Modelagem de Dados

Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico

Mapeamento Objeto Relacional

Responsável pelo Conteúdo:
Prof. Ms. Rafael Alencar Segura

Revisão Técnica:
Prof. Ms. Douglas Almendro

Revisão Textual:
Profa. Esp. Márcia Ota

UNIDADE Mapeamento Objeto Relacional

- Paradigma da Orientação a Objetos
- Programação Estruturada X Programação Orientada a Objetos
- Multiplicidades
- Classes Associativas

Objetivo de APRENDIZADO

Leia com atenção e pratique com os exemplos citados, desta forma será mais fácil entender o conteúdo.

Esta unidade trata sobre os bancos de dados e modelos orientados a objetos. Por isso, sugiro a leitura dos artigos existentes no item material complementar

Esta unidade trata sobre os bancos de dados e modelos orientados a objetos. Por isso, sugiro a leitura dos artigos existentes no item material complementar.

Lembramos a você da importância de realizar todas as atividades propostas dentro do prazo estabelecido para cada Unidade. Dessa forma, evitará que o conteúdo se acumule, trazendo problemas ao final do semestre.

Uma última recomendação: caso tenha problemas para acessar algum item da disciplina, ou dúvidas relacionadas ao conteúdo, não deixe de entrar em contato com seu professor tutor através do botão mensagens.

Contextualização

É hora de trabalharmos o paradigma da orientação a objetos, mas voltada para projetos de banco de dados. Estes tipos de SGBD são considerados recentes, muito embora o SGBD relacional ainda tenha uma boa aceitação no mercado de trabalho. Atualmente, as novas versões dos bancos de dados comerciais são Objeto Relacional, que significa que reúnem as melhores características dos 2 modelos de banco de dados.

Paradigma da Orientação a Objetos

Este paradigma parte do princípio que, no mundo em que vivemos, existem diversos objetos, os quais possuem características e realizam ações. Por exemplo, um automóvel é um objeto que possui características: cor, ano modelo e marca. E realiza algumas ações, acelerar e frear, por exemplo.

Contextualizando:

Chris precisa enviar flores para o amigo Robin que mora em outra cidade...

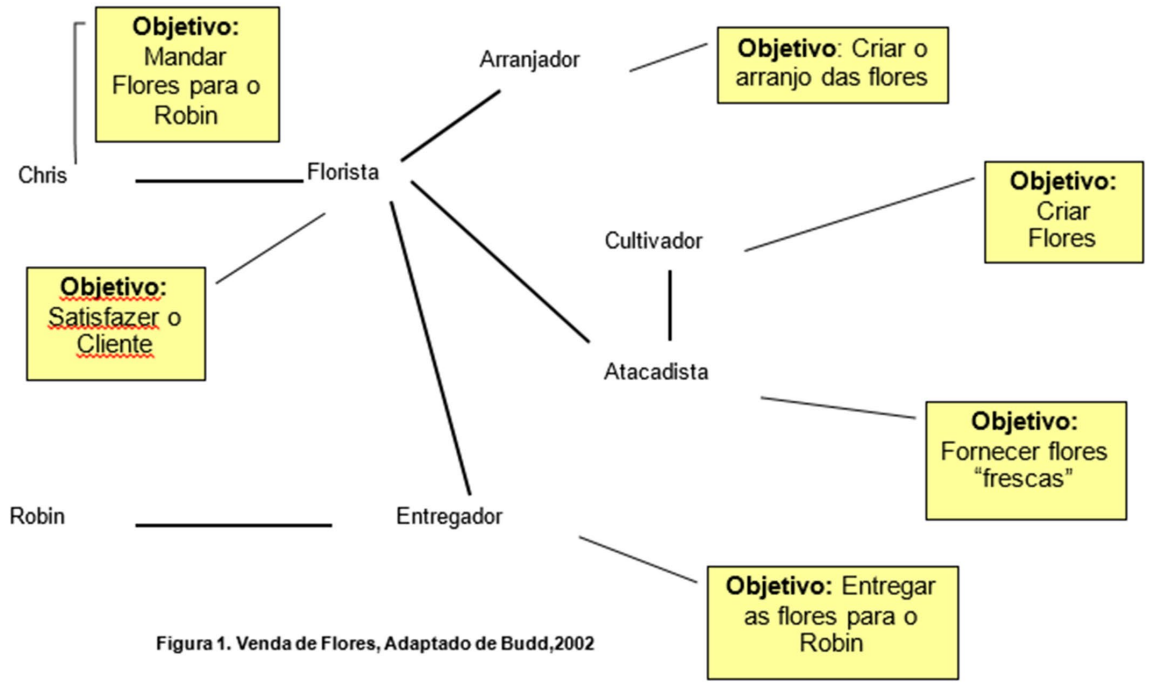


Figura 1. Venda de Flores, Adaptado de Budd,2002

Este cenário apresenta na prática alguns conceitos da orientação a objetos, conforme abaixo:

Agentes e Comunidades:

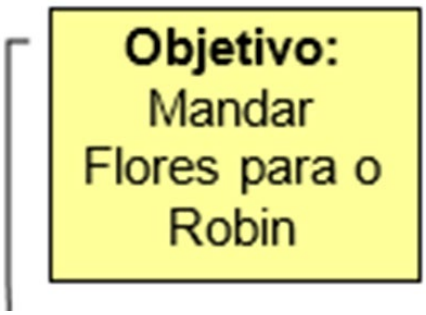


Figura 2. Agentes e Comunidades.

1. Chris passa uma mensagem para o florista contendo uma requisição (entregar flores para Robin em outra cidade).
2. É responsabilidade do florista satisfazer esta requisição.
3. Chris não sabe o método que o florista irá utilizar para entregar flores em outra cidade e, também, desconhece os detalhes desta operação.
4. A Programação Orientada a Objetos é estruturada como uma comunidade de agentes que interagem chamados de objetos.
5. Cada objeto tem um papel a ser executado.
6. Cada objeto fornece serviços ou executam ações que podem ser utilizadas por outros membros da comunidade.

Mensagens e métodos:

1. Uma ação é a transmissão de uma mensagem para um agente (objeto) responsável pela ação.
2. Quando um objeto aceita a mensagem, ele é responsável por tratar a ação.
3. Em resposta a esta mensagem, o receptor executará algum método para atender a solicitação.

Responsabilidade

Descreve o comportamento em termos de responsabilidades.

Classes e Instâncias

O Florista é um exemplo de classe, pois representa todos os floristas. O Florista Pitoco e o Bodjo são instâncias desta classe.

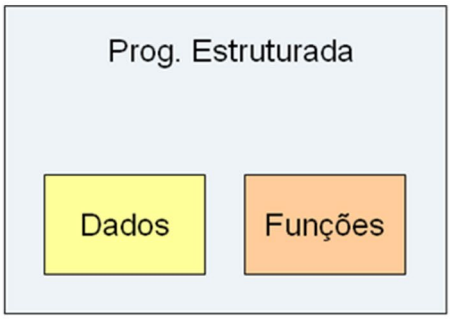
Outro exemplo de classes e instâncias é forma de gelatina e as gelatinas.

Objetivo: Criar o arranjo das flores.

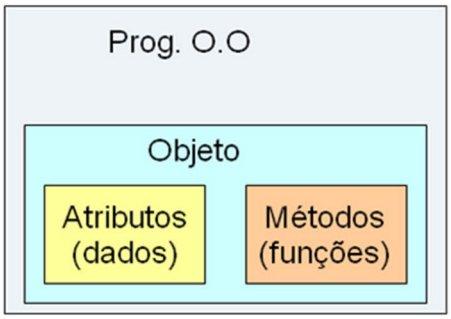
Resumindo:

Objetivo: Satisfazer o Cliente.

Programação Estruturada X Programação Orientada a Objetos



- Estruturado:
  - Ênfase nos procedimentos, implementados em blocos estruturados, com comunicação entre procedimentos por passagem de dados;



- Orientado a Objetos:
  - Dados e procedimentos fazem parte de um só elemento básico (objeto). Os elementos básicos comunicam-se entre si, caracterizando a execução do programa à Dados e procedimentos agrupados em um só elemento.

Figura 3. POO X Programação Estruturada.

UML – Unified Modeling Language

- É uma linguagem de modelagem unificada e visual usada para modelar sistemas orientados a objetos;
- Possui diversos diagramas tais, como:
  - classe;
  - sequência;
  - colaboração;
  - caso de uso;
  - transição de estados;
  - componentes.

Utilizaremos para modelagem de banco de dados, o diagrama de classes; pois, a partir dele, podemos gerar o modelo relacional, conforme apresentaremos no decorrer deste documento.

Classe

O que é uma Classe?

> "Uma classe é uma entidade que descreve um conjunto de objetos com propriedades e comportamentos semelhantes e com relacionamentos comuns com outros objetos".

Uma classe é dividida em três partes, conforme abaixo:

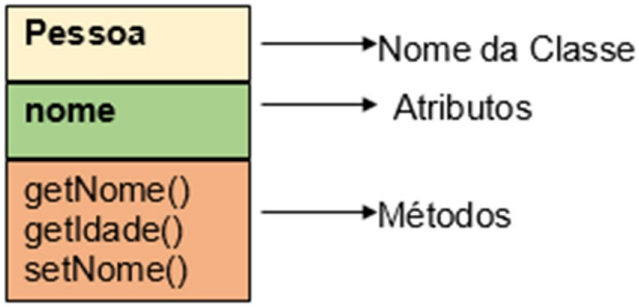
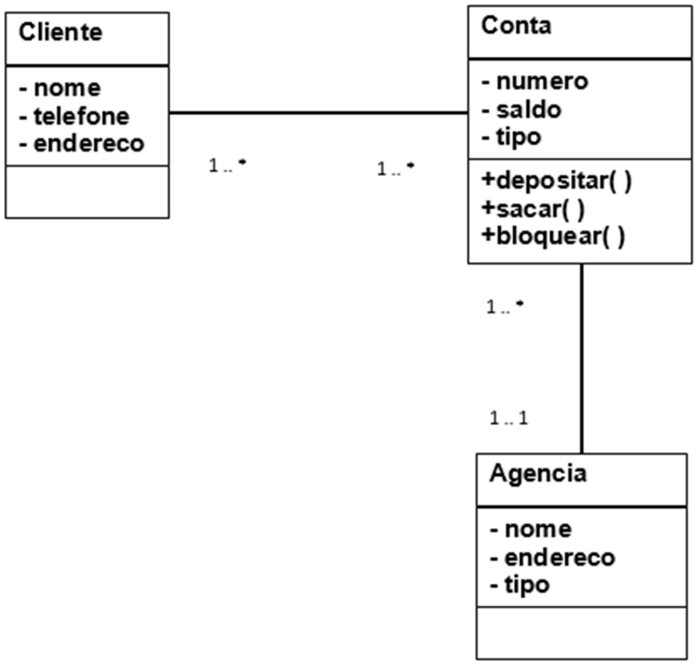


Figura 4. Exemplo da estrutura de uma Classe.

Diagrama de Classe

É um dos principais diagramas para um sistema orientado a objetos. Um diagrama de classes apresenta a estrutura e as relações entre as classes do sistema.



Informação

OBS: Todos os objetos são instâncias de uma classe.

Figura 5. Diagrama de Classes.

Multiplicidades

As multiplicidades, utilizadas nos diagramas de classe, são bem próximas dos conceitos das cardinalidades do Diagrama de Entidade Relacionamento. No diagrama de classes, representa-se a quantidade de objetos, aos quais outro objeto pode estar associado.

Nome	Simbologia
Apenas um	1
Zero ou muitos	0..*
Um ou muitos	1..*
Zero ou um	0..1

Tabela 1. Multiplicidades.

Mapeamento Objeto Relacional

Informação

OBS: Lembro que no diagrama de classes não temos um atributo identificador, pois na hora em que você instancia um objeto, ele automaticamente possui um identificador único [internamente]. Porém, na hora de realizarmos o mapeamento para o modelo relacional, existe a necessidade de criarmos um atributo identificador, ou seja, a chave primária.

Associações um para um

Neste caso, departamento recebe a chave estrangeira, código que referencia a tabela empregada.

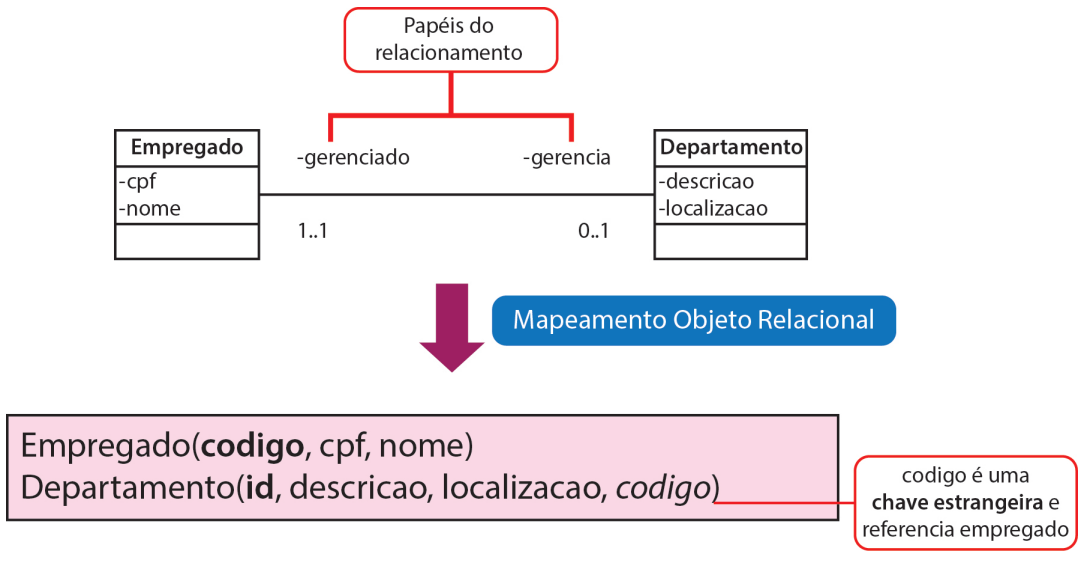


Figura 6. Mapeamento objeto relacional: um para um.

Associações muitos para muitos

Neste tipo de associação, é gerada a tabela associativa Loca, onde a chave primária é composta.

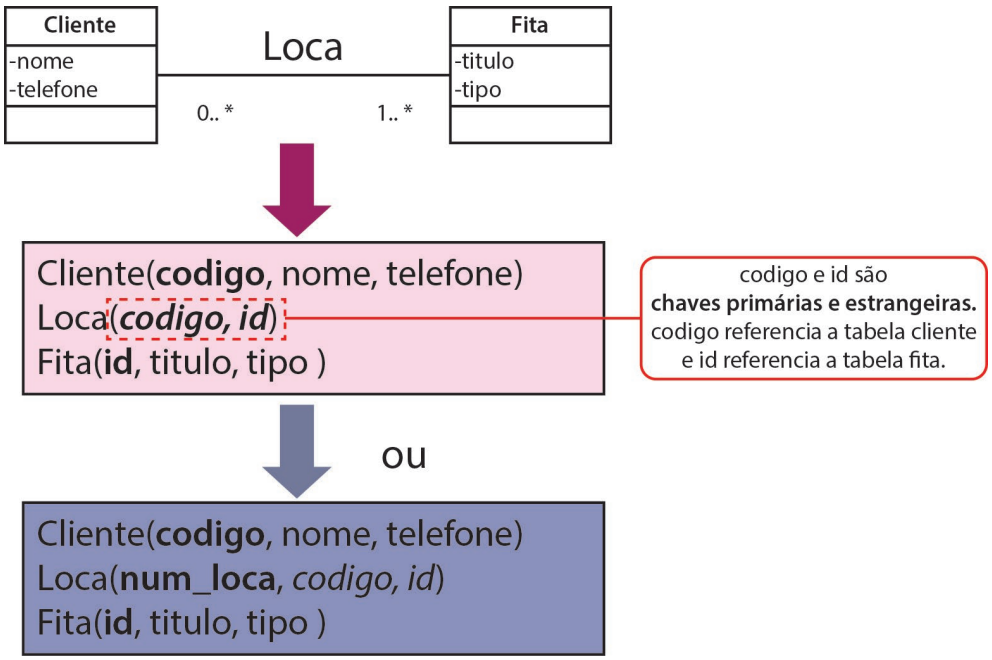


Figura 7. Mapeamento objeto relacional: muitos para muitos.

Associações um para muitos

Como podemos observar para representar o relacionamento um-para-muitos, na classe Dependente, a classe de multiplicidade muitos, devemos adicionar um atributo do tipo da classe Cliente (_cliente), a classe de multiplicidade um, e os correspondentes métodos gets/sets para os atributos.

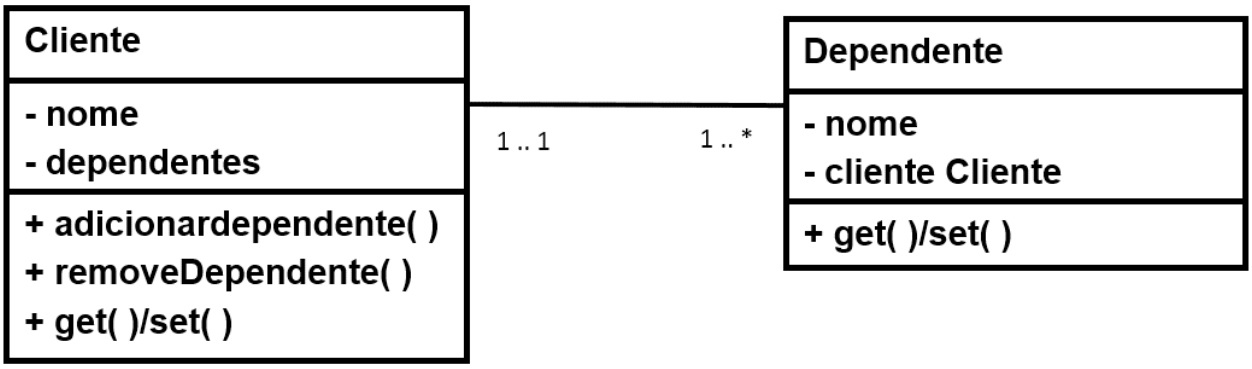


Figura 8. Mapeamento objeto relacional: um para muitos.

Na classe Cliente, a classe de multiplicidade um, devemos ter um atributo do tipo dependentes, que represente uma coleção de itens de Dependentes

Dessa forma estamos representando a associação entre as classes Cliente e Dependente.

Associações Reflexivas

É duplicada a chave primária, codigo, com outro nome codigo_ supervisor para implementar a associação reflexiva, ou seja, uma classe que se autorrelaciona.

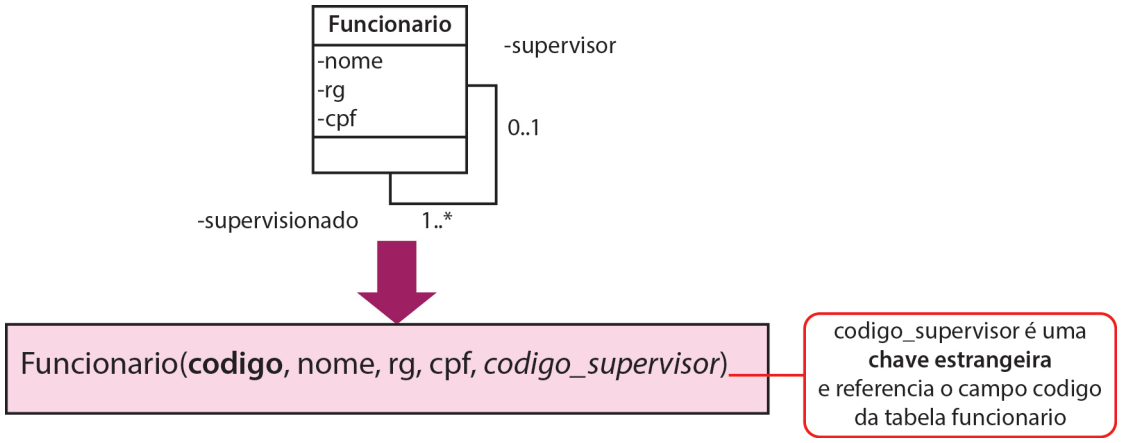


Figura 9. Mapeamento Associações reflexivas.

Associações Ternárias

Neste mapeamento, uma das alternativas de implementação é criar uma chave primária para entrega e implementar para cada classe envolvida na associação, uma chave estrangeira, conforme apresentado abaixo.

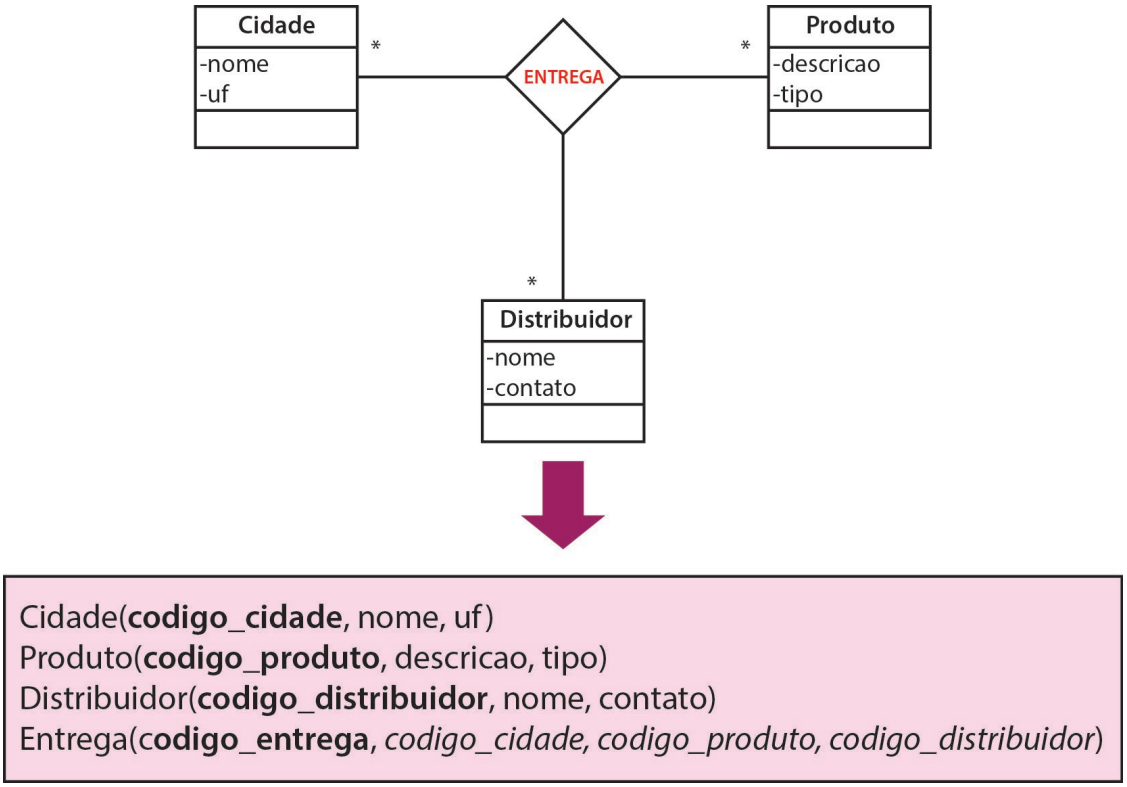
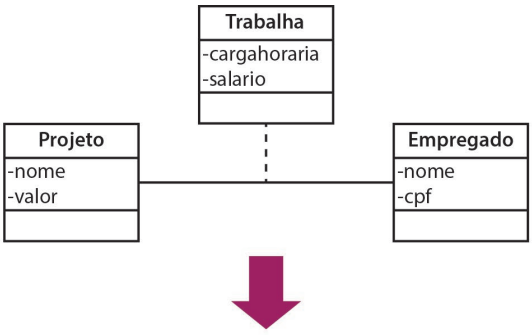


Figura 10. Mapeamento objeto relacional: associação ternária.

Classes Associativas

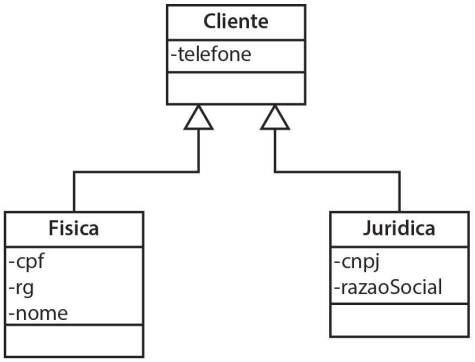
É possível termos classes associativas, mesmo que a multiplicidade não seja muitos para muitos. Como mapear? O exemplo abaixo contempla este cenário. A classe associativa Trabalha vira uma tabela e recebe as chaves estrangeiras das classes envolvidas na associação.



Projeto(**id_projeto**, nome, valor)
Empregado(**id_empregado**, nome, cpf)
Trabalha(**codigo_trabalha**, *id_projeto*, *id_empregado*, cargahoraria, salario)

Generalização

As três formas possuem vantagens e desvantagens. Por isso, é uma questão da equipe de desenvolvimento definir qual a forma a ser implantada.



1ª Forma
Cliente(**id**, telefone, cpf, rg, nome, cnpj, razaoSocial)

2ª Forma
Fisica(**id**, telefone, cpf, rg, nome)
Juridica(**id**, telefone, cnpj, razaoSocial)

3ª Forma
Cliente(**id**, endereco)
Fisica(**id**, cpf, rg, nome, *id_cliente*)
Juridica(**id**, cnpj, razaoSocial, *id_cliente*)

Figura 12. Mapeamento objeto relacional: Generalização.

Material Complementar

Como complemento desta unidade, sugiro a leitura de alguns artigos:

- **Artigo:** Bancos de Dados Orientados a Objetos
  http://homes.dcc.ufba.br/~jteles/artigoODBMS.pdf.

- **Artigo:** Sistemas de Bancos de Dados Orientados a Objetos
  http://www.fsma.edu.br/si/edicao3/banco_de_dados_orientado_a_objetos.pdf.

- **Caché** - Um SGBDOO com toda tecnologia em banco de dados Orientados a Objetos
  http://www.linhadecodigo.com.br/Artigo.aspx?id=2212.

Referências

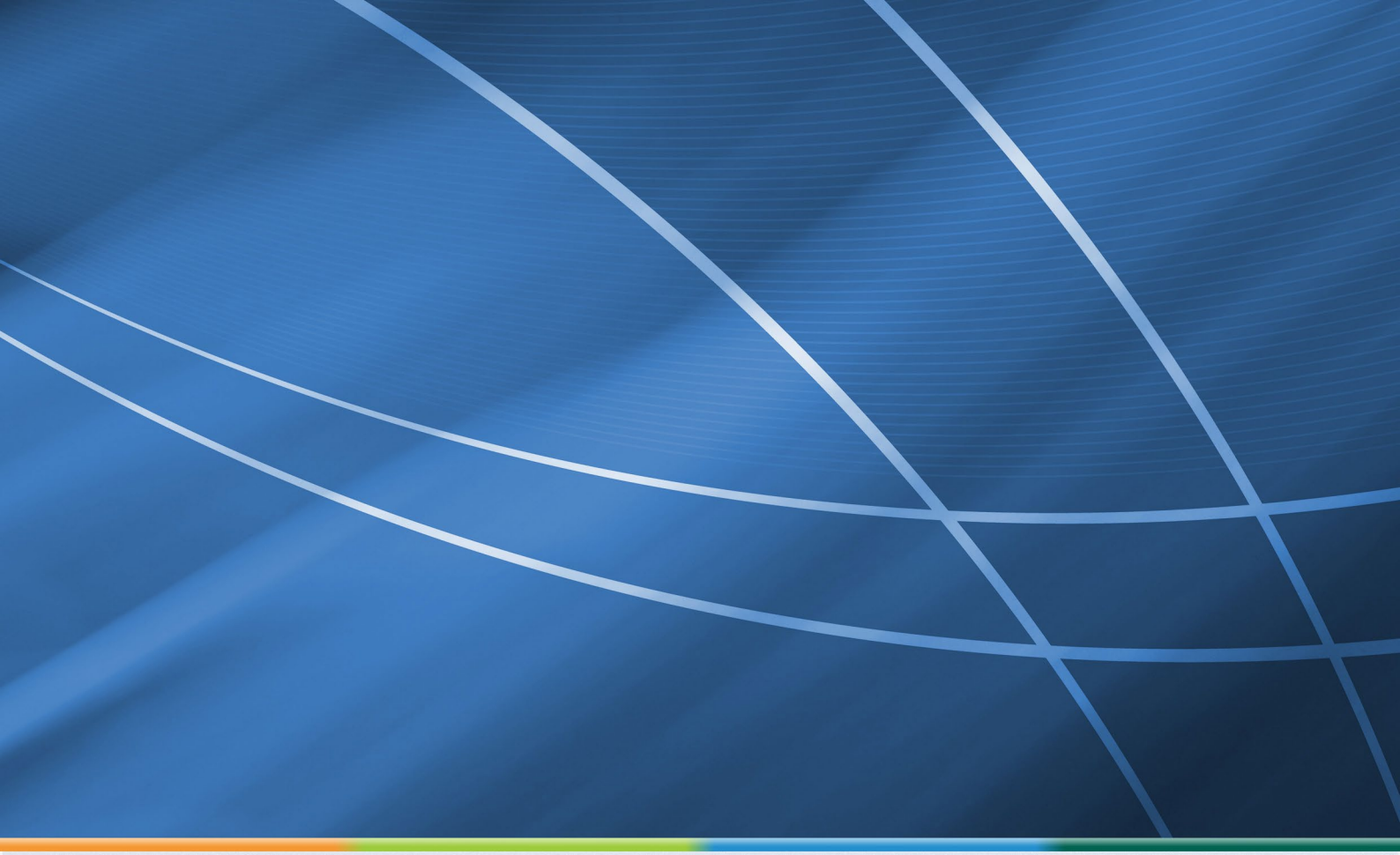
Bezerra, Eduardo. **Princípios de análise e projeto de sistemas com UML.** Rio de Janeiro: Elsevier, 2007.

Timothy Budd, **Introduction to Object-Oriented Programming,** An (3rd Edition) Addison Wesley, 2002. by.

Anotações


Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo SP Brasil
Tel: (55 11) 3385-3000



Universidade Cruzeiro do Sul

UNICID Universidade Cidade de S. Paulo

UNIFRAN Universidade de Franca

UDF Centro Universitário

Módulo Centro Universitário