

Apresentação

Para iniciar uma etapa de projeto é necessário utilizar como ponto de partida um modelo de análise. Nesta Unidade de Aprendizagem você vai conhecer alguns elementos da aplicação Web que devem ser projetados: conteúdo, interação, navegação e armazenamento dos dados, sendo que o armazenamento dos dados terá mais destaque nesta unidade. Para que seja possível o armazenamento dos dados, é necessário que seja criado um banco de dados que permita sua definição e manipulação. Para que seja possível criar um banco de dados nessas condições, é necessária a elaboração de modelos, e dois exemplos desses modelos são: Diagrama Entidade/Relacionamento (DER) e Modelo Lógico.

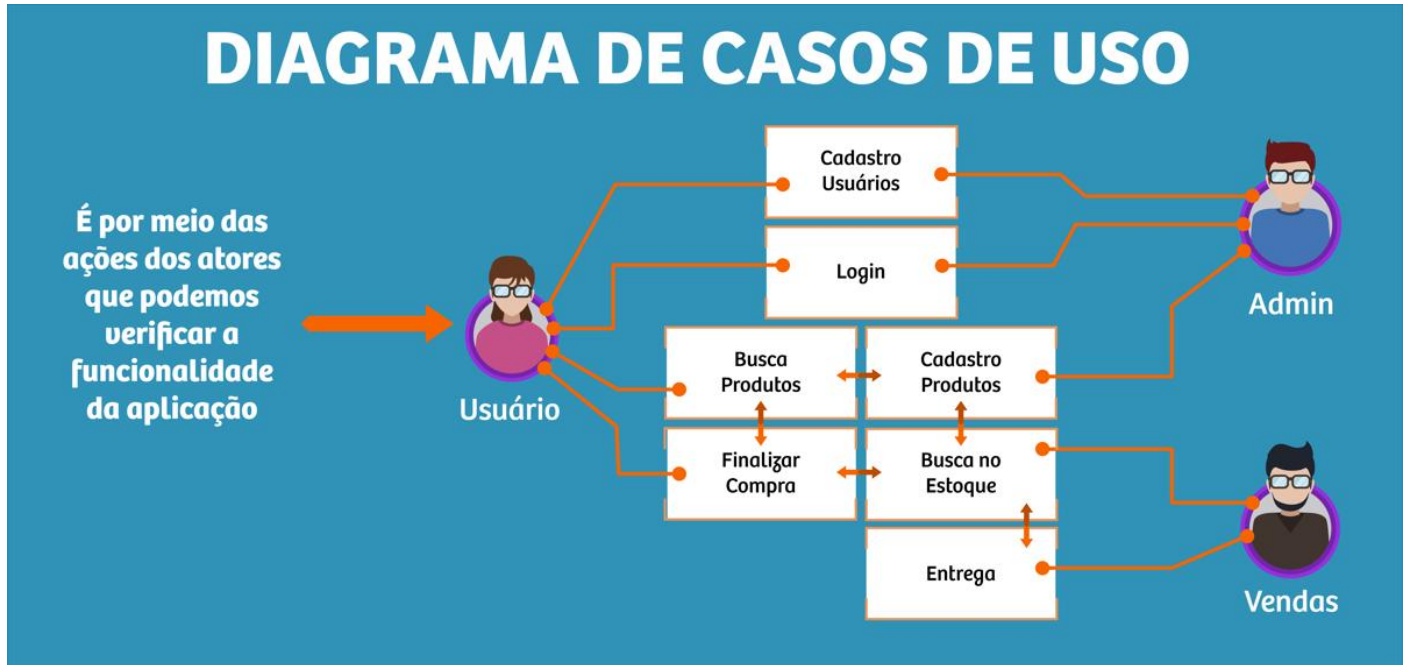
Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Explorar cenários de atores e ações.
- Identificar o projeto de armazenamento de dados.
- Reconhecer o modelo de Diagrama Entidade/Relacionamento (DER) e Modelo Lógico.

Infográfico

Neste infográfico é mostrado um diagrama de casos de uso, com as ações dentro da aplicação.



Conteúdo do Livro

Acompanhe um trecho da obra *Desenvolvimento de software II*, inicie a leitura a partir do tópico Diagrama de casos de uso: elementos e finalize em Modelo lógico.

Boa leitura.

EVANDRO MANARA MILETTO
SILVIA DE CASTRO BERTAGNOLLI



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO SUL

DESENVOLVIMENTO DE SOFTWARE II

INTRODUÇÃO AO DESENVOLVIMENTO WEB
COM HTML, CSS, JAVASCRIPT E PHP

EIXO INFORMAÇÃO E COMUNICAÇÃO

>> série tekne





D451 Desenvolvimento de software II [recurso eletrônico] : introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP / Organizadores, Evandro Manara Miletto, Sílvia de Castro Bertagnolli. – Dados eletrônicos. – Porto Alegre : Bookman, 2014.

Editado também como livro impresso em 2014.
ISBN 978-85-8260-196-9

1. Informática – Desenvolvimento de software. 2. HTML.
3. CSS. 4. JavaScript. 5. PHP. I. Miletto, Evandro Manara.
II. Bertagnolli, Sílvia de Castro.

CDU 004.41

»» Diagrama de casos de uso: elementos

O diagrama de casos de uso permite descrever os requisitos funcionais de um sistema de maneira simples e direta para usuários e desenvolvedores, pois ele deixa claro o que o sistema faz e quais são as funcionalidades que devem ser desenvolvidas.

Esse modelo é muito utilizado como ferramenta de comunicação entre os usuários e o desenvolvedor, pois, ao olhar o diagrama, fica fácil perceber quem faz o que e, com isso, todos os envolvidos percebem quando algo foi esquecido ou não está

mapeado corretamente. Esse modelo mostra o relacionamento entre os atores e os casos de uso dentro do sistema. Todo o modelo é composto por quatro elementos básicos: atores, casos de uso, relacionamentos e a fronteira do sistema.

Atores

Os atores representam um papel dentro do contexto do sistema, e não necessariamente uma pessoa. Isso se dá porque um mesmo usuário pode executar diversos papéis e um mesmo papel pode ser representado por muitos atores. Por exemplo: não podemos dizer que quem vai usar o sistema da loja virtual é “fulano”, mas sim o usuário ou o cliente da loja, logo os prováveis atores seriam “usuário” ou “cliente”.

Os atores representam o mundo externo, podendo ser pessoas, máquinas, dispositivos ou outros sistemas. Alguns exemplos típicos de atores típicos são: operador, cliente, gerente, computador, impressora e dispositivo de comunicação de rede. Assim, podemos deduzir que um ator é quem ou o que utiliza o sistema. Para facilitar a identificação de atores, você pode utilizar as seguintes questões:

- Que pessoas usarão as funcionalidades do sistema? Que papel elas exercem?
- Que pessoas serão responsáveis por manter e administrar o sistema? Que papel elas exercem?
- Quais são os componentes de hardware que o sistema necessita integrar?
- Existem outros sistemas que devem ser integrados ao sistema que está sendo desenvolvido?

Ao analisar os questionamentos acima, podemos deduzir que o nosso estudo de caso possui quatro atores, os quais estão ilustrados na Figura 2.3.

Um ator é representado por um boneco esquelético com uma descrição, que representa o seu papel no sistema.

Casos de uso

Um caso de uso é uma funcionalidade ou um serviço fornecido pelo sistema utilizado pelo(s) ator(es). Ele serve para documentar os comportamentos que são esperados do sistema.



Figura 2.3 Exemplos de atores.

Fonte: dos autores.

Na maioria das vezes, o caso de uso é um conjunto de ações que um sistema pode executar. Dessa forma, um caso de uso pode ser realizado por um botão, uma tela ou um conjunto de telas.

Em nosso estudo de caso, há diversos casos de uso. Apenas alguns são especificados na Figura 2.4. Note que a representação de um caso de uso é dada por uma elipse, com um texto em seu interior correspondendo ao nome do caso de uso.

Relacionamentos

O diagrama de casos de uso permite que os atores e os casos de uso possam representar as funcionalidades que serão disponibilizadas pelo sistema. Há três tipos de relacionamento, conforme descrito abaixo.

Relacionamento entre ator e caso de uso

É o principal tipo de relacionamento existente. Neste modelo, são informados quais casos de uso cada ator vai utilizar. A Figura 2.5 ilustra o mapeamento que deve ser realizado para indicar, por exemplo, que o Cliente é quem faz o seu próprio cadastro.

Relacionamento entre atores

Recomenda-se a utilização deste tipo de relacionamento quando existe um grande número de atores que possuem características comuns a vários grupos de usuários.

Relacionamentos entre casos de uso

Este modelo se divide em três tipos: inclusão, generalização e extensão. Cada um desses relacionamentos tem uma representação sobre o diagrama. A inclusão é representada por "<<include>>", e a extensão por "<<extend>>", enquanto a generalização utiliza uma representação gráfica (Figura 2.7).

Inclusão: é utilizada quando um caso de uso "inclui" outro caso de uso, de modo que atenda à funcionalidade prevista. Por exemplo: quando o cliente efetua a compra de um produto, deve selecionar a forma de pagamento, e o sistema deve emitir o comprovante do pedido. Logo, podemos criar três casos de uso: um que representa a ação de efetuar a

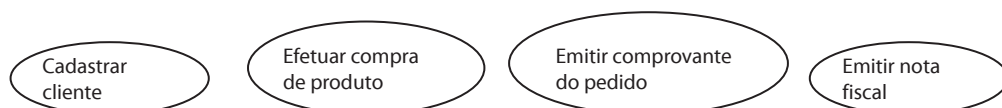


Figura 2.4 Exemplos de casos de uso.

Fonte: dos autores.

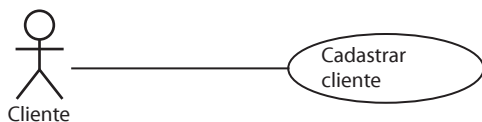


Figura 2.5 Exemplo de relacionamento ator-caso de uso.

Fonte: dos autores.

compra, um para a seleção da forma de pagamento e outro para a emissão do comprovante, como ilustra a Figura 2.6.

Generalização: indica que um caso de uso é mais geral e descreve parte das funcionalidades pertencentes a outro caso de uso mais específico, sendo que essas funcionalidades serão complementadas por esse caso de uso. A generalização entre casos de uso é representada como uma linha sólida direcionada com uma seta vazada apontando para o caso de uso mais geral (Figura 2.7). Nela, temos um caso de uso geral que possui parte da documentação relacionada à seleção da forma de pagamento, e os casos de uso específicos que completam o geral detalhando o que é específico para cada situação (pagamento com boleto ou pagamento com cartão de crédito).

Extensão: é utilizada quando um caso de uso possui uma condição para ocorrer, ou seja, ele somente ocorrerá em situações específicas. Pense em um caso de uso geral que pode permanecer isolado, mas que, sob determinadas circunstâncias, seu comportamento poderá ser estendido pelo comportamento de outro caso de uso. Considere, por exemplo, a situação de comprar um produto e possuir um cupom de desconto na loja virtual. Como você vai modelar, dizendo que o sistema deve levar em consideração o cupom de desconto, se essa situação é aplicável a somente alguns clientes? Para isso, vamos usar a extensão, que determina que o cupom só será cadastrado se o cliente possuir um, e que os cálcu-

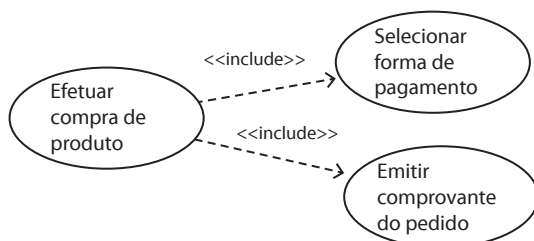


Figura 2.6 Exemplo de relacionamento de inclusão entre casos de uso

Fonte: dos autores.

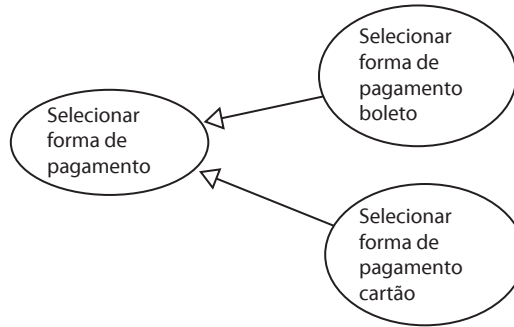


Figura 2.7 Exemplo de relacionamento de generalização entre casos de uso.

Fonte: dos autores.

los relacionados com a venda só levarão este cupom em consideração se ele existir (Figura 2.8).

>> Projeto do sistema

Para iniciar a etapa de projeto, é necessário utilizar como ponto de partida um modelo de análise. Neste livro, optamos por estabelecer como referência para o projeto o diagrama de casos de uso e as documentações vinculadas (especificações, documentação resumida e/ou documentação completa).

Existem alguns elementos da aplicação Web que devem ser projetados: conteúdo, interação, navegação e armazenamento dos dados. Neste capítulo – vamos nos concentrar no último item – você analisará aspectos relacionados aos outros tópicos no Capítulo 3.

Para possibilitar o **armazenamento dos dados**, é necessário criar um banco de dados que permita sua definição e manipulação. Para isso, resolvemos usar dois modelos clássicos da área de banco de dados: o Diagrama Entidade Relacionamento (DER) e o modelo lógico. As próximas seções apresentarão alguns aspectos teórico-práticos relacionados a esses modelos e como eles podem ser gerados a partir da análise e consideração do nosso estudo de caso.

» Diagrama Entidade Relacionamento

O **Diagrama Entidade Relacionamento** (DER) permite a modelagem dos conceitos que devem ser armazenados em um banco de dados. Nesse diagrama, as entidades representam um elemento da aplicação que deve armazenado em um banco de dados. Os relacionamentos entre entidades representam o número de elementos que uma entidade tem relacionados aos elementos de outra. Tanto entidades quanto relacionamentos podem ser caracterizados pelos seus atributos.

Para descobrir os atributos de uma entidade, devemos analisar todas as documentações relacionadas, bem como o projeto da interface gráfica com o usuário, pois, para fazer o cadastro, os dados necessários têm de ser informados. Alguns atributos da entidade vão se transformar em outras entidades, devido a dois fatores principais:

- 1) A entidade é composta por diversas outras informações, e se todas as informações fossem colocadas em um única entidade, elas ficariam sem sentido. Por exemplo: Entidade “Cliente” com Entidade “Endereço”.
- 2) A entidade está relacionada a várias outras ocorrências de outra entidade. Por exemplo: Entidade “Produto” com Entidade “Fotos”.

As entidades podem ser reconhecidas pelo identificador de entidade, que é formado por um atributo ou por um conjunto de atributos, de modo que cada ocorrência de entidade possa ser identificada de forma única.



» PARA SABER MAIS

Todos os aspectos teórico-práticos relacionados com o DER podem ser encontrados no livro *Projeto de Banco de Dados*, 6. ed., e também no livro *Desenvolvimento de software I: conceitos básicos*. (Veja as referências ao final do capítulo.)

Ao realizar uma análise do diagrama de casos de uso, podemos perceber que existem diversos elementos que devem ser armazenados no banco de dados. Como vamos priorizar o que será desenvolvido? Quais serão os primeiros casos de uso a serem desenvolvidos? Para responder a isso devemos identificar o caso de uso sem o qual os demais não irão funcionar.

Para que o cliente possa comprar os produtos, eles devem estar cadastrados. Assim, vamos priorizar a modelagem do banco de dados, começando pelo caso de uso “Cadastrar Produto”. Se você analisar o “Produto”, verá que ele tem várias características (atributos), como ilustra a Figura 2.11, que representa o exemplo no quadro a seguir.

>> EXEMPLO

Resolvemos vincular à Entidade “Produto” outras três entidades: “Fotos”, “Categoria” e “Fabricante”. Assim, representamos algumas definições do problema:

- “[...] Todos os produtos que compõem a pesquisa podem ser ordenados por fabricante [...]”

Dessa frase do problema, geramos a Entidade “Fabricante”.

- “[...] Nessa página é necessário exibir a(s) foto(s) relacionada(s) com o produto [...]”

Dessa frase, foi extraída a necessidade de que o “Produto” pode ter uma ou várias fotos.

Ao concluir a etapa de cadastro de produtos, podemos realizar o projeto do caso de uso “Pesquisar Produto”. Note que esse caso de uso não necessita de entidades para ser desenvolvido, somente do projeto da interface gráfica e de navegação.

Dando continuidade à priorização dos casos de uso, para que o cliente possa comprar, ele deve estar cadastrado no sistema. Logo, devemos mapear os seus atributos e as entidades relacionadas ao “Cliente”. O aspecto mais interessante ao modelar o “Cliente” é a sua relação com o seu endereço e com os endereços de

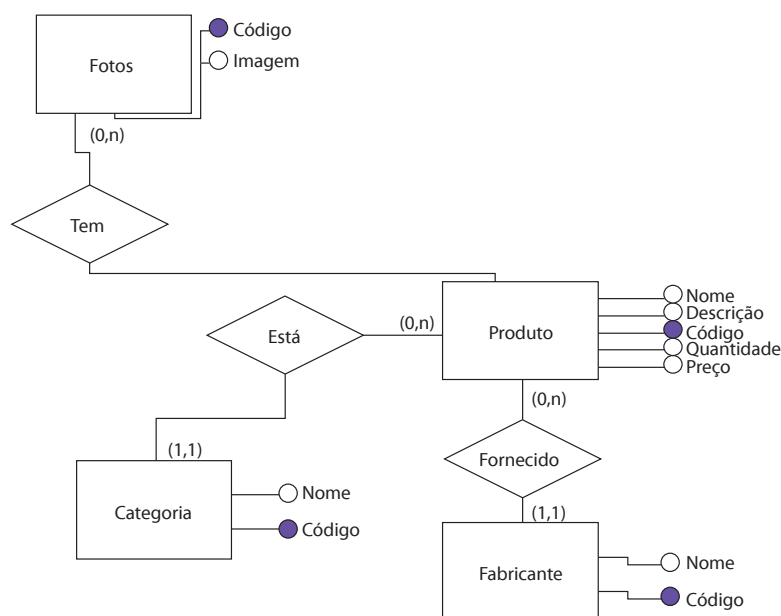


Figura 2.11 Exemplo de Entidades “Produto”, “Fotos”, “Fabricante” e “Categoria”.

Fonte: dos autores.

entrega, pois devemos garantir as restrições impostas na documentação inicial do problema: “Como muitas vezes as compras são realizadas por um cliente que deseja que o produto seja entregue como presente em outro endereço, é necessário que vários endereços de entrega possam ser cadastrados.” (Figura 2.12).

Para que o cliente faça seu cadastro, ele deve registrar um identificador de usuário (seu email) e uma senha, de modo a permitir o acesso ao sistema. Assim, para verificar se esses dados estão funcionando, podemos determinar como próximo caso de uso a ser desenvolvido o “Efetuar Login” e “Efetuar Logout”. Ambos não precisam de entidades para serem desenvolvidos, somente do projeto da interface gráfica e de navegação. E, no caso do login, os atributos já estão definidos na Entidade “Cliente”.

Assim, já mapeamos até o momento:

- O produto com suas fotos, categorias e fabricantes.
- A pesquisa de produto.
- O cliente e seus endereços.
- O login e o logout.

Agora, devemos nos preocupar com a venda do produto para o cliente, ou o caso de uso “Efetuar Compra de Produto” (Figura 2.13), uma vez que ele possui vinculados outros casos de uso. A Entidade “Produto” está relacionada com a Entidade “Cliente”, porque um pedido é realizado pelo cliente (pedido que deve ser entregue em um endereço de entrega, que não necessariamente é o endereço do cliente). O pedido deve ainda conter a lista de todos os produtos vinculados, assim como a forma de pagamento. Caso o cliente tenha optado por pagar com boleto, o pedido tem um prazo de validade. Se o cliente não pagar o boleto no prazo definido, o pedido deixa de existir.

A expedição do produto é confirmada somente após a confirmação do pagamento. Para isso, o funcionário analisa o pedido e, se tudo estiver correto, é emitida a nota fiscal para o endereço de entrega registrado. Note que para o funcionário realizar a expedição ele necessita ter permissão de acesso ao sistema, logo, deve possuir um identificador de usuário e uma senha para entrar na área restrita do sistema, como esquematiza a Figura 2.14.

Ao concluir a análise de todos os casos de uso e mapeá-los em entidades e seus atributos e relacionamentos, obtivemos como resultado o diagrama esquematizado pela Figura 2.15, o qual reúne todas as partes apresentadas previamente e mostra como elas se comunicam.

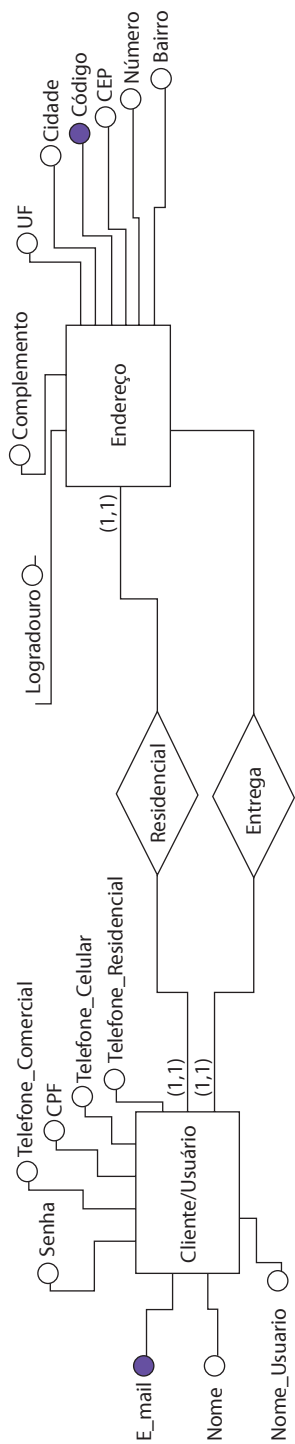
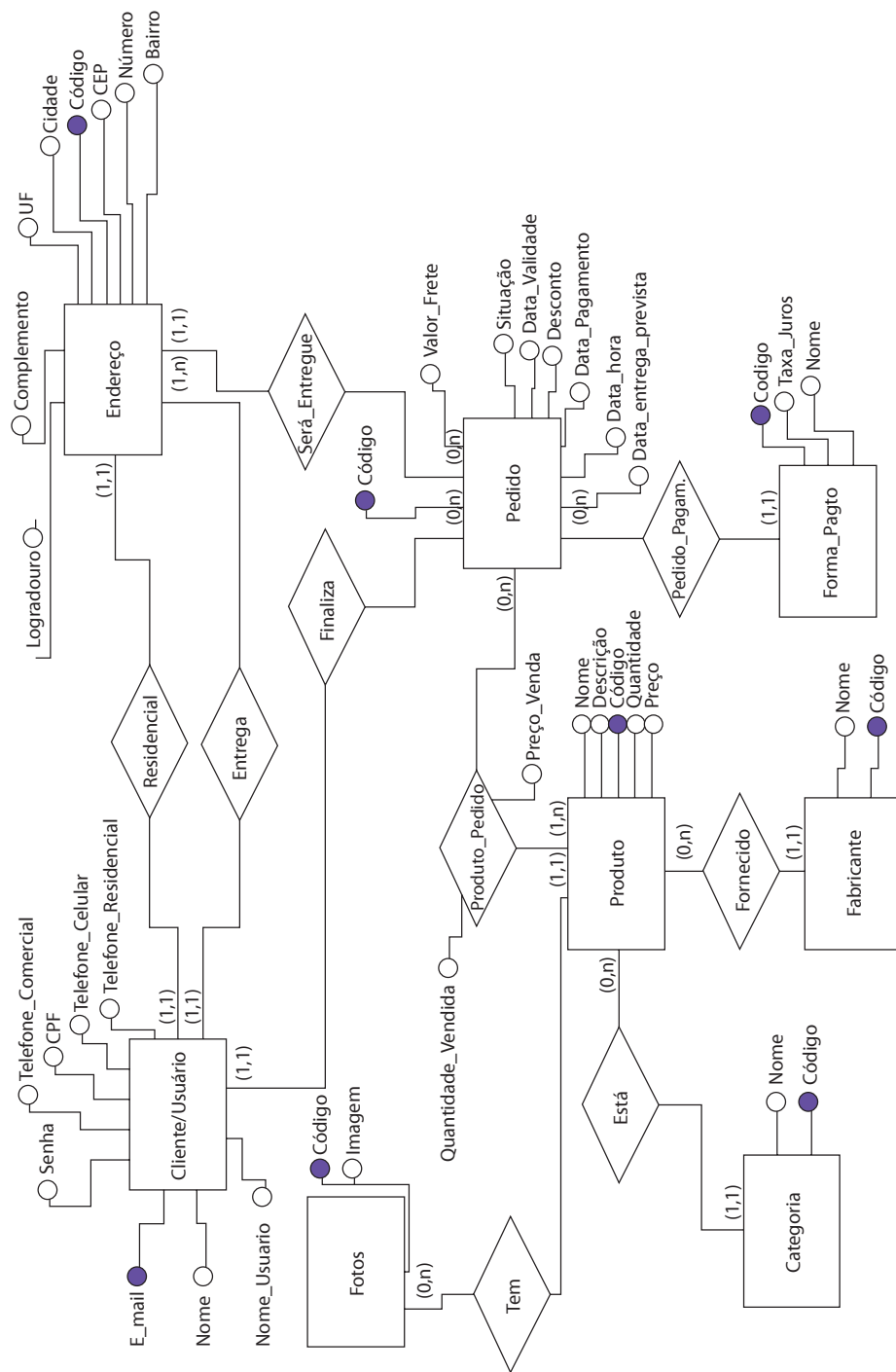


Figura 2.12 Exemplo de Entidade “Cliente” e “Endereço”.

Fonte: dos autores.

**Figura 2.13** Exemplo com as entidades relacionadas com a compra do produto.

Fonte: dos autores.

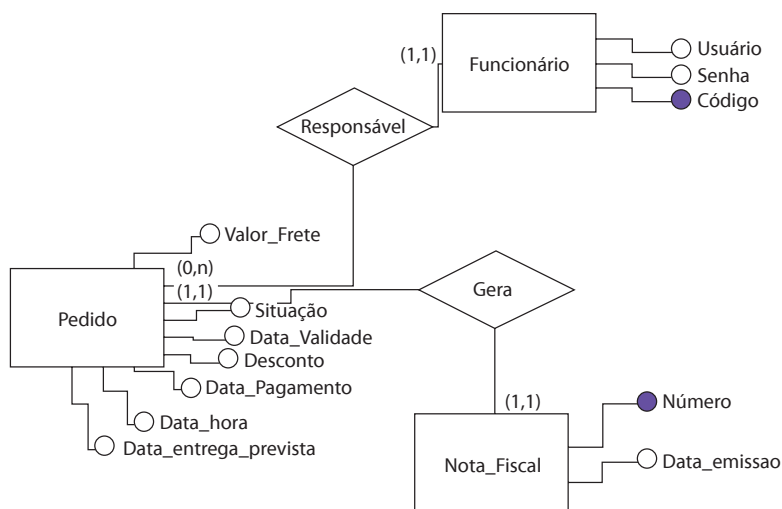


Figura 2.14 Exemplo de Entidade “Pedido” e seus relacionamentos.

Fonte: dos autores.

» Modelo lógico

O **modelo lógico** considera as características e restrições do Sistema Gerenciador de Banco de Dados (SGBD), enquanto o DER é um modelo que serve para realizar um mapeamento dos dados que fazem parte da solução do problema. Com o modelo lógico, passamos a definir as chaves primárias e estrangeiras e utilizamos regras de normalização e integridade referencial.

O modelo lógico pode ser criado a partir do DER, seguindo algumas regras:

Entidade: cada entidade é transformada em uma tabela. A tabela é composta por linhas ou tuplas, e cada linha tem um conjunto de campos, os quais são os valores dos atributos.

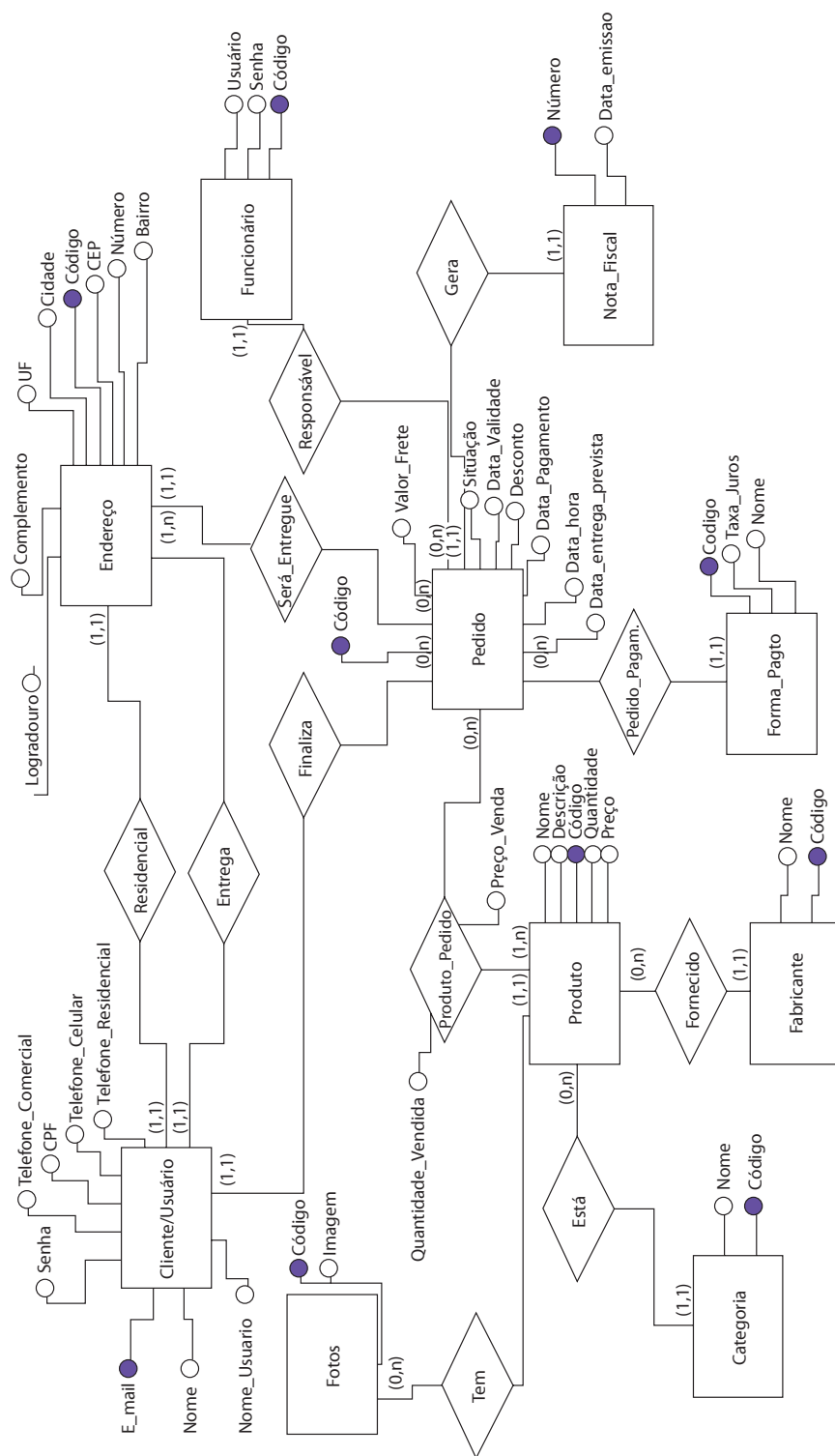
Coluna: cada atributo é traduzido como uma coluna da tabela.

Chave primária: usada para diferenciar uma linha das outras linhas na mesma tabela. Geralmente, o identificador de identidade é transformado na chave primária.

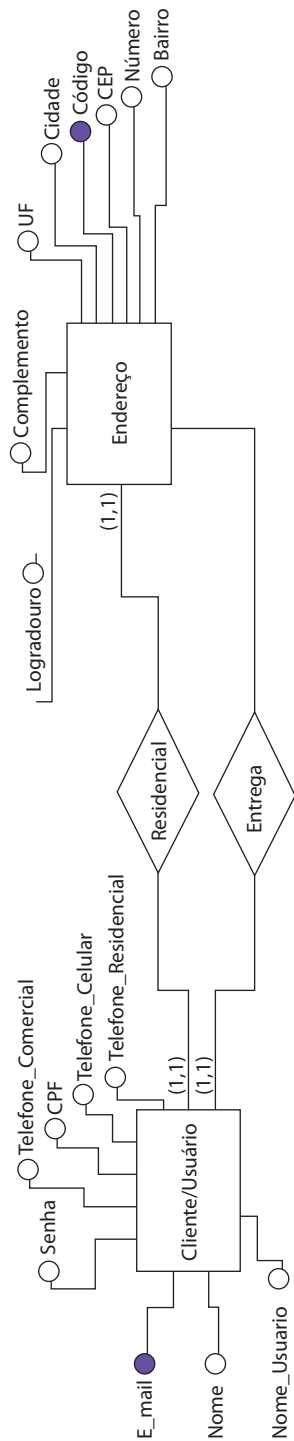
Chave estrangeira: usada para implementar os relacionamentos entre tabelas.

Vamos partir do DER, criado na seção anterior, e montar o modelo lógico. O DER será decomposto de modo a facilitar a sua compreensão.

O primeiro mapeamento selecionado para ser realizado foi o do “Cliente” com os seus “Endereços” (Figura 2.16). Observe que as entidades foram transformadas em tabelas, e seus atributos, em colunas. Os atributos identificadores foram traduzi-

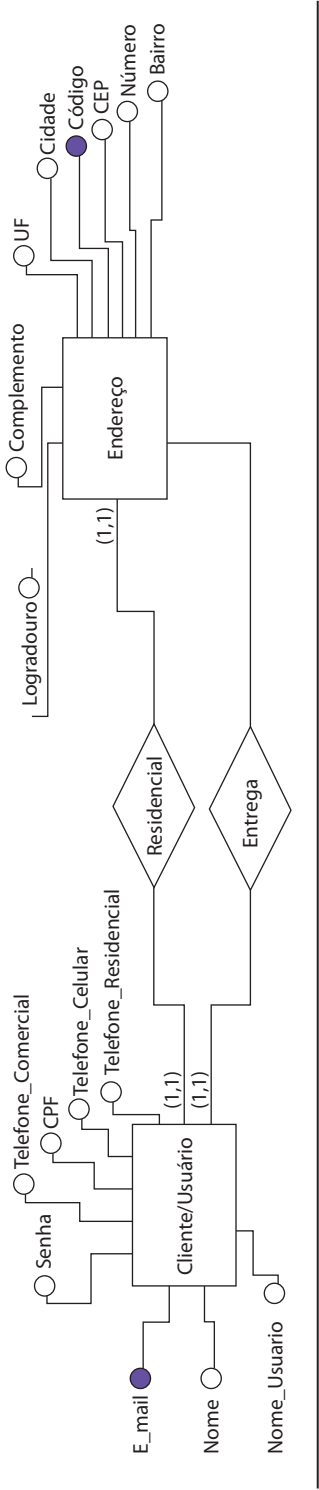
**Figura 2.15** DER do estudo de caso.

Fonte: dos autores.



Produtos	Produtos
Nome_Usuário: Texto(15)	Número: Número
Senha: Texto(10)	CEP: Número(7)
E_mail_Cliente: Texto(30)	Logradouro: Texto(30)
Nome: Texto(30)	Complemento: Texto(50)
Telefone_Residencial: Texto(11)	Cidade: Texto(30)
Telefone_Celular: Texto(11)	UF: Texto(2)
CPF: Número(11)	Bairro: Texto(20)
Telefone_Comercial: Texto(11)	Código_Endereço: Número
Código_End_Resid: Número	E_mail_Cliente: Texto(30)

Figura 2.16 Modelo lógico: “Cliente” e “Endereço”. (Continua)
 Fonte: dos autores.



Produtos	
Número: Número	
CEP: Número(7)	(1,1)
Logradouro: Texto(30)	
Complemento: Texto(50)	
Cidade: Texto(30)	
UF: Texto(2)	(1,n)
Bairro: Texto(20)	
Código_Endereço: Número	
E_mail_Cliente: Texto(30)	

Figura 2.16 Modelo lógico: “Cliente” e “Endereço”. (Continuação)
Fonte: dos autores.

dos para chaves primárias. Já os relacionamentos “Residencial” e “Entrega” foram mapeados com chaves estrangeiras: (i) a chave estrangeira “Codigo_End_Resid” na tabela “Cliente/Usuário” é usada para determinar o endereço residencial do cliente, enquanto a chave estrangeira “E_mail_Cliente” na tabela “Endereço” é usada para determinar a qual cliente pertence o referido endereço de entrega.

A Figura 2.15 ilustra o mapeamento realizado para as Entidades “Fotos”, “Categoria” e “Fabricante”. Observe que, na tabela “Fotos”, aparece a chave estrangeira “Codigo_Produto” indicando a qual produto a foto pertence. E, na tabela “Produto”, aparecem duas chaves estrangeiras: “Codigo_Fabricante” e “Codigo_Produto”. Essas chaves surgem porque a cardinalidade máxima entre “Fabricante” e “Produto” é “n”. Logo, para mapear corretamente, precisamos incluir uma coluna na tabela Produto que corresponde à chave primária da tabela “Fabricante”.

O modelo lógico final gerado a partir do DER da Figura 2.15 se encontra esquematizado na Figura 2.18.

Outras fases

Além das fases de delimitação, de análise e de projeto do sistema, há outras etapas que estão relacionadas com o desenvolvimento de aplicações Web. São elas: a implementação, os testes e a implantação. A seguir, o detalhamento de cada uma delas:

Implementação: consiste em criar o código que vai ser utilizado pelo usuário da aplicação. No caso de aplicações Web, você provavelmente utilizará tecnologias como HTML, CSS, JavaScript e PHP, e, para manipular os dados, SQL e um SGBD (MySQL, no caso deste livro).

Testes: são técnicas para mostrar a presença de erros na aplicação que está sendo desenvolvida. Existem várias estratégias de teste de software como, por exemplo, teste de unidade, teste de integração, teste de validação e teste de sistema.

Implantação: compreende todos os processos necessários para disponibilizar a aplicação para o cliente, desde aspectos de configuração de ambiente até aspectos de manutenção do software.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Dica do Professor

Neste vídeo você vai estudar o Diagrama de Casos de Uso, o Diagrama Entidade/Relacionamento e o Modelo Lógico.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

1) Qual a função de um Diagrama de Casos de Uso?

- A) Descrever os requisitos funcionais de um sistema de maneira simples e direta para usuários e desenvolvedores.
- B) Estudar casos antigos de outros projetos para que não se caia nos mesmos erros.
- C) Descrever os requisitos de hardware necessários para que o banco de dados opere em uma velocidade adequada.
- D) Diagnosticar as dificuldades do usuário quanto à utilização e à navegação dentro da aplicação.
- E) Definir a linguagem utilizada pelos desenvolvedores.

2) O que são os atores em um Diagrama de Casos de Uso?

- A) Pessoas que utilizam o site de uma forma fictícia, para divulgação.
- B) Funções automáticas da aplicação.
- C) Os papéis e funções que cada um representa dentro do contexto do sistema.
- D) Toda a equipe de manutenção do site.
- E) Usuários falsos que tentam invadir o sistema.

3) Quais são os três tipos de relacionamentos que existem no Diagrama de Casos de Uso?

- A) Entre usuários, entre usuário e desenvolvedor e entre desenvolvedores.
- B) Entre ator e caso de uso, entre atores e entre casos de uso.
- C) Relação Humano-Computador, Humano-Humano e entre computadores.
- D) Curta, mediana e extensa.
- E) Direta, inversa e mista.

4) Qual a função do Diagrama Entidade/Relacionamento?

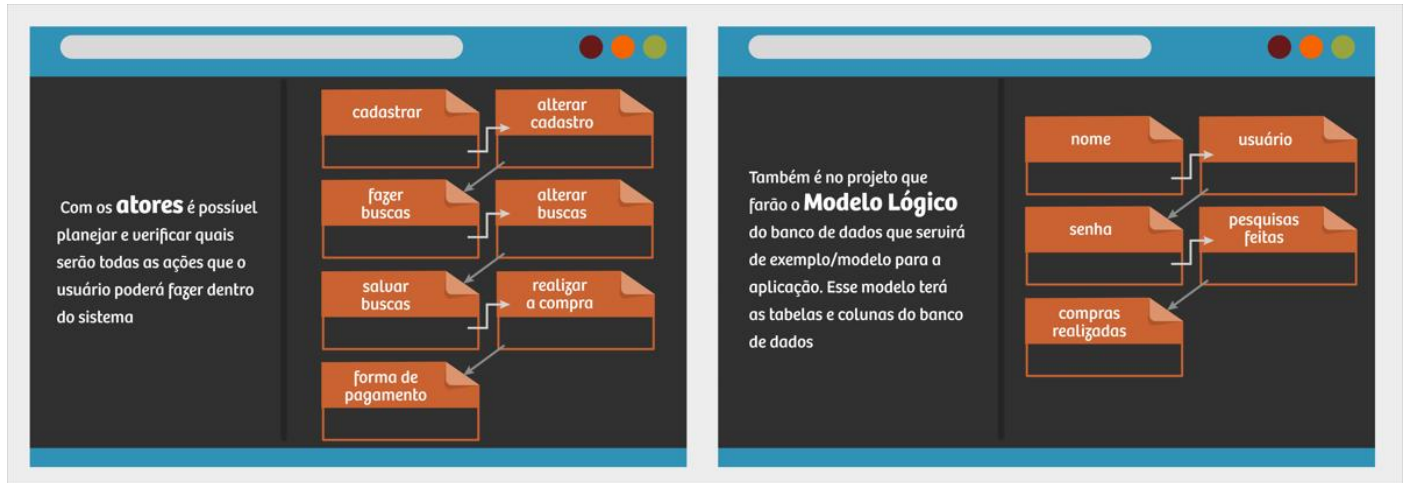
- A) Define a relação entre o usuário e a aplicação.
- B) Define qual o tipo de banco de dados a ser utilizado na aplicação (MySQL, Postgree, Oracle, PL).
- C) Define as ações do usuário dentro de uma aplicação.
- D) Define a modelagem dos conceitos que devem ser armazenados em um banco de dados.
- E) Define o número de tabelas dentro de um banco de dados.

5) O que é o Modelo Lógico?

- A) Modelo de ações do usuário.
- B) Modelo de ações do desenvolvedor.
- C) Modelo que faz um mapeamento dos dados da solução escolhida para a aplicação.
- D) Modelo de relacionamento entre tabelas.
- E) Modelo de linguagem de programação.

Na prática

Marta e sua equipe foram contratados para desenvolver um sistema de compra de passagens aéreas online. Antes do desenvolvimento propriamente dito, é necessário que se faça um projeto em que irá se planejar tanto as ações dos usuários quanto as do administrador do sistema (atores).



Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Modelagem de Dados - Projeto de um Banco de Dados



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

HEUSER, C. A. Projeto de banco de dados - V4

Conteúdo interativo disponível na plataforma de ensino!

LARMAN, C. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento interativo

Conteúdo interativo disponível na plataforma de ensino!