sistema.
- Usando casos de uso relacionados ou cenários (+1).

Padrões de arquitetura

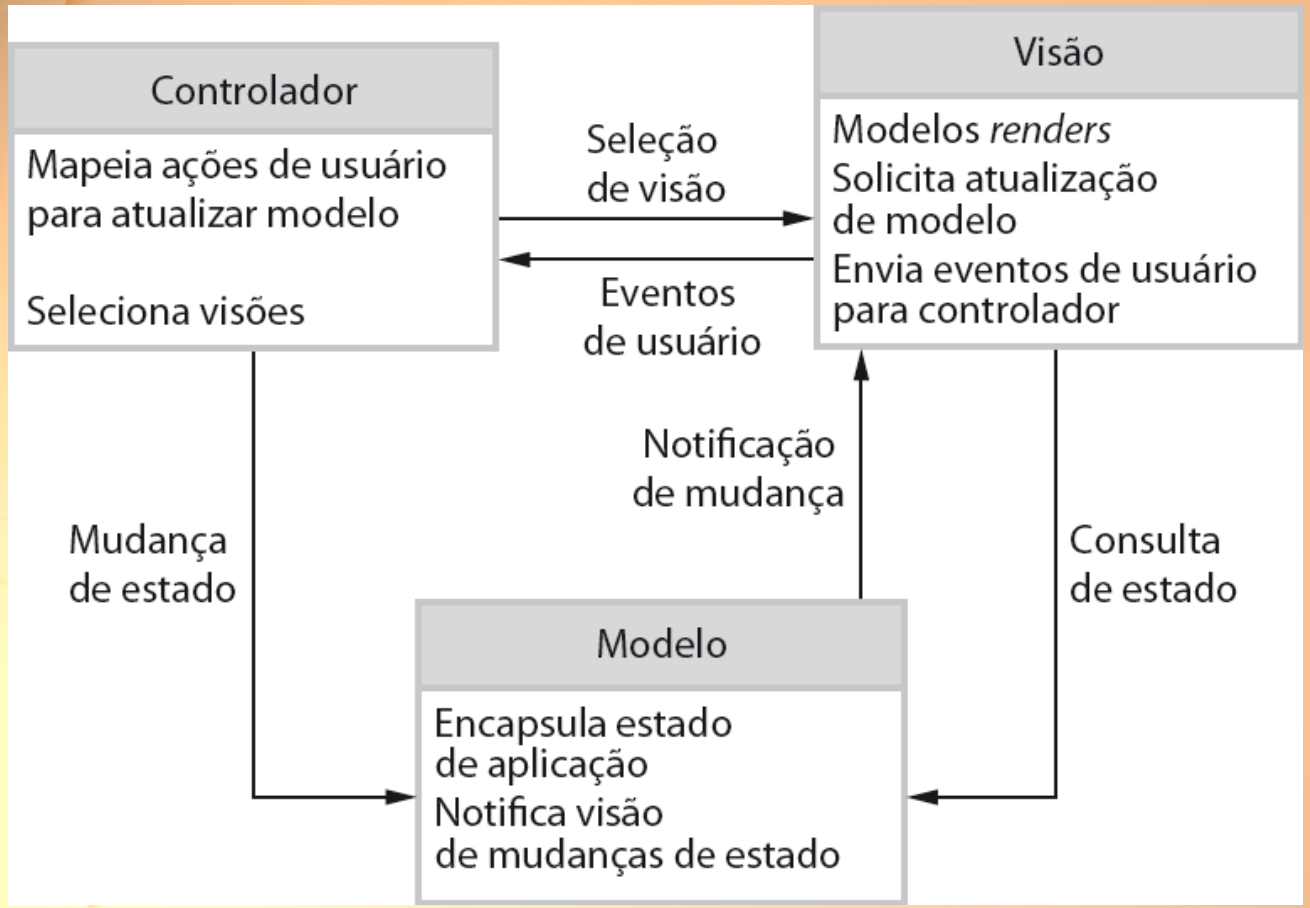
- Padrões são um meio de representar, partilhar e reusar conhecimento.
- Um padrão de arquitetura é uma descrição estilizada das boas práticas de projeto, que tem sido experimentadas e testadas em diferentes ambientes.
- Os padrões devem incluir informações sobre quando elas são úteis ou não.
- Os padrões podem ser representados usando descrições de tabelas e gráficos.

O padrão do Modelo-Visão-Controlador (MVC)

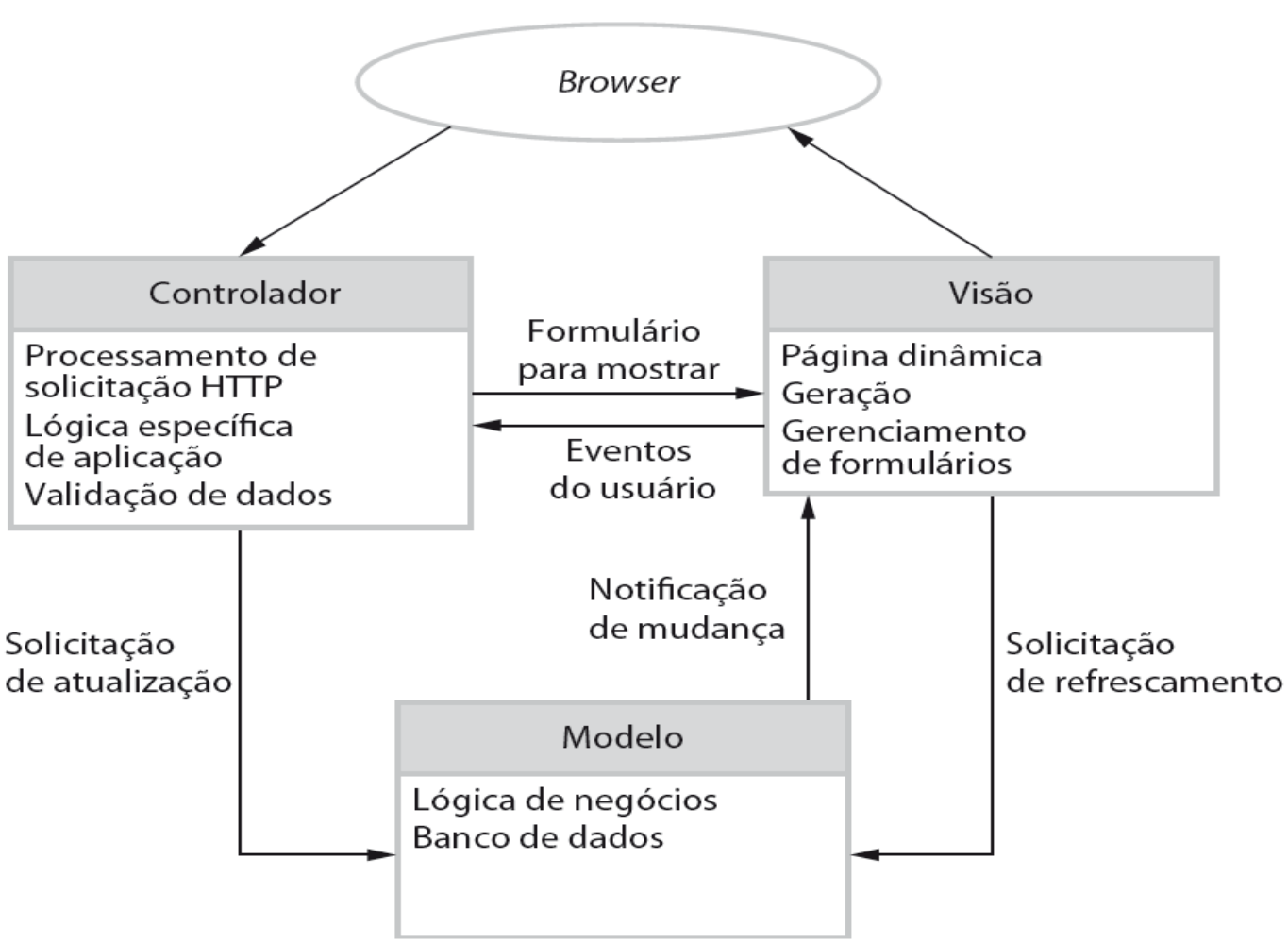
engenharia de SOFTWARE

Nome	MVC (Modelo-Visão-Controlador)
Descrição	Separa a apresentação e a interação dos dados do sistema. O sistema é estruturado em três componentes lógicos que interagem entre si. O componente Modelo gerencia o sistema de dados e as operações associadas a esses dados. O componente Visão define e gerencia como os dados são apresentados ao usuário. O componente Controlador gerencia a interação do usuário (por exemplo, teclas, cliques do mouse etc.) e passa essas interações para a Visão e o Modelo. Veja a Figura 6.2.
Exemplo	A Figura 6.3 mostra a arquitetura de um sistema aplicativo baseado na Internet, organizado pelo uso do padrão MVC.
Quando é usado	É usado quando existem várias maneiras de se visualizar e interagir com dados. Também quando são desconhecidos os futuros requisitos de interação e apresentação de dados.
Vantagens	Permite que os dados sejam alterados de forma independente de sua representação, e vice-versa. Apoia a apresentação dos mesmos dados de maneiras diferentes, com as alterações feitas em uma representação aparecendo em todas elas.
Desvantagens	Quando o modelo de dados e as interações são simples, pode envolver código adicional e complexidade de código.

A organização do MVC



A arquitetura de aplicações web usando o padrão MVC



Arquitetura em camadas

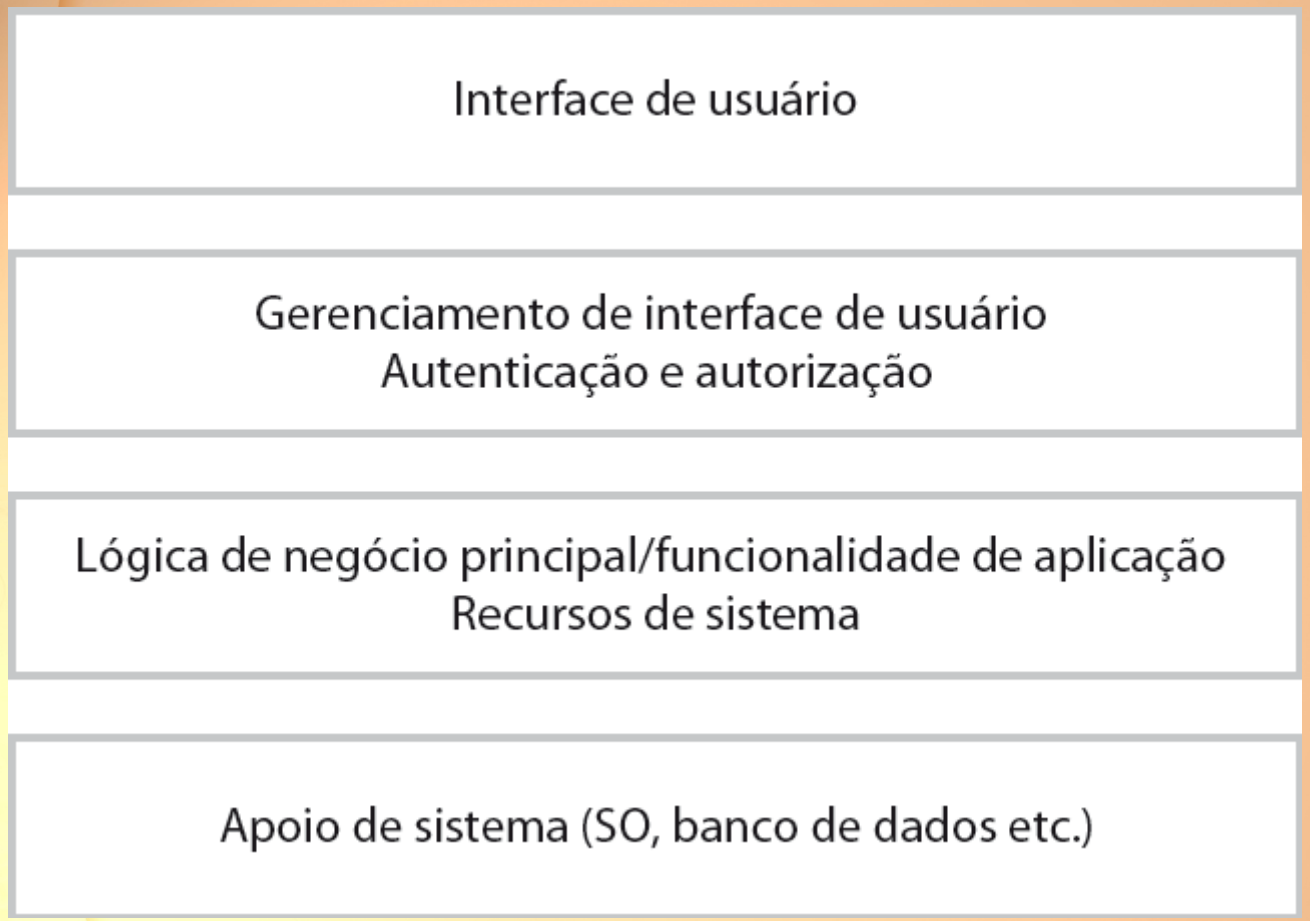
engenharia de SOFTWARE

- Usada para modelar a interface dos subsistemas.
- Organiza o sistema em um conjunto de camadas (ou máquinas abstratas) cada uma das quais fornecem um conjunto de serviços.
- Apoia o desenvolvimento incremental de subsistemas em diferentes camadas. Quando uma camada na interface muda, apenas a camada adjacente é afetada.
- No entanto, frequentemente, é artificial estruturar sistemas dessa forma.

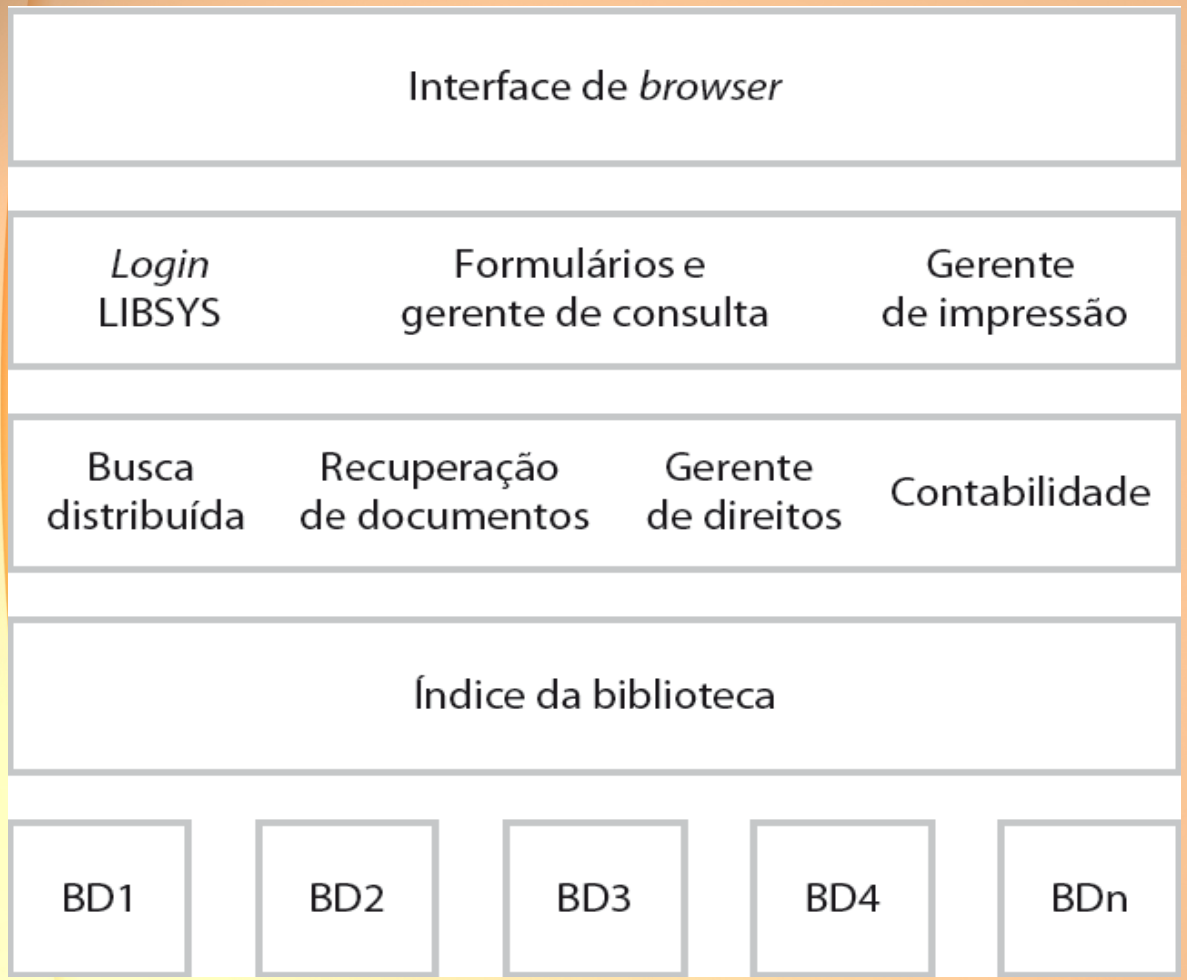
O padrão de arquitetura em camadas

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.

Uma arquitetura genérica em camadas



A arquitetura do sistema LIBSYS



Pontos Importantes

- Uma arquitetura de software é uma descrição de como um sistema de software é organizado.
- Decisões de projeto de arquitetura incluem decisões sobre o tipo de aplicação, a distribuição do sistema, e o estilo de arquitetura a ser usada.
- As arquiteturas podem ser documentadas de várias perspectivas ou visões diferentes tais como uma visão conceitual, uma visão lógica, uma visão de processo, uma visão de desenvolvimento e uma visão física.
- Os padrões de arquitetura são um meio de reusar o conhecimento sobre as arquiteturas genéricas de sistemas. Eles descrevem a arquitetura, explicam quando podem ser usados e descrevem suas vantagens e desvantagens.

Arquitetura de repositório

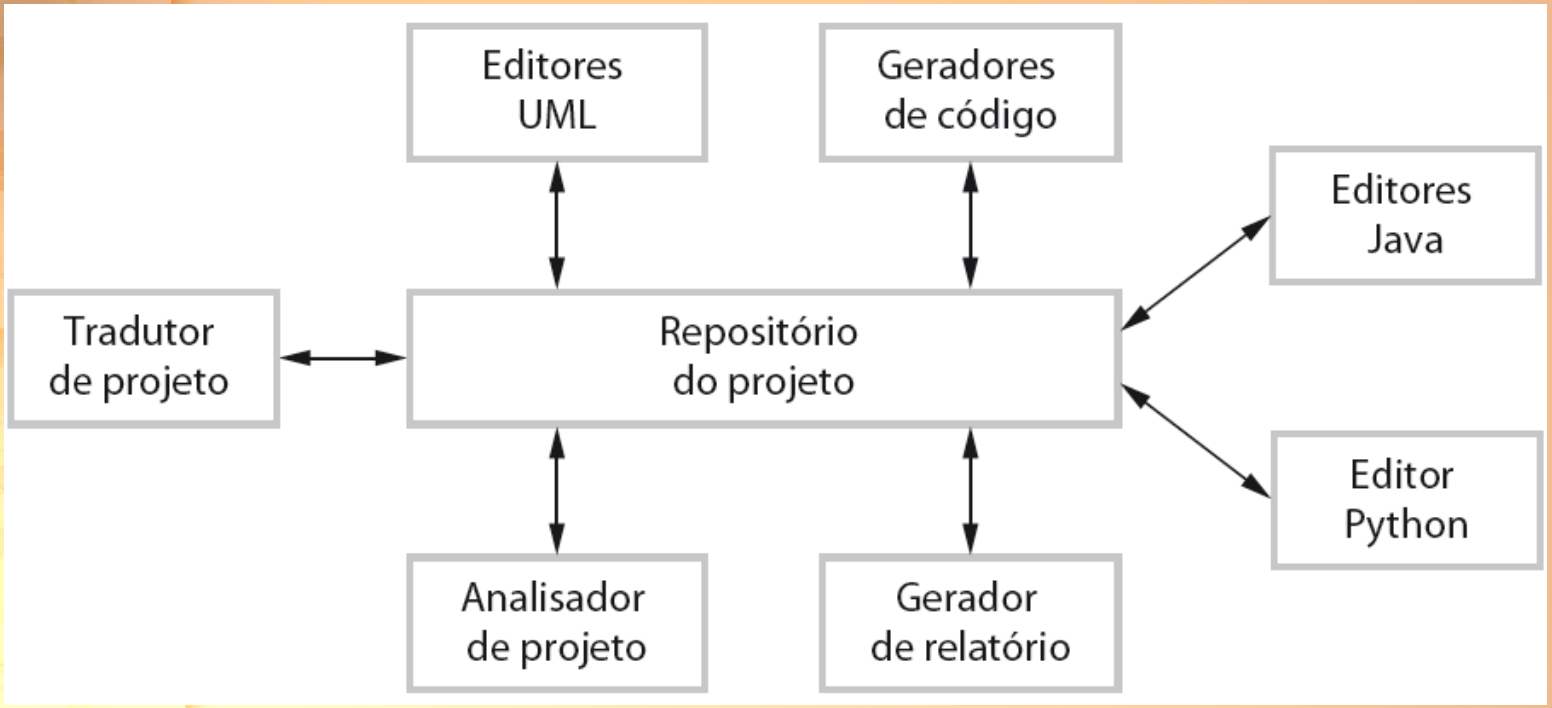
engenharia de SOFTWARE

- Subsistemas devem trocar dados. O que pode ser feito de duas maneiras:
  - ✓ Dados compartilhados são guardados em um banco de dados central ou repositório e podem ser acessados por todos os subsistemas;
  - ✓ Cada subsistema mantém seu próprio banco de dados e transmite dados explicitamente para outros subsistemas.
- Quando grandes quantidades de dados devem ser compartilhadas, é mais comum o uso do modelo de repositório compartilhado pois esse é um eficiente mecanismo de compartilhamento de dados.

O padrão Repositório

Nome	Repositório
Descrição	Todos os dados em um sistema são gerenciados em um repositório central, acessível a todos os componentes do sistema. Os componentes não interagem diretamente, apenas por meio do repositório.
Exemplo	A Figura 6.6 é um exemplo de um IDE em que os componentes usam um repositório de informações sobre projetos de sistema. Cada ferramenta de software gera informações que ficam disponíveis para uso por outras ferramentas.
Quando é usado	Você deve usar esse padrão quando tem um sistema no qual grandes volumes de informações são gerados e precisam ser armazenados por um longo tempo. Você também pode usá-lo em sistemas dirigidos a dados, nos quais a inclusão dos dados no repositório dispara uma ação ou ferramenta.
Vantagens	Os componentes podem ser independentes — eles não precisam saber da existência de outros componentes. As alterações feitas a um componente podem propagar-se para todos os outros. Todos os dados podem ser gerenciados de forma consistente (por exemplo, *backups* feitos ao mesmo tempo), pois tudo está em um só lugar.
Desvantagens	O repositório é um ponto único de falha, assim, problemas no repositório podem afetar todo o sistema. Pode haver ineficiências na organização de toda a comunicação através do repositório. Distribuir o repositório através de vários computadores pode ser difícil.

Uma arquitetura de repositório para um IDE



Arquitetura cliente-servidor

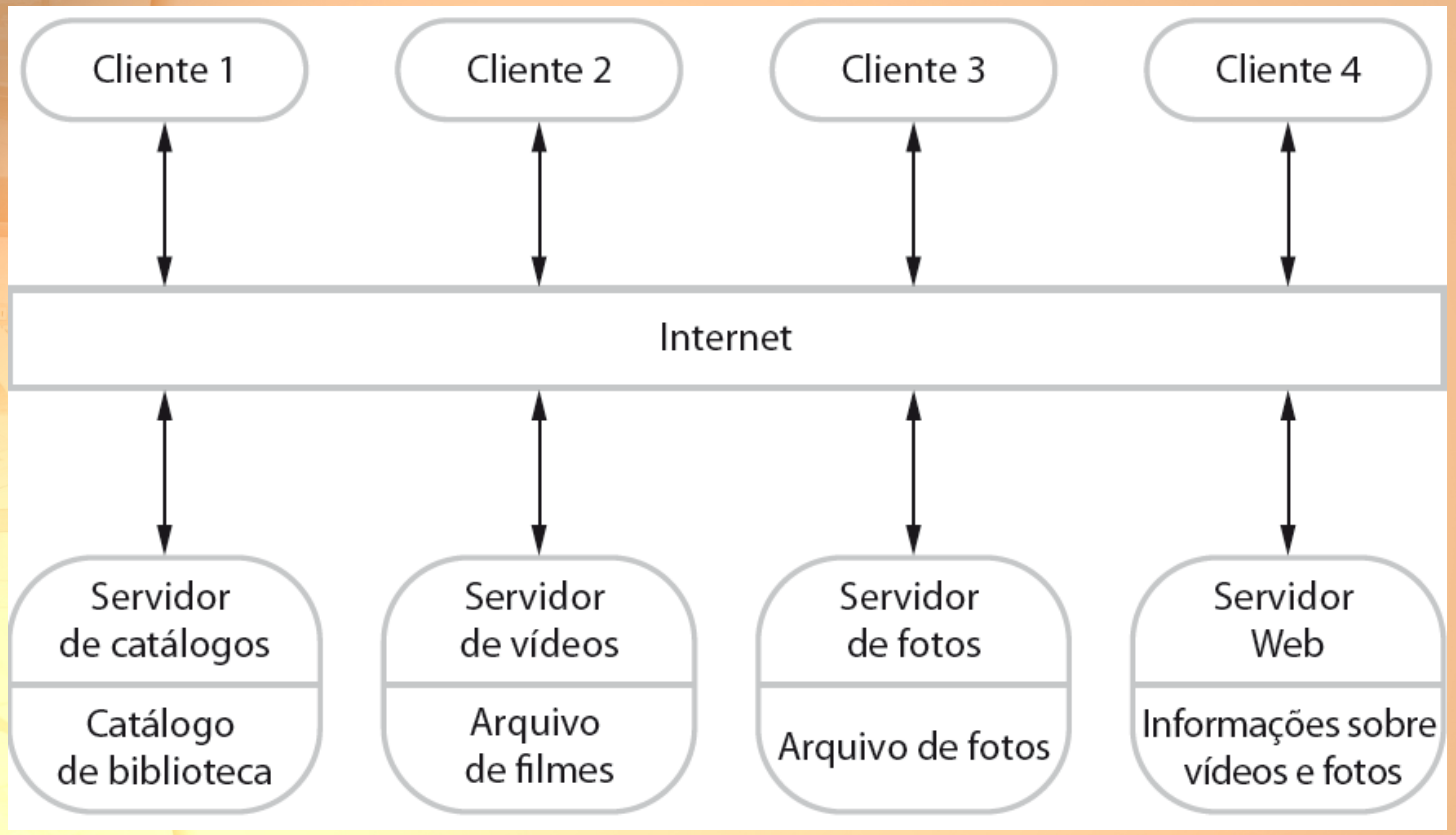
engenharia de SOFTWARE

- O modelo de sistema distribuído mostra como os dados e processamento são distribuídos através de uma série de componentes.
  - ✓ Pode ser implementado em um único computador.
- Um conjunto de servidores autônomos que prestam serviços específicos, tais como impressão, gerenciamento de dados, etc.
- Um conjunto de clientes que solicitam estes serviços.
- Rede que permite aos clientes acessar os servidores.

O padrão cliente-servidor

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.

A arquitetura cliente-servidor para uma biblioteca de filmes



Arquitetura de duto e filtro

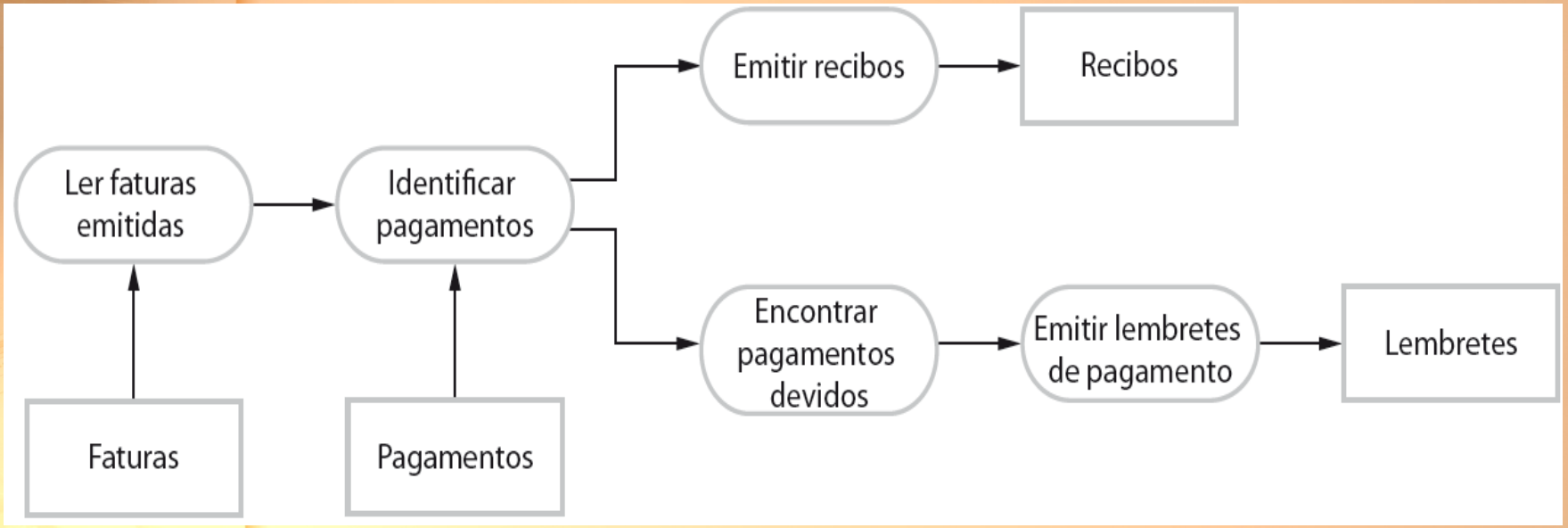
engenharia de SOFTWARE

- Transformações funcionais processam suas entradas para produzir saídas.
- Pode ser referido como um modelo de dutos e filtros (como no shell do UNIX).
- As variantes dessa abordagem são muito comuns.
- Quando as transformações são sequenciais, esse é um modelo de lote sequencial amplamente usado em sistemas de processamento de dados.
- Não é realmente adequado para sistemas interativos.

O padrão duto e filtro

Nome	Duto e filtro
Descrição	O processamento dos dados em um sistema está organizado de modo que cada componente de processamento (filtro) seja discreto e realize um tipo de transformação de dados. Os dados fluem (como em um duto) de um componente para outro para processamento.
Exemplo	A Figura 6.8 é um exemplo de um sistema de duto e filtro usado para o processamento das faturas.
Quando é usado	Comumente, é usado em aplicações de processamento de dados (tanto as baseadas em lotes como as baseadas em transações) em que as entradas são processadas em etapas separadas para gerarem saídas relacionadas.
Vantagens	O reúso da transformação é de fácil compreensão e suporte. Estilo de *workflow* corresponde à estrutura de muitos processos de negócios. Evolução por adição de transformações é simples. Pode ser implementado tanto como um sistema sequencial quanto concorrente.
Desvantagens	O formato para transferência de dados tem de ser acordado entre as transformações de comunicação. Cada transformação deve analisar suas entradas e gerar as saídas para um formato acordado. Isso aumenta o *overhead* do sistema e pode significar a impossibilidade de reúso de transformações funcionais que usam estruturas incompatíveis de dados.

Um exemplo da arquitetura duto e filtro



Arquiteturas de aplicações

- Os sistemas de aplicações são projetados para atender a uma necessidade organizacional.
- Como as empresas têm muito em comum, seus sistemas de aplicações também tendem a ter uma arquitetura comum que reflete os requisitos da aplicação.
- Uma arquitetura genérica de aplicação é uma arquitetura para um tipo de sistema de software que pode ser configurada e adaptada para criar um sistema que atenda aos requisitos específicos.

Uso de arquiteturas de aplicações

engenharia de SOFTWARE

- Como ponto de partida para o projeto de arquitetura.
- Como um checklist de projeto.
- Como uma forma de organizar o trabalho da equipe de desenvolvimento.
- Como uma forma de avaliar componentes para reuso.
- Como um vocabulário para falar sobre os tipos de aplicações.

Exemplos de tipos de aplicações

- Aplicações de processamento de dados
  - ✓ Aplicações centradas em dados que processam dados em lotes sem a intervenção explícita do usuário durante o processamento.
- Aplicações de processamento de transações
  - ✓ Aplicações centradas em banco de dados que processam solicitações dos usuários e atualizam as informações em um banco de dados do sistema.
- Sistemas de processamento de eventos
  - ✓ Aplicações em que as ações do sistema dependem da interpretação dos acontecimentos do ambiente do sistema.
- Sistemas de processamento da linguagem
  - ✓ Aplicações em que as intenções dos usuários são especificadas em uma linguagem formal, a qual é processada e interpretada pelo sistema.

Exemplos de tipos de aplicações

- O foco aqui é no processamento de transações e sistemas de processamento de linguagem.

- Sistemas de processamento de transações
  - ✓ Sistemas de comércio eletrônico;
  - ✓ Sistemas de reservas.

- Linguagem de processamento de sistemas
  - ✓ Compiladores;
  - ✓ Interpretadores de comando.

Sistemas de processamento de transações

engenharia de SOFTWARE

- Processar pedidos do usuário para informações de um banco de dados ou pedidos para atualizar o banco de dados.

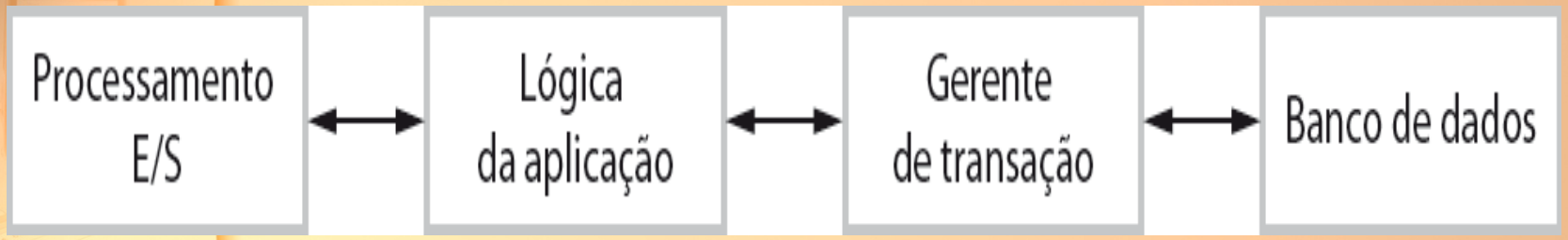
- Da perspectiva do usuário uma transação é:

  - ✓ Qualquer sequência coerente de operações que satisfaça uma meta;

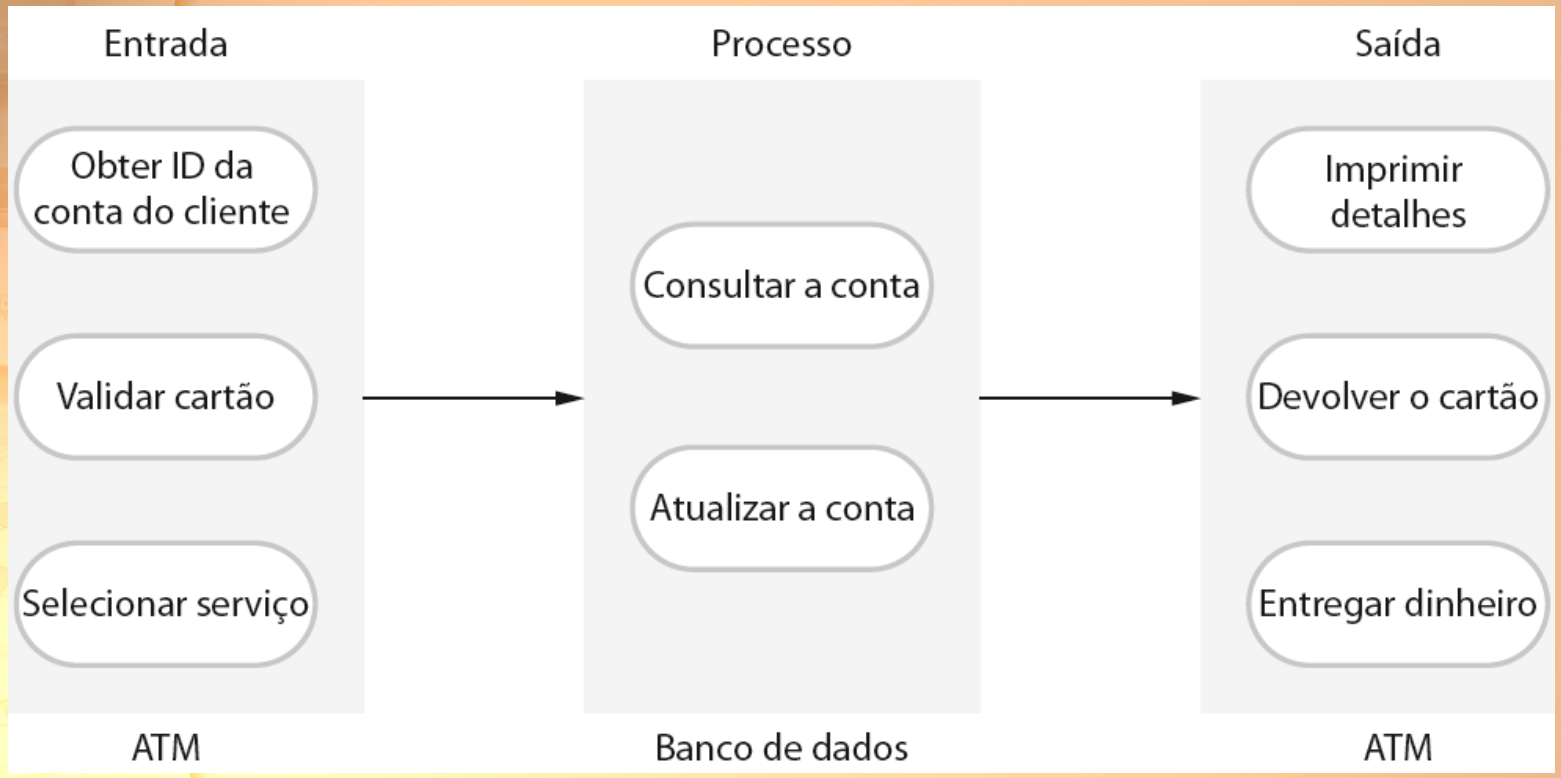
  - ✓ Por exemplo - encontrar os horários de vôos de Londres a Paris.

- Usuários fazem solicitações assíncronas de serviço que são então processadas por um gerenciador de transações.

A estrutura de aplicações de processamento de transações



A arquitetura de software de um sistema de ATM



Arquitetura de sistemas de informação

- Os sistemas de informação têm uma arquitetura genérica que pode ser organizada como uma arquitetura em camadas.

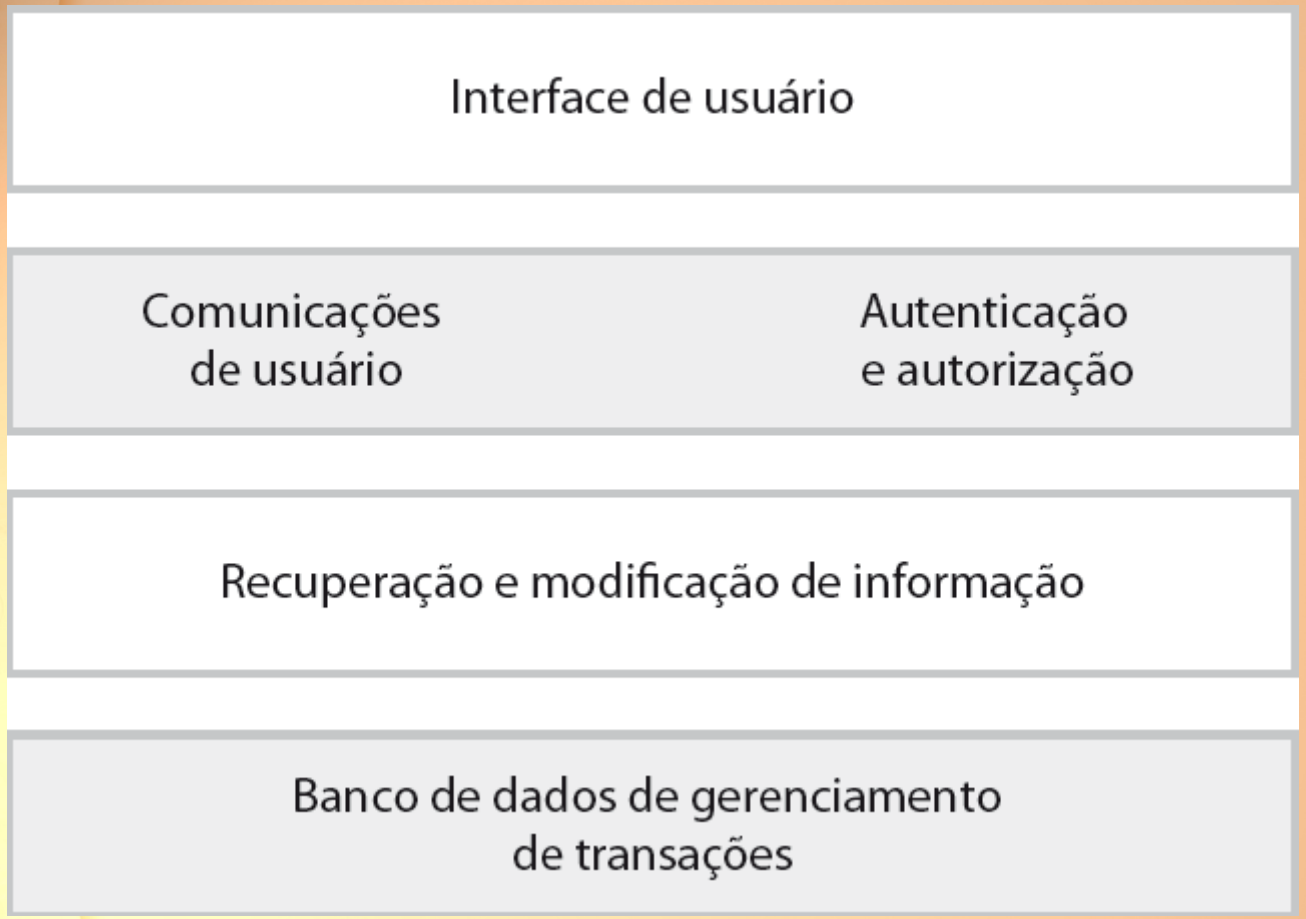
- Esses são sistemas baseados em transações pois geralmente a interação com esses sistemas envolve transações de banco de dados.

- As camadas incluem:

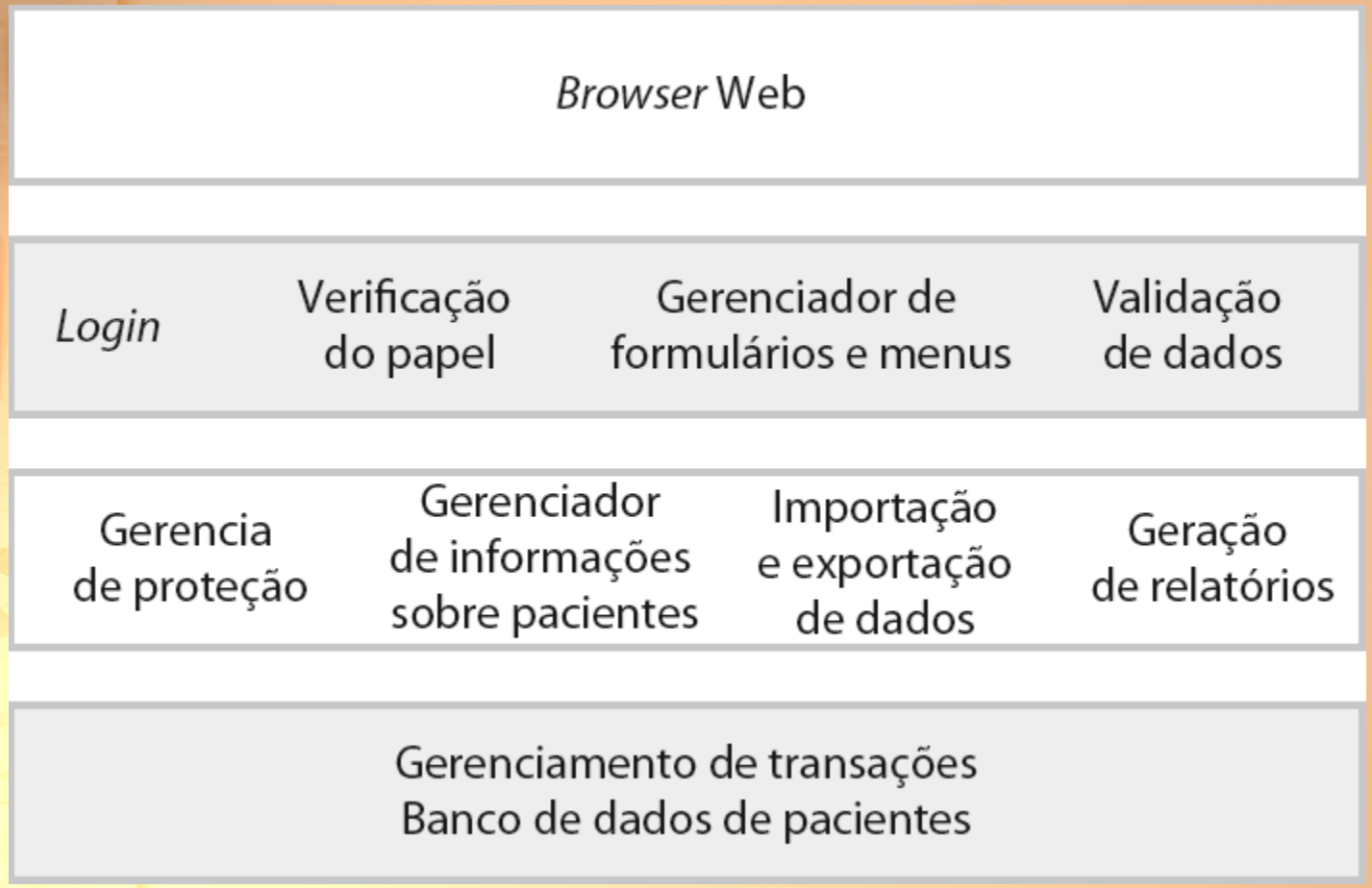
  - ✓ Interface de usuário
  - ✓ Comunicações de usuário
  - ✓ Recuperação e modificação de informações
  - ✓ Banco de dados do sistema

Arquitetura do sistema de informação em camadas

engenharia de SOFTWARE



A arquitetura do MHC-PMS



Sistemas de informação baseados na Web

- Normalmente os sistemas de informação e gerenciamento de recursos são sistemas baseados na web, onde as interfaces de usuário são implementadas usando um browser.

- Por exemplo, os sistemas de comércio eletrônico são sistemas de gerenciamento de recursos baseados na web, que aceitam pedidos eletrônicos de bens ou serviços e, em seguida providenciam a entrega desses bens ou serviços aos clientes.

- Em um sistema de comércio eletrônico, a camada específica da aplicação inclui funções adicionais de apoio ao 'carrinho de compras', no qual os usuários podem colocar um número de itens em transações separadas, e em seguida, pagar por todos juntos em uma única transação.

Implementação do servidor

- Frequentemente esses sistemas são implementados como arquiteturas cliente-servidor multicamadas

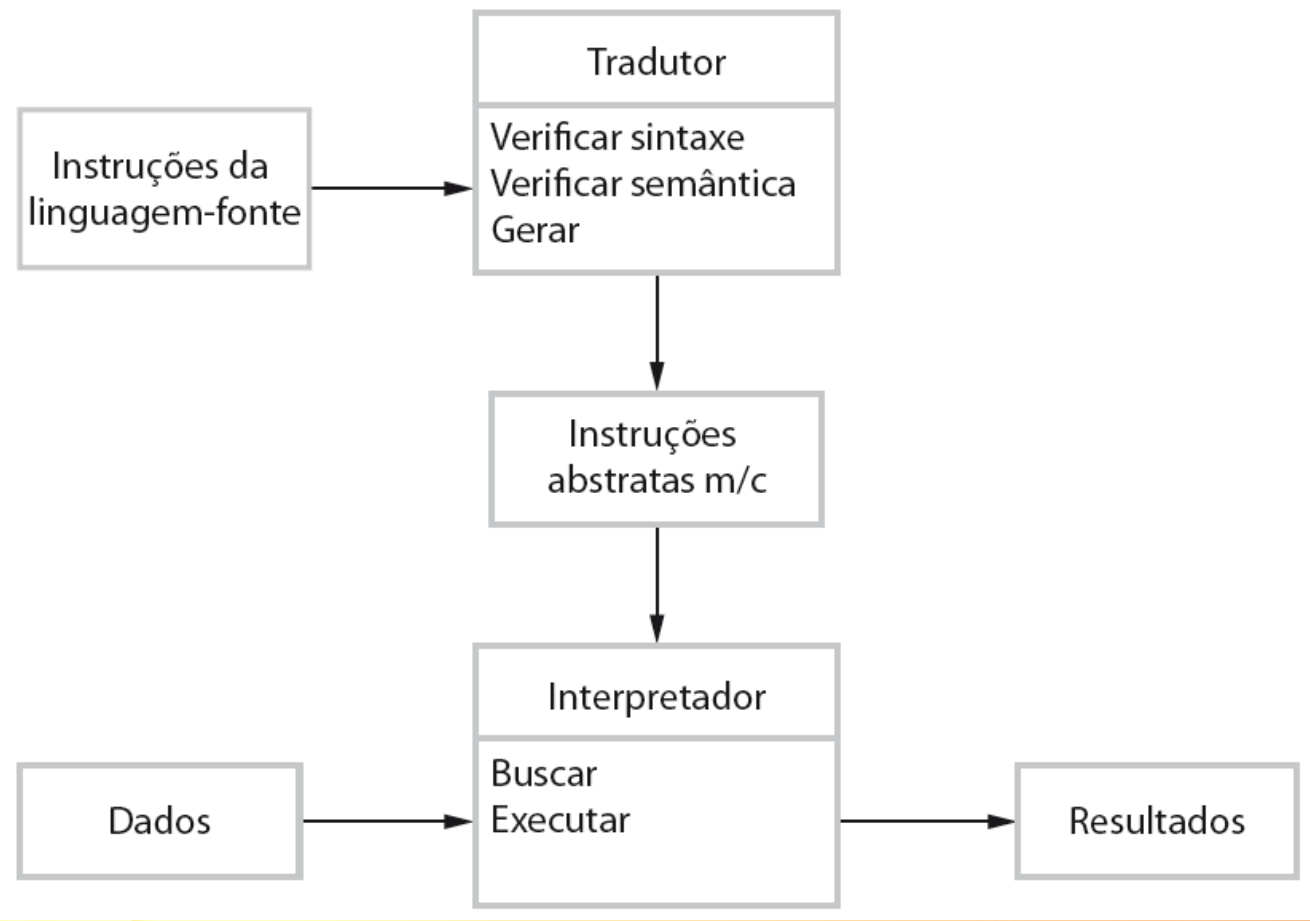
  - ✓ O servidor web é responsável por todas as comunicações do usuário, com a interface do usuário implementada usando um browser;
  - ✓ O servidor da aplicação é responsável pela implementação da lógica específica de aplicação assim como o armazenamento de informações e solicitações de recuperação;
  - ✓ O servidor do banco de dados move as informações de e para o banco de dados e lida com o gerenciamento de transações.

Sistemas de processamento de linguagem

- Aceitar uma linguagem natural ou artificial como entrada e gerar alguma outra representação dessa linguagem.
- Pode incluir um interpretador para dar seguimento nas instruções, na linguagem que está sendo processada.
- Utilizado em situações em que a maneira mais fácil para se resolver um problema é descrever um algoritmo ou descrever os dados do sistema.
  - ✓ Ferramentas meta-case processam descrições de ferramentas, regras de métodos, etc. e geram ferramentas.

A arquitetura de um sistema de processamento de linguagem

engenharia de SOFTWARE



Componentes do compilador

engenharia de SOFTWARE

- Um analisador léxico, que toma os tokens de entrada de linguagem e os converte para uma forma interna.
- A tabela de símbolos, que contém informação sobre os nomes de entidades (variáveis, nomes de classes, nomes de objetos, etc.) usadas no texto que está sendo traduzido.
- Um analisador sintático, que verifica a sintaxe da linguagem sendo traduzida.
- Uma árvore de sintaxe, é uma estrutura interna que representa o programa a ser compilado.

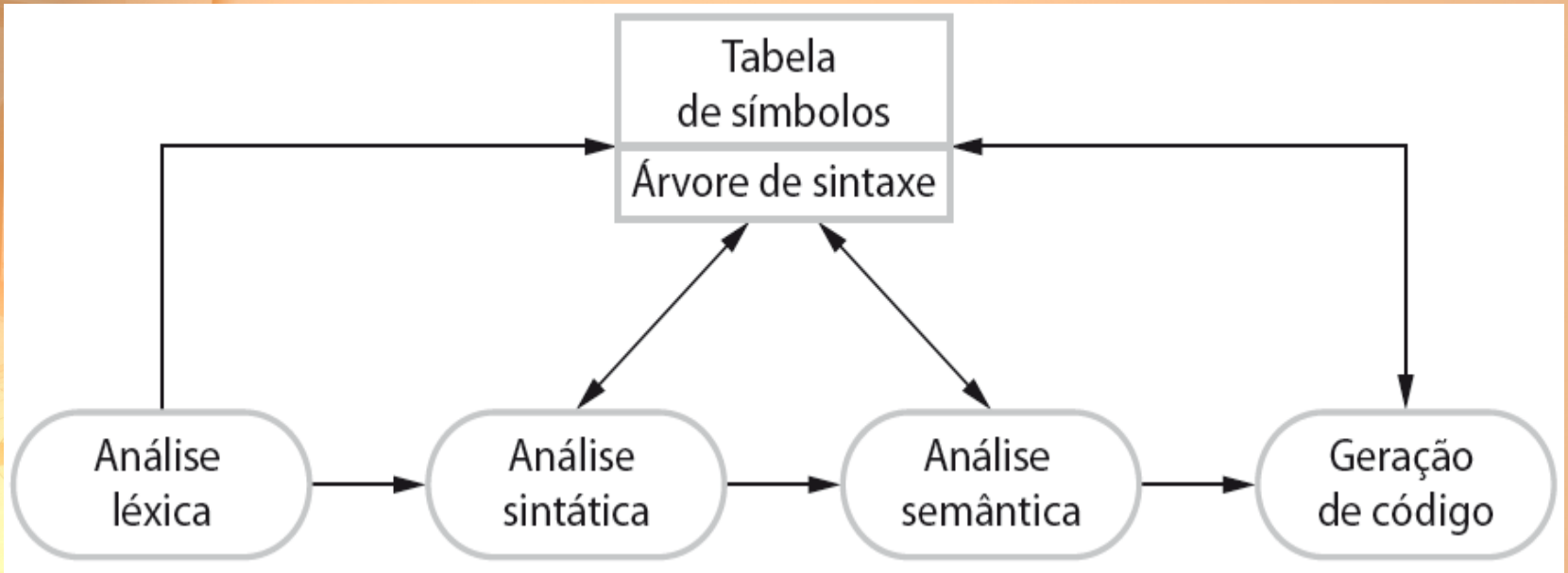
Componentes do compilador

engenharia de SOFTWARE

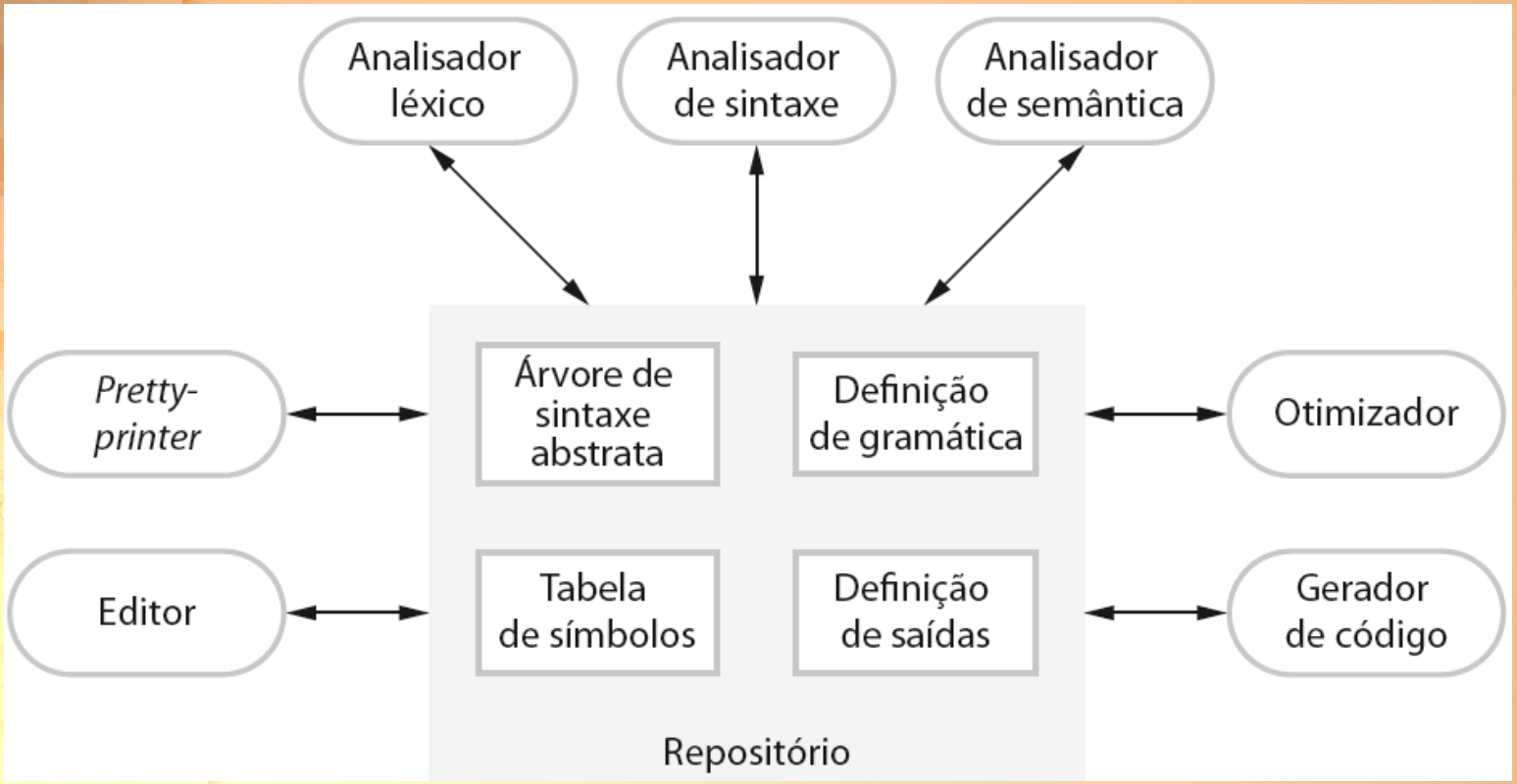
- Um analisador semântico que usa informações da árvore de sintaxe e a tabela de símbolos para verificar a correção semântica do texto da linguagem de entrada.

- Um gerador de código que 'anda' na árvore de sintaxe e gera códigos de máquina abstrata.

Uma arquitetura de compilador em duto e filtro



Uma arquitetura de repositório para um sistema de processamento de linguagem



Pontos Importantes

- Modelos genéricos de arquiteturas de sistemas de aplicação nos ajudam a entender e comparar as aplicações, validar projetos de sistemas de aplicação e avaliar componentes para reuso em larga escala.

- Os sistemas de processamento de transações são sistemas interativos que permitem que a informação em um banco de dados seja acessada remotamente e modificada por vários usuários.

- Os sistemas de processamento de linguagem são usados para traduzir textos de uma linguagens para outra e para realizar as instruções especificadas na linguagem de entrada.

- Eles incluem um tradutor e uma máquina abstrata que executa a linguagem gerada.