

# O PARADIGMA DA ORIENTAÇÃO A OBJETOS, A LINGUAGEM UNIFICADA DE MODELAGEM (UML) E A ORGANIZAÇÃO E REPRESENTAÇÃO DO CONHECIMENTO: UM ESTUDO DE CASO DE UM SISTEMA PARA BIBLIOTECAS

## EL PARADIGMA DE LA ORIENTACIÓN A OBJETOS, EL LENGUAJE DE MODELADO UNIFICADO (UML) Y LA ORGANIZACIÓN Y REPRESENTACIÓN DE CONOCIMIENTO: UN ESTUDIO DE CASO DE UN SISTEMA DE BIBLIOTECAS

**Fernanda Regebe Castro** - fernandaregebe@gmail.com  
Mestre em Modelagem Computacional pelo Centro de Pesquisa E Pós Graduação Fundação Visconde Cairu (CEPPEV). Docente do Instituto Federal da Bahia

**Fábio Marques da Cruz** - fabiomacz@gmail.com  
Mestre em Ciência da Informação pela Universidade Federal da Bahia (UFBA). Docente do Instituto Federal da Bahia.

**Nanci Elizabeth Oddone** - neoddone@gmail.com  
Pós Doutora pela Universidade de Kent (Reino Unido). Docente da Universidade Federal do Estado do Rio de Janeiro (UNIRIO).

### RESUMO

**Introdução:** A orientação a objetos trabalha com uma notação própria e possui alguns conceitos centrais. Dentre os mais importantes estão as classes, os atributos, os relacionamentos e a herança. Já a linguagem UML possui diagramas próprios.

**Objetivo:** Demonstrar a metodologia da orientação a objetos e a linguagem de modelagem UML, com seus diversos diagramas.

**Metodologia:** Para ilustrar o funcionamento dos diagramas desenvolve-se um estudo de caso envolvendo um sistema de empréstimo de livros em biblioteca.

**Resultados:** Constatou-se que há influências das teorias da classificação, especialmente a de Ranganathan, na área de modelagem de sistemas de informação, demonstrando que tudo o que hoje se desenvolve na informática, mais precisamente na área de desenvolvimento de sistemas, já foi realizado antes, de forma manual.

**Conclusões:** Nesse trabalho foi confirmado que para entender o complexo, antes de tudo é preciso dividi-lo em partes e classificá-lo.

**Palavras-chave:** Modelagem de sistemas, Orientação a objetos, UML, Classificação facetada, Ranganathan.

## 1 INTRODUÇÃO

Atualmente existe consenso em torno da ideia de que estamos vivendo em uma sociedade da informação e do conhecimento. Hoje, organizações competem em um mercado que é global, caracterizado por oportunidades e riscos, no interior do qual as mudanças e os avanços tecnológicos envolvem uma competição cada vez maior.

Para qualquer organização sobreviver em tempos de globalização e concorrência acirrada é necessário processar dados de forma automatizada, ou seja, informatizar suas rotinas e processos para que sirvam de suporte nos decisivos momentos da tomada de decisões. Para isso é necessário transcrever e transferir o ambiente organizacional, tal como ele funciona no cotidiano, para sistemas de computadores. Isto é feito criando rotinas automatizadas que repliquem as práticas de forma a alcançar agilidade e precisão nos processos organizacionais, sejam eles de qualquer natureza.

Muitos sistemas organizacionais modernos têm sido construídos através do paradigma da orientação a objetos (OO). A organização e a representação dos dados são feitas através da linguagem UML (Unified Modeling Language), que é uma linguagem visual utilizada para modelar *softwares* no paradigma da OO. Para Oliveira (2011, p. 68):

[...] a orientação a objetos surgiu com a necessidade de se criar um paradigma de programação simples, baseado na percepção humana dos objetos ao seu redor. Este novo paradigma não é apenas um modo de programar, mas uma maneira de pensar e conceber as idéias.

Ainda de acordo com os autores,

[...] a utilização do paradigma orientado a objetos está presente em todas as etapas do processo de desenvolvimento [de sistemas], indo da análise, passando pelo projeto, até as fases de teste e implementação. É um fator decisivo para a construção de sistemas, visando alcançar os objetivos esperados no determinado espaço de tempo com determinado custo (OLIVEIRA, 2011, p. 70).

A orientação a objetos trabalha com uma notação própria e possui alguns conceitos centrais que serão esclarecidos neste artigo. Dentre os mais importantes estão as classes, os atributos, os relacionamentos e a herança. Já a linguagem UML possui diagramas próprios que também serão tratados neste artigo. Examinaremos especificamente os diagramas de caso de uso, de classe, de objetos e de atividades.

Após demonstrar a teoria da orientação a objetos e a linguagem de modelagem de dados UML, o presente artigo mostrará o caso prático de modelagem de um sistema simplificado de uma biblioteca, na forma de estudo de caso. Na ocasião, os diagramas serão descritos com detalhes para um melhor entendimento.

## **2 ORIENTAÇÃO A OBJETOS**

O paradigma da orientação a objetos é um “processo conceitual independente da linguagem de programação [...]. O desenvolvimento baseado em objetos é fundamentalmente uma forma de pensar e não uma técnica de programação” (RUMBAUGH et al., 1994, p. 5).

### **2.1 Abstração**

Abstração é a habilidade de esconder detalhes e concentrar-se no geral quando se avalia as propriedades comuns a um conjunto de objetos. Na atividade de modelagem de dados, a habilidade de abstrair é essencial e deve ser exercida de maneira sistemática. De acordo com Campos (2001, p. 103, grifo nosso),

[...] a abstração consiste na seleção que um analista faz de alguns aspectos de um assunto em restrição a outros considerados

irrelevantes para o contexto em questão. É um princípio geral, que podemos afirmar que é a própria essência do ato de *classificar*.

A definição apresentada por Rumbaugh et al. (1994, p. 9) diz que

A abstração consiste na concentração dos aspectos essenciais, próprios, de uma entidade e em ignorar suas propriedades acidentais. No desenvolvimento de sistemas, isso significa concentrar-se no que um objeto é e faz, antes de decidir como ele deve ser implementado. O uso da abstração preserva a liberdade de se tomar decisões evitando, tanto quanto possível, comprometimentos prematuros com detalhes.

Em outra definição, Oliveira e Amaral (2001) afirmam que abstração é

[...] um conceito no qual não se leva em conta um valor específico determinado e sim qualquer entre todos os valores possíveis daquilo com que estamos lidando ou ao que estamos nos referindo. Por exemplo, em álgebra, quando dizemos que  $x$  é uma variável, desconsideramos o seu valor atual, mas consideramos todos os possíveis valores de  $x$  como sendo números, os quais não são objetos físicos e sim objetos linguísticos, formados pela abstração durante o ato de contar.

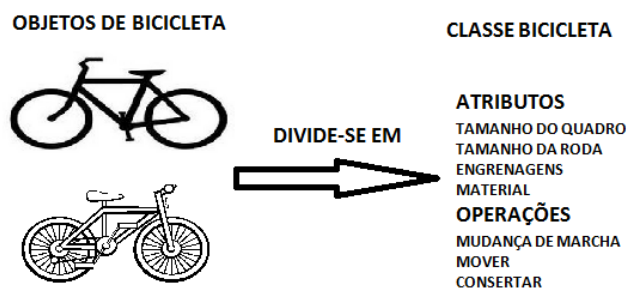
Para exemplificar podemos usar um exemplo prático. Quando o objetivo for desenvolver um sistema de informação escolar, deve-se pensar nos objetos 'alunos', 'professores', 'disciplinas', no ato de fazer matrícula, no ato de trancar matrícula, etc., sem levar em conta, por exemplo, o perfil psicológico dos alunos, que poderiam vir a ser parte de outro sistema.

## 2.2 Classes e Objetos

Para Rumbaugh et al. (1994, p.3), quando se fala em 'classificar'

[...] significa que os objetos com a mesma estrutura de dados (atributos) e o mesmo comportamento (operações) são agrupados em uma classe. [...] Uma classe é uma abstração que descreve propriedades importantes para uma aplicação e ignora o restante. [...] Cada classe descreve um conjunto possivelmente infinito de objetos individuais. Cada objeto é dito ser uma instância de sua classe. Cada instância da classe tem seu próprio valor para cada atributo, mas compartilha os nomes de atributos e operações com outras instâncias da mesma classe.

**Figura 1 – Objetos e Classes**

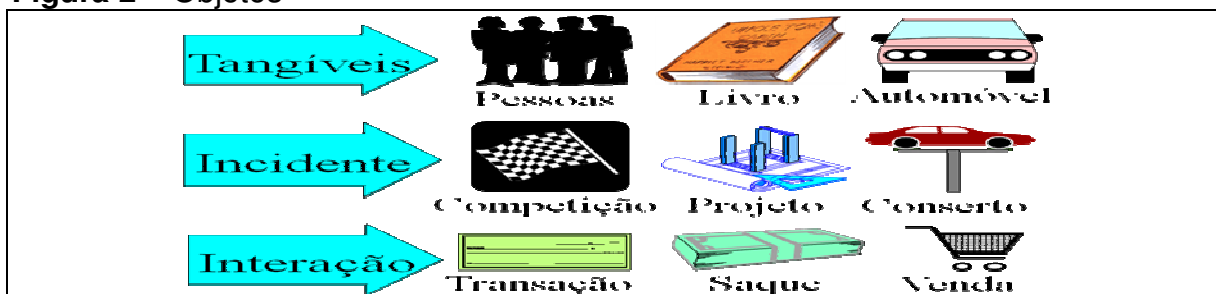


Fonte: Adaptado de Rumbaugh et al. (1994, p. 3)

No entender de Puga (2004, p. 36),

[...] Um objeto é uma extensão do conceito de objeto do mundo real em que se podem ter coisas tangíveis, um incidente (evento ou ocorrência) ou uma interação (transação ou contrato). [...] Por exemplo, em um sistema acadêmico em que João é um aluno objeto e Carlos é um professor objeto que ministra aulas objeto da disciplina objeto Algoritmos, para que João possa assistir às aulas de Algoritmos do professor Carlos, ele deve fazer uma matrícula objeto no curso objeto Computação.

**Figura 2 – Objetos**

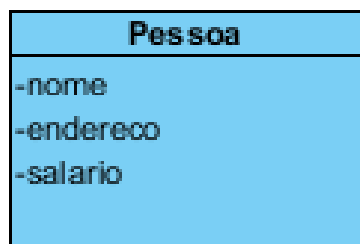


Fonte: Puga (2004, p. 36)

### 2.3 Atributos

São elementos que agregam informações aos objetos, representando qualidades, estados, características, enfim, propriedades do objeto. O objeto 'pessoa' (Figura 3) pode ter como atributos, por exemplo, o nome, o endereço e o salário. Os atributos e seus valores, juntos, descrevem as instâncias de um objeto. Cada atributo tem um valor para cada instância de um objeto, ou seja, o atributo 'salário' tem o valor 'R\$ 2.000' para o objeto 'André'. Assim, conclui-se que André ganha 2 mil reais.

**Figura 3** – Objeto ‘pessoa’ e seus atributos



Fonte: Elaborado pelos autores

Além disso, em objetos de uma mesma classe os valores dos atributos podem ser iguais ou diferentes, e duas classes diferentes podem ter atributos com o mesmo nome:

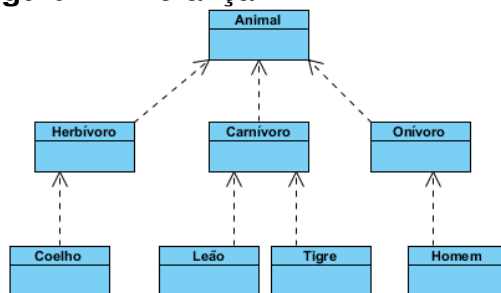
[...] diferentes instâncias de objetos podem ter valores iguais ou diferentes para um dado atributo. Cada nome de atributo é único dentro de uma classe (o que não é verdadeiro quando se consideram todas as classes). Dessa maneira, as classes pessoa e empresa podem ter simultaneamente um atributo chamado endereço. (RUMBAUGH et al., 1994, p. 34).

## 2.4 Herança

Conforme definição de Rumbaugh et al. (1994, p. 4), Herança é

[...] o compartilhamento de atributos e operações entre classes, com base em um relacionamento hierárquico. Uma classe pode ser definida de forma abrangente e depois refinada em sucessivas subclasses mais definidas. Cada subclasse incorpora, ou herda, todas as propriedades de sua superclasse e acrescenta suas próprias e exclusivas características. As propriedades da superclasse não precisam ser repetidas em cada subclasse.

Figura 4 – Herança



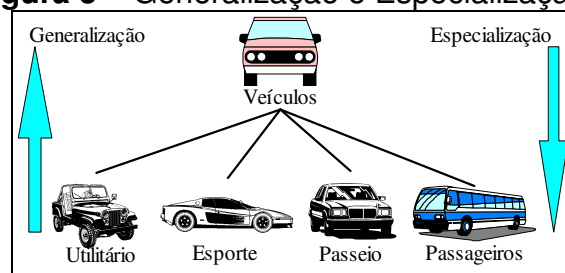
Fonte: Elaborado pelos autores

## 2.5 Generalização e Especialização

Os conceitos de Generalização e Especialização podem ser descritos da seguinte maneira:

**Generalização** é um mecanismo baseado na ligação semântica (hierárquica), onde as diferenças entre elementos similares são ignoradas para formar um tipo de alto nível de abstração, no qual as semelhanças são enfatizadas. O processo inverso é chamado de **especialização**. A generalização tem como principais objetivos representar relacionamentos semânticos do tipo “é um” entre objetos com alto nível de abstração, evitar a repetição de propriedades comuns em cada classe de objetos e possibilitar a definição de classes por refinamento das classes mais gerais (CAMPOS, 2001, p. 106).

Figura 5 – Generalização e Especialização



Fonte: Puga (2004, p. 38)

Em outra definição, Booch, Rumbaugh e Jacobson (2005, p. 66) defende que:

[...] uma generalização é um relacionamento entre itens gerais (chamados superclasses ou classes mãe) e tipos mais específicos desses itens (chamados subclasses ou classes filhas). Muitas vezes, as generalizações são chamadas relacionamentos “é um tipo de”: um

item, como a classe Janela Oval é um tipo de item mais geral (por exemplo, a classe Janela).

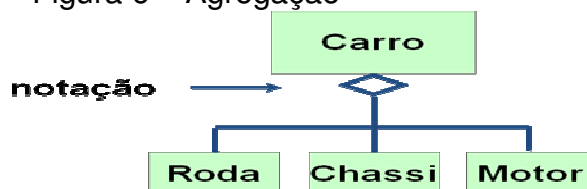
Pilone e Pitman (2005, p. 28, tradução nossa) por outro lado, afirmam que: “[...] a generalização é usada para expressar o que há de comum em diferentes classes. Por exemplo, se existe uma classe gato e uma classe cachorro, pode-se criar uma generalização de ambas as classes, chamada animal.”

## 2.6 Agregação

A agregação se caracteriza pelo

[...] relacionamento “parte todo” ou “uma parte de”, na qual objetos que representam componentes de alguma coisa são associados a um objeto que representa a estrutura inteira. Em termos gerais, uma agregação é uma forma especial de associação utilizada para mostrar um objeto composto, pelo menos em parte, de outro em uma relação todo/parte, por exemplo, a roda é parte de carro ou, usando a associação inversa, *carro tem roda* (CAMPOS, 2001, p. 106).

Figura 6 – Agregação



Fonte: Elaborado pelos autores

## 2.7 Associação

De acordo com Rumbaugh et al. (1994, p. 39), “ligações e associações são os meios para estabelecermos relacionamentos entre objetos e classes.” Já para Booch, Rumbaugh e Jacobson (2005, p. 69), uma Associação representa:

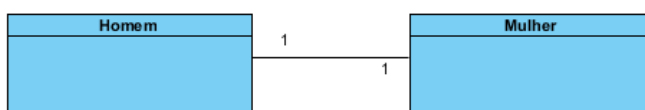
[...] um relacionamento estrutural existente entre objetos. Em muitas situações de modelagem, é importante determinar a quantidade de objetos que podem ser conectados pela instância de uma associação. Essa “quantidade” é chamada de multiplicidade do papel de uma associação. Ela representa o intervalo de inteiros que especifica o tamanho possível do conjunto de objetos relacionais. É escrita como uma expressão com valores mínimos e máximos, que



podem ser iguais; dois pontos são usados para separar os valores mínimo e máximo. [...] O número de objetos deve estar no intervalo dado. Pode ser apresentado uma multiplicidade de exatamente um (1), zero ou um (0..1), muitos (0..\*) ou um ou mais (1..\*).

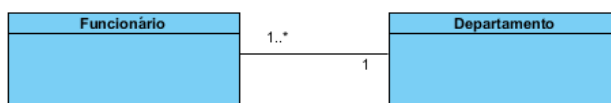
Para melhor ilustrar as associações observem-se as Figuras 7 a 10 abaixo que ilustram graficamente o conceito de Associação:

Figura 7 – Associação ilustrando que um homem é casado com uma mulher e que uma mulher é casada com um homem



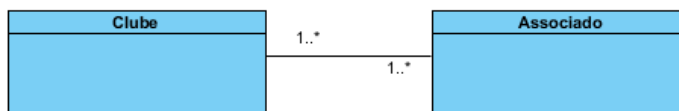
Fonte: Elaborado pelos autores

Figura 8 – Associação ilustrando que um funcionário trabalha em um departamento que pode possuir um ou mais funcionários



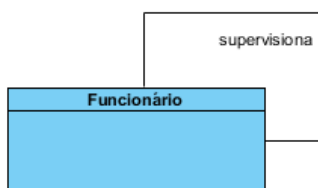
Fonte: Elaborado pelos autores

Figura 9 – Associação ilustrando que um clube pode ter de um a vários associados e que um associado pode ser sócio de um a vários clubes



Fonte: Elaborado pelos autores

Figura 10 – Associação ilustrando que um funcionário pode supervisionar de zero a vários funcionários, mas é supervisionado por um único colega



Fonte: Elaborado pelos autores

## 2.8 Operações e Métodos

De acordo com Melo (2008, p. 21), Operações e Métodos se aplicam aos objetos e ao seu comportamento:

[...] o comportamento dos objetos é representado pelas operações. Contudo, a operação para um objeto representa apenas a definição do serviço que ele oferece a outras estruturas. Quando tratamos da implementação dessa operação, ou seja, da sua representação em código, estamos nos referindo ao seu método.

## 2.9 Encapsulamento

De acordo com Oliveira (2011, p. 69), o encapsulamento “consiste em evidenciar ou não as funcionalidades do objeto ao ambiente externo”. Os autores ilustram o encapsulamento com o exemplo do motor de um carro. O exemplo começa mostrando que o motor de um carro é um objeto acionado através do método da ignição e que

[...] este método interage com outros métodos internos, como acender vela de ignição para prover a funcionalidade de ligar o carro. O método responsável por acionar a vela de ignição não está visível ao usuário do carro. Não é ele quem decide quando a vela irá acender ou não. Esse método é interno, pois a entidade usuário do carro, que interage com o carro, não interage com o motor (OLIVEIRA, 2011, p. 69).

Outro exemplo mostrado por estes autores a respeito do conceito de encapsulamento é a ação de ligar uma TV:

[...] não é interessante para o telespectador saber se o diodo da televisão foi acionado, ou se o circuito interno está respondendo ao determinado componente. Essa parte deve ficar transparente para as entidades que interagem com o objeto, sendo visíveis apenas para os métodos que realmente importam na comunicação, como ligar a TV ou ligar um canal (OLIVEIRA, 2011, p. 69).

## 3 LINGUAGEM UML

No que se refere à UML (*Unified Modeling Language*), que é uma linguagem visual utilizada para modelar softwares no paradigma da OO, Guedes (2011, p. 19) afirma que,

A UML – Unified Modeling Language ou Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral, que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de software. Deve ficar bem claro, porém, que a UML não é uma linguagem de programação e sim uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de software a definirem as características do sistema, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema será implantado.

A Linguagem Unificada de Modelagem (UML) trabalha a representação de ideias, entidades e ações em padrões classificatórios. A modelagem torna-se necessária, porque a atividade de desenvolvimento de sistemas é bastante complexa, isso sem falar que os sistemas geralmente envolvem grandes volumes de itens e são compostos por muitos módulos. Dessa forma só se deve começar a desenvolver um sistema depois de uma definição dos requisitos e das funcionalidades que o sistema deve possuir. Modelar-se então esse conjunto de dados e informações através de uma linguagem de representação; os diagramas resultantes servem de meio de comunicação entre analistas e usuários e entre os próprios membros das equipes de analistas e programadores.

A linguagem UML, ambiente no qual são usados os conceitos da metodologia da OO para modelar sistemas, constitui-se por diversos tipos de diagramas. Iremos ver neste artigo os diagramas de caso de uso, diagramas de classe, diagramas de objetos e diagramas de atividades.

### **3.4 Diagrama de Casos de Usos**

O Diagrama de Casos de Uso é

[...] o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma idéia geral de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas ou



**Figura 12** – Lista de Casos de Uso de Sistema Gestor de Papelaria

LISTA DE CASOS DE USO	
<b>Ator:</b> Departamento de estoque	
- Manter Produto	- Manter Fabricante
- Manter Tipo de Produto	- Manter Fornecedor
- Manter Unidade de Venda	- Manter Cor
- Registrar Entrada em Estoque	- Emitir etiquetas de vendas
- Gerar relatório de reposição	- Consultar Produtos e Preços
<b>Ator:</b> Vendedor	
- Consultar Produtos e Preços	- Registrar Vendas
<b>Ator:</b> Gerência	
- Manter Vendedor	- Registrar Troca e Devolução
- Consultar Faturamento	- Consultar Produtos e Preços
- Atualizar preços e lançar promoções de produtos	- Realizar retirada de capital do caixa
<b>Ator:</b> Supervisor	
- Cancelar item de venda	- Realizar retirada de capital do caixa
- Abrir o caixa	- Fechar o caixa
<b>Ator:</b> Caixa	
- Registrar pagamento de venda	- Inserir suprimento de capital

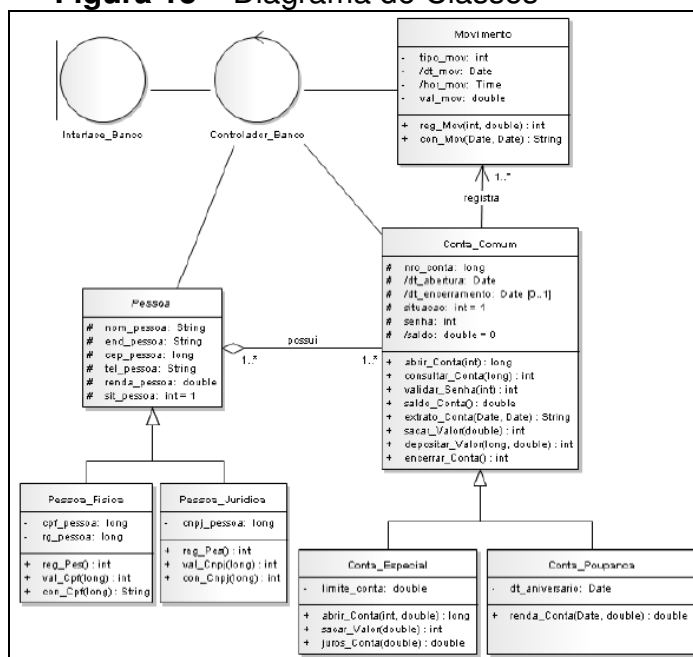
Fonte: Melo (2009, p. 20)

### 3.1 Diagrama de Classes

Guedes (2011, p. 31) nos oferece a seguinte definição para Diagrama de Classes:

[...] diagrama de classes é provavelmente o mais utilizado e um dos mais importantes da UML. Serve de apoio para a maioria dos diagramas. Como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de esclarecer como as classes trocam informações entre si.

**Figura 13 – Diagrama de Classes**



Fonte: Guedes (2011, p. 32)

Na definição de Booch, Rumbaugh e Jacobson (2005, p. 108) vê-se que:

[...] um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Graficamente, um diagrama de classes é uma coleção de vértices e arcos. [...] Os diagramas de classes são utilizados para fazer a modelagem da visão estática de um sistema. Essa visão oferece principalmente suporte para os requisitos funcionais do sistema – os serviços que o sistema deverá oferecer aos usuários finais.

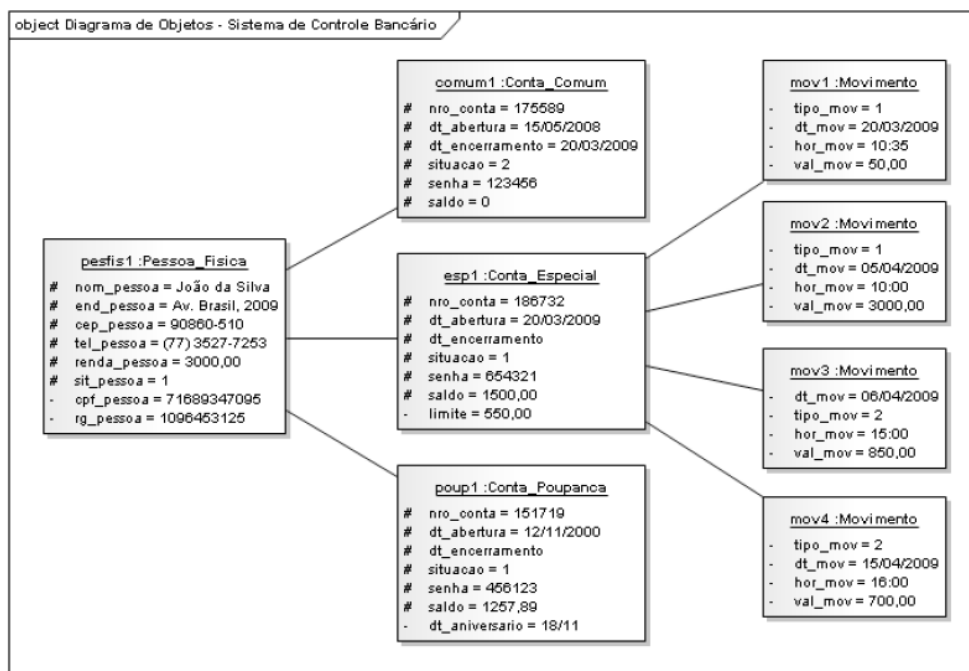
Entende-se por visão estática do sistema o conjunto de tabelas que contém os dados que o sistema necessita para funcionar. No diagrama de classes, as classes representam as tabelas que irão ser criadas no banco de dados. O diagrama de classes não tem relação com o banco de dados físico, ou seja, ao criá-lo, não é necessário saber se o sistema rodará em uma base de dados MS-SQL Server ou Oracle.

### 3.2 Diagrama de Objetos

Segundo Guedes (2011, p. 32), o Diagrama de Objetos está associado ao Diagrama de Classes.

Na verdade, o diagrama de objetos é praticamente um complemento do diagrama de classes e bastante dependente deste. O diagrama fornece uma visão dos valores armazenados pelos objetos de um diagrama de classes em um determinado momento da execução de um processo do software.

**Figura 14 – Diagrama de Objetos**



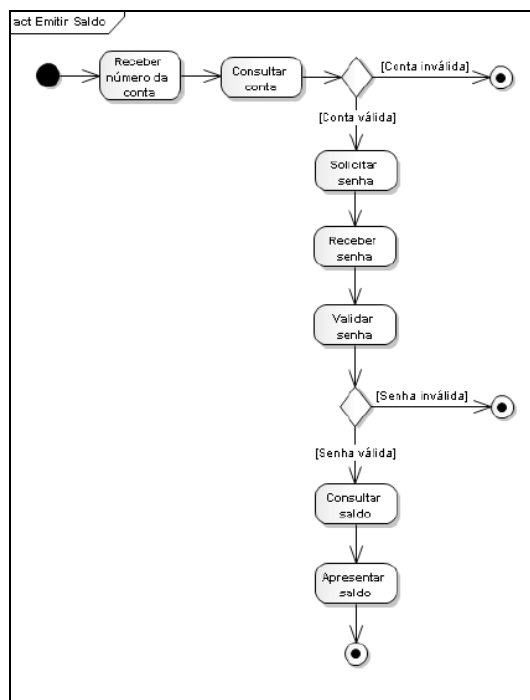
Fonte: Guedes (2011, p. 33)

### 3.3 Diagrama de Atividade

Para Guedes (2011, p. 36), o Diagrama de Atividade

[...] preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle de uma atividade.

**Figura 15** – Diagrama de Atividade



Fonte: Guedes (2011, p. 37)

#### **4 PROXIMIDADES E INFLUÊNCIAS ENTRE A CLASSIFICAÇÃO FACETADA E O PARADIGMA DA ORIENTAÇÃO A OBJETOS**

Como já foi dito anteriormente, o paradigma da orientação a objetos, que trabalha com a prática de dividir os objetos do mundo real em *classes*, agrupando os que têm características (atributos) em comum, se resume a uma *classificação*. Para Barbosa (1972, p. 73), “entre todas as linguagens existentes, a classificação ainda é considerada a melhor, por assegurar uma arrumação lógica dos assuntos dentro de uma coleção”. Lima (2004, p. 58), por sua vez, afirma que “o objetivo de toda classificação é estabelecer uma ordem ou organização das coisas e dos pensamentos”.

Os sistemas de informação lidam com coleções de dados de vários tipos e várias estruturas que precisam ser processadas. O computador nas empresas, em sistemas de automação de tarefas, não faz nada mais do que automatizar estas tarefas, ou seja, fazer o que o ser humano faria de forma mais rápida e sem erros causados pela fadiga. Numa escola, por exemplo, o sistema de matrícula dos alunos permite que os funcionários efetuem a matrícula no formulário devido e depois



emitam relatórios com a relação de alunos *matriculados por turma*, alunos *matriculados por turno*, alunos que *frequentam a disciplina A* e, no final do semestre, relatórios dos *alunos aprovados e reprovados*. Observa-se que a classificação está presente em todos esses momentos.

A contribuição de Ranganathan para as teorias e os sistemas de classificação é significativa até os dias atuais, como se percebe através de sua “idéia em dividir os assuntos em *categorias* ou *facetadas*, isto é, grupos de classes reunidas por um mesmo princípio de divisão” (BARBOSA, 1972, p. 74). Para Lima (2004, p. 58), a Classificação Facetada de Ranganathan

[...] compreende princípios e técnicas para os sistemas de organização e recuperação da informação. Um sistema facetado reconhece muitos aspectos [...] e tenta sintetizar estes aspectos de maneira a melhor descrevê-lo. [...] A classificação facetada pode mostrar um mapeamento [...] de uma determinada área, permitindo a descoberta de conexões e analogias [...] e facilitando a recuperação da informação.

Na metodologia de OO podemos ver claramente a relação das *facetadas* e *subfacetadas* entre as *classes* e *subclasses*. Podemos deduzir que as *facetadas* estão para as *classes* assim como as *subfacetadas* estão para as *subclasses*. Pode-se perceber isso através do seguinte trecho do artigo de Barbosa (1972, p. 75):

[...] Podemos definir *faceta* como uma lista de termos mantendo entre si as mesmas amplas relações com a classe que lhes deu origem, ou então, ‘como um conjunto de termos produzidos pela aplicação de um amplo princípio de divisão’. *Faceta* não é portanto uma única subclasse, mas um conjunto delas.

Outra semelhança entre os procedimentos adotados pela OO para modelagem de sistemas e pelos sistemas de classificação é que para usar a linguagem de representação de dados há necessidade de um “minucioso conhecimento na área a ser coberta” (BARBOSA, 1972, p. 73). Na modelagem de sistemas o profissional também tem que se envolver com a área fim da organização a fim de que possa criar bases de dados consistentes e que possibilitem a recuperação das informações de forma íntegra e com a máxima facilidade possível.

## 5 ESTUDO DE CASO

### 5.1 Situação Real

Para ilustrar a aplicação da metodologia e dos conceitos de OO na modelagem de sistemas de informação, apresenta-se um *estudo de caso* baseado num sistema simplificado para uma biblioteca. A situação real e os dados a partir dos quais o sistema vai ser modelado são mostrados abaixo (Figura 16). Após essa descrição, os diagramas que representam os dados serão montados, com uma breve explicação sobre o que foi representado.

Figura 16 – Situação Real

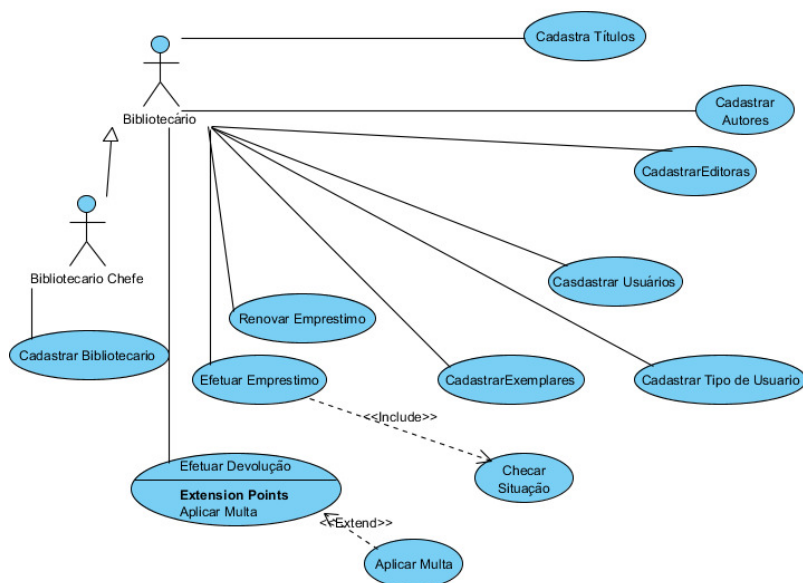
A empresa XYZ deverá desenvolver um sistema para uma biblioteca.  
O sistema deverá ser usado para administração do empréstimo de livros.  
O sistema deverá cadastrar **livros**(codLivro, nomeLivro, codAutor, codEditora), **editoras**(codEditora, nomeEditora, CNPJ, cidadeEditora), **exemplares**(codExemplar, codLivro, dataCompra), **autores**(codAutor, nomeAutor), **usuários**(codUsuario, nomeUsuario, CPF, enderecoUsuario, telefoneUsuario, tipoUsuario), **tipos de usuário**(codTipoUsuario, nomeUsuario), **bibliotecários**(codBibliotecario, nomeBibliotecario, CPF, enderecoBibliotecario, telefoneBibliotecario), **empréstimos**(codExemplar, codBibliotecario, codUsuario, dataEmprestimo, dataDevolucao, multa(s/n), valorMulta, statusMulta). O sistema deverá oferecer as funções “**Emprestar Livro**”, “**Checar situação de usuário**”, “**Aplicar multa**”, “**Devolver Livro**”, “**Checar Histórico**”. Também deverão fazer parte do sistema os seguintes relatórios: “**Relatório de Inventário**”, “**Relatório de Objetos em Circulação**”, “**Relatório de Multas Pendentes**”, “**Relatório de Multas Pagas**”.

Fonte: Elaborado pelos autores

### 5.2 Diagrama de Casos de Uso

Quando um profissional de tecnologia da informação é contratado para desenvolver um sistema informatizado, a primeira coisa que deve fazer é analisar os requisitos desse sistema. Esta atividade ocorre por meio de entrevistas cujo objetivo é “descobrir” as funcionalidades que o sistema deve oferecer. Como já foi dito antes, o diagrama de casos de uso mostra “o que” o sistema irá fazer. O diagrama de casos de uso do sistema de empréstimo de livros da biblioteca pode ser visualizado na figura abaixo (Figura 17).

Figura 17 – Diagrama de Casos de uso do Sistema de Empréstimo de Livros



Fonte: Elaborado pelos autores

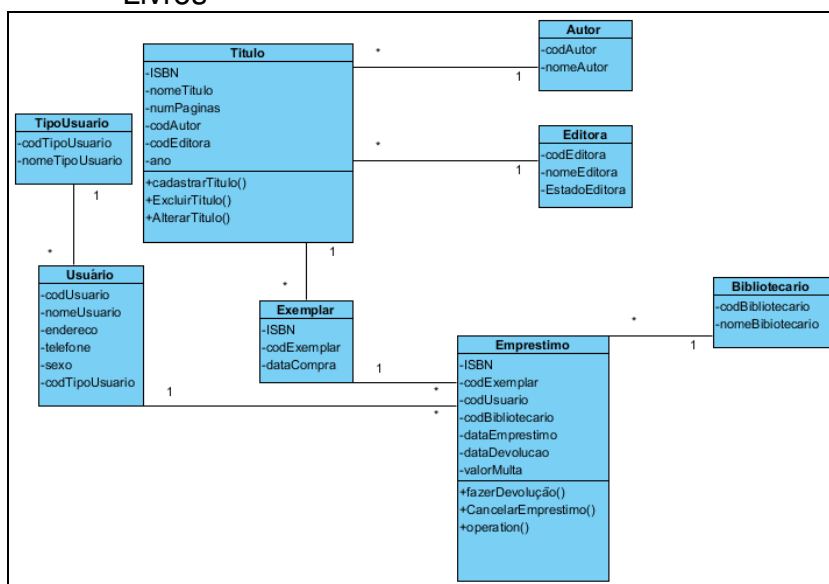
Como se pode observar, o sistema de empréstimo de livros da biblioteca tem dois atores que interagem com ele: o *Bibliotecário* e o *Bibliotecário Chefe*. Ambos têm acesso às funcionalidades do sistema, exceto a funcionalidade de *Cadastrar Bibliotecário*, que só pode ser executada pelo *Bibliotecário Chefe*. Em seguida verifica-se que os bibliotecários da instituição, representados pelo ator *Bibliotecário*, podem cadastrar títulos, cadastrar autores, cadastrar editoras, cadastrar usuários, cadastrar tipos de usuários, cadastrar exemplares, efetuar empréstimos, checar situação (do usuário), efetuar devolução e aplicar multa e finalmente cadastrar bibliotecário, função esta restrita ao bibliotecário chefe.

A funcionalidade *Efetuar Empréstimo* está ligada por uma associação <<include>> à funcionalidade *Checar Situação*. Isto acontece porque toda vez que o bibliotecário for efetuar um empréstimo ele deverá *necessariamente* checar a situação de cada usuário. Observa-se também que as funcionalidades *Efetuar Devolução* e *Aplicar Multa* estão ligadas através de uma associação do tipo <<Extend>>. Isto acontece porque *algumas vezes* em que o bibliotecário registra uma devolução ele deverá aplicar uma multa.

### 5.3 Diagrama de Classes

O diagrama de classes abaixo (Figura 18) representa as classes encontradas no sistema de empréstimo de livros de acordo com o processo de engenharia de requisitos. Podemos observar que os relacionamentos entre as classes possuem multiplicidade, representadas pelos símbolos 0, 1 ou \*.

Figura 18 – Diagrama de Classes do Sistema de Empréstimo de Livros



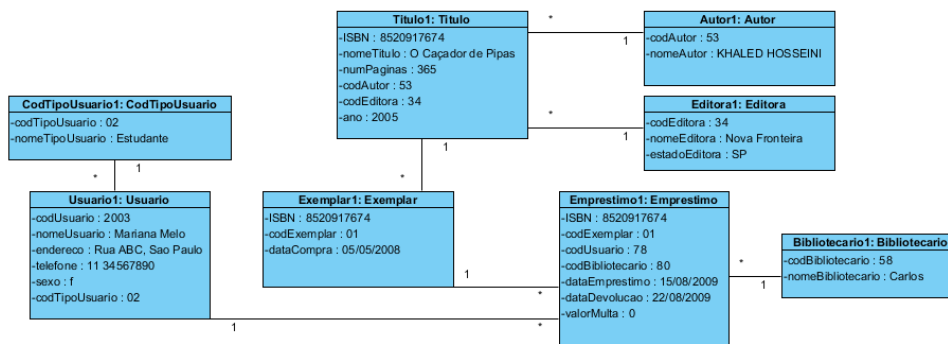
Fonte: Elaborado pelos autores

No diagrama acima percebe-se, por exemplo, que um título só pode ser de uma editora, mas uma editora pode ter vários títulos. Um exemplar só pode ser de um título, mas um título pode ter vários exemplares. O restante dos relacionamentos segue a mesma lógica.

### 5.4 Diagrama de Objetos

No diagrama de objetos abaixo (Figura 19) observa-se a representação gráfica do momento em que a estudante Mariana Melo, atendida pelo bibliotecário Carlos, toma por empréstimo domiciliar à biblioteca o exemplar de número 1 do título *O Caçador de Pipas*, do autor Khaled Hosseini, da editora Nova Fronteira, no dia 15/08/2009.

Figura 19 – Diagrama de Objetos do Sistema de Empréstimo de Livros

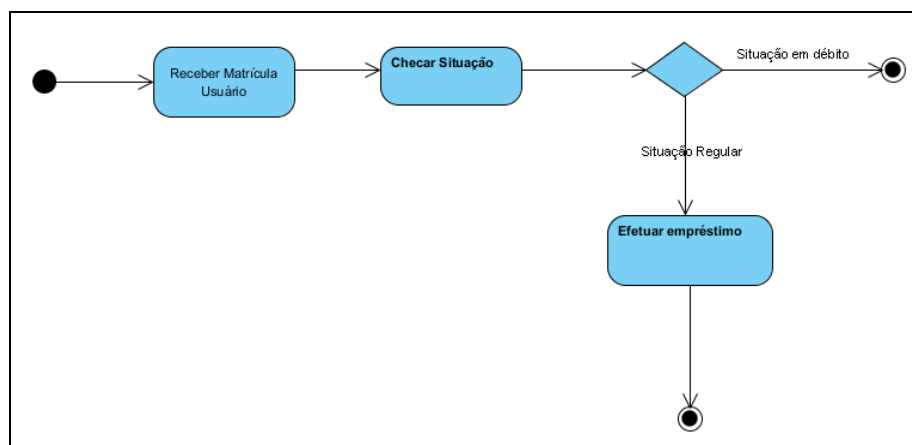


Fonte: Elaborado pelos autores

## 5.5 Diagrama de Atividades

O diagrama de atividades que se vê na figura abaixo (Figura 20) representa a ação de empréstimo de livros no sistema da biblioteca. A primeira tarefa envolve receber a matrícula do usuário e, depois, checar a situação. Se o usuário estiver em débito com a biblioteca, o processo é finalizado. Caso o usuário esteja em situação regular, a rotina do empréstimo prossegue, até sua efetivação final.

Figura 20 – Diagrama de Atividades do Sistema de Empréstimo de Livros



Fonte: Elaborado pelos autores

## 6 CONCLUSÕES

Este artigo tentou evidenciar proximidades lógicas, operacionais e técnicas entre o planejamento e a modelagem de sistemas de informação utilizando a

metodologia de orientação a objetos e a linguagem UML e as teorias e conceitos dos sistemas de classificação bibliográfica, em especial a concepção da classificação facetada, criada por Ranganathan em 1960 em sua *Colon Classification* (Classificação de Dois Pontos) (RANGANATHAN, 1960).

Constatou-se que há influências das teorias da classificação, especialmente a de Ranganathan, na área de modelagem de sistemas de informação, demonstrando que tudo o que hoje se desenvolve na informática, mais precisamente na área de desenvolvimento de sistemas, já foi realizado antes, de forma manual.

Os hábitos mudaram, as tecnologias entraram em cena, as tarefas ganharam novos nomes, mas a essência do trabalho de organização e recuperação de documentos e de informação continua a mesma. Nesse trabalho foi confirmado que para entender o complexo, antes de tudo é preciso dividi-lo em partes e classificá-lo.

## REFERÊNCIAS

BARBOSA, Alice Príncipe. Classificações facetadas. **Ciência da Informação**, Brasília, v. 1, n. 2, p. 73-81, 1972. Disponível em: <<http://revista.ibict.br/index.php/ciinf/article/view/1665/1271>>. Acesso em: 11 jul. 2011.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Rio de Janeiro: Elsevier, 2005.

CAMPOS, Maria Luiza de Almeida. **A organização das unidades de conhecimento em hiperdocumentos: o modelo conceitual como um espaço comunicacional para a realização da autoria**. 2001. Dissertação (Doutorado em Ciência da Informação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2001.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. São Paulo: Novatec, 2011.

LIMA, Gercina Ângela Borém. Modelo simplificado para análise facetada de Spiteri a partir de Ranganathan e do Classification Research Group (CRG). **Información, Cultura y Sociedad**, Buenos Aires, n. 11, p. 57-72, 2004.

MELO, Ana Cristina. Conceitos de orientação a objetos e UML. **Revista Engenharia de Software Magazine**, Rio de Janeiro, n. 8, p. 20-25, 2008.

\_\_\_\_\_. Entendendo casos de uso na prática: estudo de caso: parte I. **Revista Engenharia de Software Magazine**, Rio de Janeiro, n. 10, p. 18-23, 2009.

OLIVEIRA, Artur Elias Hayne. Introdução à orientação a objetos. **Revista SQL Magazine**, Rio de Janeiro, n. 89, p. 68-74, 2011.

OLIVEIRA, Jorge Martins; AMARAL, Júlio Rocha. O pensamento abstrato. **Cérebro e Mente**, Campinas, n. 12, 2001. Disponível em <<http://www.cerebromente.org.br/n12/opiniaio/pensamento.html>>. Acesso em: 10 jul. 2011.

PILONE, Dan; PITMAN, Neil. **UML 2.0 in a nutshell**: a desktop quick reference. New York: O'Reilly Media, 2005.

PUGA, Sandra. **Lógica de programação e estrutura de dados**. Brasil: Prentice Hall Brasil, 2004.

RANGANATHAN, Shiyali Ramamrita. **Colon classification, basic classification**. 6. ed. New York, Asia Publ. House, 1960.

RUMBAUGH, James et al. **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1994.

SOUZA, Vinicius Lourenço. Desenvolvimento de software dirigido por caso de uso. **Revista Engenharia de Software Magazine**, Rio de Janeiro, n. 2, p. 36-40, 2008.

---

### **Title**

The paradigm of object orientation, unified modeling language (UML) and the organization and representation of knowledge: a case study of a library system

### **Abstract**

**Introduction:** The object orientation works with a notation of its own and has some core concepts. Among the most important are classes, attributes, relationships and inheritance. On the other UML language, hand has diagrams own.

**Objective:** To demonstrate the methodology of object orientation and UML modeling language, with its many diagrams.

**Methodology:** To illustrate the operation the diagrams develops a case study surrounding a system of lending books in library.

**Results:** It found that there are influences of the theories of classification, especially that of Ranganathan, in modeling of information systems, demonstrating that everything is developing today in computer science, more precisely in the area of systems development, has been accomplished before manually.

**Conclusions:** In this study has confirmed that in order to understand the complex, first you need to divide it into parts and sort it.

**Keywords:** Systems modeling. Object orientation. UML. Faceted classification. Ranganathan

---

## Título

El paradigma de la orientación a objetos, el lenguaje de modelado unificado (UML) y la organización y representación de conocimiento: un estudio de caso de un sistema de bibliotecas

## Resumen

**Introducción:** La orientación a objetos trabaja con una notación propia y posee algunos conceptos centrales. Entre dichos conceptos están los rangos, los atributos, los relacionamientos y la herencia. Por su parte, el lenguaje UML posee diagramas propios.

**Objetivo:** Demostrar la metodología de la orientación a objetos y el lenguaje de modelado UML, con sus diagramas.

**Metodología:** Para ilustrar el funcionamiento de los diagramas, se desarrolla un estudio de caso relativo a un sistema de préstamo de libros en biblioteca.

**Resultados:** Se constató que hay influencias de las teorías de la clasificación, especialmente de Ranganathan, en el modelado de sistemas de información, lo que demuestra que todo lo que se está desarrollando hoy en día en la informática, más precisamente en el área de desarrollo de sistemas, se ha hecho antes, manualmente.

**Conclusiones:** En este estudio se confirmó que con el fin de entender el complejo, antes que nada, es necesario dividirlo en partes y clasificarlo.

**Palabras Clave:** Modelado de sistemas. Orientación a Objetos. UML. Clasificación por facetas. Ranganathan.

---

Recebido em: 27.08.2011

Aceito em: 10.05.2013