



**UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS HUMANAS, LETRAS E ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM LINGUÍSTICA**

LILIANE CARVALHO FÉLIX CAVALCANTE

**COMPREENSÃO DE LEITURA EM TEXTOS MULTIMODAIS
IMPRESSOS E SUPORTADOS PELO COMPUTADOR: uma
abordagem cognitivo-textual**

João Pessoa - PB
2010



**UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS HUMANAS, LETRAS E ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM LINGUÍSTICA**

LILIANE CARVALHO FÉLIX CAVALCANTE

**COMPREENSÃO DE LEITURA EM TEXTOS MULTIMODAIS
IMPRESSOS E SUPORTADOS PELO COMPUTADOR: uma
abordagem cognitivo-textual**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Linguística da Universidade Federal de João Pessoa - PB como requisito parcial para a obtenção do título de Mestre em Linguística.

Orientador: Prof. Dr. Jan Edson Rodrigues Leite.

João Pessoa - PB
2010

C376c Cavalcante, Liliane Carvalho Félix

Compreensão de leitura em textos multimodais impressos e suportados pelo computador: estudo de caso com alunos do ensino médio integrado ao técnico / Liliane Carvalho Félix Cavalcante. – João Pessoa, 2010.
189f. : il.

Dissertação (Mestrado) - Universidade Federal da Paraíba, Programa de Pós-Graduação em Língua Portuguesa, 2010.

Orientador: Prof. Dr. Jan Edson Rodrigues Leite

Bibliografia e anexos

1. Língua portuguesa. 2. Compreensão na leitura. 3. Cognição. I. Universidade Federal da Paraíba. II. Título

CDD 469

Ficha Catalográfica: Rosana Maria Santos de Oliveira
Bibliotecária CRB2-810

LILIANE CARVALHO FÉLIX CAVALCANTE

**COMPREENSÃO DE LEITURA EM TEXTOS MULTIMODAIS
IMPRESSOS E SUPORTADOS PELO COMPUTADOR: uma
abordagem cognitivo-textual**

Dissertação apresentada ao Programa de Pós-Graduação em Linguística da Universidade Federal da Paraíba como requisito parcial para obtenção do título de Mestre em Linguística.

Data: 23/09/2010

Conceito: Aprovado

BANCA EXAMINADORA

Dr. Jan Edson Rodrigues Leite (Orientador) - UFPB

Dra. Maria de Fátima Alves - UFCG

Dra. Regina Celi Mendes Pereira da Silva - UFPB

Dra. Marianne Carvalho Bezerra Cavalcante - UFPB

*Ao Evandro, companheiro incansável.
Aos meus filhos Murilo e Ana Carolina,
que me proporcionaram a plenitude do
amor incondicional.*

AGRADECIMENTOS

Em especial ao professor Dr. Jan Edson Rodrigues Leite, pela dedicação e pela seriedade com que conduziu a elaboração deste trabalho.

Às professoras Dr.^a Marianne Cavalcante e Dr.^a Regina Celi, membros da banca de qualificação, pelas importantes contribuições que resultaram nesta dissertação.

Com o mesmo apreço, agradeço aos professores com os quais fiz disciplinas no mestrado, cada um de forma particular contribuiu para meu crescimento. Em especial agradeço à professora Dr.^a Regina Baracuí, que, com a irreverência das leituras indicadas, me proporcionou uma nova visão do mundo.

Aos alunos que aceitaram participar dos experimentos deste trabalho, meus sinceros agradecimentos.

E, por fim, agradeço aos meus pais, que conduziram o trajeto da minha formação básica para que eu pudesse alçar voos maiores.

RESUMO

Esta pesquisa trata da compreensão leitora como resultado de vários processos cognitivos integrados. Discute a relação da dinâmica das atividades cognitivas que subsidiam o trabalho de elaboração dos modelos cognitivos idealizados a partir de leituras de textos. Para considerar essa problemática, a compreensão foi investigada através de experimentos e instrumentos nos quais observamos as variáveis: estratégias cognitivas e conhecimento prévio por meio do uso do método hipotético-dedutivo. O estudo foi realizado com alunos do curso técnico em informática na modalidade médio-integrado da primeira e da segunda série. O experimento se constituiu de textos multimodais em duas formas: digitais (hipertexto) e impressos. Controlamos o índice de multimodalidade presente no texto com o intuito de observar a interferência desse fator nas variáveis citadas. Os instrumentos utilizados foram questionários e testes de resolução de problemas aplicados após a leitura dos textos. Como avaliação dos instrumentos, confrontamos as respostas dos alunos, que chamamos de MCI *qualis* com as expectativas ou os protótipos do autor, que nomeamos de MCI *expert*. Para tratar esse fenômeno, buscamos sustentação teórica nas teorias: Modelos Cognitivos Idealizados, de Lakoff (1987), Categorias Conceituais, de Lakoff e Johnson (1980), Espaços Mentais, de Fauconier e Sweetser (1996), Redes de Integração Conceitual, de Fauconier e Turner (2002); nos estudos de leitura, de Kleiman (2004), e processamento de leitura, de Kato (2007). Os resultados apontaram que a constituição do MCI *qualis*, ou da compreensão leitora pode ser condicionada pela presença e pela atuação de estratégias cognitivas e conhecimento prévio que são evocados no processamento de leitura; e que o texto em suporte digital não implica um maior condicionamento da compreensão, ao passo que a presença da multimodalidade, tanto em textos impressos como digitais, favorece a disponibilidade das atividades cognitivas que culminam em um maior grau de compreensão.

Palavras-chave: Leitura; Compreensão; Multimodalidade; Modelos cognitivos idealizados.

ABSTRACT

This research deals with reading comprehension as a result of several integrated cognitive processes . It discusses the relationship of the dynamics of cognitive activities that support the preparation of these idealized cognitive models from reading texts. To consider this issue, the understanding was investigated through experiments and instruments in which observed variables: cognitive strategies and prior knowledge, using the hypothetical-deductive method. The study was conducted with students of the first and second grade of a technical course of ensino médio integrado in computing. The experiment consisted of multimodal texts in two ways: digital (hypertext) and printed. Control the rate of multimodality in this text in order to observe the effect of this factor in the variables mentioned. The instruments used were questionnaires and tests of problem solving applied after reading the texts. Keywords: reading comprehension, multimodality, idealized cognitive models. As evaluation tools we confronted the students' answers, which we call MCI qualis with expectations or prototypes of the author, who named MCI expert. To address this phenomenon, we sought theoretical support in the theories: Idealized Cognitive Models of Lakoff (1987), Conceptual Categories, by Lakoff and Johnson (1980), Mental Spaces in Fauconier and Sweetser (1996), Conceptual Integration Networks, in Fauconier and Turner (2002), in studies of reading, Kleimann (2004), and processing of reading, Kato (2007). The results indicate that the constitution of MCI qualis, or reading comprehension, can be conditioned by the presence and role of cognitive strategies and prior knowledge that are evoked in the processing of reading, and that the text in digital does not imply a higher conditioning understanding, whereas the presence of multimodality, both in printed and digital favors the availability of cognitive activities that culminate in a greater degree of understanding.

Keywords: reading comprehension, multimodality, idealized cognitive models.

LISTA DE GRÁFICOS

- Gráfico 1 – Questão 1 do experimento I
- Gráfico 2 – Questão 2 do experimento I
- Gráfico 3 – Questão 3 do experimento I
- Gráfico 4 – Questão 8 do experimento I
- Gráfico 5 – Questão 1 do experimento II
- Gráfico 6 – Questão 2 do experimento II

LISTA DE FIGURAS

- Figura 1 – Experimento I
- Figura 2 – Experimento II
- Figura 3 – Tela de acesso ao sistema *web*
- Figura 4 – Tela da primeira aula do experimento I
- Figura 5 – Teste de compreensão
- Figura 6 – Legenda dos hipertextos
- Figura 7 – Hipertexto *Visão geral da UML*
- Figura 8 – Hipertexto *Diagrama de classes*
- Figura 9 – Organização do hipertexto do experimento II

LISTA DE QUADROS

- Quadro 1 – Formulário de MCI *expert* do experimento I
- Quadro 2 – Formulário de avaliação de alunos do experimento I
- Quadro 3 – Formulário de MCI *expert* do experimento II
- Quadro 4 – Formulário de avaliação de alunos do experimento II
- Quadro 5 – Respostas e estratégias cognitivas esperadas no experimento
- Quadro 6 – Avaliação da compreensão do AI1 - grupo experimental impresso
- Quadro 7 – Avaliação da compreensão do AI2 - grupo experimental impresso
- Quadro 8 – Avaliação da compreensão do AI3 - grupo experimental impresso
- Quadro 9 – Avaliação da compreensão do AI4 - grupo experimental impresso
- Quadro 10 – Avaliação da compreensão do AI5 - grupo experimental impresso
- Quadro 11 – Avaliação da compreensão do AI6 - grupo experimental impresso
- Quadro 12 – Avaliação da compreensão do AI7 - grupo experimental impresso
- Quadro 13 – Avaliação da compreensão do AI1 - grupo experimental digital
- Quadro 14 – Avaliação da compreensão do AI2 - grupo experimental digital
- Quadro 15 – Avaliação da compreensão do AI3 - grupo experimental digital
- Quadro 16 – Avaliação da compreensão do AI4 - grupo experimental digital
- Quadro 17 – Avaliação da compreensão do AI5 - grupo experimental digital
- Quadro 18 – Avaliação da compreensão do AI6 - grupo experimental digital
- Quadro 19 – Avaliação da compreensão do AI7 - grupo experimental digital
- Quadro 20 – Respostas e estratégias cognitivas esperadas do experimento II
- Quadro 21 - Avaliação da compreensão do AI1 - grupo experimental impresso
- Quadro 22 - Avaliação da compreensão do AI2 - grupo experimental impresso
- Quadro 23 - Avaliação da compreensão do AI3 - grupo experimental impresso
- Quadro 24 - Avaliação da compreensão do AI4 - grupo experimental impresso
- Quadro 25 - Avaliação da compreensão do AI1 - grupo experimental digital
- Quadro 26 - Avaliação da compreensão do AI2 - grupo experimental digital
- Quadro 27 - Avaliação da compreensão do AI3 - grupo experimental digital
- Quadro 28 - Avaliação da compreensão do AI4 - grupo experimental digital

LISTA DE TABELAS

Tabela 1 – Verificação de MCI

Tabela 2 – Resumo da avaliação da questão 1 - grupo experimental impresso

Tabela 3 – Resumo da avaliação da questão 2 - grupo experimental impresso

Tabela 4 – Resumo da avaliação da questão 3 - grupo experimental impresso

Tabela 5 – Resumo da avaliação da questão 8 - grupo experimental impresso

Tabela 6 – Resumo da avaliação da questão 1 - grupo experimental digital

Tabela 7 – Resumo da avaliação da questão 2 - grupo experimental digital

Tabela 8 – Resumo da avaliação da questão 3 - grupo experimental digital

Tabela 9 – Resumo da avaliação da questão 8 - grupo experimental digital

Tabela 10 – MCI *qualis* da questão 1

Tabela 11 – MCI *qualis* da questão 2

Tabela 12 – MCI *qualis* da questão 3

Tabela 13 – MCI *qualis* da questão 8

Tabela 14 – Tempo de leitura do experimento I - grupo experimental impresso

Tabela 15 – Tempo de leitura do experimento I - grupo experimental digital

Tabela 16 – Lista de blocos do hipertexto 1

Tabela 17 – Lista de blocos do hipertexto 2

Tabela 18 – Lista de janelas *popup* do hipertexto 2

Tabela 19 – Navegação dos alunos no hipertexto 1

Tabela 20 – Navegação dos alunos no hipertexto 2

Tabela 21 – Resumo da avaliação da questão 1 - grupo experimental impresso

Tabela 22 – Resumo da avaliação da questão 2 - grupo experimental impresso

Tabela 23 – Resumo da avaliação da questão 1 - grupo experimental digital

Tabela 24 – Resumo da avaliação da questão 2 - grupo experimental digital

Tabela 25 – MCI *qualis* da questão 1

Tabela 26 – MCI *qualis* da questão 2

Tabela 27 – Tempo de leitura do texto do experimento II

Tabela 28 – Tempo de leitura do hipertexto do experimento II

Tabela 29 – Lista de blocos do hipertexto do experimento II

Tabela 30 – Lista de *popup* do hipertexto do experimento II

Tabela 31 – Navegação dos alunos no hipertexto do experimento II

LISTA DE ABREVIATURAS

CP – Conhecimento prévio

EC esperadas – Estratégias cognitivas esperadas

EC alcançadas – Estratégias cognitivas alcançadas

INF – Inferência

MCI – Modelos cognitivos idealizados

MCI qualis – Modelos cognitivos idealizados qualis

MCI expert – Modelos cognitivos idealizados expert

UML – Unified modeling language

PR – Pressuposição

PD – Predição

Q1 – Questão 1

Q2 – Questão 2

Q3 – Questão 3

Q4 – Questão 4

SUMÁRIO

INTRODUÇÃO	13
1 PERSPECTIVAS TEÓRICAS COGNITIVAS E SUAS IMPLICAÇÕES NA INVESTIGAÇÃO DE LEITURA E COGNIÇÃO.....	16
1.1 TEORIAS COGNITIVAS E O POSICIONAMENTO DA INVESTIGAÇÃO	16
1.2 DOMÍNIOS ESTÁVEIS E LOCAIS	21
1.2.1 Modelos Cognitivos Idealizados	21
1.2.2 Categorias Conceituais	23
1.2.3 Espaços Mentais	27
1.2.4 Redes de Integração Conceitual	29
1.3 TEXTO, GÊNERO, HIPERTEXTO E MULTIMODALIDADE	32
1.3.1 Caracterizando Texto	32
1.3.2 Uma Noção de Gênero Textual	33
1.3.3 O Hipertexto: um novo espaço da escrita.....	34
1.3.4 Definindo Multimodalidade	35
1.4 LEITURA E COMPREENSÃO: ASPECTOS COGNITIVOS E METACOGNITIVOS	37
1.4.1 Definindo Leitura.....	37
1.4.2 O Conhecimento Prévio.....	39
1.4.3 Processamento da Leitura	41
1.4.4 Entendendo a Memória.....	42
1.4.5 Estratégias Cognitivas e Metacognitivas.....	43
2 METODOLOGIA	46
2.1 CLASSIFICAÇÃO DA PESQUISA	46
2.2 DEFINIÇÃO DA PESQUISA	47
2.3 OBJETIVO DA PESQUISA	48
2.3.1 Objetivo Geral.....	48
2.3.2 Objetivos Específicos	49
2.4 DESCRIÇÃO DOS EXPERIMENTOS.....	49
2.4.1 Experimentos I e II.....	50
2.5 SUJEITOS DA EXPERIÊNCIA.....	51
2.6 PROCEDIMENTOS DA EXPERIÊNCIA.....	51
2.7 FERRAMENTAS EXPERIMENTAIS UTILIZADAS.....	52
2.7.1 Suporte Impresso	52
2.7.2 Suporte Digital	53
2.8 CORPUS DE ANÁLISE.....	54
2.8.1 Procedimento de Exclusão	55

2.9 PROCEDIMENTO DE ANÁLISE (VERIFICAÇÃO DAS VARIÁVEIS) ..	55
2.9.1 Procedimentos de Análises dos Instrumentos	55
2.9.2 Especificidades da Análise dos Dados	58
3 ANÁLISE DOS DADOS	62
3.1 EXPERIMENTO I.....	62
3.1.1 Análise do experimento I.....	62
3.1.1.1 Questões Analisadas	62
3.1.1.2 Análise por Aluno	64
3.1.1.3 Resumo por Questão	64
3.1.1.4 Tempo gasto nas leituras dos dois textos do experimento I.....	87
3.1.1.5 Organização dos Hipertextos	88
3.2 EXPERIMENTO II.....	92
3.2.1 Análise do Experimento II	92
3.2.1.1 Questões do Questionário do Experimento II	92
3.2.1.2 Análise por Aluno	93
3.2.1.3 Resumo por Questão	103
3.2.1.4 Tempo gasto nas leituras dos textos do experimento II.....	103
3.2.1.5 Organização do Hipertexto	104
3.3 PONDERAÇÕES GLOBAIS	1045
CONSIDERAÇÕES FINAIS	109
REFERÊNCIAS.....	112
ANEXOS.....	117

INTRODUÇÃO

A compreensão é foco de estudos de grande parte das ciências humanas em seus vários aspectos. Entendemos esse fenômeno como resultado de várias atividades cognitivas integradas que culminam na faculdade humana de compreender algo, ou mesmo na aquisição do conhecimento.

Para investigarmos a compreensão, filiamo-nos às Ciências Cognitivas, que abrange diversas áreas do conhecimento: neurociência, linguística, psicologia cognitiva, sociologia, antropologia, biologia e ciência da computação.

Cada uma das ciências citadas colabora, conforme suas perspectivas, com insumos que enriquecem a discussão em torno do tema e ainda ajuda para um entendimento macro da fenomenologia da capacidade cognitiva humana.

Para investigar a compreensão leitora, direcionamos nossos esforços para o estudo do processo, ou mesmo, das atividades cognitivas que resultam no compreender. Entendemos que tal compreensão é um fenômeno natural do sujeito socialmente situado que se instaura a partir das interações desse sujeito com os inúmeros textos em variados formatos presentes no mundo.

Esta investigação tem caráter experimental analítico e compreende a aplicação de experimentos de compreensão a alunos do curso médio integrado ao técnico em informática do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins Campus de Palmas. O primeiro experimento se constitui de textos do gênero didático, revestidos essencialmente da natureza pretendida – a multimodalidade – em duas versões, sendo uma impressa e a outra hipertextual apresentada no microcomputador; o segundo, do gênero artigo científico, que compreende um artigo em texto puro impresso e uma adaptação dele em hipertextos semióticos.

A presença de textos, como os mencionados, é marcante na atualidade. No ambiente educacional, alunos convivem com textos assim em toda sua trajetória acadêmica. Dessa forma, acreditamos que a pesquisa se faz relevante por pretender identificar a modalidade de texto, entre as mencionadas, que melhor se adapta ao público estudantil moderno. Em outras palavras, a que possibilita um maior grau de

compreensão, tendo em vista a composição e a organização do texto e também o suporte em que é apresentado.

Acreditamos contribuir com os estudos da Linguística Cognitiva ao apontarmos os efeitos de texto que favorecem a internalização¹ branda e amigável de conhecimentos por alunos da faixa etária do ensino médio.

A investigação girou em torno da busca de elementos científicos que esclarecessem as indagações que emergiram do estudo. Pensando na relação entre diferentes textos e seus efeitos na recepção e na disposição de aprendizagem do compreendedor, é que buscamos repostas para as perguntas:

- Qual dos dois suportes de textos multimodais (impressos ou digitais) favorece a compreensão de alunos do ensino médio?
- A multimodalidade em textos suportados pelo computador facilita a compreensão?

Acreditamos na hipótese de que textos multimodais suportados por computador favorecem em maior grau a compreensão de alunos da fase do ensino médio devido à proximidade desse público com os aparatos tecnológicos.

Nessa ótica, a relação dos alunos com a tecnologia propicia a formação cognitiva de modelos que os auxiliam na construção de novos conceitos. Esse fato pode ocorrer até mesmo devido à natureza hipertextual dos textos multimodais apresentados no suporte digital, que induz o leitor a outra lógica, em que, diferentemente dos impressos, nos quais a linearidade predomina, são os labirintos e as associações entre outras formas de comunicação visual que abrem ao leitor novas possibilidades de leitura. Assim, é possível que os textos multimodais no suporte micro favoreçam a compreensão por possibilitar outras dinâmicas de leituras apoiadas em mecanismos semióticos, como as cores, os sons, as formas plásticas, os gráficos, os desenhos e os *links*.

Com a pesquisa, objetivamos em panorama geral: analisar a compreensão leitora em textos em suportes impressos e digitais com características diferenciadas de multimodalidade. Especificamente ojetivamos: investigar o processamento da compreensão leitora em textos multimodais impressos com maiores e menores

¹ Chamamos de internalização branda e amigável o ato espontâneo de compreender.

índices de multimodalidade; analisar o processo de compreensão de leitura em textos multimodais suportados por computador; proceder à investigação do processo de construção de MCI relativos aos conceitos apresentados nos textos experimentais; analisar as estratégias cognitivas empregadas no processo de compreensão leitora.

Para tratar o fenômeno, buscamos sustentação teórica nas teorias: Modelos Cognitivos Idealizados, de Lakoff (1987); Categorias Conceituais, de Lakoff e Johnson (1980); Espaços Mentais, de Fauconier e Sweetser (1996); Redes de Integração Conceitual, de Fauconier e Turner (2002); e nos estudos de leitura, de Kleiman (2004); e processamento de leitura, de Kato (2007).

Em termos de organização, esta pesquisa traz, além da introdução, quatro capítulos. No próximo capítulo, apresentamos o aporte teórico, que dá sustentação à pesquisa, em que os temas centrais são: Leitura e Compreensão. Inicialmente, fazemos um passeio panorâmico sobre os paradigmas teóricos da cognição: cognitivismo, connexionismo e atuacionismo; em seguida, focamos os aspectos da compreensão em seus domínios estáveis e locais, discutimos a estruturação do conhecimento em Modelos Cognitivos Idealizados, Categorias Conceituais, Espaços Mentais e Redes e Integração Conceitual; e, por fim, explanamos sobre os aspectos cognitivos e metacognitivos da Leitura.

No capítulo 3, apresentamos o desenho metodológico da pesquisa, debatemos sobre a classificação da pesquisa, descrevemos os objetivos, os sujeitos experienciados, a composição do experimento, as ferramentas utilizadas e ainda o procedimento de análise dos dados. No quarto, analisamos os dados coletados, suas implicações com as teorias e observamos se a hipótese testada se comprovava. No último, apresentamos as considerações finais, as opiniões geradas no decorrer da pesquisa e a relevância delas para a produção de conhecimento, uma vez que se buscou investigar o fenômeno compreensão em leituras em diferentes suportes.

1 PERSPECTIVAS TEÓRICAS COGNITIVAS E SUAS IMPLICAÇÕES NA INVESTIGAÇÃO DE LEITURA E COGNIÇÃO

1.1 TEORIAS COGNITIVAS E O POSICIONAMENTO DA INVESTIGAÇÃO

Segundo Koch e Cunha-Lima (2007), a abordagem das ciências cognitivas surgiu na década de 50 com o intuito de combater as pesquisas behaviorista que se preocupavam única e exclusivamente com as reações aos estímulos observáveis e mensuráveis.

Para os estudiosos da ciência, a mente e seus estados eram vistos como uma “caixa preta”. Não havia explicação que levasse em consideração “estados mentais”, “intenções”, “vontades” e outros elementos subjetivos, o que não é aceito pela metodologia científica.

Atualmente, o estudo da cognição está baseado em três paradigmas teóricos: o simbolismo, conhecido como cognitivismo clássico; o conexionismo; e o atuacionismo, conhecido também como o enacionismo da escola chilena.

O paradigma simbolista, segundo Bonini (2002, p. 27), “é conhecido como IA forte porque pretendeu simular toda a atividade cognitiva humana através da manipulação de símbolos em algoritmos computacionais”. Assim, os cognitivistas clássicos assumiam que representações mentais² são símbolos, que carregam em si valores semânticos. Dessa forma, as atividades mentais são operações sobre esses símbolos, ou mesmo uma computação exercida sobre os valores que eles representam com a finalidade de melhor simbolizar uma situação (KOCH; CUNHA-LIMA, 2007).

A teoria permitiu que os cientistas estudassem os processos internos da mente, deixando de lado a forma como os estímulos são captados e transformados em símbolos. Como os cognitivistas acreditavam que os símbolos representavam a realidade tal como ela era, acreditavam também que esse processo se explicava por um mecanismo inato, uma vez que seres, de modo geral, são capazes de reagir a

² Assumimos representações mentais como comportamento inteligente, que pressupõe a representação do mundo por meio de padrões relevantes (VARELA; THOMPSON; ROSCH, 2003).

estímulos e responder a eles de forma a atingir sucesso em tentativas como sobreviver, reproduzir, caçar ou escapar de predadores (KOCH; CUNHA-LIMA, 2007).

Os estudiosos dessa linha também concebiam a separação entre mente e corpo, focando somente na mente. Consideravam o corpo como responsável pela recepção de dados e envio de mensagens para a mente.

Um grande avanço do paradigma cognitivista foi a criação da máquina de Turing, que propôs um modelo para mecanizar qualquer cálculo ou procedimento lógico-matemático, resultando no computador atual. O objetivo da máquina não era revolucionar a informática, mas compreender a mente humana como uma espécie de máquina de Turing (KOCH; CUNHA-LIMA, 2007).

Com a revolução e o surgimento da máquina, entendeu-se que calcular era um ato mecânico e que podia ser realizado por qualquer sistema sem se preocupar com sua formação material e muito menos com o conteúdo dos símbolos. Isso levou os estudiosos a crerem que “pensar é calcular” e, ainda por se tratar de uma atividade mecânica, poderia ser apreendida pelo homem por meio da matemática, da lógica e da física (KOCH; CUNHA-LIMA, 2007).

As autoras expõem que

A tendência no interior das perceptivas cognitivas clássicas era descrever todo o trabalho mental a partir de representações simbólicas do real, e construir hipóteses seguindo regras dedutivas, a fim de chegar a conclusões sobre tais hipóteses (KOCH; CUNHA-LIMA, 2007, p. 264).

Com o amadurecimento das ciências cognitivas clássicas, houve uma corrente de pesquisadores que considerava o modelo como um todo inadequado, em especial, alguns dos pressupostos básicos que norteavam o trabalho da ciência, da sua concepção aos seus grandes avanços. A maior crítica estava na separação entre as dicotomias mente e corpo, processo interno e processo externo, individual e social e, por fim, o sistema de representação simbólica como cognição.

Paralelo ao cognitivismo, surgiu também o modelo conexionista. Este difere do cognitivista clássico por substituir processamento simbólico localizado por operações distribuídas dentro da rede de componentes interconectados. No fim do processo, segundo esse modelo, podem ser visualizadas propriedades globais que podem identificar e corrigir erros dos funcionamentos individuais dos componentes.

Para Bonini (2002), conexionismo ou funcionalismo neurocomputacional parte do princípio de que as representações são distribuídas em unidades predispostas a um processamento paralelo, de forma que a ativação de uma rede de neurônios ocorre quando um *input* ativa certos neurônios e inibe outros. Esse processamento é realizado simultaneamente e em paralelo, equiparado à força de conexão dos componentes até que se atinja um equilíbrio capaz de processar o *output*.

Ainda conforme Bonini (2002), o sistema conexionista trouxe duas características inovadoras:

- a possibilidade de operar com dados incompletos: como as representações operam de forma distribuída, um input incompleto pode ativar outros componentes de significado produzindo uma saída completa;
- a possibilidade de o sistema aprender: a performance da rede em uma tarefa pode ser treinada, por meio de testes consecutivos e, com isso, gerar o aprendizado metafórico e, ainda, acarretar no fornecimento de uma solução possivelmente melhor no output desse processamento.

Para Koch e Cunha-Lima (2007), a fonte de inspiração do conexionismo foi o sistema nervoso humano, caracterizado pelo uso de um conjunto de nódulos ou nós com capacidade limitada que, interligados entre si, formam uma rede neural. Dizem, ainda, que essas redes são capazes de aprender se as alimentarmos com dados, pois as ligações entre os nós mudam na proporção em que a rede conhece os dados e, com isso, altera sua configuração interna.

Esse paradigma, apesar de dar outra roupagem para o que chamamos cognição, não mais como operações de representações simbólicas, mas agora como processo sistematizado de redes integradas, exibindo como modelo inspirador o cérebro em oposição da mente, ainda não põe fim às dicotomias citadas anteriormente, pelo contrário, só as alimenta porque tem suas bases fincadas em uma dicotomia específica: a mente e o cérebro. Portanto, esse modelo também teve suas críticas, em especial no que diz respeito à dicotomia “individual e social”.

Smolensky (1992, p. 76) afirma que “Dentro dos sistemas conexionistas, o conhecimento é codificado não nas estruturas simbólicas, mas na configuração dos

coeficientes numéricos, que caracterizam a eficiência das conexões entre os processadores”.

Por isso, outros estudiosos colocam em xeque a noção computadorizada dos paradigmas apresentados até aqui, em especial, por não admitirem mais que conhecimento é algo computado ou no cérebro ou na mente. Para eles, há fatores no entorno de tudo isso que juntos corroboram em grande escala para a confecção do tão estudado conhecimento humano.

Como oposição tanto ao simbolismo quanto ao connexionismo, surgiu então o atuacionismo, ou mesmo, enacionismo. Para o novo paradigma, segundo Bonini (2002, p. 29), “o conhecimento é circular não podendo separar claramente o ambiente e o sujeito, sob pena de construir uma explicação que não leve em conta a experiência individual e o papel da linguagem”.

As atividades da mente, nesse caso, o ato de refletir, não têm suas origens no vazio, ao contrário, são condicionadas a partir do *background*: crenças, práticas biológicas, sociais e culturais.

O conhecimento, nessa ótica, conforme Teixeira (1998, p. 143), “advém do fato de eu estar num mundo que é inseparável do meu corpo, de minha linguagem de toda minha história social”. Na visão do atuacionismo, o ser, que integra o mundo, é o agente que coordena e interage com o ambiente a partir das suas experiências e das atividades internas da mente.

Lakoff e Jonhson (2002) apontam fenômenos como a formação do conceito das cores para fortalecer a concepção trazida pelo atuacionismo de que corpo e mente são um todo. A percepção das cores, segundo os autores, vem da união de quatro fatores, em que dois dependem do ambiente (comprimento de onda de luz e as condições de iluminação do ambiente) e os outros dois do nosso organismo (os cones que receptam os estímulos luminosos da retina). À medida que os fatores variam, variam também as condições de recepção e identificação das cores. Portanto “a cor não está nos objetos nem na cabeça de quem a vê” (KOCH; CUNHA-LIMA, 2007, p. 275), mas trata-se de uma maneira em que o organismo significa o mundo que o envolve.

Um grande diferencial desse modelo está no fato de entender que a cognição é um fenômeno situado e social. O foco não está em entender como o conhecimento

está estruturado na mente de um indivíduo, mas como a cultura e a vida representam conhecimentos para a mente. Um indivíduo, para se estabelecer dentro da sua cultura, precisa, no mínimo, conhecê-la. Dessa forma, a cultura é um conjunto de dados que pode ser aprendido, ou mesmo, uma relação de procedimentos e noções dispostas socialmente para ser armazenada individualmente (KOCH; CUNHA-LIMA, 2007).

Para Bonini (2002), o atuaçãoismo enxerga a inteligência e a linguagem de modos diferentes, de forma que a inteligência não está fundamentada na capacidade de resolver problemas, mas na forma de ingresso no mundo compartilhado. Defende também que “a comunicação também não consiste na transmissão de informação de um indivíduo receptor” (BONINI, 2002, p. 29), mas na interação em comum por meio de ações conjuntas no mundo.

Dessa forma, o atuaçãoismo rompe barreiras até então sustentadas pelos modelos anteriores e vislumbra uma proposta com bases na relação do indivíduo dotado de suas capacidades com o mundo em que atua e modifica.

A pesquisa apresentada investiga leitura e compreensão e se posiciona dentro do paradigma atuaçãoista, por entender que o agente que lê e compreende é um ser humano que atua no mundo com seus saberes e crenças, e que o produto gerado pela leitura, a compreensão do lido, é composto de uma série de atividades cognitivas que reflete a atuação do leitor no ambiente e vice-versa. Para que haja compreensão, a história, a cultura e a vida de um ser atuam integradamente para condicionar sentido ao objeto a ser compreendido.

Assim, compreender é antes de tudo estar no mundo e significá-lo através das experiências receptadas pelo corpo no ambiente. Exemplos disso podem ser constatados nas respostas dos alunos a questões relacionadas às leituras efetuadas nos experimentos realizados neste relato. Ao responderem às questões, os alunos se utilizaram não só do conhecimento veiculado no texto, mas também do seu conhecimento prévio sobre o assunto, além disso, inferiram, pressupuseram, predisseram e previram. Cada uma dessas atividades requereu dos alunos que suas experiências de vida funcionassem como gatilhos que ativassem a capacidade cognitiva individual de categorizar, recuperar modelos cognitivos idealizados e criar

situações de integração conceitual que juntos culminaram na compreensão do que foi lido.

Portanto, entendemos que o compreendedor é um ser imerso no mundo com sua bagagem linguística, histórica e social e não pode ser reduzido a operador mecânico de símbolos e nem mesmo a receptor de comandos *inputs*, que ativam redes distribuídas de neurônios e processam *outputs*.

1.2 DOMÍNIOS ESTÁVEIS E LOCAIS

O produto do processo de compreensão é derivado de complexas atividades de construções cognitivas que se embasam em conjuntos de conhecimentos estruturados em domínios estáveis e locais.

Os domínios estáveis correspondem a estruturas disponíveis socialmente conhecidas como modelos cognitivos idealizados, molduras comunicativas e esquemas genéricos (MIRANDA, 1999). Eles podem ser entendidos como mecanismos de organização do conhecimento, ou mesmo como base para formulação e reformulação de hipóteses categoriais que direcionam a tentativa de compreender o mundo. Já os domínios locais, segundo Miranda (1999), também chamados de espaços mentais, são estruturas individuais e dinâmicas que se fazem e desfazem na ação do processamento cognitivo, fundam-se nos domínios estáveis e têm a função de construir significados.

1.2.1 Modelos Cognitivos Idealizados

A Teoria dos Modelos Cognitivos Idealizados (MCI) foi proposta por Lakoff dentro da área de conhecimento Semântica Cognitiva, oriunda do campo da Ciência Linguística Cognitiva. Ela vislumbra uma percepção gestáltica, que conduz o conhecedor a perceber, em primeira instância, o todo como sendo a realidade, o que torna mais simples conhecer as partes integrantes.

Segundo Lakoff (1987), MCI são construtos idealizados, uma vez que não precisam se ajustar necessariamente ao mundo, porque são resultados da experiência do homem com a sua realidade. Logo, os modelos cognitivos idealizados são a base de sustentação para o fenômeno humano da categorização porque as categorias são formadas a partir de pequenos grupos de MCI: "as categorias conceptuais humanas têm propriedades que são, pelo menos em parte, determinadas pela natureza corporal das pessoas que as categorizam" (LAKOFF, 1987, p. 371). Nessa ótica, a concepção semântico-conceitual, proposta por Lakoff, tem origem em uma epistemologia experiencialista, conforme registrado na citação:

[...] os modelos cognitivos, em nosso sentido, não são representações internas da realidade externa. Não são por duas razões: primeiro, porque eles são entendidos em termos de corporalidade, não em termos de uma conexão direta com o mundo externo; e segundo, porque eles incluem aspectos imaginativos da cognição como a metáfora e a metonímia (LAKOFF, 1987, p. 341).

Os MCI, devido à sua natureza experiencialista, podem ser construídos e interpretados de diferentes formas, tendo em vista necessariamente o ator da ação corporificada. Dessa forma, para um mesmo evento, podem ser construídos modelos cognitivos diferentes e até mesmo contraditórios. Isso mostra que os motivadores do surgimento dos modelos cognitivos estão imersos no ambiente e são atributos naturais do experienciador, como necessidades, valores, propósitos ou crenças.

Ainda sobre MCI, Feltes (2007, p. 11) afirma que

Temos em mente que um modelo cognitivo é um construto que nos permite entender o modo como as experiências bio-psico-sócio-culturais são categorizadas e organizadas pelo aparato cognitivo humano, tornando-as, dessa forma, significativas para nós, e não são apenas modelos metafóricos e metonímicos que realizam essa tarefa.

McCauley (1987, p. 292) define Modelos Cognitivos Idealizados como "construtos mentais simplificados que organizam vários domínios da experiência humana, tanto prática quanto teórica". Para o autor, a superestrutura do conhecimento se constitui do somatório dos MCI.

Pensando à luz dos autores, MCI podem ser definidos como repositórios de propriedade pública, construídos de forma colaborativa tendo em vista os aspectos culturais do ambiente que os subsidiam. Dessa forma, justifica-se a sua natureza reusável e, ainda, a sua função específica de guiar os significados. Lakoff (1987, p.

292) assevera que “O significado não é uma coisa; ele envolve o que é significativo para nós. Nada é significativo em si mesmo. A significatividade deriva da experiência da atuação como um ser de certo tipo em uma ambiente de um certo tipo”.

Lakoff (1987) diz que uma compreensão mais ampla de MCI pode se dar tendo em vista a teoria de frame de Fillmore, que parte da observação de uma categoria como 'terça-feira', que, para ele, só foi assim categorizada partindo do princípio de uma formatação convencional de um calendário cíclico com sete dias, que reflete o modelo cognitivo idealizado 'Semana'. O modelo convencionaliza o conceito como sendo um todo dividido em sete partes sequenciais em que a 'terça-feira' reflete a sétima parte do todo. Essa convenção é resultado da organização da nossa cultura, algo criado ou idealizado que de fato não existe na natureza e nem mesmo em todas as culturas.

Um MCI pode também refletir uma categoria de forma imperfeita, como ilustrado por Lakoff (1987) no exemplo da categoria 'solteirão'. Essa categoria foi convencionalizada tendo em vista um modelo de sociedade que pratica o casamento monogâmico e ainda institui uma idade prévia para isso. Portanto, solteirões são homens com determinada idade que ainda não contraíram o matrimônio. Essa convenção não contempla casos como o celibato, homens que vivem em união estável e ainda casos de isolamento involuntário como o exemplo do Tarzan.

Assim, os MCI são usados para compreender e teorizar o mundo, que reflete o ambiente cultural em que uma sociedade está inserida. Por essa razão, modelos cognitivos são, em sua maioria, modelos culturais.

1.2.2 Categorias Conceituais

Segundo Lakoff e Johnson (1980), para compreendermos o mundo e atuarmos nele, é necessário que categorizemos os objetos e as experiências, a fim de produzirmos algum tipo de significado a esses artefatos. Os autores dimensionam categorização como

[...] uma forma natural de identificar um tipo de objeto ou de experiência iluminando certas propriedades, atenuando outras e até escondendo outras. Cada uma das dimensões indica as propriedades que são iluminadas. Para iluminar determinadas propriedades, desviamos nossa atenção de outras.

Ao descrevermos fatos do dia a dia, por exemplo, usamos categorizações para pôr em evidência determinadas propriedades que correspondem às nossas intenções. Cada descrição irá iluminar, atenuar ou esconder [...] (LAKOFF; JOHNSON, 1980, p. 163).

Para Saraiva (2008), categorizar significa ordenar um número infinito de estímulos encontrados nas nossas experiências cotidianas em categorias finitas e de fácil acesso. Na mesma linha, Jacob e Shaw (1998) acrescentam que a categorização é um processo cognitivo de divisão das experiências de mundo em entidades (objetos, ideias, ações etc.) que resulta na construção de uma ordem física e social do mundo.

Apoiados nos autores, identificamos as categorias como unidades essenciais que encapsulam informações suficientes capazes de descrever eventos e fenômenos que vivenciamos no curso natural de nossas vidas. Dessa forma, a atividade cerebral se torna mais econômica uma vez que a recuperação de categorias implica necessariamente um conjunto de informações acerca do objeto percebido.

Assim, a capacidade de categorizar é a base para os processamentos cognitivos mais simples, como a percepção, a organização e o reconhecimento. E isso torna o aprimoramento do estudo das categorias imprescindível uma vez que é fator condicionante para entendermos como pensamos, significamos ou compreendemos o mundo. Sobre isso, Lakoff (1987, p. 5) expõe que

A categorização não é um processo que deva ser estudado superficialmente. Não há nada mais básico do que a categorização para o nosso pensamento, percepção, ação e discurso. Cada vez que nós vemos algo como um “tipo de coisa”, por exemplo, uma árvore, nós estamos categorizando.

As categorias, por influenciarem primariamente as ações cognitivas, foram e ainda são objetos de estudo de muitos estudiosos. A primeira hipótese acerca do assunto tem suas raízes calçadas na filosofia de Aristóteles, chamada hipótese clássica, que diz que os seres e os objetos podem pertencer a uma categoria específica se e somente se compartilharem as mesmas propriedades necessárias e suficientes. A hipótese aristotélica foi de extrema contribuição por iniciar a discussão sobre categorias, visto que, a partir dela, vieram muitas críticas que trouxeram novas propostas que se adequaram à realidade do assunto.

Para a hipótese clássica, um conjunto de atributos reflete uma classe que agrupa membros com as mesmas características, ou mesmo, as instâncias dessa

classe devem obrigatoriamente valorar os atributos da classe como condição de pertença. Assim, os exemplares só são considerados quando apresentarem todas as características de uma categoria. Caso deixem de apresentar qualquer um dos atributos, automaticamente não são exemplares da categoria em questão.

Conforme Rodrigues-Leite (2004), na hipótese aristotélica, as categorias são definidas apenas pelas propriedades inerentes a seus membros, independente de quem estiver categorizando. Portanto o processo independe da influência do movimento do corpo humano, da neurofisiologia humana pela capacidade humana de percepção, pelo aprendizado, pelas lembranças, pela comunicação ou pelos fatores socioculturais.

Essa abordagem foi extremamente criticada, em especial pela descorporificação das instituições das categorias. Segundo Lakoff (1987), a teoria clássica era vista como uma verdade definida e inquestionável e não como uma hipótese empírica.

Wittgenstein (1961, 1979) questionou a ineficácia das propriedades necessárias e suficientes no exemplo de conceito de jogo. O autor compara a instituição dos membros de uma categoria a uma família, em que os indivíduos compartilham da rede entrecruzada de semelhanças, mesmo não tendo todos os atributos inerentes à classe, o que não os desprezam da pertença à categoria, ao contrário, são ligados por relacionamentos de semelhanças sobrepostas.

Wittgenstein (1961, p. 66) sugere que consideremos,

[...] por exemplo, as atividades a que chamamos de jogos [...]. O que é comum a elas? Não diga 'deve haver algo em comum, senão não seriam chamadas de jogos', – mas veja se há algo comum a todas elas – pois se você observá-las, não verá algo comum a todas, apenas similaridades, relações, e toda uma série delas. Repito: não pense, veja! – Por exemplo, veja os jogos de tabuleiro, com seus múltiplos parentescos. Agora observe os jogos de carta; aqui você encontra muitas correspondências com o primeiro grupo, mas muitos traços comuns se perdem e outros surgem. Quando você passa para os jogos de bola, muito do que é comum é mantido, mas outro tanto se perde. São todos jogos 'divertidos'? Compare xadrez com jogo da velha. Ou há sempre perdedores e ganhadores, ou competição entre os jogadores? Pense em paciência. Nos jogos de bola há perdedor e vencedor, mas quando uma criança atira a bola contra a parede e a apanha na volta, este traço desaparece. Observe aqueles em que há habilidade e aqueles em que há sorte e veja a diferença entre habilidade no xadrez e habilidade em tênis. Pense agora em outros jogos de crianças; aqui há um elemento de diversão, mas quantos outros traços característicos desapareceram! E nós podemos percorrer os muitos, muitos outros grupos de jogos da mesma forma; vemos como as semelhanças surgem e desaparecem.

Outra teoria sobre categorização é a hipótese prototípica desenvolvida por Eleanor Rosch. Essa hipótese criticou o modelo clássico e questionou o formato das categorias, que são definidas pelos atributos que todos os membros da classe obrigatoriamente devem apresentar. Portanto nenhum exemplar poderia representar uma categoria melhor que outra uma vez que é condição de pertença (ROSCH, 1975, 1978).

A hipótese prototípica classifica a organização das categorias não por apresentar certos atributos em comum a uma entidade, mas por apresentar dimensões próximas ao qualificado ideal para uma dada categoria. A característica gradual dos exemplares demonstra a posição deles em relação ao centro da categoria, que, nesse ponto, são considerados membros mais prototípicos e, à medida que se afastam da centralidade, são chamados exemplares periféricos. Tendo em vista a categoria “Ave”, um pardal é um exemplar mais prototípico da classe em questão do que um pinguim, por exemplo. Essa comparação pode ser feita com muitas outras “Aves” que apontarão graus diferentes de prototipicidade, comprovando, assim, a gradualidade das categorias. O que não é claro nessa teoria, segundo Rodrigues-Leite (2004), é a linha que separa dentro de uma categoria os exemplares mais próximos de seus não exemplares.

A categorização prototípica foi revista por Varela, Thompson e Rosch (1991) que propuseram uma versão com ênfase na experiência. Para os autores, a cognição é ação incorporada e não pura representação de um mundo dado a priori, anterior a qualquer atividade cognitiva. É a atuação em um mundo, pautada em uma história da diversidade de ações desempenhadas por um ser no mundo (VARELA; THOMPSON; ROSCH, 2003).

Os autores nos dizem ainda que a habilidade de categorizar é inevitável a todos os seres. Até mesmo os mais simples são capazes de selecionar o que é vital a seu organismo, como alimentação, predadores, reconhecimento de membros da espécie, acasalamento.

Segundo Saraiva (2008), categorizar, ao contrário de ser um processo mental descontextualizado, é uma condição imprescindível à integração bio-psico-sócio-cultural-ecológica do indivíduo ao mundo. A categorização prototípica, nessa perspectiva, é condicionada pelo ambiente e pelo ator que nele atua.

1.2.3 Espaços Mentais

Tenuta (2010) afirma que a metáfora do *iceberg*, muito utilizada por linguistas cognitivos, mostra a relação entre a parte aparente da linguagem e as operações cognitivas que ocorrem durante o desenvolver dela. Diz também que a produção ou a interpretação de expressões linguísticas requerem a utilização de habilidades psicológicas gerais do tipo memória, atenção, percepção e realização de processos como categorização, abstração, mapeamento, projeção e integração conceitual.

Segundo Fauconnier e Sweetser (1996), durante os eventos de fala entre pessoas, espaços mentais são construídos, estruturados e ligados, motivados por fatores como a gramática, o contexto e a cultura. Nesse processo, o significado é construído a partir de uma rede de espaços que se formam no momento da enunciação, enquanto o discurso ocorre.

A partir disso, a estrutura linguística pode ser entendida como condutora do trabalho de estruturação dos espaços mentais que proliferam no desenvolver das atividades cognitivas inerentes aos seres humanos, como pensar e falar. Também fica clara a natureza dinâmica dos espaços mentais que se configuram e reconfiguram a cada processamento cognitivo em função da expressão linguística e do contexto.

Fauconnier e Sweetser (1996, p. 8-9) expõem que

A linguagem nos permite falar não só sobre o que é, mas também sobre o que poderia ser, o que será, do que se espera, do que se acredita, de hipóteses, do que é visualmente esperado, do que aconteceu, do que deveria ter acontecido, entre outros.

Conforme as intenções que pretendemos, referenciamo-nos a diversos fatos por meio da linguagem, que tem o poder de acionar os meios para construir os significados.

As estruturas linguísticas também têm a função de conectar espaços mentais, sendo chamadas de construtores de espaços mentais. Segundo Miranda (1999), em nível gramatical, as estruturas apresentam formas variadas, como sintagmas preposicionais, sintagmas adverbiais, conectivos, sentenças, marca de tempo e

modo verbal, que introduzem as marcas de diferença entre espaços e projetam um domínio em outro. Ainda conforme a autora, esses espaços são tipificados como:

- **crença:** *Eu acho* que a Maria casou-se;
- **imagens:** *Na foto*, Maria é morena;
- **hipótese, contrafactualidade:** *Se eu fosse você*, ia ao ABC;
- **escala:** Ele é *um tipo* de herói;
- **tempo** (situações temporais): A TV *anunciou à meia noite* a vitória do PMDB em Minas;
- **drama:** *No filme*, o ator brasileiro é americano;
- **lugar:** *No Brasil*, temos uma bela música;
- **modelo cultural** (domínio de atividades-jogo, campo da ciência, religião, esporte...): *Em sua religião*, Nossa Senhora não existe.

A ferramenta teórica espaços mentais não pode ser entendida como representações mentais preexistentes, ao contrário, ela se apoia nas estruturas estáveis como os modelos cognitivos idealizados. Os espaços mentais trabalham de forma *on-line* no processo de construção discursiva. Portanto, durante o discurso, eles vão sendo construídos, estruturados e reformulados e, por isso, são representações temporárias armazenadas na memória de curto prazo com a função de desenvolver conceituação aos atores do discurso.

Os domínios estáveis fornecem insumos para o trabalho dos espaços mentais, e estes muitas vezes se constituem de domínios diferentes ou de vários domínios. Nessa ótica, o processo de significação operado pelos espaços resulta de projeções entre domínios. Fauconnier (1997) afirma que, para falar ou pensar sobre determinados domínios (domínio-alvo), usamos estruturas de outro domínio (domínio-fonte) e do correspondente vocabulário. Dessa forma, a projeção entre os domínios ajuda-nos a entender as intenções dos atores do discurso, como também evidencia a presença de negociação conceptual da linguagem.

As projeções tornam possível a transferência de informações de um domínio a outro. Isso se torna visível no exemplo da metáfora do *tempo como espaço*, presente em muitas culturas. Em nossa concepção cotidiana, nos valemos de

espaço e movimento (domínios concretos – experienciados pelo corpo) para organizar nossa concepção de tempo (domínio abstrato): *As férias estão chegando, O ano está indo embora, O natal se aproxima...* (MIRANDA, 1999).

A projeção conceitual, segundo a teoria proposta por Fauconnier e Turner (2002), necessita de pelo menos quatro espaços mentais envolvidos no processo de projeção conceitual: espaço correspondente ao domínio-fonte (espaço influente 1), espaço que corresponde ao domínio-alvo (espaço influente 2), espaço genérico (onde ficam as estruturas abstratas resultantes da partilha entre os dois domínios) e o espaço mescla (onde se verifica a mistura dos domínios em evidência). O resultado desse processo é a eminência de um novo conceito que a nomenclatura oficial da teoria chama de estrutura emergente. Assim, o novo surgido não resulta da soma dos domínios projetados, mas é convencionado pela instância da projeção efetuada, bem como pelas nossas experiências cotidianas, pelas molduras culturais (MCI) e pelo nosso conhecimento prévio.

1.2.4 Redes de Integração Conceitual

Fauconnier e Turner (2002, p. 217) asseveram que

As pessoas fingem, imitam, mentem, fantasiam, iludem, enganam, consideram alternativas, simulam, constroem modelos e propõem hipóteses. Nossa espécie tem uma extraordinária habilidade para operar mentalmente sobre o irreal, e essa habilidade depende de nossa capacidade de efetuar integração conceitual.

As operações cognitivas efetuadas por seres humanos projetam domínios, realizam mapeamentos analógicos e integram ações em domínios emergentes. São essas atividades as responsáveis por atributos humanos inatos como a criatividade, por exemplo, que é originária das formas de pensamentos que produzem novas relações de significações e conceptualizações.

Salomão (2005, p. 159) acrescenta que mesclagem conceitual “São processos cognitivos, identificados e postulados no seio da teoria dos espaços mentais. Tratam das relações projetivas entre vários domínios conceptuais, que levam à emergência de um novo domínio (o domínio mescla)”. A compreensão de uma simples composição linguística pode ter inúmeros desdobramentos no que

tange à configuração de espaços mentais, bem como às redes que se formam em função da integração dos conceitos que culminam no significado. Para ilustrar esse processo, utilizamos um exemplo de Tenuta (2010) com o enunciado: *No meu sonho, Ana era namorada do Robin Willians*. Compreender o enunciado não nos causa nenhuma dificuldade. A nossa capacidade de desenvolver operações cognitivas cria inicialmente um espaço de cenário hollywoodiano, em que Robin Willian é o famoso ator americano, de outro lado cria um espaço que corresponde à minha vida pessoal com minhas amigas e seus namorados, e o espaço que corresponde ao sonho no qual Robin Willians é namorado de Ana é resultado da mesclagem. Cada espaço de entrada de dados do exemplo se filia a um domínio estável presente na cultura.

No processo de construção de significado resultante da integração conceitual, não conseguimos prever a ordem exata dos domínios, como em uma receita de bolo em que os passos são enumerados, essencialmente porque tudo ocorre de forma simultânea resultante de um processamento paralelo.

Isso é vantajoso para nos darmos com situações envolvendo múltiplos domínios e mesclas, do contrário, não se configurariam as redes de integração conceitual. Nessa ótica, podemos imaginar o processamento cognitivo serial no qual identificaríamos uma metáfora, para depois de outra mescla resultar em outra e assim sucessivamente até formar o resultado final, que possivelmente não seria a intenção do objeto provocador do significado. Portanto fica evidente que o processo de significação envolve uma rede de ações integradas que trabalham em níveis de frequências sincrônicas. Dessa forma, por mais que não seja possível prever a ordem dos acontecimentos cognitivos, eles obedecem a uma espécie de organização. Assim a integração conceitual circunstancialmente se filia a um dos quatro tipos de rede convencionada nos construtos teóricos de Fauconnier e Turner (2002), que são identificadas como redes Simplex, Especular, de Alcance Único e de Alcance Duplo.

As redes de integração conceitual apresentam as características expostas na sequência.

- Simplex: apresenta conexões de tipos de enquadres para tipos de valores, ou mesmo, um espaço influente com a função sem valores (ex. relação de

parentesco: pai), e um espaço influente que contém elementos sem enquadre (ex. Pedro, Sandra). No espaço mescla, as funções enquadradas se fundem aos elementos do segundo espaço, formando a relação função e valor (ex. Pedro é pai de Sandra).

- Especular: todos os espaços componentes compartilham o mesmo enquadre, uma vez que a mescla herda a estrutura organizacional. Por exemplo, para entendermos o enunciado “Em 1984, João do Pulo bateu seu próprio recorde de 1980, no salto triplo olímpico”, construímos uma rede de integração conceitual em que o espaço influente 1 (João do Pulo, 1980, salto triplo) tem um enquadre organizacional que espelha no espaço influente 2 (João do Pulo, 1984, salto triplo), o enquadre é herdado pela mescla resultando na estrutura emergente João do Pulo vence a si mesmo.
- Alcance único: os dois espaços de entradas são de enquadres distintos, mas somente um é projetado para organizar a mescla. No exemplo “Empresários de TV boxeiam entre si para ganhar audiência”, temos o espaço influente 1 (competição, boxeadores, nocaute), espaço influente 2 (empresários, audiência) e a mescla que herda o enquadre do espaço 1 formando o resultado “empresário boxeador”, que, na competição profissional, luta com outros profissionais por audiência.
- Alcance duplo: os dois espaços de entrada são projetados por enquadres diferentes e, na maioria das vezes, divergentes. O enquadre que orienta a mescla, nesse caso, se organiza com conteúdos dos dois espaços de entradas e forma uma estrutura emergente própria. Os sistemas aplicativos comerciais ilustram bem esse tipo de rede. Um sistema de controle de uma locadora de veículos, por exemplo, é estruturado de forma que os espaços de entrada são organizados, de um lado por um enquadre que representa o ambiente natural com elementos do mundo real (ficha de clientes, ficha de veículos, comprovante de locação em papel), e do outro lado por um enquadre formado por comandos de entrada de dados e processamento de informações, que permitem a estruturação no banco de dados virtual. O espaço mescla, nesse exemplo, é estruturado tanto pelo enquadre do ambiente de trabalho real da locadora (preencher ficha de cliente, preencher ficha de veículos) quanto pelo ambiente virtual (cadastrar cliente, cadastrar veículo, listar novos clientes) (RODRIGUES-LEITE, 2004).

1.3 TEXTO, GÊNERO, HIPERTEXTO E MULTIMODALIDADE

1.3.1 Caracterizando Texto

Para Marcuschi (2008, p. 72), “O texto pode ser tido como um tecido estruturado, uma entidade significativa, uma entidade de comunicação e um artefato sócio-histórico”. Para o autor, texto³ é uma reconstrução do mundo, uma entidade comunicativa que gera uma unidade de sentido.

Pensando assim, texto é a unidade linguística comunicativa básica, utilizada pelas pessoas para se comunicarem umas com as outras, uma vez que tudo que alguém tem a dizer a outrem não são palavras nem frases, mas são textos. Dessa forma, entendemos que textos são manifestações naturais da linguagem humana configuradas em uma língua, produzidas em uma situação concreta, dotadas de sentido e que visam a um objetivo comunicativo.

Segundo Beaugrande (1997, p. 10), “O texto é um evento comunicativo em que convergem ações linguísticas, sociais e cognitivas”. Consoante ao autor, falamos de texto como sendo um evento comunicativo com toda a composição que o institui, como regras, espaço, atores, funções e forma; e, para gerar sentido, é necessário haver colaboração entre os atores que constroem e que consomem esse produto. Assim, textos caracterizam uma atividade sociointerativa porque “os falantes e escritores da língua, ao produzirem textos, estão enunciando conteúdos e sugerindo sentidos que devem ser construídos, inferidos, determinados mutuamente” (MARCUSCHI, 2008, p. 77).

Ainda para Marcuschi (2008), o que atribui sentido a um texto é a discursividade, inteligibilidade e articulação que ele dispara, tendo em vista que eles operam em contextos comunicativos e que seja processado por alguém, ou mesmo, produza sentido em um ator sócio-historicamente situado.

³ A caracterização de texto apresentada é oriunda dos estudos da linguística textual que se conceitua tecnicamente, segundo Marcuschi (2008, p. 73), como sendo “o estudo das operações linguísticas, discursivas e cognitivas reguladoras e controladoras da produção, construção e processamento de textos escritos ou orais em contextos naturais de uso”.

Assim, a existência de um texto está subordinada à condição de interpretação de alguém, caso contrário não reflete sentido e automaticamente não é um texto. O que determina a atribuição de sentido é o ambiente em que o evento comunicativo se instaura em consonância com as condições socioculturais do ator que processa, através das estratégias, das expectativas, dos conhecimentos linguísticos e não-linguísticos e, com isso, interpreta a informação veiculada no texto.

Portanto, compreensão é uma atividade colaborativa que se dá na interação autor-texto-leitor, condicionada pelo meio histórico-cultural em que o sujeito interpretativo está imerso. Durante o processamento de um texto (evento), a compreensão ocorre em instanciamentos de textualidade na proporção em que o leitor tem acesso à configuração do sentido.

1.3.2 Uma Noção de Gênero Textual

A comunicação realizada diariamente dentro da sociedade é constituída de ações comunicativas que usam estratégias com o fim de alcançar objetivos e metas específicas. Essas ações se constituem de um conjunto de instruções que formam enquadres com funções comunicativas bem definidas capazes de ser reconhecidas pela sua forma e sua configuração. Assim, os gêneros são as estratégias comunicativas que subsidiam a linguagem em uso dentro de um contexto social em seus diversos formatos e eventos comunicativos.

Segundo Mascuschi (2008, p. 149), gênero é “uma categoria cultural, um esquema cognitivo, uma forma de ação social, uma estrutura textual, uma forma de organização social e uma ação retórica”. Para o autor, um gênero funciona dentro de um fato social, em que cada um deles tem um propósito claro que determina o meio de circulação. Por exemplo, a finalidade de uma monografia é obter um título, uma chamada publicitária promove a venda de um produto, uma receita culinária orienta no preparo de um tipo de alimento, entre outros (MARCUSCHI, 2008)

O autor defende ainda que toda comunicação verbal se dá por algum gênero e também por algum texto, ou ainda, por meio de textos realizados através de um gênero. Assim, “os gêneros textuais operam, em certos contextos, como formas de

legitimação discursiva, já que se situam numa relação sócio-histórica como fontes de produção que lhes dão sustentação” (MARCUSCHI, 2008, p. 154)

Ainda para Bronckart (2001), os gêneros são entidades poderosas que, na produção textual, não nos permitem escolhas livres nem aleatórias, mas limitam nossas ações condicionando a identificação de um gênero que melhor se adapta a fatores como o léxico, o grau de formalidade e a natureza do fato comunicativo. Nessa ótica, seguindo a discussão do autor, os gêneros constituem ações de linguagem que exigem do agente produtor uma série de decisões para as quais ele precisa atuar com competência. São elas: a) a escolha do gênero a partir do rol de gêneros existentes que julgue mais adequado para o contexto e a intenção comunicativa; b) a aplicação do gênero que facultará ao produtor acrescentar algo à forma ou dentro da necessidade até mesmo recriá-la.

Portanto, entendemos gêneros textuais como o envoltório modelador do evento comunicativo, a entidade capaz de projetar o texto dentro do seu objetivo comunicativo de forma universalizada com suas características imanentes que o discrimina e o qualifica.

1.3.3 O Hipertexto: um novo espaço da escrita

Como discutido anteriormente sobre texto e sua demanda de sentido, observamos que, para ele se constituir de fato, é imprescindível estar socioculturalmente situado, de forma que a sociedade se use dos textos e, conseqüentemente, dos gêneros para exercer as atividades comunicativas. Tanto os textos como os gêneros estão em constante atualização por terem seu ciclo de vida condicionado à evolução da sociedade.

Diante do surgimento da cultura eletrônica, o texto ganha um novo suporte tecnológico que é a tela dos computadores, e, com as novas tecnologias digitais, a sociedade parece tornar-se “textualizada”⁴, ou mesmo, passa a operar no plano da escrita (YATES, 2000).

⁴ Termo utilizado por Yates (2000) para designar a radicalização do uso da escrita no suporte computacional.

Para Marcuschi (2001, p. 82), hipertexto é o “[...] espaço cognitivo que exige a revisão de nossas estratégias de lidar com o texto. Sobretudo as estratégias que dizem respeito à continuidade textual. Pois o novo espaço não é mais linear nem se comporta numa direção definida”.

Já Xavier (2005, p. 171) conceitua hipertexto como sendo “uma forma híbrida, dinâmica e flexível de linguagem que dialoga com outras interfaces semióticas, adiciona e condiciona à superfície formas outras de textualidade”. Acrescenta que, na esteira da leitura, emergimos numa tecnologia de linguagem em que a apreensão de sentido não advém somente das palavras, mas unidas a elas outras semioses como sons, gráficos e diagramas,

[...] todos lançados sobre uma mesma superfície perceptual, amalgamados uns sobre os outros formando um todo significativo e de onde sentidos são complexicamente disponibilizados aos navegantes do oceano digital (XAVIER, 2005, p. 171).

Cavalcante (2005) defende que a natureza do hipertexto está nos seus constituintes internos: os nós e os *links*; são eles que permitem a composição arquitetônica do hipertexto de forma a remeter o leitor a outros espaços de informação. Para a autora, os *links* promovem ligações entre nós (blocos informacionais) que podem ser: outros textos, palavras, parágrafos, endereçamento. Dentro dessas ligações, os nós não necessariamente precisam estabelecer relações sêmicas entre si, ou mesmo, “as ligações possíveis não formam necessariamente a tessitura daquele texto específico, mas promovem a abertura para outros textos, mas nunca qualquer texto” (CAVALCANTE, 2005, p. 166).

Logo, o hipertexto apresenta-se como um novo espaço da escrita, revestido de multimodalidade por natureza e evoca novas atitudes cognitivas de seus usuários, tanto do produtor do texto quanto do leitor que deverá ser mais crítico ao construir a leitura pelos vastos caminhos que entrelaçam os *links* e os nós dentro da rede mundial.

1.3.4 Definindo Multimodalidade

Os gêneros textuais, em geral, no seu estado de uso, são compostos, além dos caracteres alfanuméricos, por imagens, figuras, rodapés, gráficos, entre outros,

o que representa a presença da multimodalidade nesses eventos comunicativos. Todo arranjo visual existente no gênero, ou seja, diagramação, cores, figuras, tipo de papel são chamados de multimodalidade.

Descardesi (2002, p. 20) afirma que

Qualquer que seja o texto escrito, ele é multimodal, isto é, composto por mais de um modo de representação. Em uma página, além do código escrito, outras formas de representação como a diagramação da página (layout), a cor e a qualidade do papel, o formato e a cor (ou cores) das letras, a formatação do parágrafo, etc. interferem na mensagem a ser comunicada. Decorre desse postulado teórico que nenhum sinal ou código pode ser entendido ou estudado com sucesso em isolamento, uma vez que se complementam na composição da mensagem.

Já para Kress e Van Leeuwen (1996), o texto multimodal é aquele cujo significado se realiza por mais de um código semiótico. Segundo os autores, um conjunto de modos semióticos está envolvido nas fases de produção e leitura dos textos. Cada espécie de semiose tem suas potencialidades de representação e comunicação e são produzidas culturalmente. Dessa forma, o autor do texto, ao produzi-lo, considera contextos históricos, sociais e a até mesmo as perspectivas do construtor do signo usado.

Com as novas práticas sociais condicionadas pela tecnocracia, o evento comunicativo (texto) se utiliza cada vez mais de multissemos e de tecnologias digitais. Kress (1998) denominou esse fenômeno de “guinada para o visual”. O autor parte do pressuposto de que a expressão material do texto é sempre importante e que são os objetos materiais que resultam de uma variedade de práticas representacionais que condicionam o leque de significação disposta nos textos com essas características.

O suporte do texto está intimamente ligado à tecnologia disponível e, conseqüentemente, ao grau de multimodalidade que se pode utilizar. O suporte do texto digital e a *World Wide Web*, sem dúvida, são o espaço que melhor permite um amplo trabalho com imagens em movimento, cores, disposição, som e interatividade.

A multimodalidade não é uma exclusividade dos hipertextos digitais e muito menos da cibercultura. Já na Idade Média, os textos eram compostos por caracteres alfanuméricos e *layout* e se utilizavam de sumários e índices para tornar possíveis referências entre um e outro ponto do texto, citar um novo exemplo e outras formas

de indexação. Com essa configuração, aliada ao surgimento da imprensa, em meados do século XV, é que resultou no livro tal como conhecemos hoje.

Portanto, a multimodalidade está presente em todo processo de evolução da escrita e é no estágio atual – a escrita digital – que ela se exerce em sua plenitude, devido à natureza multimídia do hipertexto acrescentada da possibilidade de movimentos e sons.

1.4 LEITURA E COMPREENSÃO: ASPECTOS COGNITIVOS E METACOGNITOS

1.4.1 Definindo Leitura

A leitura, tradicionalmente, é entendida como um processo de decodificação do código linguístico com a finalidade de decifrar sinais gráficos e organizar as estruturas sintáticas nas sentenças. Esse conceito é tributário da concepção de l

eitura que vê o aprendizado dessa prática como o acesso às primeiras letras, que seria acrescido linearmente do reconhecimento das sílabas, das palavras e das frases, que, em conjunto, formariam os textos e, após o conhecimento dessas unidades, o aluno estaria apto a ler e a escrever (CAGLIARI, 1989). A concepção de leitura como decifração de signos linguísticos percebe a leitura como um processo linear e cumulativo.

Para Allende e Condemarin (1987), o processo de leitura compreende um conjunto de operações parciais e, nesse contexto, a decodificação é uma operação inicial e não pode ser confundida como a totalidade do processo. Na concepção dos autores, o processo de leitura compreende etapas que invocam capacidades humanas, como: a) decodificação para identificar um signo gráfico por um nome ou por um som; b) transformação dos signos em linguagem oral; e c) decifração do código de uma mensagem para captar seu significado.

Segundo Kleiman (2004), a leitura é uma atividade complexa devido aos múltiplos processos cognitivos utilizados pelo leitor ao construir o sentido de um texto, especialmente porque “não se dá linearmente, de maneira cumulativa, em que a soma do significado das palavras constituiria o significado do texto” (TERZI, 2002, p. 15).

Assim, ler é, antes de tudo, atribuir sentido ao texto decodificado posto em interação, e esse é o gatilho que desencadeia uma série de processos cognitivos e metacognitivos que norteiam as atividades cerebrais que culminarão no significado que um dado leitor atribuirá a um texto.

É importante ressaltarmos que o significado atribuído por um leitor a um texto dificilmente será idêntico ao de outro leitor. Esse fato ocorre devido à carga de memória aplicada aos processos cognitivos variar de leitor para leitor, uma vez que também variam as experiências de cada um dentro do mundo.

Nessa ótica, a leitura depende exclusivamente da ação cognitiva do leitor, o qual atua nesse processo com sua própria “bagagem” cultural e participa efetivamente da construção do significado nas interações com o texto e com o autor. As interações comunicativas ocorrem porque

[...] a leitura não se configura como um processo passivo [...]. Por exigir descoberta e recriação, a leitura coloca-se como produção e sempre supõe trabalho do sujeito-leitor [...], então o leitor, além de partilhar e recriar referenciais de mundo, transforma-se num produtor de acontecimentos, em função do aguçamento da compreensão e de sua consciência crítica (SILVA, 1991, p. 25).

Kleiman (2004, p. 27) assevera que

[...] leitura implica uma atividade de procura pelo leitor, no seu passado de lembranças e conhecimentos, daqueles que são relevantes à compreensão de um texto, que fornece pistas e sugere caminhos, mas que certamente não explicita tudo o que seria possível explicitar.

Na visão da autora, leitura é um processo interativo, porque interliga níveis de conhecimento como o linguístico, o textual e o de mundo. Na compreensão de um texto, o leitor utiliza-se do seu conhecimento prévio que é constituído pelo conjunto de conhecimentos armazenados durante as experiências cotidianas.

Segundo Foucambert (1994, p. 5), “Ler significa ser questionado pelo mundo e por si mesmo, significa que certas respostas podem ser encontradas na escrita,

significa poder ter acesso a essa escrita, significa construir uma resposta que integra parte das novas informações ao que já se é”.

A leitura, para Soares (1996), é entendida como fruto de uma interação, de forma que o texto não preexiste à sua leitura, não é aceitação passiva, mas construção ativa. Dessa forma, é no processo de interação disparado pela ação da leitura que o texto se constitui.

Ainda nessa linha de raciocínio, Solé (1998) afirma que a leitura é um processo interativo e construtivo que se configura nas relações e nas negociações entre as diferentes partes do texto e o conhecimento prévio do leitor. Portanto, o processo de compreensão envolve a cumplicidade entre as particularidades do texto, as intenções do leitor, as circunstâncias em que ocorre a leitura e as características pessoais do leitor.

Somando as perspectivas dos autores citados neste item, entendemos que a leitura é uma ação interativa, ou mesmo um processo constituído de fases que depende da motivação do leitor, em especial de relacionar o conhecimento percebido aos armazenados nos construtos do cérebro; é ainda a função maior de autor quando cria o texto paralelo ao posto em interação; é também a oportunidade suprema de navegar nos nós e nos *links* das estruturas de conhecimento, de reorganizar e reestruturar o que existe e ainda construir o que ainda não há. Todos esses aspectos encontram-se em um imbricamento que resulta na compreensão da leitura que é o objeto de estudo desta pesquisa, e o essencial para esse resultado é, sem dúvida, a ação do conhecimento prévio, obviamente disparado na ação da leitura.

1.4.2 O Conhecimento Prévio

O leitor, ao interagir com os textos, percebe e atribui sentido a lacunas e informações incompletas que lá são colocadas pelo autor de forma intencional ou não. Isso acontece para que, nas atividades de interpretação, o leitor lance mão de suas experiências, crenças, opiniões, interesses, sua maturidade em relação a leituras anteriores e ainda seu conhecimento de mundo (BRITO; VERRI, 2004).

Segundo Brito e Verri (2004), esse conjunto de elementos subjetivos e individuais é que determina o tipo de leitura que será realizado, porque é por meio dele que o leitor interfere no texto, atribuindo-lhe sua visão pessoal, a fim de construir o mundo textual a partir de suas vivências.

Para Braga e Silvestre (2002), a leitura não é compreendida como um processo de mera decodificação do código impresso, por envolver o conhecimento prévio, como as experiências cotidianas e com leituras anteriores que integram o conhecimento adquirido no momento da leitura. Assim, o conhecimento anterior do leitor se une às informações novas contidas no texto, possibilitando a transposição do estímulo textual para um nível mais elaborado de inferências (BRANDÃO; SPINILLO, 1998).

Sobre a ação do conhecimento prévio, Kleiman (2004, p. 25) acrescenta que

A ativação do conhecimento prévio é, então, essencial à compreensão, pois é o conhecimento que o leitor tem sobre o assunto que lhe permite fazer inferências necessárias para relacionar diferentes partes discretas do texto num todo coerente. Este tipo de inferência, que se dá como decorrência do conhecimento de mundo e que é motivado pelos itens lexicais no texto é um processo inconsciente do leitor proficiente.

Para Goulemot (2001, p. 113), em uma perspectiva sociocultural, ler é “fazer emergir a biblioteca vivida, quer dizer, a memória de leituras anteriores e de dados culturais”. A ideia de biblioteca, evidenciada pelo autor, refere-se à cultura coletiva, ou mesmo a um conjunto de valores que entra em cena nas atividades de significação, em que “nasce, em grande parte, tanto desse exterior cultural quanto do próprio texto e é bastante certo que seja de sentidos já adquiridos que nasça o sentido a ser adquirido” (GOULEMOT, 2001, p. 114).

Nessa ótica, o leitor, que uma vez foi considerado um mero receptor de mensagens, assume o papel de coautor, uma vez que o sentido construído na leitura ocorre na proporção em que o leitor desempenha uma função ativa no processo e estabelece relações entre o seu conhecimento anterior e o conhecimento que está sendo tecido durante a atividade de leitura.

A familiaridade do leitor com vários tipos de textos e sua carga de memória, que é enriquecida gradativamente com experiências vivenciadas no cotidiano, interfere de maneira significativa na compreensão de leitura.

1.4.3 Processamento da Leitura

Segundo Kato (2007), o processamento da informação se dá por dois processos o *top-down* (descendente) e o *bottom-up* (ascendente). No processamento descendente, a abordagem se dá de forma macro, não linear com maior uso das capacidades de inferências e deduções dos leitores, ao passo que, no ascendente, o *input* visual é analisado linearmente decomposto em partes que somadas compõem o significado. Esses processos servem para categorizar leitores em ascendentes e descendentes.

Para Kato (2007, p. 50), o descendente “é o leitor que aprende facilmente as ideias gerais e principais do texto, é fluente e veloz, mas, por outro lado, faz excessos de adivinhações, sem procurar confirmá-las com os do texto, através de uma leitura ascendente”. A autora diz ainda que esses leitores utilizam mais seu conhecimento prévio em oposição à informação que o texto veicula. Já o ascendente “[...] constrói o significado com base nos dados do texto, fazendo pouca leitura nas entrelinhas”, mas que, ao contrário do descendente, não antecipa conclusões, mas é vagaroso e tem dificuldade de condensar ideias por não saber distinguir o que é mais importante do que é ilustrativo (KATO, 2007, p. 51).

A autora fala também em uma nova categoria de leitores, os maduros, os que avaliam estrategicamente quando usar um ou outro processo, fazendo com que eles se complementem na busca do objetivo final. Esse leitor usa as estratégias metacognitivas quando controla conscientemente o seu comportamento e controla ativamente o processamento da leitura (KATO, 2007).

A utilização das estratégias de leitura não se dá de forma aleatória, o leitor proficiente usa-as de acordo com a situação de interação instanciada no momento. Em condições em que o estímulo visual condiciona relações de informação armazenada na memória por meio do conhecimento prévio, é provável que a opção do leitor seja por leitura descendente, por haver condições de inferências, levantamento e checagem de predições, ao passo que, em situações opostas, o leitor opta pelo processo ascendente, que possibilita o desprendimento de algum significado por meio da sintetização das unidades decompostas do *input* visual.

1.4.4 Entendendo a Memória

A memória também exerce influência no processo de compreensão leitora. Ao lermos, temos a intenção de dar sentido ao texto, e isso gera a ativação de diferentes memórias que, em um trabalho coordenado, propicia o objetivo almejado. O próprio conhecimento prévio evocado durante a significação das leituras é um tipo de memória.

A memória, segundo Gonçalves (2008), é um armazém que tem uma secção de arquivo (memória de longo prazo) e uma secção ativa (memória de trabalho), com a função de: identificar objetos, reconhecer problemas, tomar decisões e executar ações. Para realizar essas atividades, o cérebro busca da memória de longo prazo para a memória de trabalho todos os conhecimentos relevantes para a questão posta em interação, e, após o processamento do que foi confrontado, os conhecimentos regressam ao arquivo, muitas vezes modificados, devido às novas aquisições derivadas da experiência e da reflexão (GONÇALVES, 2008).

Izquierdo (2002, p. 9) assevera que “memória é a aquisição, a formação, a conservação e a evocação de informações”. Para o autor, aquisição é sinônimo de aprendizagem e evocação é o mesmo que recordação. Diz também que somos o que somos devido ao acervo de nossa memória, ninguém pode fazer o que não sabe, nem mesmo comunicar o que não conhece e, ainda, não fazem parte de nós episódios que nunca vivemos. Portanto não podemos ser o que não está em nossa memória (IZQUIERDO, 2002).

Ainda segundo Izquierdo (2002, p. 27), as memórias classificam como:

- **memória de trabalho:** é responsável pela gestão de todas as atividades cognitivas. Para executar essa gerência, a memória de trabalho aciona as memórias preexistentes para confrontar com as informações recepcionadas e então classificá-las em novas ou velhas e se relevantes ou não para agregarem ao repositório, ou mesmo, o sujeito adquirir o conhecimento;
- **memórias declarativas:** são as que registram fatos, eventos ou conhecimentos. São subclassificadas em episódicas ou bibliográficas, que registram os eventos de que participamos (uma formatura, um filme), e semânticas, que armazenam os conhecimentos gerais alcançados nas experiências (conhecimentos de português);
- **procedurais:** registram capacidades ou habilidades motoras ou sensoriais, são hábitos que efetuamos automaticamente, como andar de bicicleta, nadar, saltar e soletrar;

- **memória de curta e longa duração:** são aquelas que levam tempo para ser consolidadas e podem ser evocadas a qualquer momento para atuar em um dado processamento, e a de curta duração é aquela que dura poucas horas, ou mesmo, o tempo necessário para que as de longa duração se consolidem.

Compreender um texto exige, entre outros fatores, o trabalho das memórias. Ao ler, a memória de trabalho recupera das memórias de armazenamento todos os conhecimentos e repertórios necessários para realizar a correlação de dados e construir um significado para o texto. Dessa forma, a informação é processada e interpretada e, após esse estágio, será novamente conduzida às fontes de armazenamento do cérebro.

A investigação realizada neste relato reconhece a importância da memória no processo de compreensão leitora, porém não a evidencia como variável de observação.

1.4.5 Estratégias Cognitivas e Metacognitivas

Em leitura, estratégias cognitivas designam os princípios que regem o comportamento automático e inconsciente do leitor, são elas que guiam o desempenho do compreendedor durante a atividade leitora. Em complementação a esse processo, as estratégias metacognitivas designam os princípios que regulam a desautomatização consciente das estratégias cognitivas (KATO, 2007).

As estratégias cognitiva e metacognitiva são usadas pelo leitor com o fim de conseguir um bom entendimento do texto. A primeira atua para facilitar o processamento textual em termos de produção ou compreensão sem a intenção do leitor; e a segunda instiga a reflexão do leitor sobre o próprio processo de compreensão a partir do monitoramento consciente das estratégias cognitivas.

O que faz diferir as estratégias cognitivas das metacognitivas é o grau de consciência dispensado à atividade. Como podemos analisar no exemplo dado por Kato (2007), em que ela aponta um equívoco natural oriundo de uma estratégia cognitiva de interpretar a primeira oração como principal e não subordinada. Esse

fato foi detectado em um experimento realizado pela autora no trabalho *Estratégias gramaticais e lexicais em leitura em língua estrangeira*⁵.

No experimento, o aluno realizou uma leitura silenciosa de um texto em inglês de conteúdo do seu interesse que, obviamente, carregava o efeito de ter a primeira oração com subordinada contrariando a regra geral de ser a principal. Na sequência, o aluno deveria traduzir livremente o texto lido. Durante a tradução em voz alta, ele fez pausas que indicaram inicialmente perplexidade ou planejamento ao se deparar com o problema das orações, que evidencia a tendência à interpretação a partir da regra geral adotada pela ação cognitiva. Em seguida, sua percepção detectou falha na sua interpretação e ele se autocorrigiu, configurando, assim, que o aluno usou estratégias metacognitivas para monitorar a compreensão daquela parte do texto e atribuir sentido em conformidade à sua língua materna.

Dessa forma, identificamos que o fato se fundamenta na estratégia cognitiva da *pressuposição* no momento em que, inconscientemente, o aluno assume, por regra automática, que a primeira oração é a principal e não subordinada até perceber o equívoco e, evidentemente, iniciar a operação de avaliação. Nessa ótica, a estratégia metacognitiva entra em ação ao perceber a fuga do padrão, que dispara algo como um módulo de consciência, que assume a tarefa a partir do estado de alerta e inicia um trabalho de recopilação da informação.

Como estratégias cognitivas utilizadas por leitores, segundo Nunes (2008), identifica-se:

- inferência: organiza traços mínimos significativos a partir da experiência e do conhecimento prévio, linguístico e de mundo. O conhecimento engramado na memória do leitor, gera inferências que são fornecidas em certo contexto, resultando em informações semânticas novas;
- pressuposição: recupera as informações dadas, ou mesmo o que já é conhecido pelo leitor, difere da inferenciação que atua nas informações novas;
- predição: é a habilidade de selecionar um grande número de pistas produtivas necessárias à elaboração das adivinhações, ocorre de forma rápida e com pouco esforço consciente. A predição é uma habilidade de antecipação vital

⁵ Ver Kato (1984).

na leitura, demonstra conhecimento prévio por meio da soma total de todas as experiências e desenvolvimento da linguagem e pensamento. O que parece ser intuitivo é, na verdade, o resultado do conhecimento aprendido;

- previsão: é a antecipação com base em informações sintáticas e semânticas, sua importância se dá por afastar as opções incorretas, evitando custo desnecessário no momento do processamento;
- seleção: habilidade de eleger somente os índices mais produtivos entre todos os disponíveis no texto, uma vez que o processamento cognitivo os traduz em conceitos. Essa atividade se responsabiliza pela economia do procedimento.

Conforme Feltes (2008, p. 249), “a metacognição é uma operação de segundo nível, tomando representações e processos de primeiro nível como objeto cognitivo”. Assim “[...] é também cognição, apenas atuando em níveis mais altos de representação, de forma altamente recursiva”.

Portanto processo metacognitivo ocorre após o cognitivo, como monitoramento e avaliação do objetivo estabelecido no ato da leitura. Sabemos como os processos cognitivos operam, tanto que podemos descrevê-los. Dentro de cada um deles, a ação metacognitiva é disparada em função do grau de consciência empregado com o fim de julgar a eficácia do conhecimento.

2 METODOLOGIA

Cervo, Bervian e Silva (2007, p. 27) definem método nas ciências como “o conjunto de processos empregados na investigação e na demonstração da verdade”. Para os autores, um método não pode ser inventado por depender efetivamente do objeto da pesquisa e também não deve ser considerado um modelo sem margem de erros, mas “um conjunto ordenado de procedimentos que se mostrou eficiente, ao longo da história, na busca do saber” (CERVO, BERVIAN; SILVA, 2007, p. 28).

Ainda segundo os autores, técnicas são os meios corretos de executar as operações que constituem o método de forma geral. Existem operações que têm efeito em qualquer pesquisa, como a observação, a descrição, a comparação, a análise e a síntese (CERVO, BERVIAN; SILVA, 2007).

2.1 CLASSIFICAÇÃO DA PESQUISA

Quanto à definição de tipologia de pesquisa apresentada por Cervo, Bervian e Silva (2007), a presente pesquisa se classifica como experimental por manipular diretamente as variáveis relacionadas ao objeto de estudo. Nesse tipo de pesquisa, manipulamos variáveis independentes com o fim de observar o que ocorre nas dependentes. Nesta pesquisa, manipulamos as variáveis independentes MCI, EC (estratégias cognitivas) para observarmos a variável dependente **compreensão**.

A pesquisa pode também ser considerada bibliográfica e direta, segundo Medeiros (2009), por procurar dados em variadas fontes secundárias, levantando-os no local em que o fenômeno ocorre (pesquisa de laboratório).

O método científico adotado é o hipotético-dedutivo, proposto por Popper (1978), devido ao trabalho ter início em um problema para o qual são elaboradas hipóteses a partir das quais as conclusões obtidas podem ser deduzidas e também podemos fazer previsões que podem ser falseadas ou aceitas.

Em relação à forma de abordagem do problema, apostamos na pesquisa qualitativa. Assim, os dados coletados tiveram tratamento qualitativo para a inferenciação nas fases de interpretação e análises.

2.2 DEFINIÇÃO DA PESQUISA

Partindo do pressuposto de que a maior parte da informação veiculada nos tempos atuais é expressa por meio de textos, que, em sua maioria, se apresentam em forma multimodal, acreditamos que eles medeiam a interação entre conhecedor e conhecimento. E pensando que o suporte utilizado pelo texto, impresso ou hipertexto, interfere no grau de compreensão da leitura é que se pretende discutir essa variável dentro do conjunto de funções cognitivas desenvolvidas pelo cérebro no processamento da leitura.

Para isso, observamos as relações de instanciamento de conteúdos provenientes dos processos de captação da informação no meio, inter-relacionamentos dos objetos candidatos ao trabalho de internalização, processamento, tratamento e depósito na estrutura cognitiva dos seres humanos a partir de textos multimodais expressos em suportes impressos e digitais.

A pesquisa de caráter analítico tem suas bases em instrumentos e experimentos que foram aplicados a alunos do curso técnico integrado ao médio da área de informática do Instituto de Educação, Ciência e Tecnologia do Tocantins Campus Palmas.

O primeiro experimento se constituiu de dois textos revestidos essencialmente da natureza pretendida (um impresso como um livro e um hipertexto formado por labirintos hipertextuais semióticos, ambos com altos índices de multimodalidade) com a função de testar o condicionamento do aluno quanto à compreensão do objeto posto em interação.

O segundo experimento se constituiu também de dois textos com características diferentes: um impresso (artigo científico) formado somente por caracteres alfanuméricos e poucas marcas de multimodalidade (somente negrito e

enumeração das partes do texto) e outro hipertexto multimodal, com uso de *links*, cores, gráficos e imagens.

Além dos textos que provocam a disposição do aluno à atividade cognitiva, foi criado um instrumento de checagem para mensurar o quantitativo de compreensão que ele conseguisse alcançar, tendo em vista o conhecimento adquirido no experimento. Esse experimento se resumiu em questionários e testes de resolução de problemas que necessitavam da compreensão da informação posta em interação nos textos.

Os dois experimentos se uniram para contribuir com a solução dos questionamentos que norteiam esta pesquisa:

- Qual dos dois suportes de textos multimodais (impressos ou digitais) favorece a compreensão de alunos do ensino médio?
- A multimodalidade em textos suportados pelo computador facilita a compreensão?

2.3 OBJETIVO DA PESQUISA

Os objetivos desta pesquisa, já mencionados na introdução, estavam sintonizados com as questões de pesquisa e serviram de base para a elaboração dos instrumentos e dos experimentos utilizados – coleta de dados – e para a elaboração de todas as etapas metodológicas que nortearam este estudo.

2.3.1 Objetivo Geral

- Analisar a compreensão leitora em textos em suportes impressos e digitais com características diferenciadas de multimodalidade.

2.3.2 Objetivos Específicos

- Investigar o processamento da compreensão leitora em textos multimodais impressos com maiores e menores índices de multimodalidade.
- Analisar o processo de compreensão de leitura em textos multimodais suportados por computador.
- Proceder à investigação do processo de construção de MCI relativos aos conceitos apresentados nos textos experimentais.
- Analisar as estratégias cognitivas empregadas no processo de compreensão leitora.

2.4 DESCRIÇÃO DOS EXPERIMENTOS

Como já mencionado anteriormente, o trabalho se fundamentou em dois experimentos, em que a diferença entre eles estava no contraste dos textos postos em interação: no primeiro, os dois textos tinham a mesma natureza multimodal, o que mudou foram os suportes; e no segundo, os textos eram caracteristicamente diferentes tanto na composição quanto no suporte.

2.4.1 Experimentos I e II

Os textos apresentados no primeiro experimento apresentavam aspectos multimodais tanto no impresso quanto no digital, ao passo que no segundo, os textos tinham índices de multimodalidade distintos (no impresso, baixo índice de multimodalidade e, no digital, alto índice). O gênero utilizado no primeiro

experimento foi um livro-texto elaborado a partir de elementos de discurso científico, didático e cotidiano, enquanto do segundo, foi um artigo científico.

No impresso do experimento I, foram confeccionados dois textos com aproximadamente onze páginas configuradas em papel A4, retirados do livro *Desenvolvendo aplicações com UML 2.0 - Unified Modeling Language*, de Ana Cristina Machado. O livro tem como público-alvo analistas de sistemas, gerentes e programadores de computador, publicado no Rio de Janeiro, em 2004. A escolha dos textos se deu justamente pela presença marcante da multimodalidade expressa por meio de imagens, gráficos, diagramas, esquemas e figuras. Nesses textos, a presença do hipertexto é evidente em várias notas de rodapé e ainda em caixas de saiba mais. Durante o desvendar do assunto, a autora se preocupa em associar conceitos técnicos a semioses estáticas, principalmente a diagramas em virtude especialmente da natureza abstrata dos conteúdos. Ao final das leituras, o aluno conta, além da expressão linguística, com vários outros artefatos semióticos que proporcionam a completude da informação disponibilizada.

Também no impresso do experimento II, confeccionamos um texto com cinco páginas, configuradas em papel A4, adaptado do artigo científico *A história da UML e seus diagramas*, de Thânia Clair de Souza Vargas, que tem como público-alvo comunidades acadêmicas de áreas afins a informática e a estatística. Desse texto, retiramos a maior parte das marcas de multimodalidade, ficaram somente a enumeração dos capítulos e o negrito.

No digital dos dois experimentos, os textos apresentados para os alunos eram exatamente os mesmos dos impressos, com os mesmos assuntos retirados de dois capítulos do livro UML de Ana Cristina Melo e de Thânia Clair de Souza Vargas, o que acrescentamos foram: movimentos, formas, sons, cores e *links*. Assim, os textos continuaram com todos os seus aspectos semióticos, porém com outras formas de comunicação visual que propiciam ao leitor uma leitura mais dinâmica e desassociada da linearidade.

Ainda como parte dos experimentos, desenvolvemos questionários e testes que foram aplicados após a leitura dos textos com a função de mensurar o grau de compreensão alcançado pelos alunos nas leituras realizadas nos dois diferentes

suportes. Os testes compreendiam questionamentos e resolução de problemas que, para serem efetuados, eram necessários os conhecimentos adquiridos nas leituras.

2.5 SUJEITOS DA EXPERIÊNCIA

Os participantes da pesquisa foram alunos do curso de informática da modalidade ensino médio integrado do Instituto Federal de Educação, Ciência e Tecnologia do Campus Palmas. Foram utilizados, no primeiro experimento, 29 alunos divididos em duas turmas de 15 e 14 para a aplicação simultânea dos dois textos do primeiro experimento, e, no segundo, 21 alunos, também divididos em duas turmas de 10 e 11.

A primeira e a segunda série do curso técnico em Informática na modalidade médio-integrado foram eleitas para participar da pesquisa devido ainda não terem acesso ao conteúdo Unified Modeling Language – UML, que é estudado na terceira série, mais precisamente na componente disciplinar Técnicas e Projetos de Sistemas. Esse fato diminuía o índice de conhecimentos prévios sobre o assunto e nos garantia maiores possibilidades de testar a compreensão por meio dos textos.

2.6 PROCEDIMENTOS DA EXPERIÊNCIA

Para os experimentos, foram escolhidos dois tópicos do livro *Desenvolvendo Aplicações com UML*⁶ 2.0: *do conceitual à implementação*, de Ana Cristiana Melo, e o artigo científico *A história da UML e seus diagramas*, de Thânia Clair de Souza Vargas. Os capítulos do livro escolhidos foram: *Visão Geral da UML* e *Diagrama de Classes*. Esses capítulos foram aplicados em dias diferentes na ordem apresentada e, em data posterior à aplicação dos capítulos do livro, aplicamos o artigo. A escolha

⁶ Unified Modeling Language.

dos assuntos se deu especialmente pela sua natureza teórica, que representa uma exceção nos conteúdos técnicos do curso de Informática.

Para aplicar o experimento I, dividimos a turma de 29 alunos em dois grupos de 15 (grupo experimental impresso) e 14 (grupo experimental digital). A primeira turma foi conduzida a um laboratório de informática e a segunda à sala de aula convencional. No experimento II, fizemos exatamente a mesma divisão, porém com grupos de 10 (grupo experimental impresso) e 11 alunos (grupo experimental digital).

Os alunos que ficaram no laboratório leram o hipertexto no computador e se utilizaram do *mouse* e das teclas de movimentos para se locomoverem no texto. Já os alunos que foram para a sala de aula receberam o texto impresso e, durante a leitura, puderam usar o lápis e a caneta para fazer marcações e ticamentos.

Por fim, depois de realizadas as leituras, tanto no impresso quanto no computador, os alunos foram submetidos a testes de questionamentos e resolução de problemas. Para os grupos dos dois experimentos, os questionários e os testes foram realizados em papel e, para responder às questões, os alunos se utilizaram de canetas e lápis.

2.7 FERRAMENTAS EXPERIMENTAIS UTILIZADAS

2.7.1 Suporte Impresso

Como mencionado anteriormente, para o experimento I, o suporte impresso se deu por meio de texto formatado em um editor de texto, com figuras, gráficos, notas de rodapé, diagramas e desenhos, impresso em folha de papel A4, com margens tradicionais de orientação do papel retrato; e, para o experimento II, o texto teve as mesmas configurações de formato, porém com ausência quase total de multimodalidade.

A história de UML e se os diagramas*

**Adaptado do artigo de Thibaut Chab de Sousa Vargas
Disponível em: <http://www.inf.ufes.br/~thibaut/teaching/uml/uml-history.html>
Universidade Federal do Rio de Janeiro (UFRJ) - Universidade, RJ - Brasil
thibaut@inf.ufes.br*

1. Introdução

Modelagem de software é a atividade de construir modelos que expõem as características ou o comportamento de um software ou de um sistema de software. No contexto de software, os modelos podem ser usados na identificação das características funcionais/operacionais do software antes de provê-los (antes de requisitos), e no planejamento de sua construção. Frequentemente a modelagem de software usa algum tipo de notação gráfica e é de fato a principal fonte de comunicação entre os membros de uma equipe.

A modelagem de software normalmente implica a construção de modelos gráficos que modelam as estruturas dos componentes de software utilizados e os seus interrelacionamentos. Uma forma comum de modelagem de software orientada a objeto é através da linguagem unificada UML. A UML (Unified Modeling Language) é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos. Similiter os princípios metodológicos, sendo construída, em sua linguagem, as estruturas para modelagem de sistemas orientados a objetos. Por meio de seus diagramas é possível representar sistemas de software sob diversas perspectivas de interação. Refere a construção e todas as demais passadas envolvidas no processo de desenvolvimento de um sistema - pontos coordenados, análise, desenvolvimento - por apresentar um vocabulário de Notação Unificada (OMG, 2005a) (OMG, 2005b) (OMG, 2005c) (OMG, 2006).

2. História de UML

No início da utilização da paradigma de orientação a objetos, diversos métodos e formas agrupamentos para a comunidade. Chegaram a mais de cinquenta entre os anos de 1985 a 1994, porém a maioria disseminou o uso de tomar consideração as mudanças estruturais de época. Com isso os maiores pesquisadores foram os unidos que começaram a desenvolver uma maneira estruturada de modelar sistemas. Para a parte de década de 90 que começaram a surgir todos que procuravam validar a forma mais atraente com o paradigma da orientação a objetos. Alguns desses autores fizeram contribuições com publicações de seus respectivos métodos.

Por volta de 1992 criaram três métodos que mais se destacaram no mercado, eram o Booch '93 de Gary Booch, OMT '92 de James Rumbaugh e OOSE de Peter Jacobson. Cada um deles possuía pontos fortes em algum aspecto. O OMT '92 era mais forte em casos de uso (use cases), OMT '93 se destaca na fase de análise de sistemas de informações e Booch '93 era mais forte na fase de projeto. O mesmo dia esses métodos foi, principalmente, devido ao fato de não terem tido consideração suficiente a estrutura.

Suas estruturas já convergiam de maneira independente, cada uma mais produtiva quanto a forma, com o intuito de (SAMPAIO, 2007). Em outubro de 1994, começaram a trabalhar para unificação dos métodos. Já em outubro de 1995, Booch e Rumbaugh lançaram um rascunho de "Tábua Definidora" unificando o Booch '93 e o OMT '92. Após isso, Jacobson se juntou a equipe do projeto e o "Tábulo Definidor" passou a incorporar o OOSE. Em

julho de 1996 os três amigos começaram a colaborar, lançaram a primeira versão com orientamentos - a versão 0.9 que foi batizada como UML (UMLRT, 2007). Posteriormente, foram lançadas várias novas versões. OMTG lançou uma RFP (Request for Proposal) para que outros especialistas colaborassem com a criação da UML, chegando a versão 1.1. Após alcançar essa versão, a OMTG passou a admitir como padrão e o seu responsável (através da RTT - Rational Trade Force) para revisões. Essa revisão deu, de certa forma, "consolidação" e não provocou uma grande mudança no categorismo. Se observarmos as diferenças entre as versões anteriores, vemos que de uma para a outra não houve grande impacto, o que facilitou sua disseminação pelo mundo.

3. Evolução da Especificação

A especificação de UML é composta por quatro documentos inter-relacionados de UML (OMG, 2006), especificações de UML (OMG, 2005c), Oligar General Language (OCL) (OMG, 2005a) e Interrelação de Diagramas (OMG, 2005b).

Objeto-orientado de UML: O conjunto de diagramas de UML constitui uma linguagem definida a partir de uma linguagem que define os elementos fundamentais. Essa linguagem que aponta a definição dos diagramas é apresentada no documento inter-relacionado de UML.

Objeto-orientado de UML: Documento que coordena o documento de inter-relacionado e que define os elementos da linguagem no nível de sintaxe.

Objeto-orientado para Especificação de Oligar (OCL): Documento que apresenta a linguagem usada para descrever expressões em modelos UML, com pré-condições, pós-condições e invariantes.

Inter-relação de diagramas de UML: Apresenta uma extensão do meta-modelo voltado a informações técnicas. A extensão permite a criação de uma descrição no nível de objetos e informações técnicas que, em conjunto com o XML original, permite produzir representações por meio de especificações UML.

4. Organização dos diagramas de UML 2

A linguagem UML 2 é composta por treze diagramas, classificados em diagramas estruturais, diagramas de comportamento. Os diagramas estruturais mostram o aspecto estrutural tanto do ponto de vista do sistema quanto das classes. Têm como finalidade visualizar, organizar, construir e descrever os aspectos estruturais de um sistema, no qual, a representação de seu conteúdo é construída "relativamente passiva". Os diagramas estruturais de um sistema de software abrangem a estrutura e a coleção de itens como classes, interfaces, colaborações, componentes.

Os diagramas de comportamento do sistema descrevem os aspectos comportamentais quando em execução, isto é, como as mudanças dinâmicas do sistema. No modo para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema que a representação dos diagramas que "estão ativos", como por exemplo o fluxo de mensagens ao longo do tempo e a movimentação de fluxos de componentes em uma rede.

5. Diagramas de UML 2

Um diagrama é uma representação gráfica de um conjunto de elementos físicos: interfaces, colaborações,

2.7.2 Suporte Digital

O suporte digital, para os dois experimentos, o microcomputador, abrigou um sistema *web* criado por nós, desenvolvido a partir da tecnologia Java e banco de dados Postgressql, que disponibilizou o texto em formato de hipertexto e as suas várias multimídias que foram inseridas para dar movimento, som e cores aos aspectos multimodais.

Para acessar o sistema, o aluno precisava digitar *login* e senha como na ilustração abaixo.

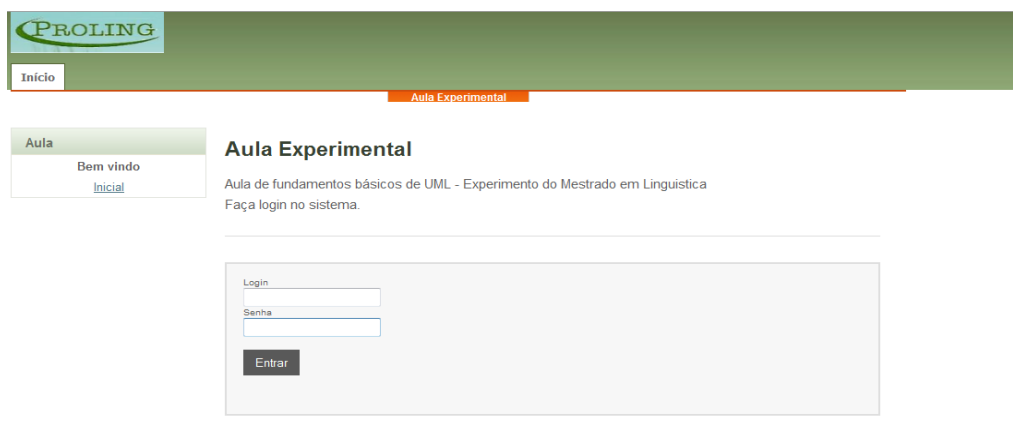


Figura 3 – Tela de acesso ao sistema web

Após a realização do acesso, o sistema disponibilizava as telas com os conteúdos que os alunos manipularam por meio de botões de comando.



Figura 4 – Tela da primeira aula do experimento I

2.8 CORPUS DE ANÁLISE

O corpus desta pesquisa constitui-se do resultado da aplicação de questionários e testes de solução de problemas realizados após a leitura dos experimentos. Os dados foram compostos de respostas a questões que corresponderam à identificação e à descrição de conceitos e resolução de

problemas. No experimento I, como foram duas leituras diferentes em dois dias, também foram dois instrumentos de coleta de dados (um questionário e um teste) referentes a um mesmo aluno. No experimento II, trabalhamos com uma leitura e um instrumento de coleta – um questionário.

2.8.1 Procedimento de Exclusão

Para o experimento I, foram excluídos do corpus os instrumentos de coleta de dados referentes a alunos que não presenciaram os dois momentos de aplicação e, para o experimento II, foram excluídos os instrumentos que não constaram as duas respostas do questionário.

2.9 PROCEDIMENTOS DE ANÁLISE (VERIFICAÇÃO DAS VARIÁVEIS)

2.9.1 Procedimentos de Análises dos Instrumentos

Tendo em vista a verificação da compreensão de textos multimodais impressos e suportados pelo computador, muitos são os fatores que incidem na composição da variável compreensão. Por isso, analisamos os instrumentos (questionário e o teste) do primeiro experimento nas questões em que o aluno exteriorizou seu raciocínio e desprezamos as objetivas e de completar. No segundo experimento, analisamos as duas questões do questionário. Nos dois experimentos, procedemos às análises por aluno. A discussão se deu tendo em vista os dois grupos experimentais dentro de cada questão.

Como discutido no capítulo teórico, MCI são molduras pelas quais identificamos situações dos vários domínios do mundo experienciados por nós e categorizadas pelo aparato cognitivo humano. Nessa mesma lógica, *MCI expert* nada mais é do que a compreensão amadurecida de um estudioso de um determinado assunto, no caso UML. Assim, se compreendemos também por meio

dos MCIs, acreditamos que um novo aprendizado pode ser subsidiado pelos MCIs *experts*.

No experimento I, cada aluno realizou dois momentos de submissão aos instrumentos de coleta de dados e estava inserido em somente um grupo experimental (impresso ou digital). O primeiro instrumento aplicado foi um questionário com cinco questões, das quais as duas primeiras objetivavam verificar se o aluno, com a leitura do texto, conseguiria aproximar o seu MCI *qualis*⁷, ou mesmo, a sua compreensão leitora referente ao texto experimental lido do que chamamos de Modelo Cognitivo (MCI) *expert* do autor, ou melhor, do que julgamos ser a compreensão do autor.

O enunciado dessas duas primeiras questões do questionário fornecia pistas que favoreciam o relacionamento dos novos conceitos internalizados pelo aluno a outros já existentes que estão presentes no censo comum cotidiano dos alunos da área de informática. Isso foi assim pensado para que os alunos tivessem maiores possibilidades de aproximar o seu MCI *qualis* do MCI *expert*.

Para proceder à análise do experimento I, classificamos as respostas (MCI *qualis*) dos alunos nas questões 1 e 2, conforme a tabela a seguir.

Tabela 1 - Verificação de MCI

Conceito	Descrição
A	Aproxima-se em totalidade do MCI <i>expert</i> .
B	Aproxima-se do MCI <i>expert</i> em proporção maior que 50%.
C	Aproxima-se do MCI <i>expert</i> em proporção menor que 50%.
D	Tem alguns pontos de contato com o MCI <i>expert</i> .
E	Em nenhum aspecto, há uma aproximação do MCI <i>expert</i> .

Com essa classificação, acreditamos mensurar a equivalência da resposta do aluno, o MCI *qualis*, junto a que seria ideal, ou ao MCI *expert*.

As últimas três questões do questionário, das quais duas eram objetivas, visavam fixar conceitos que foram usados no teste de resolução de problemas desse experimento.

⁷ Entendemos a expressão MCI *qualis* como sendo a construção cognitiva efetuada pelo aluno do seu próprio modelo cognitivo idealizado correlacionado ao MCI *expert* mediato pela leitura do texto.

Na terceira questão, foi pedido ao aluno que identificasse uma determinada categoria da UML no texto e a transcrevesse. Para essa questão, a atividade mental máxima exigida era a localização no texto. As duas outras questões objetivas suscitavam levemente a estratégia de *seleção* e *inferenciação* ao deparar com combinações de respostas em que parte delas podia estar certa e outras não e, em somente uma, o aluno poderia validar toda a afirmativa conforme as informações novas que adquiriu no texto. Devido à natureza binária dessas respostas não a utilizamos para as análises.

O segundo instrumento de coleta de dados, o teste de resolução de problemas, referia-se ao conteúdo de diagrama de classe da UML e tinha oito questões com objetivos diferentes. As duas primeiras questões eram objetivas de múltipla escolha. Na primeira, o aluno podia optar por mais de uma resposta, até porque duas delas estavam corretas. Essas duas primeiras tinham o objetivo de fazer com que o aluno identificasse conceitos centrais da UML dentro da sua notação oficial.

A terceira questão foi do tipo discursivo e pedia que o aluno desenvolvesse um exemplo de um conceito de classe da UML conforme as definições da notação oficial apresentada no texto do experimento. Para elaborar essa resposta, o aluno precisava ativar estratégias cognitivas como *inferenciação*, *pressuposição* e *predição*. A análise dessa questão comparou a resposta do aluno, ou seja, o MCI *qualis* a um MCI *expert* e foi classificada conforme tabela 1.

As questões 4, 5, 6 e 7 visavam a verificar se, com a leitura do texto, o aluno conseguiria analisar um dado esquema analítico da UML, as relações entre os artefatos da notação e o efeito deles no contexto global do projeto. Apesar de evocarem estratégias cognitivas consideráveis, as respostas eram objetivas e, por isso, como também a 1 e a 2, não consideramos na análise devido à natureza binária.

A última questão, considerada mais relevante para concretização dos conhecimentos em UML, era uma análise que demandava a construção de um protótipo de um sistema computacional. Requeria do aluno estratégias cognitivas e metacognitivas como discutido no capítulo teórico. O procedimento de análise foi a classificação conforme a tabela 1.

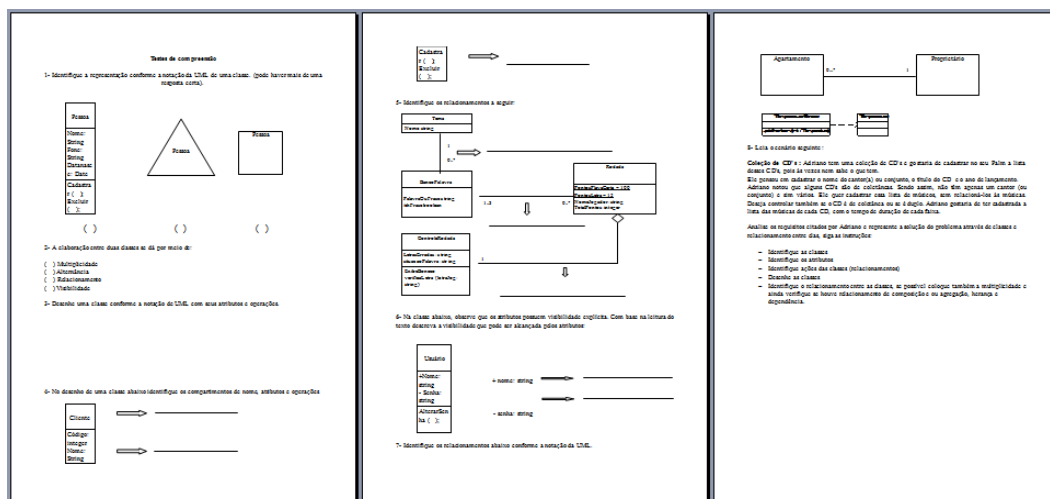


Figura 5 – Teste de compreensão

Para o experimento II, os alunos também foram divididos em dois grupos experimentais (impresso e digital) e realizaram somente um momento de submissão a um questionário que continha duas questões. As questões eram descritivas e cobravam um posicionamento do aluno quanto à leitura do experimento. Essas questões foram elaboradas com o intuito de fornecer pistas que favorecessem o condicionamento de estratégias cognitivas que auxiliassem no processo de integração conceitual⁸ (FAUCCONIER; TURNER, 2002) dos alunos.

A avaliação do experimento se deu comparando as respostas dos alunos, ou o MCI *qualis*, ao MCI *expert*, conforme tabela 1.

2.9.2 Especificidades da Análise dos Dados

A avaliação dos resultados foi realizada somente nas questões dos instrumentos em que o aluno expôs sua análise, ou por respostas escritas ou por construção de protótipos. Nesse caso, foram utilizadas, do primeiro experimento, as questões 1 e 2 do questionário e 3 e 8 do teste de resolução de problemas; e, do segundo experimento, as duas questões do questionário, observando a divisão dos dois grupos experimentais (impresso e digital).

⁸ Ver capítulo teórico no item 2.2.4.

A partir dessas questões, avaliamos as repostas dos alunos com o objetivo de obter um MCI *qualis* para cada resposta. Chegamos ao MCI *qualis* dos alunos comparando as estratégias cognitivas projetadas nas repostas com as estratégias cognitivas esperadas para solucionar a questão; e, ainda, equacionando o MCI *qualis* de cada resposta com o MCI *expert* idealizado para a resposta correspondente.

Para responder às questões dos instrumentos, os alunos utilizaram determinadas estratégias cognitivas que nortearam sua compreensão e interpretação. Para cada questionamento, identificamos uma quantidade de estratégias cognitivas esperadas nas respostas dos alunos. Desse cruzamento de dados, identificamos o MCI *qualis* do aluno.

As estratégias cognitivas investigadas nos MCIs *qualis* são as citadas por Nunes (2008), no capítulo teórico, acrescidas de atividades cognitivas, como o conhecimento prévio, a localização de informação no texto, o levantamento e a validação de hipóteses.

Essa análise buscou elementos nas repostas dos alunos que nos auxiliassem na investigação do fenômeno compreensão que é caracteristicamente individual e subjetivo. O procedimento de análise também seguiu a linha da subjetividade. Para isso, utilizamos os formulários expostos na sequência.

Para o experimento I:

Quadro 1 – Formulário de MCI *expert* do experimento I

Questões		MCI <i>expert</i>	EC esperadas ⁹
QUESTIONÁRIO	Q1		
	Q2		

⁹ Estratégias cognitivas esperadas

TESTE	Q3		
	Q8		

Quadro 2 – Formulário de avaliação de alunos do experimento I

Avaliação da compreensão do AI 1					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas ¹⁰
	Q1				
	Q2				
T E S T E	Q3				
	Q8				

Para o experimento II:

Quadro 3 – Formulário de MCI *expert* do experimento II

Questões		MCI <i>expert</i>	EC esperadas
Q U E S T I O N Á R I O	Q1		
	Q2		

¹⁰ Estratégias cognitivas alcançadas

Quadro 4 – Formulário de avaliação de alunos do experimento II

Avaliação da compreensão do AI 1					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
	Q1				
	Q2				

3 ANÁLISE DOS DADOS

3.1 EXPERIMENTO I

Consideramos os dados relativos a sete alunos do grupo experimental do texto impresso e sete alunos do grupo experimental no suporte digital para proceder à análise das categorias explicadas no item da metodologia da pesquisa.

3.1.1 Análise do Experimento I

Como descrito no procedimento de análise, a observação qualitativa dos dados se deu nas questões 1 e 2 do questionário e 3 e 8 do teste. A análise foi realizada por aluno. O formulário de MCI *expert*, que contém a reposta base para comparar com as dos alunos, foi preenchido somente uma vez no início da análise.

3.1.1.1 Questões Analisadas

Q1. Através do texto, observa-se que a UML é extremamente adotada em desenvolvimento de sistemas por propor um modelo único que externa, de forma gráfica, os artefatos do mundo real, conforme o exemplo dado no texto da classe ALUNO. De posse da leitura, fale de forma sucinta o que você julga que favorece essa adoção.

Q2. A UML trouxe uma proposta metodológica de padronização da arquitetura de projetos de sistemas em resposta a tantos problemas vividos nessa área ocasionados pela quantidade de métodos diferentes adotados e desenvolvidos por entidades dessa natureza. Pensando assim e tendo em vista a analogia de um projeto arquitetônico de uma casa (onde qualquer pessoa, sendo ou não profissional

da área da engenharia civil pode entender), fale sobre a importância do padrão no desenvolvimento de sistemas.

Q3. Desenhe uma classe conforme a notação da UML com seus atributos e operações.

Q8. Leia o cenário seguinte:

¹¹Coleção de CDs: Adriano tem uma coleção de CDs e gostaria de cadastrar no seu Palm a lista desses CDs, pois às vezes nem sabe o que tem. Ele pensou em cadastrar o nome do cantor(a) ou conjunto, o título do CD e o ano de lançamento.

Adriano notou que alguns CDs são de coletâneas. Sendo assim, não têm apenas um cantor (ou conjunto) e sim vários. Ele quer cadastrar essa lista de músicos sem relacioná-los às músicas. Deseja controlar também se o CD é de coletânea ou se é duplo. Adriano gostaria de ter cadastrada a lista das músicas de cada CD, com o tempo de duração de cada faixa.

Analise os requisitos citados por Adriano e represente a solução do problema através de classes e relacionamento entre elas, siga as instruções:

- identifique as classes;
- identifique os atributos;
- identifique as ações das classes (relacionamentos);
- desenhe as classes;
- identifique o relacionamento entre as classes, se possível coloque também a multiplicidade e ainda verifique se houve relacionamento de composição e ou agregação, herança e dependência.

Como as questões 1 e 2 do questionário são respostas expositivas, opiniões dos alunos, apontamos em que situações foram utilizadas as estratégias cognitivas; já nas questões 3 e 4 do teste, por serem desenhos gráficos que devem obedecer a uma notação oficial, avaliamos os desenhos quanto à forma, ao conteúdo e às ligações e discutimos as estratégias utilizadas.

¹¹ Adaptado do livro Exercitando Modelagem em UML de Ana Cristina Melo.

3.1.1.2 Análise por Aluno

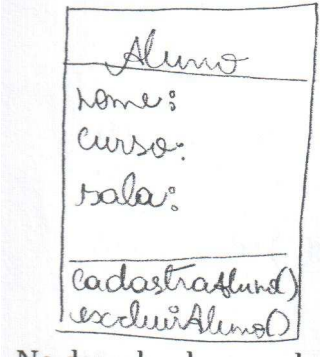
Quadro 5 - Respostas e estratégias cognitivas esperadas do experimento I

Questões		MCI <i>expert</i>	EC ¹² esperadas
QUESTIONÁRIO	Q1	Esperava-se que os alunos ressaltassem a importância da estruturação de modelos universais em forma gráfica que favorece a interpretação de desenvolvedores e usuários sem ambigüidade.	Conhecimento prévio, inferência, pressuposição, predição
	Q2	Esperava-se que os alunos reconhecessem que um projeto que obedece a um rigor metodológico padrão facilita o entendimento tanto de profissionais da área quanto de meros usuários no que se remete ao futuro <i>software</i> .	Conhecimento prévio, inferência, pressuposição, predição
TESTE	Q3	Aluno matricula: string nome: string dataNascimento: date dataMaticula: date valorMensalidade: real MatricularAluno() Reajustar Mensalidade(percentual: real) TrancarMatricula() EmitirBoletoPagto(mesAno: string)	Conhecimento prévio, pressuposição, inferência, predição
	Q8	<pre> classDiagram class Cd { /musico: Classe Musico titulo: string ano: integer ehDuplo: boolean ehColetanea: boolean /listaMusicas: Coleção de Musica listarCdsPorMusica(musica: Classe Musica) } class Musica { nome: string tempoFaixa: real Cadastrar() } class Musico { nome: string ehSolo: boolean Cadastrar() listarCdsMusico(musico: Classe Musico) } Cd "0..*" -- "1..*" Musico Cd "1" *-- "1..*" Musica </pre>	Conhecimento prévio, inferência, pressuposição, predição, previsão, seleção

¹² Estratégias cognitivas esperadas.

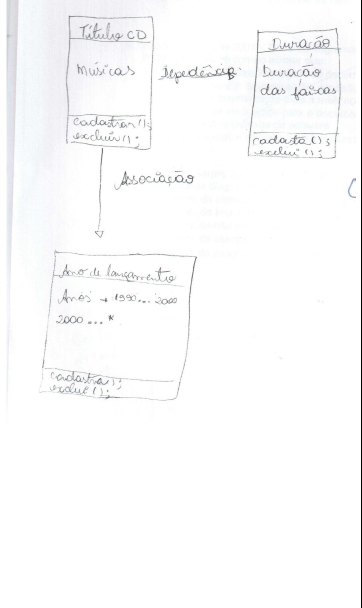
a) Alunos do grupo experimental impresso

Quadro 6 – Avaliação da compreensão do AI1 - grupo experimental impresso

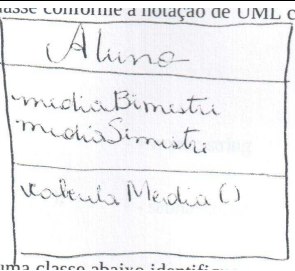
Avaliação da compreensão do AI1					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação ¹³	EC ¹⁴ alcança- das
Q U E S T I O N Á R I O	Q1	Para facilitar o entendimento do usuário.	D	Opinião vaga, porém suscitou um ponto importante que é a facilidade de entendimento pelo usuário, mas não expôs a razão disso. Acreditamos que o aluno apresentou PR , baseado no conhecimento que <i>softwares</i> são para usuários e que a notação facilita o entendimento do usuário.	Pressupo- sição
	Q2	O padrão é muito importante porque fez com que não houvesse mais confusão entre os programadores e facilitou a compreensão dos programas.	B	Aqui percebemos uma indicação de CP . O aluno sabe que programadores codificam sistemas e que é um processo confuso entre esses profissionais, devido cada um ter uma forma de desenvolver as instruções dos programas. Assim ele usou INF e PR quando apostou que um padrão poderia resolver o problema.	Conheci- mento prévio, inferência, pressupo- sição
T E S T E	Q3		A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Apresentou CP quando demonstrou conhecer os atributos de aluno e ainda PR quando apostou que esses atributos e esses métodos compõem a classe.	Conheci- mento prévio, pressupo- sição

¹³ Para economizar espaço na célula avaliação, utilizamos as siglas: PR – pressuposição, CP – conhecimento prévio, INF – inferência e PD – predição.

¹⁴ Estratégias cognitivas alcançadas.

Q8		D	O aluno identificou uma classe importante (CD), sendo que as demais que ele criou não configuram classes e sim atributos. Portanto, o relacionamento presente no desenho não reflete a ligação uma vez que não existe ligação de associação entre classe e atributo. Acreditamos que houve CP quando o aluno mostrou entender a relação de CD com música e com lançamento do CD, e INF quando ele apostou na subordinação das classes apresentadas.	Conhecimento prévio, Inferência
----	---	---	--	---------------------------------

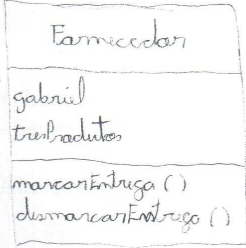
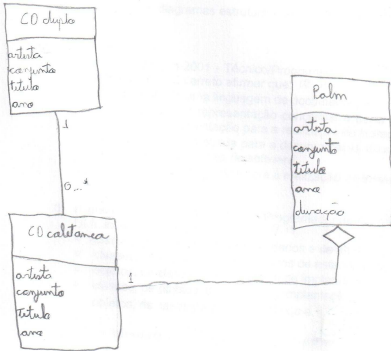
Quadro 7 – Avaliação da compreensão do AI2 - grupo experimental impresso

Avaliação da compreensão do AI2					
		Resposta do aluno	MCI qualis	Avaliação	EC alcançadas
QUESTIONÁRIO	Q1	O fato de ser um modelo único ajuda na sua adoção, pois todos utilizando a UML podem se entender melhor sem ambiguidades. Uma padronização facilita a comunicação.	B	A resposta é eficiente, porém o aluno não atentou para o fator importante de que é a exposição do método de forma gráfica que gera o resultado apontado. Aqui identificamos a presença de INF quando ele apostou que um modelo único evitará ambiguidades.	Inferência, conhecimento prévio
	Q2	É importante para que os outros também entendam o seu projeto.	C	Resposta vaga, porém observamos o processo de INF quando o aluno comparou um projeto arquitetônico como um de <i>software</i> . Esses elementos são fornecidos no enunciado da questão.	Inferência
TESTE	Q3	na classe conforme a notação de UML com seu  de uma classe abaixo identi-	B	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Mas não atentou para atributos que identificam a classe, como nome ou matrícula do	Conhecimento prévio, pressuposição, inferência

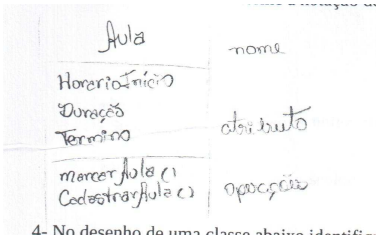
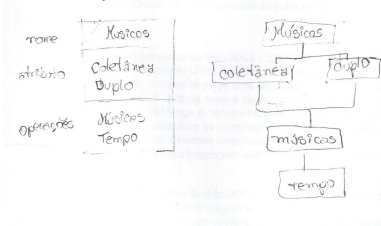
			aluno. Acreditamos apresentar CP quando demonstrou conhecer os atributos de aluno dentro do contexto escolar do ano letivo, e INF quando presumiu que o cálculo da média é a operação de aluno, e PR, ao apostar na constituição do grupo formado pelos atributos e métodos relacionado à aprovação do aluno.	
Q8		B	O aluno identificou três classes importantes e seus relacionamentos, porém não utilizou a notação adequada para demonstrar a função dos relacionamentos. Apresentou CP quando mostrou conhecer a composição de um CD, PR quando definiu a lógica de relação entre as classes, e INF quando apostou no conteúdo das classes.	Conhecimento prévio, pressuposição, inferência

Quadro 8 – Avaliação da compreensão do AI3 - grupo experimental impresso

Avaliação da compreensão do AI3					
		Resposta do aluno	MCI qualis	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	Favorece o entendimento do conteúdo apresentado, pois representa uma forma mais organizada.	A	A resposta é eficiente e eficaz. O aluno mostrou ter CP de que demonstrações em forma gráfica são mais entendíveis e PR de que um projeto gráfico é mais visual que outros de forma não gráfica.	Conhecimento prévio, pressuposição
	Q2	Sem adotar um padrão único, as pessoas teriam que conhecer todos os padrões existentes para que não ficassem limitadas a um único tipo de serviço, mas, existindo um único padrão bem estipulado, as pessoas podem variar bastante o tipo de serviço.	B	O aluno atentou para a dificuldade de trabalho em que não há um padrão definido e pressupôs , inclusive, que nesses casos se devam conhecer todos os padrões existentes.	Pressuposição

T E S T E	Q3	 - No desenho de uma classe abaixo ide	C	O aluno construiu o desenho da classe obedecendo à forma com as divisões adequadas, porém não compreendeu a divisão dos atributos e os nomeou como se fossem objetos da classe. Acreditamos apresentar CP quando ele sugeriu que o fornecedor seria o Gabriel, INF quando presumiu as operações que podem ocorrer com a classe fornecedor e PR ao mostrar que conhece a lógica que envolve fornecedor que pressupõe produto, entrega.	Conhecimento prévio, pressuposição, inferência
	Q8		D	O aluno identificou uma classe importante (CD), sendo que a outra (CD coletânea) deveria ser um atributo da classe CD e a classe Palm simplesmente não existir no sistema, devido não representar uma entidade, mas um suporte para o sistema. O relacionamento presente no desenho não refletiu a ligação uma vez que não existe ligação de associação entre classe e atributo. Assim cremos apresentar CP quando demonstrou entender categorias como CD coletânea e PR quando expôs a lógica que implementou entre as classes.	Conhecimento prévio, pressuposição

Quadro 9 – Avaliação da compreensão do AI4 - grupo experimental impresso

Avaliação da compreensão do AI4					
		Resposta do aluno	MCI qualis	Avaliação	EC alcançadas
Q U E S T I O N Á R I O	Q1	Porque a UML utiliza ferramentas mais simples, imagens que é muito importante para chamar a atenção, fornece mais detalhes.	B	A resposta mostra que o aluno entendeu que uma demonstração, por meio de imagens (gráficos), é mais eficiente para mostrar um projeto de <i>software</i> que outra forma escrita. Esse apontamento configura que o aluno, ao afirmar isso, utilizou o CP e a PR de que as imagens podem fornecer mais detalhes.	Conhecimento prévio, pressuposição
	Q2	Com o sistema padronizado, fica mais fácil de entender e não haverá confusão no entendimento, padrões foram criados para facilitar a vida das pessoas.	A	A resposta, apesar de confusa, é eficaz, porque justifica que o padrão elimina a dificuldade de compreensão pelos envolvidos. Acreditamos que aluno relacionou com outras experiências vivenciadas por ele, configurando CP e INF ao emitir a opinião de que padrões são criados para facilitar a vida.	Conhecimento prévio, Inferência
T E S T E	Q3		A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Apresentou CP sobre a rotina de aulas, PR quando imaginou a lógica dos atributos e INF quando presumiu as operações que podem ocorrer na classe aula.	Conhecimento prévio, pressuposição, inferência
	Q8		D	O aluno identificou somente uma classe relevante para o sistema e apresentou um organograma completamente descontextualizado da proposta da UML. O aluno demonstrou CP sobre a composição e o suporte da música e PD quando, baseado nas	Conhecimento prévio, predição

				informações de música, músicos, faixa, elaborou a proposta da modelagem que mais se parece com uma tentativa de adivinhação.	
--	--	--	--	--	--

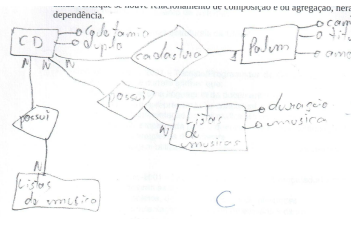
Quadro 10 – Avaliação da compreensão do AI5 - grupo experimental impresso

Avaliação da compreensão do AI5					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	De acordo com o texto, é mais fácil compreender uma imagem organizada que um texto.	A	A resposta é objetiva e reflete a ideia central do texto no que se refere à UML como uma forma gráfica. Para responder pontualmente, acreditamos que o aluno usou o CP , algo em que ele experienciou com demonstração por meio de imagens e de PR .	Conhecimento prévio, pressuposição
	Q2	Quer dizer que ele não determina o que deve ser feito. Uma pessoa pode desenvolver o seu próprio projeto sem ter de se prender a limitações.	E	Aqui acreditamos que houve um problema de interpretação quanto ao que representa padrão para o texto e para o aluno. Pela resposta, identificamos ainda que o aluno tem CP de desenvolvimento de projetos de um modo geral.	Conhecimento prévio
T E S T E	Q3	3- Desenhe uma classe conforme Aluno nome notaBimestre recuperacao matricula() fecharMatricula()	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Acreditamos que apresentou CP quando demonstrou conhecer os atributos de aluno dentro do contexto escolar do ano letivo, INF quando presumiu que o aluno precisa ser matriculado ou mesmo pode encerrar a matrícula e PR ao apostar no conjunto de dados identificados pelo nome e ainda nas operações vitais para a existência do aluno.	Conhecimento prévio, pressuposição, inferência

Q8		C	O aluno identificou as três principais classes (CD, cantor e músico), porém não descreveu atributos e métodos das mesmas. As ligações de relacionamento não contemplam a multiplicidade presente na associação, e as classes coletânea e duplo deveriam ser atributos da classe CD e não colocadas como classes ligadas a cantor. Demonstrou CP ao descrever as categorias que se relacionam com CD, PR e INF quando presumiu a relação entre as classes.	Conhecimento prévio, inferência, pressuposição
----	--	---	---	--

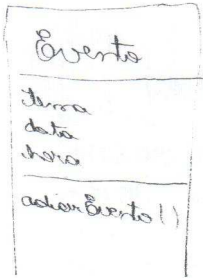
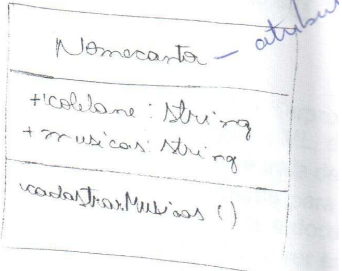
Quadro 11 – Avaliação da compreensão do A16 - grupo experimental impresso

Avaliação da compreensão do A16					
		Resposta do aluno	MCI qualis	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	Favorece por o usuário ter uma visão de entendimento melhor, por ser tudo bem classificado.	D	Resposta vaga, porém percebemos que o aluno entendeu que a organização do projeto facilitará a compreensão dos usuários. Acreditamos que ele emitiu essa opinião devido a uma experiência próxima, ou CP de organização de um modo geral, e ainda é isso que o levou à PR de que é a organização que favorecerá aos usuários uma visão melhor.	Conhecimento prévio, pressuposição
	Q2	“A palavra chave é padrão,” o problema ocorrido no desenvolvimento de sistemas era a falta de padronização, pois sem uma padronização correta existiam muitas ambiguidades na hora do desenvolvimento.	A	O aluno apontou a causa central da necessidade de adoção de um padrão. Como ele apresentou o problema com propriedade, acreditamos que ele tenha algum tipo de CP e, por isso, apostou na PR de que a dificuldade vivida pelos desenvolvedores que usavam métodos ambíguos.	Conhecimento prévio, pressuposição

T E S T E	Q3	3- Desenhe uma classe com	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Apresentou CP sobre o conceito e o funcionamento de uma sapataria e, com isso, trabalhou PR e INF ao definir os atributos, tendo em vista que ele se preocupou em personalizar o objeto pelo nome e ainda com as operações ligadas à classe que se referencia ao objeto.	Conhecimento prévio, pressuposição, inferência
	Q8		D	O aluno identificou uma classe relevante ao sistema e fugiu completamente da notação. Ainda assim demonstrou CP de conceito, constituintes e conteúdo de um CD e PR quando propôs uma lógica de relação entre o que ele imagina que são as classes e os atributos.	Conhecimento prévio, pressuposição

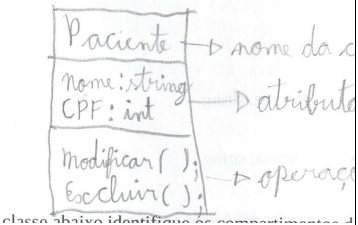
Quadro 12 – Avaliação da compreensão do AI7 - grupo experimental impresso

Avaliação da compreensão do AI7					
		Resposta do aluno	MCI qualis	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	Eu acho que o principal fator que favorece esta adoção é que este foi desenvolvido de modo claro e específico é de fácil entendimento, o que ajuda bastante.	D	Apesar de o aluno não citar os pontos principais da adoção da UML, ele usou INF quando apostou que a metodologia veio para facilitar o trabalho do desenvolvimento de <i>software</i> .	Inferência
	Q2	Existe a necessidade de um padrão, pois uma casa pode ser projetada de várias maneiras, seria um erro fazer o banheiro maior que o quarto ou a sala, da mesma forma que existe certo padrão para a construção de casas de modo que o banheiro geralmente é o menor cômodo da casa. A UML criou um padrão para a confecção de <i>software</i> para não serem criados entendimentos de forma errônea.	A	O aluno demonstrou ter compreendido a necessidade de adoção de um padrão no desenvolvimento de <i>software</i> através de uma analogia com os projetos da construção civil. Para isso, ele usou CP sobre projetos arquitetônicos, INF e PR ao comparar um projeto de uma casa com os projetos de <i>software</i> .	Conhecimento prévio, pressuposição, inferência

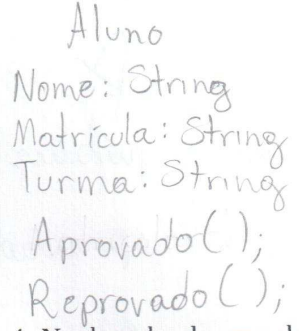
T E S T E	Q3	 <pre> classDiagram class Evento { data hora adicionarEvento() } </pre>	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Utilizou CP sobre o conceito e o funcionamento de eventos de um modo geral; ainda, utilizou PR quando compôs os atributos da classe esquecendo-se da identificação unária do evento; e INF quando designou a operação “adiar evento ()” para preparar o sistema para uma possibilidade de imprevistos e cancelamento de data.	Conhecimento prévio, pressuposição, inferência
	Q8	 <pre> classDiagram class NomeCantor { +colelame: String +musicas: String cadastraMusica() } </pre>	E	O aluno identificou uma classe descontextualizada que não tem representatividade para o sistema.	

b) Alunos do grupo experimental digital

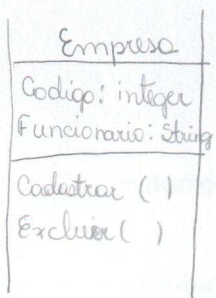
Quadro 13 – Avaliação da compreensão do AI1 - grupo experimental digital

Avaliação da compreensão do AI1					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	A facilidade de compreensão, padronização de métodos e facilidade no desenvolvimento.	C	O aluno identificou que a padronização de métodos facilitará o desenvolvimento (talvez por uma experiência que viveu com algum tipo de padrão de alguma metodologia), porém não trouxe elementos que ratificassem sua posição. Acreditamos haver a presença de CP sobre modelos visuais.	Conheci- mento Prévio
	Q2	Facilita na engenharia de <i>software</i> por causa do padrão na modelagem gráfica, e por causa do suporte à engenharia inversa.	D	O aluno demonstrou algum tipo de CP ao mencionar o termo técnico engenharia inversa que não foi tratado no texto, porém a resposta está descontextualizada da pergunta.	Conheci- mento prévio
T E S T E	Q3	conforme a notação de UML com seus atributos  classe abaixo identifique os componentes	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Acreditamos que utilizou CP sobre o conceito e o funcionamento do comportamento do paciente dentro do seu ambiente de assistência a saúde. Dentro dessa lógica, usou PR ao constituir os atributos da classe paciente que remete à identificação do objeto "paciente" para o acesso ao tratamento; e INF ao indicar as operações, crendo que o paciente pode ter alterações no cadastro, excluir consulta. Quanto à PD , utilizou a estratégia ao identificar o tipo de dado de cada campo baseado no CP de programação e tentou adivinhar o tipo de dado.	Conheci- mento prévio, pressupo- sição, inferência, predição.
	Q8		E	O aluno não respondeu.	

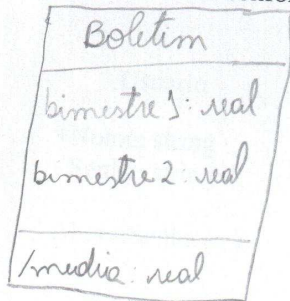
Quadro 14 – Avaliação da compreensão do AI2 - grupo experimental digital

Avaliação da compreensão do AI2					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	O que favorece a adoção da Uml como padrão foi, além da união entre James Rumbaugh, Grady Booch e Ivan Jacobson, que juntaram seus métodos (OMT, Booch e OOSE) e fizeram com que mais pessoas o utilizassem, pois cada um tinha uma fatia do mercado, foi que depois que eles se juntaram, outros se aliaram a eles e começaram a colaborar com o projeto.	D	A resposta atentou para a história da unificação dos métodos. Quando diz que mais pessoas usaram o método unificado, pressupôs (PR) a soma dos que usavam os diferentes métodos.	Pressupo- sição
	Q2	Com um padrão tudo fica mais fácil, pois com um padrão qualquer pessoa que entende o assunto poderá utilizá-lo, sem um padrão os profissionais terão de se virar para saber o que os outros fizeram.	A	O aluno apontou um problema real vivido pelos desenvolvedores antes da existência da UML. Ele demonstrou CP de funcionamento de padrões; INF quando disse que o padrão permite o entendimento de qualquer pessoa; também pressupôs (PR) que a ausência de um padrão culmina no desentendimento entre os profissionais.	Conheci- mento prévio, pressupo- sição, inferência
T E S T E	Q3	 Aluno Nome: String Matrícula: String Turma: String Aprovado(); Reprovado(); 4- No desenho de uma classe abstrata	B	O aluno construiu algo parecido com o desenho da classe, observou corretamente atributos e métodos, mas não atentou para a notação oficial. Utilizou CP de conceito e funcionamento do aluno dentro do ambiente escolar e também inferiu (INF) os atributos, atentando pelo atributo identificador (nome).	Conheci- mento prévio, inferência.
	Q8		E	O aluno não respondeu.	

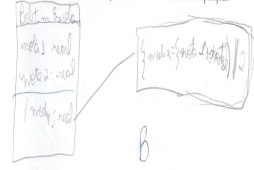
Quadro 15 – Avaliação da compreensão do AI3 - grupo experimental digital

Avaliação da compreensão do AI3					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	É uma forma diferente das outras, que meio que acaba a disputar entre as “empresas”, e é uma unidade de modelagem mais fácil de compreender e mais simples também já que é padrão.	D	Resposta vaga, pois não justificou a pergunta. Porém inferiu (INF) que a modelagem UML é mais fácil de entender por ser padrão.	Inferência
	Q2	A importância do <u>padrão</u> no desenvolvimento é que fica mais fácil a compreensão, podendo assim qualquer pessoa entender melhor até quem é um pouco “leigo” na área.	A	Acreditamos que, quando o aluno afirmou que qualquer pessoa pode entender um projeto padrão, é devido ao padrão de UML estar associado com o padrão arquitetônico mencionado no enunciado. Ele fez inferências (INF) baseado no CP sobre projetos arquitetônicos.	Conhecimento prévio, inferência
T E S T E	Q3		A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Acreditamos que ele usou CP sobre o conceito e o funcionamento de uma empresa, inferiu (INF) os atributos da classe, inclusive um identificador (código) e, ainda, pressupôs (PR) as operações de empresa (cadastrar e excluir), mas não em relação à empresa, mas a funcionário, o que é um erro de estrutura, porém foi precisa a estratégia para abstrair a relação.	Conhecimento prévio, pressuposição, inferência.
	Q8		E	O aluno não respondeu.	

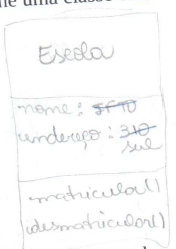
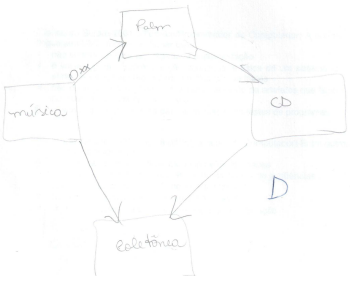
Quadro 16 – Avaliação da compreensão do AI4 - grupo experimental digital

Avaliação da compreensão do AI4					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
	Q1	A fácil absorção de ideias e o fácil entendimento.	D	Resposta vaga, porém levemente suscitou PR no que se refere à facilidade de compreensão no uso do método, mas não argumentou a razão.	Pressupo- sição
	Q2	Adotando-se um padrão, torna-se bem mais simples o uso da UML, pois de uma versão para outra, pouca coisa é alterada.	D	Resposta vaga, desconectada da pergunta.	
T E S T E	Q3	Desenhe uma classe conforme 	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Acreditamos haver a presença de CP de funcionamento de um boletim escolar; PR ao referenciar a média como operação da classe, pois, tendo em vista sua ligação direta um boletim, pressupôs no mínimo a média; e INF na composição dos atributos da classe formada por bimestres, o que gerou a inferência de que devem ter valores até no máximo quatro, conforme o ano letivo.	Conheci- mento prévio, pressupo- sição, inferência
	Q8		E	O aluno não respondeu.	

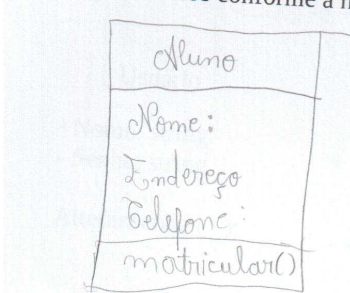
Quadro 17 – Avaliação da compreensão do AI5 - grupo experimental digital

Avaliação da compreensão do AI5					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	O que favorece essa adoção é que a UML faz uma representação gráfica do que é para ser feito, faz uma descrição do que é para se feito.	A	O aluno identificou exatamente a ideia do texto. Para chegar a essa resposta, acreditamos que ele usou CP sobre demonstrações gráficas, inferiu (INF) que a UML faz o papel da representação gráfica e pressupôs (PR) que as representações gráficas são mais visíveis e facilitam a interpretação.	Conhecimento prévio, pressuposição, inferência
	Q2	É importante ser padronizado, pois os sistemas são parecidos, porém cada um tem suas particularidades, favorecendo áreas diferentes e para cada sistema uma manutenção diferente.	D	Resposta vaga com somente um ponto de contato com a pergunta que é a importância do padrão. O aluno demonstrou CP de sistemas computacionais quando afirmou que são parecidos.	Conhecimento prévio
T E S T E	Q3		A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Acreditamos haver a presença de CP de funcionamento de um boletim escolar; PR ao referenciar a média como operação da classe, pois, tendo em vista sua ligação direta com um boletim, pressupôs no mínimo a média e ainda explicou, através de nota, o funcionamento da média que também advém de CP sobre funções matemáticas; e INF na composição dos atributos da classe formada por bimestres, o que gerou a inferência de que devem ter valores até no máximo quatro conforme o ano letivo.	Conhecimento prévio, pressuposição, inferência
	Q8		E	O aluno não respondeu.	

Quadro 18 – Avaliação da compreensão do AI6 - grupo experimental digital

Avaliação da compreensão do AI6					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
Q U E S T I O N Á R I O	Q1	Por ter uma facilidade de entendimento e absorção de ideias.	D	Resposta vaga com somente um ponto de contato com o texto, a facilidade de entendimento que a visualidade gráfica do método traz. Presença de PR devido ao aluno entender que a UML veio para facilitar a implementação do trabalho.	Pressuposi- ção
	Q2	Pois através de seus modelos preciosos, completos e sem ambiguidades, atende as decisões que necessitam ser tomadas para o desenvolvimento de sistemas.	B	O aluno demonstrou ter compreendido a função da UML quando afirmou que os modelos são preciosos, completos e sem ambiguidades e atende às necessidades do desenvolvimento de <i>software</i> . Demonstrou CP de desenvolvimento de <i>software</i> .	Conheci- mento prévio
T E S T E	Q3	3- Desenhe uma classe 	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Apresentou CP do conceito e do funcionamento de uma escola, inferiu (INF) os atributos da classe e pressupôs (PR) as operações de escola (matricular e desmatricular), mas não em relação à escola e sim ao aluno, o que é um equívoco.	Conheci- mento prévio, pressuposi- ção, inferência
	Q8		D	O aluno identificou duas classes relevantes ao sistema (CD, música), porém não identificou nenhum relacionamento válido entre essas classes e, em todo o esboço feito, não obedeceu à notação oficial da UML. Demonstrou ter CP sobre composição de um CD, PR quando ele definiu a lógica de relação entre as classes.	Conheci- mento prévio, pressuposi- ção

Quadro 19 – Avaliação da compreensão do AI7 - grupo experimental digital

Avaliação da compreensão do AI7					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcança- das
Q U E S T I O N Á R I O	Q1	A facilidade de entendimento e absorção de ideias.	D	Resposta vaga com somente um ponto de contato com o texto que é a facilidade de entendimento que a visualidade gráfica do método traz. Presença de PR devido ao aluno entender que a UML veio para facilitar a implementação do trabalho.	Pressupo- sição
	Q2	O padrão no desenvolvimento de sistemas é importante porque possibilita o entendimento e o uso facilitado da linguagem por pessoas de diferentes níveis de conhecimento na área da programação.	A	O aluno identificou exatamente a ideia do texto. Para chegar a essa resposta, acreditamos que ele usou o CP de projetos arquitetônico que foi mencionado no enunciado e, com isso, inferiu (INF) o entendimento do projeto por qualquer pessoa, uma vez que se trata de um modelo padrão.	Conheci- mento prévio, Inferência
T E S T E	Q3	 o desenho de uma classe	A	O aluno construiu o desenho da classe obedecendo à forma e aos constituintes internos da classe conforme a notação apresentada no texto do experimento. Acreditamos que apresentou CP quando demonstrou conhecer os atributos de aluno, INF quando presumiu que o aluno precisa ser matriculado e PR ao apostar no conjunto de dados identificados pelo nome e ainda na operação vital para a existência do aluno – matrícula.	Conheci- mento prévio, pressupo- sição, inferência
	Q8		E	O aluno não respondeu.	

3.1.1.3 Resumo por Questão

a) Grupo experimental impresso

Tabela 2 – Resumo da avaliação da questão 1 - grupo experimental impresso

QUESTÃO 1 – Questionário					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	D			X	
AI2	B	X	X		
AI3	A	X		X	
AI4	B	X		X	
AI5	A	X		X	
AI6	D	X		X	
AI7	D		X		

Tabela 3 – Resumo da avaliação da questão 2 - grupo experimental impresso

QUESTÃO 2 – Questionário					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	B	X	X	X	
AI2	C		X		
AI3	B			X	
AI4	A	X	X		
AI5	E	X			
AI6	A	X		X	
AI7	A	X	X	X	

Tabela 4 – Resumo da avaliação da questão 3 - grupo experimental impresso

QUESTÃO 3 – Teste					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	A	X		X	
AI2	B	X	X	X	
AI3	C	X	X	X	
AI4	A	X	X	X	
AI5	A	X	X	X	
AI6	A	X	X	X	
AI7	A	X	X	X	

Tabela 5 – Resumo da avaliação da questão 8 - grupo experimental impresso

QUESTÃO 8 – Teste						
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição	Previsão

AI1	D	X	X			
AI2	B	X	X	X		
AI3	D	X		X		
AI4	D	X			X	
AI5	C	X	X	X		
AI6	D	X		X		
AI7	E					

b) Grupo experimental digital

Tabela 6 – Resumo da avaliação da questão 1 - grupo experimental digital

QUESTÃO 1 – Questionário					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	C	X			
AI2	D			X	
AI3	D		X		
AI4	D			X	
AI5	A	X	X	X	
AI6	D			X	
AI7	D			X	

Tabela 7 – Resumo da avaliação da questão 2 - grupo experimental digital

QUESTÃO 2 – Questionário					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	D	X			
AI2	A	X	X	X	
AI3	A	X	X		
AI4	D				
AI5	D	X			
AI6	B	X			
AI7	A	X	X		

Tabela 8 – Resumo da avaliação da questão 3 - grupo experimental digital

QUESTÃO 3 – Teste					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	A	X	X	X	X
AI2	B	X	X		
AI3	A	X	X	X	
AI4	A	X		X	
AI5	A	X	X	X	
AI6	A	X	X	X	
AI7	A	X	X	X	

Tabela 9 – Resumo da avaliação da questão 8 - grupo experimental digital

QUESTÃO 8 – Teste							
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição	Previsão	Seleção
AI1	E						
AI2	E						
AI3	E						
AI4	E						
AI5	E						
AI6	D	X		X			
AI7	E						

Para melhor visualizar as estratégias cognitivas empregadas nas repostas que geraram o MCI *qualis* dos alunos, apresentamos os gráficos dos dois grupos experimentais (impresso e digital) por questão, sendo as duas primeiras (1 e 2) do questionário e as duas últimas do teste (3 e 8).

Tabela 10 – MCI *qualis* da questão 1

Questão 1	
MCI <i>qualis</i> (Impresso)	MCI <i>qualis</i> (Digital)
D	C
B	D
A	D
B	D
A	A
D	D
D	D

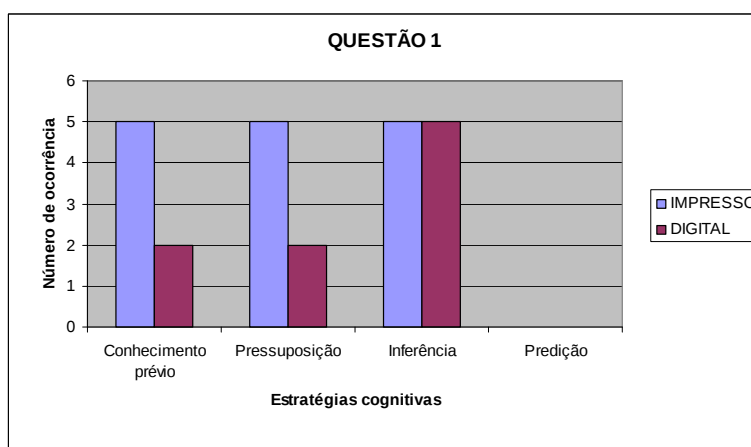


Gráfico 1 – Questão 1 do experimento I

Conforme o gráfico 1, para processar a resposta da questão 1, os alunos do grupo do impresso utilizaram mais seu conhecimento prévio que os do digital. Talvez, por consequência disso, percebemos nas repostas um maior grau de inferência sobre o texto que os do digital, porém identificamos que ambos os grupos tiveram a mesma proporção de pressuposição. Isso justifica o que Kleimam (1997) apontou, no capítulo teórico, quando disse que o conhecimento prévio é essencial para a compreensão por permitir que o leitor faça as inferências necessárias que relacionam partes que geram o todo coerente. A partir desse resultado, também identificamos, segundo os resumos expostos anteriormente, que os alunos do grupo

experimental do impresso, de modo geral na questão 1, alcançaram MCI *qualis* mais aproximados do *expert* do que os do grupo experimental digital.

Tabela 11 – MCI *qualis* da questão 2

Questão 2	
MCI <i>qualis</i> (Impresso)	MCI <i>qualis</i> (Digital)
B	D
C	A
B	A
A	D
E	D
A	B
A	A

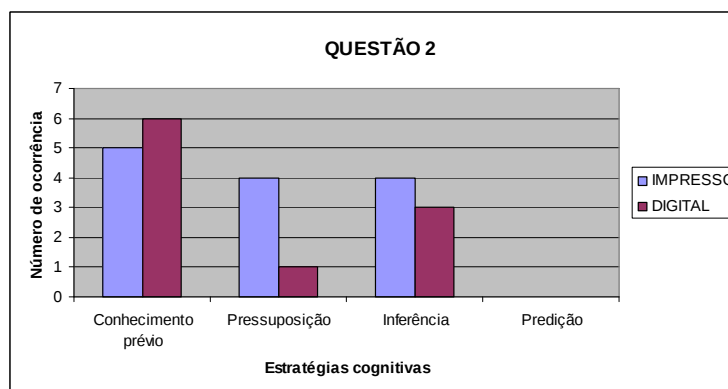


Gráfico 2 – Questão 2 do experimento I

No gráfico 2, observamos que os alunos do grupo experimental digital utilizaram mais o conhecimento prévio do que as demais estratégias cognitivas. Já os do grupo do impresso, apesar de utilizarem um pouco menos o conhecimento prévio, usaram em maior proporção as estratégias de pressuposição e inferência. Em um contexto geral para essa questão, os alunos do grupo experimental do impresso tiveram maior proximidade do MCI *expert* do que os do grupo digital. Acreditamos que o resultado se configurou assim devido aos alunos do grupo experimental digital demonstrarem conhecimento prévio do assunto, mas utilizarem em menor grau sua capacidade de inferir e pressupor.

Essa situação pode ser identificada, segundo a teoria dos espaços mentais de Fauconnier, apresentada no capítulo teórico, como resultado de uma projeção de um domínio no outro, em que os alunos não precisaram do domínio-alvo, porém tinham o domínio-fonte estabilizado. Para Fauconnier (1997), o processo de significação opera pelos espaços mentais projetados entre domínios estáveis (no caso da questão o próprio conhecimento prévio) e locais que, segundo Miranda (1999), são estruturas individuais e dinâmicas que se fazem e desfazem na ação do processamento cognitivo (estratégias cognitivas).

Tabela 12 – MCI *qualis* da questão 3

Questão 3	
MCI <i>qualis</i> (Impresso)	MCI <i>qualis</i> (Digital)
A	A
B	B
C	A
A	A
A	A
A	A
A	A

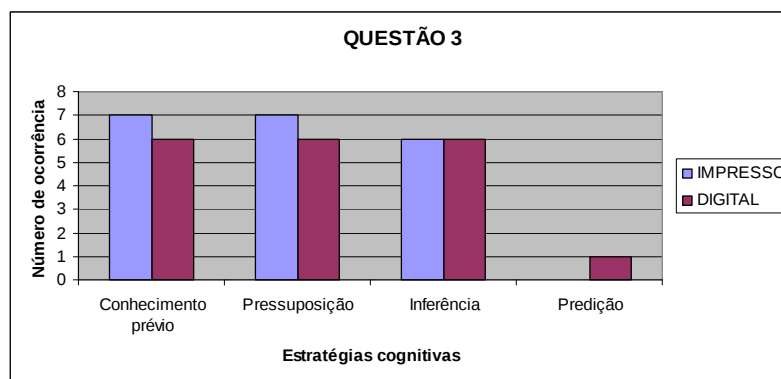


Gráfico 3 – Questão 3 do experimento I

A questão 3 do teste do experimento I solicitava a exposição da resposta conforme a notação gráfica da UML apresentada no texto. Os alunos submetidos ao experimento, por serem da segunda série do curso técnico em informática médio-integrado, ainda não tinham visto nenhum assunto que se assemelhava a projetos de sistemas com UML. Assim, acreditamos que o uso de conhecimento prévio nessa questão, como demonstra o gráfico, é oriunda de outros domínios do conhecimento, como da geometria, da relação de conjuntos e categorias morfológicas da gramática que serviram de insumos para a integração conceitual (FAUCCONIER; TURNER, 2002) e, conseqüentemente, para a formulação das respostas.

Pelo gráfico, identificamos que ambos os grupos experimentais utilizaram as estratégias cognitivas em uma proporção bem aproximada. Esse fato desencadeou também a similaridade encontrada nos MCI *qualis* dos alunos dos dois grupos. Portanto, acreditamos que a utilização aproximada das estratégias cognitivas culminou também na solução aproximada apresentada pelos alunos à questão, ou mesmo, ao MCI *qualis*.

Tabela 13 – MCI *qualis* da questão 8

Questão 8	
MCI <i>qualis</i> (Impresso)	MCI <i>qualis</i> (Digital)
D	E
B	E
D	E
D	E
C	E
D	D
E	E

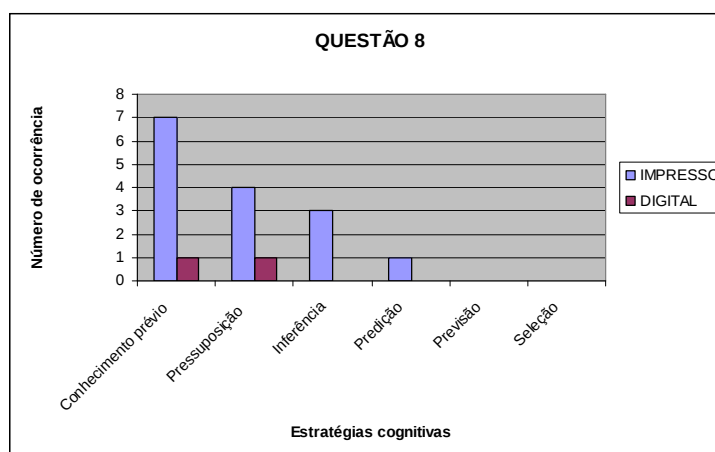


Gráfico 4 – Questão 8 do experimento I

A questão 8 do teste do experimento I, também de representação gráfica, ansiava por uma diversidade maior no uso de estratégias cognitivas, uma vez que suscitava uma análise complexa das relações dos elementos da UML. Como demonstra o gráfico e o resumo, somente um aluno do grupo experimental digital respondeu à questão e, com isso, utilizou algumas estratégias cognitivas, alcançando um MCI *qualis* de ordem D. Os demais do mesmo grupo não responderam.

Quanto aos do grupo experimental do impresso, percebemos, pela demonstração, que usaram em maior grau o conhecimento prévio, que acreditamos ser de outras áreas do conhecimento, como as citadas na questão anterior, devido à natureza das repostas. Com isso, tiveram elementos que os subsidiaram para que inferissem, pressupusessem e ainda predissessem, porém usaram essas estratégias em uma proporção menor que a do conhecimento prévio.

Vale ressaltar que, apesar haver utilização de estratégias cognitivas e conhecimento prévio, os alunos do grupo experimental do impresso, de um modo geral, alcançaram em torno de 50% do MCI *expert* para essa questão, o que nos faz crer que, para que houvesse um índice maior de proximidade ao MCI *expert*, os alunos teriam de projetar mais estratégias cognitivas na leitura do texto. Isso se justifica porque o que atribui sentido a um texto, segundo Marcuschi (2008)¹⁵, é a articulação que ele dispara que visa a produzir sentido a um ator sócio-historicamente situado.

¹⁵ Ver capítulo teórico item 2.3.1.

4.1.1.4 Tempo gasto nas leituras dos dois textos do experimento I

Os títulos dos textos do experimento I são: 1) *Visão Geral da UML* e 2) *Diagrama de Classes*. Utilizamos a numeração indicada para identificar a referência aos hipertextos, uma vez que são dois.

Tabela 14 – Tempo de leitura do experimento I - grupo experimental impressos

Tempo gasto nas leituras dos dois textos impressos do experimento I				
Aluno	Hipertexto	Tempo início	Tempo fim	Tempo gasto
Al1	1	16:26	16:51	00:25
	2	15:58	16:27	00:29
Al2	1	16:26	16:56	00:30
	2	15:59	16:40	00:41
Al3	1	16:26	16:51	00:25
	2	16:00	16:25	00:25
Al4	1	16:27	16:54	00:27
	2	15:59	16:23	00:24
Al5	1	16:27	17:02	00:35
	2	15:59	16:33	00:34
Al6	1	16:26	16:57	00:31
	2	15:58	16:26	00:28
Al7	1	16:29	16:57	00:28
	2	16:01	16:23	00:22
Média tempo de leitura			00:28	

Tabela 15 – Tempo de leitura do experimento I - grupo experimental digital

Tempo gasto nas leituras dos dois hipertextos digitais do experimento I				
Aluno	Texto	Tempo início	Tempo fim	Tempo gasto
Al1	1	16:21	16:58	00:37
	2	15:59	16:47	00:48
Al2	1	16:23	17:06	00:43
	2	16:03	16:46	00:43
Al3	1	16:23	16:53	00:30
	2	16:01	16:37	00:36
Al4	1	16:23	16:54	00:31
	2	16:01	17:10	01:09
Al5	1	16:23	16:45	00:22
	2	16:07	16:45	00:38
Al6	1	16:23	16:46	00:23
	2	16:00	16:36	00:36
Al7	1	16:22	16:59	00:37

	2	16:02	17:09	01:07
Média tempo de leitura			00:40	

Nos textos 1 e 2 do experimento I, os alunos tiveram tempo de leitura descrito na coluna “Tempo gasto” das tabelas 14 e 15. Em média, os alunos do grupo experimental impresso utilizaram menos tempo para ler os textos do que os dos hipertextos (digital). A diferença do tempo de leitura médio entre os grupos é de 12 minutos. Ainda observamos que, no grupo do hipertexto, dois alunos utilizaram quase duas vezes o tempo médio dos demais do mesmo grupo.

Esses dados nos levam a crer que os alunos do grupo experimental digital, talvez pela natureza do hipertexto, foram levados a usar tempo nas navegações oferecidas pelos textos. Esse fato nos comprova o que foi dito por Marcuschi (2001) sobre hipertexto no capítulo teórico deste relato. Para o autor, o hipertexto exige que revisemos nossas estratégias de lidar com os textos, em especial as que dizem respeito à continuidade textual, porque o hipertexto não se comporta numa direção definida.

3.1.1.5 Organização dos Hipertextos

Como o experimento I se constitui de dois hipertextos, chamaremos o hipertexto do capítulo *Visão geral da UML* de hipertexto 1 e outro do capítulo *Diagrama de classes* de hipertexto 2.

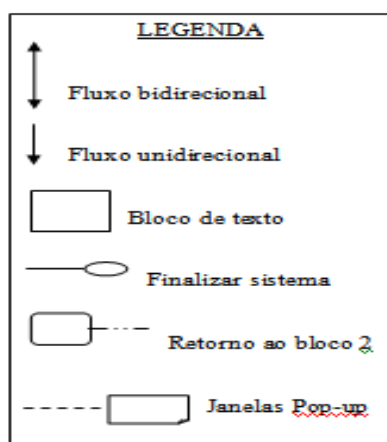
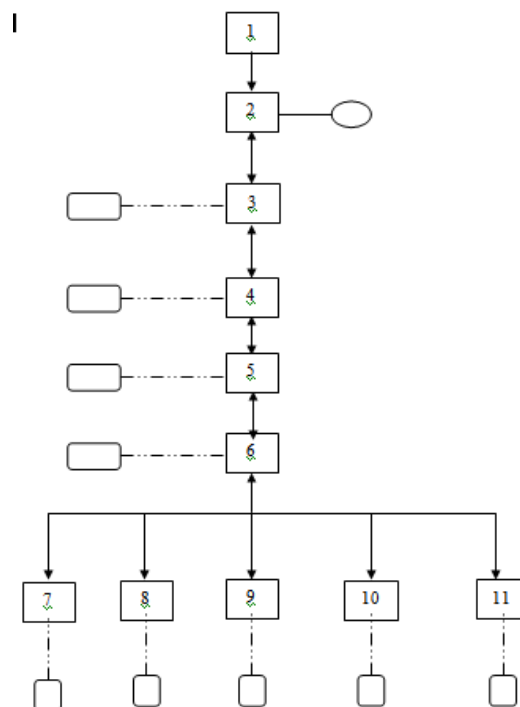


Figura 6 – Legenda dos hipertextos

Tabela 16 – Lista de blocos do hipertexto 1

Lista de bloco de texto do hipertexto 1	
1	Login e senha (início)
2	Boas-vindas – Registro do início da navegabilidade
3	Visão geral da UML
4	Histórico da UML
5	Conhecendo a UML
6	Modelando com a UML
7	Elementos básicos do modelo
8	Relacionamentos
9	Diagramas
10	Regras de formação
11	Finalização – Agradecimentos

Abaixo, expomos o fluxograma de funcionamento do hipertexto relativo ao capítulo *Visão geral da UML* para visualizarmos o condicionamento dos fluxos possíveis de leitura.

Figura 7 – Hipertexto *Visão geral da UML*

A tabela 19 demonstra, através dos passos de navegação dos alunos nos blocos de textos, a trajetória realizada por cada aluno no hipertexto 1 do experimento I.

Tabela 19 – Navegação dos alunos no hipertexto 1

Aluno	Trajeto																											
Bloco	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20	b21	b22	b23	b24	b25	b26	b27	b28
AI1	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	6	7	6										
AI2	1	2	3	4	5	6	7	6	8	6	8	6	9	6	10	6	11	6										
AI3	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	6	5	4	3	2	3	4						
AI4	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	6	5	4	3	2								
AI5	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	6	2	3	4	5	6	9	6	9	6			
AI6	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	6	2	3	4	5	6							
AI7	1	2	3	4	5	6	7	6	9	6	10	6	11	6	2	3	4	5	6	9	6							

No hipertexto 1, percebemos, pelo fluxo do trajeto, que, de modo geral, os alunos cumpriram a mesma trajetória até o bloco 8, a partir daí começaram a diferenciar o passeio pelo texto. O aluno 5 foi quem mais navegou no texto iniciando seu regresso ao início do texto no bloco 17. Os alunos 5, 6 e 7 iniciaram novamente a navegação pelo topo do hipertexto, já os alunos 3 e 4 reiniciaram o texto, porém não concluíram a trajetória. Os alunos 1 e 2 navegaram de forma sequencial e não regressaram ao topo como os demais, e o aluno 7 não passou pelo bloco 8 que falava de relacionamentos na UML, assunto presente em grande parte das questões dos instrumentos.

Na tabela 20, descrevemos a navegação dos alunos no hipertexto 2 do experimento I. Assim, podemos acompanhar o trajeto dos alunos dentro dos blocos de textos disponíveis no hipertexto.

Tabela 20 – Navegação dos alunos no hipertexto 2

Aluno	Trajeto																											
Bloco	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20	b21	b22	b23	b24	b25	b26	b27	b28
AI1	1	2	3	4	5	4	6	4	7	9	7	10	11	10	12	10	13	10	14	10	7	4	8					
AI2	1	2	3	4	5	4	6	4	7	10	11	10	12	10	13	10	14	10	7	4	8	2	3	4				
AI3	1	2	3	4	5	4	6	4	7	9	7	10	11	10	12	10	13	10	14	10	7	4	8	2	3	4		
AI4	1	2	3	4	3	4	5	4	6	4	7	9	7	10	11	10	13	10	12	10	14	10	7	4	8	4	8	4
AI5	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	6	2	3	4	5	6	9	6	9	6			
AI6	1	2	3	4	5	4	6	4	7	9	7	10	12	10	11	10	13	10	14	10	7	4	8	4	8	2		
AI7	1	2	3	4	5	4	6	4	7	10	11	10	12	10	13	10	14	10	7	9	7	4	8					

Já no hipertexto 2, as trajetórias começaram se diferenciar no bloco 5. Exatamente nesse ponto, o aluno 4 regressou um bloco e, a partir daí, os caminhos foram se modificando. Nesse hipertexto, os alunos também tinham a opção de

consultar *popups* nos blocos 7 e 9, e todos visitaram as duas *popups* presentes em cada bloco, com exceção do aluno 2 que não visitou o bloco 9. Nesse hipertexto, os regressos se deram em blocos diferentes e todos regressaram em algum momento e seguiram novos trajetos.

3.2 EXPERIMENTO II

O experimento II se constitui de um artigo científico sobre UML em duas versões: uma adaptada do artigo com redução do índice de multimodalidade; e outra, o mesmo artigo organizado em um hipertexto digital com aumento do índice de multimodalidade em função do suporte. Para esse experimento, consideramos os dados relativos a quatro alunos do grupo experimental do texto impresso e quatro alunos do grupo experimental no suporte digital e procedemos à análise segundo as categorias explicadas no item da metodologia da pesquisa.

3.2.1 Análise do Experimento II

3.2.1.1 Questões do Questionário do Experimento II

1. Tendo em vista a leitura do artigo *A história da UML e seus diagramas*, fale o que você entendeu sobre modelagem de *software* e o papel da UML dentro desse trabalho.
2. Para fabricar uma bicicleta, é necessário que se tenha o modelo da estrutura com cada elemento que a compõe e ainda são necessários estudos que comprovem a eficiência no funcionamento dos componentes da bicicleta quando ela está em movimento (a relação da catraca com a corrente e dos componentes com o aro, o acionamento dos freios e a repercussão dessa ação no movimento da bicicleta). Para modelar um sistema orientado a objetos, a UML utiliza treze diagramas. Parte deles tem por finalidade

representar aspectos de estrutura e composição do *software* e a outra parte representa o comportamento dos componentes do sistema quando está em execução. Sabendo disso e de posse da leitura do artigo, fale sobre o que você entende por modelar um sistema computacional de forma estrutural e comportamental.

Na questão 1, o conceito modelagem de *software* era facilmente localizado no texto, assim o aluno teria facilidade em responder e, com isso, talvez evitasse a atividade cognitiva de interpretação e usasse o conceito do autor. Para avaliar a presença das estratégias cognitivas, observamos se o aluno compreendeu o trabalho da UML dentro da modelagem.

A questão 2 forneceu informações do senso comum dos alunos (funcionamento da bicicleta) que os ajudariam a entender os conceitos propostos através de projeção entre domínios. Portanto, avaliamos, em especial, se o aluno usou o conceito do senso comum para ativar as estratégias cognitivas.

3.2.1.2 Análise por Aluno

O quadro 20 apresenta as respostas e as estratégias cognitivas esperadas.

Quadro 20 – Respostas e estratégias cognitivas esperadas no experimento II

Questões		MCI <i>expert</i>	EC esperadas
QUESTIONÁRIO	Q1	Esperava-se que os alunos respondessem que a modelagem de <i>software</i> pode ser entendida como a atividade de construir modelos que expliquem as características ou o comportamento (funcionalidades) de um <i>software</i> , e a UML é a notação gráfica que padroniza esses modelos e a interligação entre eles.	Conhecimento prévio, inferência, pressuposição, predição
	Q2	Esperava-se que os alunos respondessem que modelar um sistema de forma estrutural é atender para os aspectos estáticos que compõem o sistema como as classes, as interfaces, as colaborações e os componentes; e comportamental é uma descrição do funcionamento dos aspectos dinâmicos do seu estado de funcionamento, ou mesmo artefatos que sofrem alteração em função de um evento.	Conhecimento prévio, inferência, pressuposição, predição

a) Alunos do grupo experimental impresso

Quadro 21 - Avaliação da compreensão do AI1 - grupo experimental impresso

Avaliação da compreensão do AI1					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
	Q1	Modelagem serve para construir modelos que expliquem as características de um <i>software</i> . UML é a linguagem usada para especificar o desenvolvimento de um sistema de <i>software</i> .	A	O aluno demonstrou ter compreendido os pontos centrais que o texto aponta: construção de modelos que representam o <i>software</i> e a UML como notação oficial desses modelos. Apresenta CP sobre representatividade através de modelos; infere (INF) que a UML é uma linguagem, tendo em vista as linguagens de programação; PR de que a UML é a linguagem que especifica o desenvolvimento de <i>software</i> e PD quando usou as pistas dadas pelo texto para construir o conceito de UML.	Conhecimento prévio, inferência, pressuposição e predição
	Q2	Comportamental: tem função de descrever o sistema operacional representando as alterações que nele podem ser feitas. Estrutural: tem função de representar a estrutura do modelo do sistema operacional.	D	O aluno interligou fragmentos dos conceitos apresentados no texto a sistemas operacionais, o que o texto não trata, e não utilizou o exemplo da bicicleta. Nesse caso, apesar de haver alguns pontos de contato entre o que ele descreveu sobre que é comportamental e estrutural, o aluno e o texto não estão falando do mesmo objeto. Trata-se de um equívoco em que o aluno, de posse do CP sobre informática, pressupôs (PR) os conceitos de diagrama comportamental e estrutural baseado no funcionamento do sistema operacional.	Conhecimento prévio, pressuposição

Quadro 22 - Avaliação da compreensão do AI2 - grupo experimental impresso

Avaliação da compreensão do AI2					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
	Q1	A modelagem de <i>software</i> criou um “manual” para um determinado <i>software</i> . UML é uma “linguagem” usada na modelagem de sistemas.	B	Quando o aluno diz que a modelagem criou um manual, foi a categoria mais próxima que ele encontrou para falar sobre a condução do trabalho de modelagem, porém limitou a um determinado <i>software</i> , sendo que o manual é para qualquer <i>software</i> . O mais importante é que identificou a UML como linguagem para modelar, o que acreditamos ser resultado do CP sobre linguagens de programação e modelos de <i>software</i> e também inferiu (INF) que UML é um manual.	Conhecimento prévio, inferência
	Q2	Forma estrutural trata de aspecto estrutural e o comportamental, do comportamento do <i>software</i> , como ele se comporta depois de ser posto para funcionar.	E	O aluno não esclareceu os conceitos que evocou e ainda se confundiu ao dizer que o aspecto comportamental é como se comporta depois de posto para funcionar. O aluno não compreendeu a essência da modelagem, que é projetar, mas pressupôs (PR) que a estrutura e o comportamento são aspectos observados após o <i>software</i> entrar em execução.	Pressuposição

Quadro 23 - Avaliação da compreensão do AI3 - grupo experimental impresso

Avaliação da compreensão do AI3					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
Q U E S T I O N Á R I O	Q1	Modelagem de <i>software</i> é toda a linguagem que você usa para programar um <i>software</i> , e UML é o tipo de linguagem de comunicação que você usa para fazer isso.	E	Modelagem e UML não são linguagens para programar, mas para construir modelos conforme a notação oficial UML. O aluno equivocou-se quanto aos conceitos, porém, mesmo dentro do equívoco, pressupôs (PR) conceitos de modelagem e UML e ainda demonstrou CP ao citar programação de <i>software</i> .	Conhecimento prévio, pressuposição
	Q2	Quando você está montando um sistema, você tem que se preocupar em montar a estrutura corretamente, fazendo com que todo sistema funcione normalmente.	E	O aluno não atentou para a função da modelagem e falou de sistemas como se estivesse na fase de construção de códigos. Em nenhum momento, deu pistas de que usou o exemplo da bicicleta. Apresentou CP de montagem e estruturação de um modo geral.	Conhecimento prévio

Quadro 24 - Avaliação da compreensão do AI4 - grupo experimental impresso

Avaliação da compreensão do AI4					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
Q U E S T I O N Á	Q1	A modelagem de <i>software</i> é a atividade de construir <i>software</i> . A UML é uma linguagem para especificação.	B	O aluno expôs com clareza o conceito e o objetivo da UML, a única falha foi atribuir modelagem à construção de <i>software</i> , quando na verdade é para construir modelos. Utilizou CP sobre desenvolvimento de <i>software</i> e linguagem de programação, pressupôs (PR) que a UML é uma linguagem e ainda inferiu (INF) que essa linguagem tem a função de especificar os dados do novo <i>software</i> .	Conhecimento prévio, pressuposição, inferência

R I O	Q2	Estrutural: existem para visualizar, especificar, construir e documentar os aspectos do sistema, ou seja, a representação de seu esqueleto. Comportamental: são voltados para descrever o sistema computacional modelado quando em execução.	E	O aluno copiou literalmente do texto tudo o que respondeu. Assim não conseguimos identificar as estratégias utilizadas.	
-------------	----	---	---	---	--

b) Alunos do grupo experimental digital

Quadro 25 - Avaliação da compreensão do AI1 - grupo experimental digital

Avaliação da compreensão do AI1					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
Q U E S T I O N Á R I O	Q1	Modelagem de <i>software</i> é a construção de modelos que expliquem as características ou comportamentos. E o papel da UML foi na criação de vários documentos como o de infraestrutura, superestrutura e outros.	D	O conceito de modelagem citado pelo aluno é o mesmo do autor, porém ele atribuiu o papel da UML para a criação de documentos como os citados por ele. O aluno retirou essas informações de forma literal do texto, o que nos leva a crer que ele não inferiu ou pressupôs o papel da UML, mas demonstrou CP de documentação de <i>software</i> ao se referir aos vários documentos de um <i>software</i> .	Conhecimento prévio
	Q2	De forma estrutural, é como que a aparência, que forma que é organizado e comportamental a forma que se comporta.	D	De certa forma a aparência está ligada à estrutura. É possível que o aluno tenha pressuposto (PR) isso; e quanto ao comportamental, a resposta não trouxe elementos que sustentassem o conceito, mas demonstrou algum tipo de CP sobre comportamento e estrutura.	Conhecimento prévio, pressuposição

Quadro 26 - Avaliação da compreensão do AI2 - grupo experimental digital

Avaliação da compreensão do AI2					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
	Q1	É a atividade de construir modelos que explique as características e o comportamento do <i>software</i> .	D	O aluno respondeu com as palavras do autor e simplesmente se omitiu em responder a outra parte, que solicitava o papel da UML. Analisamos que o aluno usou PD , pois emitiu uma opinião baseada em pistas presentes no texto.	Predição
	Q2	Modelar um sistema computacional de forma estrutural e comportamental é definir de modo específico cada função e qual comportamento o sistema deve operar.	D	Apesar de a resposta não ter relevância para a pergunta, o aluno demonstrou CP de funções de sistemas e modelos computacionais e pressupôs (PR) que um sistema tem várias funções que precisam ser definidas na fase de modelagem.	Conhecimento prévio, pressuposição

Quadro 27 - Avaliação da compreensão do AI3 - grupo experimental digital

Avaliação da compreensão do AI3					
Q U E S T I O N Á R I O		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
	Q1	Modelagem de <i>software</i> é a ação de criar diagramas que expliquem os aspectos de um <i>software</i> , o papel da UML foi facilitar a interação entre o usuário e o <i>software</i> , tal como a junção dos métodos OMT, OOSE e BOOCH, aproveitando as vantagens de cada um.	C	O aluno expôs que modelagem é a criação de diagramas, o que na verdade é. Para responder assim, acreditamos que ele conheça o trabalho dos diagramas que genericamente são chamados de modelos. Isso representa o CP . Quanto a outra parte da questão, perdeu-se ao dizer o papel da UML na modelagem, porém foi essa a proposição que o aluno pressupôs (PR). Apresentou também predição ao tentar adivinhar o papel da UML.	Conhecimento prévio, pressuposição e predição
	Q2	A forma estrutural consiste em descrever a estrutura do sistema tal como suas características, já a forma comportamental descreve as consequências dos atos	A	O aluno compreendeu a essência do texto, inclusive pressupôs (PR) a dinamicidade da forma comportamental ao atribuir o comportamento às ações do	Conhecimento prévio, pressuposição, inferência,

		executados pelo usuário, assim acontece com a bicicleta, a ação de pedalar ocasiona no movimento da engrenagem e consequentemente no movimento da bicicleta, avalia-se então a eficiência do sistema.		usuário. Em especial demonstrou inferência (INF) por ter percebido a relação do exemplo da bicicleta com os conceitos de modelagem, o que nos apontou também CP dos dois domínios: modelagem e funcionamento da bicicleta. Por fim, apresentou PD ao utilizar um grande número de pistas do texto para compor a resposta.	predição
--	--	---	--	--	----------

Quadro 28 - Avaliação da compreensão do AI4 - grupo experimental digital

Avaliação da compreensão do AI4					
		Resposta do aluno	MCI <i>qualis</i>	Avaliação	EC alcançadas
Q U E S T I O N Á R I O	Q1	Modelagem de <i>software</i> é construir modelos que expliquem o comportamento de <i>softwares</i> . UML foi juntar três sistemas nele (facilitou a interação entre usuário e <i>software</i>).	D	O aluno respondeu ao conceito modelagem com as palavras do autor e fugiu do eixo da pergunta quando deveria falar do papel da UML. Apresentou indício de CP ao falar em usuário, modelos e <i>software</i> .	Conheci- mento prévio
	Q2	É necessário estudo, para modelar um sistema computacional, de forma estrutural, ou seja, a estrutura desse sistema e de forma comportamental, para que o sistema tenha um bom entendimento, entre o usuário e sistema computacional.	E	Resposta completamente desconectada da pergunta. Identificamos CP de funcionamento de sistemas quando o aluno remeteu o comportamento do <i>software</i> às ações do usuário.	Conheci- mento prévio

3.2.1.3 Resumo por Questão

a) Grupo experimental impresso

Tabela 21 – Resumo da avaliação da questão 1 - grupo experimental impresso

QUESTÃO 1					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	A	X	X	X	X
AI2	B	X	X	X	
AI3	E	X		X	
AI4	B	X	X	X	

Tabela 22 – Resumo da avaliação da questão 2 - grupo experimental impresso

QUESTÃO 2					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	D	X		X	
AI2	E			X	
AI3	E	X			
AI4	E				

b) Grupo experimental digital

Tabela 23 – Resumo da avaliação da questão 1 - grupo experimental digital

QUESTÃO 1					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	D	X			
AI2	D				X
AI3	C	X		X	X
AI4	D	X			

Tabela 24 – Resumo da avaliação da questão 2 - grupo experimental digital

QUESTÃO 2					
Alunos	MCI <i>qualis</i>	Conhecimento prévio	Inferência	Pressuposição	Predição
AI1	D	X		X	
AI2	D	X		X	
AI3	A	X	X	X	X
AI4	E	X			

Para facilitar a visualização do uso das estratégias cognitivas no desenvolvimento das respostas dos alunos, o MCI *qualis*, apresentamos o gráfico relativo às duas questões do experimento II, contemplando os dois grupos experimentais (impresso e digital).

Tabela 25 – MCI *qualis* da questão 1

Questão1	
MCI <i>qualis</i> (Impresso)	MCI <i>qualis</i> (Digital)
A	D
B	D
E	C
B	D

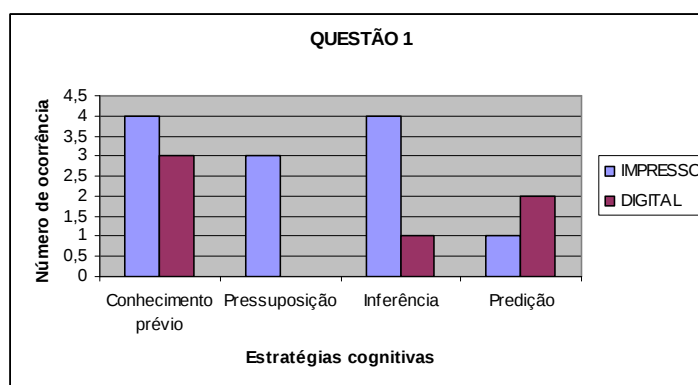


Gráfico 5 – Questão 1 do experimento II

Observando o gráfico 5, percebemos que os alunos do grupo experimental impresso utilizaram estratégias cognitivas como: conhecimento prévio, pressuposição e inferência em uma proporção maior que os do grupo experimental digital. De um modo geral, esses alunos também obtiveram MCI *qualis* mais próximos do MCI *expert* que os do grupo experimental digital. Esses dados nos levam a atestar que o ator da leitura que efetivar uma proporção maior das estratégias cognitivas terá uma possibilidade maior de compreensão. A teoria dos espaços mentais interpretada por Tenuta (2010) reza que a produção ou a interpretação de expressões linguísticas requer a utilização de habilidades psicológicas gerais, das quais, identificamos algumas e demonstramos seu papel na formação da compreensão.

Somente a estratégia predição foi usada em maior proporção pelo grupo experimental impresso. A explicação disso está nas avaliações das repostas presentes no formulário, nas quais identificamos que os alunos, para procederem à resposta, buscaram pistas no texto e elaboraram adivinhações intuitivas.

Esse fato pode ter ocorrido devido à natureza do hipertexto¹⁶ favorecer o surgimento dessa estratégia. Assim, no momento da navegação entre os blocos de

¹⁶ Definido por Cavalcante (2005) no item 2.3.3 do capítulo teórico.

textos, o aluno capturou as pistas que julgou mais produtivas e, a partir de seu conhecimento prévio, elaborou as intuições, o que nos leva a crer que o hipertexto suportado pelo computador favoreceu essa estratégia.

Tabela 26 – MCI *qualis* da questão 2.

Questão 2	
MCI <i>qualis</i> (Impresso)	MCI <i>qualis</i> (Digital)
D	D
E	D
E	A
E	E

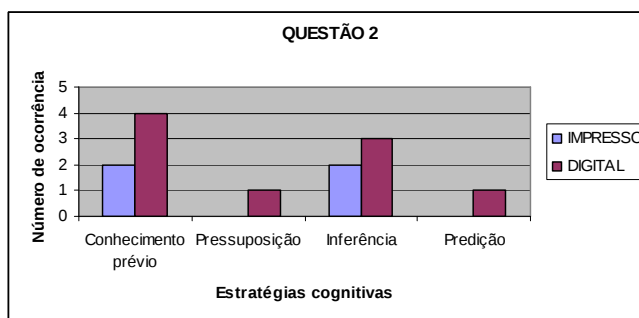


Gráfico 6 – Questão 2 do experimento II

Na questão 2, os alunos do grupo experimental digital superaram os do impresso na utilização de estratégias cognitivas, o que ainda não tinha acontecido nas análises deste relato.

É importante lembrarmos que o enunciado dessa questão fornecia informações do conhecimento cotidiano dos alunos para ajudá-los a processar o novo conceito. A informação citada no enunciado, na verdade, é um MCI da estrutura e do funcionamento de uma bicicleta, que está na experiência diária de alunos adolescentes. A intenção era que os alunos de posse do conhecimento do MCI da bicicleta constituíssem o MCI de modelos de UML.

A teoria de redes de integração conceitual de Fauconnier e Turner (2002) assegura que operações cognitivas efetuadas por seres humanos projetam domínios, realizam mapeamentos analógicos e integram ações em domínios emergentes. Essa foi a nossa intenção quando propusemos a questão 2.

Como resultado, tivemos uma ação mais enérgica do grupo experimental digital na utilização das estratégias cognitivas e, comprovado nas experiências deste relato, um alcance mais próximo do MCI *expert* do que o grupo que utilizou as estratégias em menor proporção.

Quanto à integração conceitual, o AI3 do grupo experimental digital alcançou em sua totalidade a projeção de domínios e conseguiu com excelência a formação do novo conceito a partir do MCI cotidiano.

3.2.1.4 Tempo gasto nas leituras dos textos do experimento II

Tabela 27 – Tempo de leitura do texto do experimento II

Tempo gasto na leitura do texto impresso do experimento II			
Aluno	Tempo início	Tempo fim	Tempo gasto
Al1	15:15	15:45	00:30
Al2	15:13	15:51	00:38
Al3	15:10	15:43	00:33
Al4	15:12	15:30	00:18
Média tempo de leitura			00:29

Tabela 28 – Tempo de leitura do hipertexto do experimento II

Tempo gasto na leitura do texto digital do experimento II			
Aluno	Tempo início	Tempo fim	Tempo gasto
Al1	15:21	15:53	00:32
Al2	15:21	15:54	00:33
Al3	15:21	15:58	00:37
Al4	15:43	16:16	00:33
Média tempo de leitura			00:33

No experimento II, os alunos tiveram tempo de leitura descrito nas tabelas 27 e 28. Observamos que, assim como no experimento I, os alunos do grupo experimental digital utilizaram mais tempo que os do impresso, porém agora a diferença não é tão significativa quanto no experimento anterior. A explicação disso está no índice de multimodalidade que foi reduzido ao extremo no impresso. Nessa leitura, os alunos só contaram com negrito e enumeração das partes do texto.

3.2.1.5 Organização do Hipertexto

Tabela 29 – Lista de blocos do hipertexto do experimento II

Lista de bloco de texto no hipertexto 3	
1	Login e senha (início)
2	Boas-vindas – Registro do início da navegabilidade
3	Introdução ao assunto do artigo
4	Histórico da UML
5	Estrutura da UML
6	Diagramas da UML
7	Diagrama de classes / pacote
8	Diagrama de componente / implantação
9	Diagrama de caso de uso / sequência / máquina de estados
10	Diagrama de comunicação / atividades
11	Diagrama de temporização
12	Finalização – Agradecimentos

Tabela 30 – Lista de *popup* do hipertexto do experimento II

Lista de janelas <i>popup</i> 3	
1	Conceito de UML
2	Mais sobre história da UML

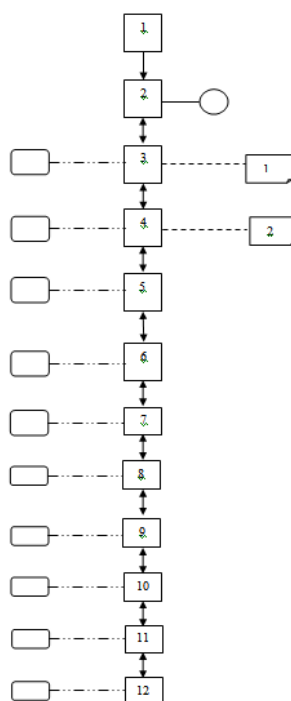


Figura 9 – Organização do hipertexto do experimento II

Para melhor visualizarmos a trajetória dos alunos dentro do hipertexto do experimento II, ilustramos a sequência de navegação nos blocos de texto utilizada por cada aluno na tabela 31.

Tabela 31 – Navegação dos alunos no hipertexto do experimento II

Aluno	Trajeto																											
Bloco	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20	b21	b22	b23	b24	b25	b26	b27	b28
AI1	1	2	3	4	5	6	7	8	9	10	11	10	9	8	7	8	9	10	11									
AI2	1	2	3	4	5	6	7	8	9	10	11	12	11	10	9	8	7	2	3	4								
AI3	1	2	1	2	3	4	5	6	7	8	9	10	11	12	11	10	11	12	11	10	9	2						
AI4	1	2	3	4	5	6	7	8	9	10	11	10	9	10	11	12	2	3										

Na figura 9, podemos observar o organograma da adaptação do artigo científico *A história da UML e seus diagramas*, de Thânia Clair de Souza Vargas, em hipertexto. Devido à natureza teórica do assunto do artigo, ao desenvolver a versão hipertextual, não tivemos muitas opções de bifurcação ou de novos caminhos para gerar um hipertexto mais hierárquico. Porém essa característica não inibiu os alunos de fluírem nos blocos de textos de forma a irem, virem e refazerem trajetos, como mostra a tabela 31, eles já começaram a diferir os caminhos uns dos outros no bloco 3. Os passeios que os alunos realizaram com vários regressos podem ser causados pela presença de artefatos de multimodalidade que os ajudaram a visualizar os conceitos apresentados no texto.

3.3 PONDERAÇÕES GLOBAIS

Assim, diante da completude dos experimentos, pudemos constatar os seguintes resultados a partir da pesquisa:

a) No experimento I: com os dois grupos experimentais com o foco no suporte.

- **Grupo experimental impresso:** após analisar os dados dos alunos que realizaram a leitura no suporte impresso, identificamos no corpus, em sua maioria, respostas com algum tipo de conexão com o texto base e, em alguns casos (em mais da metade), os alunos ressaltaram pontos importantes da leitura. Por essa razão, esses alunos alcançaram um nível de MCI *qualis* mais aproximados do MCI *expert* que os do grupo digital, a exceção está na questão 3, na qual os MCIs *qualis* dos dois grupos se aproximaram.

Mensuramos também a presença de estratégias cognitivas e conhecimento prévio nas questões dos instrumentos e, mais uma vez, percebemos que o grupo do impresso utilizou um maior índice de estratégias cognitivas e de conhecimento prévio que o outro. Ainda observamos o tempo gasto nas leituras do texto do experimento e concluímos, através dos números, que os alunos do grupo experimental impresso utilizaram um tempo consideravelmente menor que o grupo digital.

Outro fator interessante foi que todos os alunos desse grupo experimental de alguma forma tentaram implementar o protótipo sugerido na última questão. Nos seus desenhos, ficou evidente a presença de estratégias cognitivas utilizadas para tentar validar a compressão dos artefatos de um projeto de *software*.

Dessa forma, cremos que o texto impresso, com a presença marcante de multimodalidade, como foi o proposto, é favorável a instigar o trabalho das estratégias cognitivas e demais atividades do cérebro que favorecem a compreensão de textos.

- **Grupo experimental digital:** os alunos do grupo digital demonstraram, nas respostas ou no MCI *qualis*, um maior grau de desconexão com os pontos centrais do texto comparado aos do impresso. Inferimos isso porque as respostas, de modo geral, não contemplaram a essência do que foi perguntado. Assim, o MCI *qualis* desse grupo foi, de modo geral, menor que o do impresso. Da mesma forma, a atuação das estratégias cognitivas ocorreu em menor volume, somente na questão 2 do questionário esses alunos superaram minimamente os do impresso na atividade conhecimento prévio. Ainda na questão 3, esses alunos tiveram uma pequena vantagem na avaliação do MCI *qualis*, também tão mínima que preferíamos dizer que estão empatados nesse ponto.

Quanto ao tempo de leitura, utilizaram consideravelmente mais tempo que os do impresso. Esse fator se confirma na descrição da trajetória dos alunos nos hipertextos, que, segundo demonstração das tabelas 19 e 20, eles iniciaram trajetos parecidos e seguiram caminhos diferentes, uns navegaram

mais entre os blocos, outros menos, outros fizeram um trajeto linear e ainda outros regressaram aos blocos.

Outro fato que nos intrigou nesse grupo foi que somente um aluno tentou responder à questão 8 do teste, que era o desenvolvimento do protótipo. Ele desenvolveu timidamente algumas classes da notação UML e não se preocupou com a relação entre elas. Isso nos fez deduzir que os alunos não se atentaram para o ponto central da leitura do texto *Diagrama de classes*, que demonstrava toda a composição e o funcionamento de um projeto de sistemas. Entender o processo do *software* demandaria dos alunos atividades cognitivas de integração conceitual (FAUCCONIER; TURNER, 2002) e categorização (LAKOFF; JOHNSON, 1980), o que não nos foi possível identificar devido não apresentarem respostas e, com isso, não pudemos avaliar as estratégias cognitivas que alimentam essas atividades.

b) Experimento II: com os dois grupos experimentais com foco no índice de multimodalidade.

- **Grupo experimental impresso:** ao analisar as respostas e qualificar os MCI *qualis* desse grupo, observamos de modo geral que, em relação à primeira questão do instrumento, os alunos confirmaram os resultados que tivemos até agora nos experimentos deste relato. Esses superaram os do grupo digital na qualidade das repostas, conseqüentemente, o índice de utilização de estratégias cognitivas e conhecimento prévio foi maior que os do outro grupo. Porém, nesse experimento, o maior fator de observação foi o índice de multimodalidade presente nos textos, que, para o impresso, foi reduzido ao extremo e ainda assim, nessa questão, os alunos tiveram maior desempenho que os do digital.

Na segunda questão, os resultados se diferenciaram, e os alunos do grupo experimental do impresso, pela primeira vez na experimentação, apresentaram resultados inferiores aos do digital. Nessa questão, os alunos apresentaram respostas confusas, equivocadas, cópias fidedignas de parte do texto e não atentaram para o exemplo da bicicleta que poderia ajudá-los a compreender os conceitos do texto através da integração conceitual. Acreditamos que o fato ocorreu pelo baixo índice de multimodalidade presente no texto, o que se confirma em Descardesi (2002), quando assegura

que a presença de outras formas de representação além do código escrito interfere na mensagem a ser comunicada.

- **Grupo experimental digital:** como citado anteriormente, esse grupo apresentou destaque na segunda questão do experimento. A constituição dos MCI *qualis* forneceu evidências de os alunos terem conhecimento prévio sobre assuntos que subsidiou o conhecimento sobre UML e ainda interligou atribuições aos atores da engenharia de *software*. O mais importante é que os alunos utilizaram o exemplo da bicicleta para relacionar conceitos antigos a novos, favorecendo a integração conceitual entre domínios e o surgimento do novo conceito sobre modelagem com a UML, como se confirma nessa experimentação. Para isso, utilizaram um maior grau de estratégias cognitivas.

CONSIDERAÇÕES FINAIS

Inicialmente, lembramos que o tema compreensão é extremamente complexo, com alto grau de subjetividade, para o qual esta pesquisa desejou ser apenas o ponto de partida. Trata-se de um estudo de caso experimental, cuja questão central foi identificar qual dos dois suportes de texto utilizados (impresso ou digital) é mais propenso a favorecer a aprendizagem de alunos do ensino médio.

Durante a condução da experimentação, trabalhamos a dualidade dos grupos, submetendo-os a instrumentos manipulados, dos quais controlamos o índice de multimodalidade e a forma de apresentação, ora no suporte impresso ora no digital. A transformação dos textos em suporte digital foi desenvolvida por nós, através de programação web, com o único fim de conduzir essa experimentação.

Entendemos que trabalhamos com particularidades dos seres humanos, apesar de os conduzirmos para um ideal único, cada um, distintivamente, trilhou seus caminhos conforme suas concepções, o que foi um fator intrigante para esta investigação e nos fez criteriosamente observar a individualidade, mesmo que, para nossa proposta, o resultado esteja no conjunto de dados.

Para as conclusões finais, observamos os dois experimentos realizados, porque os resultados que buscamos dependiam da visão geral do experimento I, que focaliza o suporte do texto, e do experimento II, que trabalha o índice de multimodalidade presente no texto.

No foco suporte do texto, identificamos que os alunos do grupo experimental impresso tiveram melhores resultados que os do digital, isso ocorreu porque ambos os textos (impresso e digital) tinham presença marcante de multimodalidade, o que assessorou o trabalho das estratégias e das atividades cognitivas dos alunos.

Nesta ótica, chegamos a conclusão que diante da presença de índices maiores de multimodalidade os textos impressos contribuem em maior grau para a compreensão leitora. Explicamos esse fenômeno tendo em vista a prática escolar, que de modo geral, privilegia a cultura impressa. Para o aluno, o objeto a ser estudado geralmente está em livros, revistas, texto impressos e outros. Já os textos

digitais, ainda que no ambiente educacional, representam momentos de descontração, de busca de informações fáceis e rápidas que na maioria dos casos são desconexas da vida escolar dos alunos.

Já no foco índice de multimodalidade, que no experimento II para o texto impresso foi quase nulo, implicou em perda de estratégias e atividades cognitivas culminando em menor grau de compreensão dos alunos do grupo experimental impresso. Por outro lado, os alunos submetidos ao texto digital com alto índice de multimodalidade obtiveram melhor desempenho quanto à compreensão leitora, o que nos faz crer que a multimodalidade é fator consubstancial no processo de compreender.

Acreditamos que mais fatores influenciaram os resultados obtidos como: a organização hipertextual idealizada para o experimento e navegabilidade individual desenvolvida pelos alunos no ato da leitura. As tabelas de navegabilidade comprovam que houve variações nas consultas aos blocos de textos demandando tempos de leituras diferentes, e ainda, a disposição dos blocos de textos demonstradas nos mapas dos hipertextos instigaram diversos trajetos de leitura e releitura dentro da estrutura hipertextual montada. Assim, para futuros trabalhos nesta linha de pesquisa, aconselhamos uma investigação detalhada desses fatores como forma de ampliarmos o conhecimento na área de leitura e cognição.

Com esse estudo, podemos concluir que quando um leitor busca conhecimentos em textos, ele mobiliza estratégias cognitivas e conhecimento prévio capazes de oferecer elementos suficientes para a proliferação dos espaços mentais originários de MCI e de categorias de várias origens que, nas atividades de integração conceitual, suscitaram o surgimento do novo conhecimento.

Percebemos também que esse ciclo faz com que a cada pequena experimentação do sujeito no mundo, no nosso caso, a cada leitura, novos MCIs e novas categorias conceituais agreguem às existentes, e, automaticamente, o conhecimento evolui.

De certa maneira, inferimos que compreender é consequência do conhecer, porque todas as atividades e as estratégias cognitivas empregadas na compreensão necessitam, antes de tudo, de bases de conhecimentos diversos para iniciar o trabalho.

Agora, apreciaremos brevemente as questões que originaram a pesquisa, procurando, assim, produzir uma síntese final. As questões centrais foram: qual dos dois suportes de textos multimodais (impressos ou digitais) favorece a compreensão de alunos do ensino médio? E a multimodalidade em textos suportados pelo computador facilita a compreensão?

Sem pretensão de encerrar essa discussão, podemos dizer, a partir dos dados analisados quanto ao suporte do texto e à presença da multimodalidade, o que inicialmente hipotetizamos sobre o suporte digital (que poderia favorecer a compreensão em maior grau de alunos do ensino médio) não se confirmou quando o texto utilizado no experimento, no caso o I, tinha a mesma proporção de multimodalidade para os dois suportes. Nessa comparação, percebemos que os alunos do grupo experimental impresso tiveram mais sucesso nas respostas empreendidas. Porém, ao reduzir a presença de multimodalidade no texto impresso (experimento II), observamos que os alunos submetidos a esse texto reduziram o seu desempenho nas respostas em contraposição aos do grupo digital, que avançaram na qualidade das respostas.

Entendemos que a presença de multimodalidade em maior ou menor índice nos textos nos dois suportes de fato contribuiu para que tivéssemos os resultados alcançados. Portanto, podemos concluir que a presença em maior grau de multimodalidade, seja no suporte impresso ou no digital, favorece a ação das atividades e das estratégias cognitivas que culminam na compreensão leitora.

REFERÊNCIAS

- ALLIENDE, F.; CONDEMARIN, M. **Leitura**: teoria, avaliação e desenvolvimento. Porto Alegre: Artes Médicas, 1987.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências: elaboração. Rio de Janeiro, 2002.
- _____. **NBR 6024**: informação e documentação: numeração progressiva das seções de um documento escrito: apresentação. Rio de Janeiro, 2003.
- _____. **NBR 6027**: informação e documentação: sumário: apresentação. Rio de Janeiro, 2003.
- _____. **NBR 6028**: informação e documentação: resumo: apresentação. Rio de Janeiro, 2003.
- _____. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2002.
- _____. **NBR 14724**: informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2005.
- BEAUGRANDE, R. de. **New Foundation for a science of text and discourse**: cognition, communication, and the freedom of knowledge and society. Norwood: Ablex, 1997.
- BONONI, A. **Gêneros textuais e cognição**: um estudo sobre a organização cognitiva da identidade dos textos. Florianópolis: Insular, 2002.
- BRAGA, R. M.; SILVESTRE, M. F. B. **Construindo o leitor competente**: atividades de leitura interativa para a sala de aula. São Paulo: Global, 2002.
- BRANDÃO, A. C. P.; SPINILLO, A. G. Aspectos gerais e específicos na compreensão de textos. **Psicologia: Reflexão e Crítica**, Porto Alegre, v. 11, n. 2, p. 253-272, 1998.
- BRITO, G. M. de; VERRI, V. S. da S. A leitura e o universo do leitor: uma experiência em sala de aula. **Linguagem & Ensino**, Pelotas, v. 7, n. 1, p. 53-80, jan./jul. 2004.
- BRONCKART, J. P. L'enseignement des discours. De l'appropriation partique à la maîtrise formelle. In: ALMGREN M. et al. (Org.). **Research on child language acquisition**. New-York: Cascadilha Press, 2001. p. 1-16.
- CAGLIARI, L. C. **Alfabetização e linguística**. São Paulo: Scipione, 1989.
- CAVALCANTE, M. C. B. Mapeamento e produção de sentido: os links no hipertexto. In: MARCUSCHI, L. A.; XAVIER, A. C. **Hipertexto e gêneros digitais**. 2. ed. Rio de Janeiro: Lucerna, 2005. p. 163-169.

CERVO, A. L.; BERVIAN P. A.; SILVA, R. **Metodologia científica**. 6. ed. São Paulo: Pearson Prentice Hall, 2007.

DESCARDECI, M. A. A. de S. Ler o mundo: um olhar através da semiótica social. ETD – **Educação Temática Digital**, v. 3, n. 2, p. 19-26, Campinas: Unicamp, jun. 2002.

FAUCCONIER, G. **Mappings in thought and language**. Cambridge: Cambridge University Press, 1997.

_____; SWEETSER, E. **Spaces, worlds and grammar**. Chicago: The University of Chicago Press, 1996.

_____; TURNER, M. **The way we think**: conceptual bending and the mind's hidden complexit. New York: Basic Books, 2002.

FELTES, H. P. M. Cognição e metacognição: aplicação em uma atividade psicolinguística com teste Cloze. In: MACEDO, A. C. P. de; FELTES, H. P. de M.; FARIAS, E. M. P. (Org.). **Cognição e linguística**: explorando territórios, mapeamentos e percursos. Caxias do Sul: Educus; Porto Alegre: Edipucrs, 2008. v. 1. p. 249-283.

_____. **Semântica cognitiva**: ilhas, pontes e teias. Porto Alegre: Edipucrs; São Paulo: Annablume, 2007.

FOUCAMBERT, J. **A leitura em questão**. Porto Alegre: Artes Médicas, 1994.

GONÇALVES, S. Aprender a ler e compreensão do texto: processos cognitivos e estratégias de ensino. **Revista Iberoamericana de Educación**, n. 46, p. 135-151, 2008. Disponível em: <www.rieoei.org/rie46a07.htm>. Acesso em: 18 jul. 2010.

GOULEMOT, J. M. Da leitura como produção de sentidos. In: CHARTIER, R. (Org.). **Práticas de leitura**. 2. ed. São Paulo: Estação Liberdade, 2001.

IZIQUIERDO, I. **Memória**. Porto Alegre: Artmed, 2002.

JACOB, E.; SHAW, D. Sócio-cognitive perpectives of representation. **Annual Review Information Science Technology**, v. 33, p. 131-185, 1998.

KATO, M. A. Estratégias gramaticais e lexicais em leitura em língua estrangeira. **Cadernos PUC**, p. 132-141, 1984.

_____. **O aprendizado da leitura**. 6. ed. São Paulo: Martins Fontes, 2007.

KLEIMAN, A. **Texto e leitor**: aspectos cognitivos da leitura. 9. ed. Campinas: Pontes, 2004.

KOCH, I. V.; CUNHA-LIMA, M. L. Do cognitivismo ao sociocognitivismo. In: MUSSALIM, F.; BENTES, A. C. (Org.). **Introdução à linguística**: fundamentos epistemológicos. 3. ed. São Paulo: Cortez, 2007.

KRESS, G. Visual and verbal modes of representation. In: SNYDER, I. (Ed.) **Page to screen: taking literacy into the electronic era.** London: Routledge, 1998.

_____; VAN LEEUWEN, T. **Reading images: a grammar of visual design.** Londres: Routledge, 1996.

LAKOFF, G. **Women, fire and dangerous things: what categories reveal about the mind.** Chicago: University of Chicago Press, 1987.

_____; JOHNSON, M. **Metáforas da vida cotidiana.** Campinas: Mercado de Letras; São Paulo: EDUC, 2002.

_____. **Philosophy in the flesh: the embodied mind and its challenge to western thought.** New York: Basic Books, 1999.

LEITE, F. **Comunicação e cognição: os efeitos da propaganda contraintuitiva no deslocamento de crenças e estereótipos.** Disponível em: <http://www.cienciasecognicao.org/pdf/v13/cec_v13-1_mar_2008.pdf>. Acesso em: 29 maio 2010.

MARCUSCHI, L. A. O hipertexto como um novo espaço de escrita em sala de aula. **Linguagem & Ensino**, Pelotas, v. 4, n. 1, p. 79-112, 2001.

_____. **Produção textual, análise de gêneros e compreensão.** 2. ed. São Paulo: Parábola Editorial, 2008.

MCCAULEY, R. The role of theories in a theory of concepts. In: NEISSER, U. (Ed.). **Concepts and conceptual development: ecological and intellectual factors in categorization.** New York: Cambridge University Press, 1987. p. 288-308.

MEDEIROS, J. B. **Redação científica: a prática de fichamento, resumo, resenha.** 11. ed. São Paulo: Atlas, 2009.

MELO, A. C. **Desenvolvendo aplicações com UML 2.0: do conceitual à implementação.** 2. ed. Rio de Janeiro: Brasport, 2004.

_____. **Exercitando Modelagem em UML.** Rio de Janeiro: Brasport, 2006.

MERVIS, C. B.; ROSCH, E. Categorização of natural objects. **Annual Review of Psychology**, n. 32, p. 89-115, 1981.

MIRANDA, N. S. Domínios conceptuais e projeções entre domínios: uma introdução ao modelo dos espaços mentais. **Revista de Estudos Linguísticos**, Juiz de Fora, v. 3, n. 1, p. 81-95, 1999.

NUNES, E. K. **A formação de esquemas emergentes significativos como fonte de compreensão leitora: uma abordagem conexcionista.** Disponível em: <<http://www.liberato.com.br/upload/arquivos/0131010716355216.pdf>>. Acesso em: 29 maio 2010.

POPPER, K. S. **A Lógica das ciências sociais.** Rio de Janeiro: Tempo Brasileiro, 1978.

RODRIGUES-LEITE, J. E. **Construção pública do conhecimento**: linguagem e interação na cognição social. 2004. 248f. Tese (Doutorado em Letras e Linguística). Centro de Artes e Comunicação, Universidade Federal do Pernambuco, Recife, 2004.

_____. Construindo versões de mundo: reflexões sobre a atividade de categorização em aulas de português. **Revista do GELNE**, Fortaleza, v. 6, p. 69-90, 2005.

ROSCH, E. Cognitive representation of semantic categories. **Journal of Experimental Psychology**, n. 104, p. 192-233, 1975.

_____. Principles of categorization. In: ROSCH, E.; LLOYD, B. B. (Ed.) **Cognition and categorization**. Hillsdale: Erlbaum, 1978.

SALOMÃO, M. M. Razão, realismo e verdade: o que nos ensina o estudo sociocognitivo da referência. In: BENTES, A. C.; KOCH, I. V.; MORATO, E. M. (Org.). **Referenciação e discurso**. São Paulo: Contexto, 2005.

SARAIVA, A. M. S. Cognição e categorização: uma revisão teórica. In: MACEDO, A. C. P. de; FELTES, H. P. de M.; FARIAS, E. M. P. (Org.). **Cognição e linguística**: explorando territórios, mapeamentos e percursos. Caxias do Sul: Educ; Porto Alegre: Edipucrs, 2008. v. 1. p. 39-70.

SAUSSURE, F. **Curso de linguística geral**. São Paulo: Cultrix, 1977.

SILVA, E. T. **Leitura na escola e na biblioteca**. 3. ed. Campinas: Papirus, 1991.

SMOLENSKY, P. Connexionnisme, l. A. symbolique et cerveau. In: FÁVERO, L. L.; PASCHOAL, M. S. Z. (Org.). **Introduction aux sciences cognitives**. Paris: Gallimard, 1992.

SOARES, M. B. **Linguagem e escola**: uma perspectiva social. 14. ed. São Paulo: Ática, 1996.

SOLÉ, I. **Estratégias de leitura**. Porto Alegre: Artes Médicas Sul, 1998.

TEIXEIRA, J. de F. **Mentes e máquinas**: uma introdução à ciência cognitiva. Porto Alegre: Artes Médicas, 1998.

TENUTA, A. M. Uma breve apresentação das teorias dos espaços mentais e da mesclagem. In: HERMOND, A. B.; ESPÍRITO SANTO, R. S. do; CAVALCANTE, S. M. S. (Org.). **Linguagem e cognição**: diferentes perspectivas, de cada lugar um outro olhar. Belo Horizonte: PUC-MINAS, 2010. p. 85-103.

TERZI, S. B. **A Construção da leitura**. 3. ed. Campinas: Pontes, 2002.

VARELA, F.; THOMPSON, E.; ROSCH, E. **A mente incorporada**: ciências cognitivas e experiência humana. São Paulo: Artmed, 2003.

_____. **The embodied mind**: cognitive science and human experience. Cambridge: The MIT Press, 1991.

VARGAS, T. C. de S. **A história de UML e seus diagramas**. Florianópolis: UFSC/Departamento de Informática e Estatística. Disponível em: <http://projetos.inf.ufsc.br/arquivos_projetos/projeto_721/artigo.tcc.pdf>. Acesso em: 29 maio 2010.

WITTGENSTEIN, L. **Investigações filosóficas**. Trad. José Carlos Bruni. São Paulo: Abril Cultural, 1979.

_____. **Notebooks 1914-16**. Oxford: B. Blackwell, 1961.

XAVIER, A. C. Leitura, texto e hipertexto. In: MARCUSCHI, L. A.; XAVIER, A. C. **Hipertexto e gêneros digitais**. 2. ed. Rio de Janeiro: Lucerna, 2005. p. 170-180.

YATES, S. J. Computer-Mediated Communication. The Future of the Letter? In: BARTON, D.; HALL, N. (Ed.). **Letter writing as a social practice**. Amsterdam/Philadelphia: John Benjamins, 2000. p. 233-251.

ANEXOS

ANEXO I – DESCRIÇÃO DA TECNOLOGIA UTILIZADA NO SISTEMA WEB DOS EXPERIMENTOS.

Para proceder aos experimentos, desenvolvemos um sistema *web* com as tecnologias: a) Java, b) Adobe Flash e c) Postgresql. Descrevemos abaixo as ferramentas para conhecimento do leitor.

- a) Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "bytecode" que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem;
- b) Adobe Flash, ou simplesmente Flash, é um software primariamente de gráfico vetorial - apesar de suportar imagens bitmap e vídeos - utilizado geralmente para a criação de animações interativas que funcionam embutidas num navegador web.
- c) PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (SGBDOR), desenvolvido como projeto de código aberto. Hoje, o PostgreSQL é um dos SGBDs (Sistema Gerenciador de Bancos de Dados) de código aberto mais avançados.

ANEXO II – TEXTOS DO SUPORTE IMPRESSO EXPERIMENTO I E II

TEXTO 1:

Visão Geral da UML

Já diz o ditado que "uma imagem vale por mil palavras" (ou algo parecido!). Duvida? Então vamos fazer o teste! O que a Figura 2.1 representa?

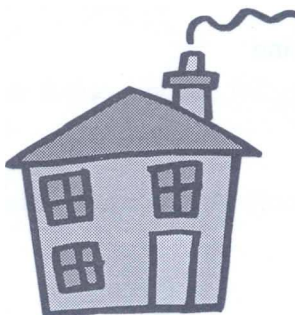


Figura 2.1 - Representação de uma casa

É uma casa. Melhor: é uma casa com dois andares, três janelas (sendo uma no andar inferior), uma porta, chaminé e telhado.

Reparou como uma descrição pode ser extensa, enquanto a imagem pode ser muito mais simples. Todavia, a imagem deve ser simples sem perder a expressividade, pois senão o que seria uma ajuda se transforma em pesadelo.

Assim, todas as boas características de uma representação gráfica, ou no nosso caso, de uma modelagem gráfica, estão presentes nos diagramas da UML.

Por exemplo, leia a descrição a seguir:

"Classe Aluno possui os atributos: matrícula (string), nome (string), data de nascimento (date), data de matrícula (date), valor da mensalidade (real). Possui, também, as operações: Matricular Aluno, Reajustar Mensalidade (recebendo por parâmetro o percentual de reajuste), Trancar Matrícula, Emitir Boleto de Pagamento (recebendo por parâmetro o mês/ano de referência)."

Agora vamos observar a representação gráfica desta descrição, na Figura 2.2:

Alunomatricula: string nome: string dataNascimento: date data Matricula: date
valorMensalidade: real MatricularAlunoO ReajustarMensalidade (percentual: real)
TrancarMatriculaO EmitirBoletoPagto (mesAno: string) Figura 2.2 - Representação da Classe Aluno

É essa facilidade de entendimento e absorção de idéias, não só pelo desenvolvedor mas também e principalmente pelo usuário, que faz com que a UML seja cada vez mais adotada.

2.1 HISTÓRICO DA UML

Desde os primeiros conceitos de orientação a objetos, diversos métodos foram apresentados à comunidade (chegaram a mais de 50 no período compreendido entre 1989 e 1994). Todavia, a grande maioria pecou por tentar estender os métodos estruturados. Os maiores prejudicados eram os usuários que não conseguiam encontrar uma completa satisfação no uso do que havia disponível. Essa situação gerou o que se conheceu como "a guerra dos métodos". Aproximadamente, a partir de meados da década de 90, passamos a conviver com novas abordagens que procuravam trabalhar mais ativamente a idéia do novo paradigma. Nessa época,

vários nomes consagrados de nossa área, por meio de suas publicações e métodos, deram grandes contribuições à orientação a objetos, como: Ward Cunningham e Kent Beck (cartões CRC - Class-Responsibility Collaboration¹¹); Sally Shlaer e Steve Mellor; Jim Odell e James Martin; Peter Coad e Ed Yourdon; James Rumbaugh (OMT - Object Modeling Technique); Grady Booch (método Booch); Ivar Jacobson (OOSE - Object-Oriented Software Engineering).

Com o passar do tempo, cada método ganhava uma fatia diferente do mercado. Juntamente com a divisão do mercado, pairava a acirrada competição. Tentativas de padronização foram propostas, mas não obtiveram sucesso. Por volta de 1993, os métodos que mais cresciam no mercado eram: Booch'93, OMT-2 e OOSE. Todavia, apesar das semelhanças, existiam pontos significativos e fortes em cada método. Resumidamente, o OOSE possuía seu foco em casos de uso (use cases), provendo excelente suporte à engenharia de negócios e análise de requisitos. OMT2 era expressivo na fase de análise dos sistemas de informação. Booch'93 já se destacava na fase de projeto. Ao invés de seguir a linha dos primeiros autores (que procuravam redefinir ou estender os métodos já existentes), Booch, Rumbaugh e Jacobson decidiram unir forças e criar um método único. Seus métodos já estavam evoluindo um em direção ao outro, de maneira independente. Seria mais sensato essa evolução prosseguir de forma conjunta do que individualmente. Estava terminando a "guerra dos métodos".

Os esforços para essa unificação tiveram início em outubro de 1994. James Rumbaugh deixou a General Electric e se juntou à Grady Booch na Rational Software, no intuito de unir seus métodos (Booch e OMT). Em outubro de 95 eles lançaram publicamente o rascunho de seu Método Unificado (assim chamado na época) na versão 0.8. Nesta época, Jacobson (da Objectory) se juntou à equipe e o projeto inicial passou a incorporar o método OOSE. Em junho de 1996, os três autores (que já passaram a ser conhecidos como os três amigos) lançaram uma nova versão. O Método Unificado passou a se chamar UML - Unified Modeling Language. A nova versão foi lançada como UML 0.9. Em outubro de 1996, mais uma versão: a UML 0.91.

Durante todo o ano de 1996, a UML já era vista pelas organizações como uma ótima estratégia para seus negócios. Um requerimento de proposta de padronização denominado RFP (Request for Proposals), emitido pela OMG, veio gerar uma união de forças em prol de produzir uma resposta a este RFP. Houve uma participação ativa da comunidade de engenharia de software, além de várias empresas de software²² que passaram a contribuir no intuito de fortalecer a UML. Essas colaborações muito contribuíram para a UML. Conseguiu-se uma linguagem bem definida, expressiva, poderosa e genericamente aplicável. Uma força de trabalho foi formada para fazer a padronização da UML na área de metodologias. Alguns autores já apresentavam sinais de abrir mão de seus métodos em favor desse padrão.

Em janeiro de 1997 a Rational lançou a versão 1.0 da UML como proposta para a padronização no OMG (Object Management Group). Entre janeiro e julho de 1997, o grupo original de parceiros se expandiu, passando a incluir outros participantes e colaboradores³³, que separadamente haviam submetido respostas ao RFP (OMG). Juntos produziram a versão 1.1 revisada da UML, que tinha por foco principal melhorar a limpidez das semânticas da UML 1.0 e incorporar todas as contribuições dos novos parceiros. Essa versão foi oferecida para padronização ao OMG em Julho de 1997 e em setembro do mesmo ano foi aceita. Em 14 de novembro de 1997, a UML 1.1 foi adotada como padrão pelo OMG.

¹ Desenvolvido em 1989, por Cunningham e Beck, consiste em representar classes em cartões de índice. Registram-se nesses cartões as responsabilidades das classes (propósito) em vez de seus atributos e operações. Além disso, representam-se as outras classes que colaboram com a classe em estudo.

² Entre os parceiros que contribuíram para a definição da UML 1.0 estavam a Digital Equipment Corporation, Hewlett-Packard, i-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, Texas Instruments e Unisys.

³ Juntaram-se ao grupo inicial, entre outros: Andersen Consulting, Ericsson, ObjectTime Limited, Platinum Technology, Ptech, Reich Technologies, Softteam, Sterlink Software e Taskon.

A manutenção da UML passou a ser responsabilidade da RTF (Revision Task Force), pertencente à OMG e dirigida por Cris Kobryn. O objetivo da RTF é aceitar comentários da comunidade em geral a fim de realizar revisões nas especificações, referentes a erros, inconsistências, ambigüidades e pequenas omissões. Essas revisões são guiadas de forma a não provocar uma grande mudança no escopo original da proposta de padronização. Nestes últimos anos novas revisões foram editadas e apresentadas à comunidade: em julho de 1998, a UML 1.2; no final de 1998, a UML 1.3; em maio de 2001, a UML 1.4. Em agosto de 2001, a RTF submeteu ao OMG um relatório provisório da UML 1.5, publicada em março de 2003. Entretanto, a grande mudança está na versão seguinte - a UML 2.0, que hoje é nossa versão atual.

Conhecendo um pouco mais ...

A OMG é uma organização internacional, fundada em 1989. Promove a teoria e prática da tecnologia orientada a objeto em desenvolvimento de sistemas. A patente da organização inclui o estabelecimento de normas industriais e especificações de gerenciamento de objetos para prover um padrão comum para desenvolvimento de aplicações. As principais metas são a reutilização, portabilidade e interoperabilidade de softwares baseados em objeto, em ambientes distribuídos heterogêneos.

2.2 CONHECENDO A UML (UNIFIED MODELING LANGUAGE)

Os métodos, na sua maioria, são compostos de uma linguagem de modelagem (notação gráfica) e de um processo (passos para elaboração de um projeto). A UML é uma linguagem de modelagem; não é um método.

A UML, através de sua estrutura, conduz à criação e leitura de seus modelos, mas não determina quais e nem quando esses modelos precisam ser criados. Essa é uma responsabilidade do processo de desenvolvimento⁴. Desta forma, pode-se usar qualquer processo com a UML, pois esta é independente de processo.

UML (Unified Modeling Language - Linguagem de Modelagem Unificada) é uma linguagem para especificação, visualização, construção e documentação de artefatos⁵ de sistemas de software.

Booch, Rumbaugh e Jacobson acrescentam:

"A UML proporciona uma forma padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis".

Quando falamos em especificação, dizemos que a UML através de modelos precisos, completos e sem ambigüidades, atende às decisões que necessitam ser tomadas para o desenvolvimento e implantação de sistemas.

No desenvolvimento da maioria dos sistemas precisamos de um padrão para modelagem gráfica, a fim de que um desenvolvedor possa escrever seu modelo e qualquer outro possa interpretá-lo sem ambigüidades. Imagine se um analista modelasse uma classe como um

⁴ Cabe ao processo decidir quais artefatos serão produzidos, a equipe e atividades necessárias para criar e gerenciar esses artefatos.

⁵ Exemplos de artefatos: requisitos, arquitetura, projeto, código-fonte, planos do projeto, testes, protótipos, versões, etc.

retângulo e outro resolvesse representar essa mesma classe como um cubo. Seria o caos! A UML, através de sua semântica bem definida, atende plenamente a essa necessidade. Essa visualização precisa dá aos desenvolvedores e às ferramentas condições de interpretar os modelos sem ambigüidades.

A UML auxilia na construção de software, pois permite um mapeamento de seus modelos para as linguagens de programação, ou vice-versa. Assim, é possível gerar código a partir de modelos da UML, ou com um suporte adequado realizar a engenharia reversa, reconstruindo um modelo a partir de sua implementação.

A UML facilita a documentação, pois possui suporte para criação e documentação de vários dos artefatos que são gerados durante o desenvolvimento de um sistema, como: requisitos, arquitetura, projeto, código-fonte, planos de projeto, testes, protótipos e versões.

A UML alcançou dois aspectos muito importantes. Primeiramente, ela terminou com as diversas diferenças existentes entre os métodos de modelagem anteriores. Em segundo lugar, unificou as perspectivas entre muitos sistemas de tipos diferentes (negócios x software), fases de desenvolvimento (análise de requisitos, projeto e implementação) e conceitos internos.

O projeto da UML procurou desenvolver uma linguagem de modelagem que atingisse as seguintes metas:

- Prover à comunidade uma linguagem de modelagem visual pronta para o uso e expressiva, possibilitando desenvolver e intercambiar modelos significativos.
- Fornecer extensibilidade e mecanismos de especialização para estender os conceitos centrais.
- Suportar especificações que são independentes de processos de desenvolvimento e linguagens de programação particulares.
- Prover uma base formal para entendimento da linguagem de modelagem.
- Encorajar o crescimento do mercado de ferramentas de objetos.
- Suportar alto nível de conceitos de desenvolvimento como componentes, colaborações, estruturas e padrões.
- Integrar melhores práticas.

2.3 MODELANDO COM A UML

Para modelarmos sistemas com a UML, trabalhamos com **elementos básicos do modelo, relacionamentos, diagramas e regras de formação**.

2.3.1 Elementos Básicos do Modelo

Podemos citar como os principais elementos básicos para utilização na modelagem de qualquer sistema (em ordem alfabética):

Ações (Action)* - uma unidade básica de especificação de comportamento que representa alguma transformação ou processamento em um sistema. Ações estão contidas em atividades, que provêm seu contexto.

Artefatos (Artifact)* - um pedaço físico de informação que é usado ou produzido por um processo de desenvolvimento. São exemplos de artefatos: modelos, arquivos fontes, scripts e arquivos executáveis.

Atividades (Activity)* - é a especificação de um comportamento parametrizado que é expresso como um fluxo de execução a partir de elementos seqüenciados, cujos elementos primitivos são ações individuais.

Caso de Uso (Use Case) - representa a funcionalidade provida por um sistema. Consiste na descrição de um conjunto de ações organizadas sequencialmente e que são executadas pelo sistema, interagindo com os atores⁶ do mesmo.

Classe (Class) - representa a descrição de um conjunto de objetos que dividem os mesmos atributos, operações (com suas implementações em métodos), relacionamentos e semântica.

Classes Ativas (Active Class) - representam atividades de controle, como threads.

Colaboração (Collaboration) - indica as instâncias e cooperações de elementos que estão envolvidos na realização de alguma tarefa.

Componente (Component) - corresponde a uma parte significativa de um sistema, que é representado modularmente, de forma totalmente substituível.

Estado (State) - é uma condição durante a vida de um objeto ou uma interação durante a qual alguma condição executa alguma ação ou espera por algum evento.

Interação (Interaction) - é um padrão de comunicação com o objetivo de realizar algum propósito.

Interface (Interface) - especifica as operações externamente visíveis de uma classe ou componente. Interfaces não possuem implementação.

Nó (Node) - é um objeto físico existente em tempo de execução que representa um recurso computacional.

Nota (Note) - é um símbolo gráfico contendo informação textual que possibilita a representação de informações do tipo restrições, comentários e descrição de métodos.

Pacote (Package) - é o agrupamento de elementos de modelo. Pacotes podem ser aninhados a outros pacotes.

Partes (Part)* - um elemento que representa um conjunto de instâncias que são pertencentes à instância de um classificador ou ao papel de um classificador.

Portas (Port)* - uma característica de um classificador que especifica um ponto de Interação distinto entre o classificador e seu ambiente ou entre o ambiente do classificador e suas partes internas.

Os elementos marcados com asterisco (*) são novos na versão 2.0.

A UML possui ainda mecanismos de extensão que são os **estereótipos, valores de etiqueta e restrições**.

Estereótipo (Stereotype) - permite que sejam introduzidos novos elementos no modelo, derivados dos elementos básicos da UML, com o objetivo de atender a Situações não descritas explicitamente na linguagem.

Valores de Etiqueta (tag definition) - definem novos tipos de propriedades que podem ser atribuídos a elementos de modelo. Um exemplo de uso é a representação de propriedades de gerenciamento de informações, como autor e data de confecção.

Restrições (constraints) - podem ser anexados a qualquer elemento de modelo com o objetivo de refinar sua semântica, através de novas ou modificadas definições.

⁶ *Papel responsável por enviar e/ou receber informações do sistema.*

◆ RELACIONAMENTOS

Realizam a ligação, entre si, dos elementos do modelo. São eles: **dependência, associação, generalização e realização**.

Dependência (dependency) - é um relacionamento entre dois elementos de modelagem, que indica que a mudança em um elemento afetará o outro.

Associação (association) - é um relacionamento entre dois ou mais classificadores que envolve conexões entre suas instâncias.

Generalização (generalization) - é um relacionamento entre um elemento mais, genérico e outro mais específico.

Realização (realization) - é um relacionamento entre uma especificação e sua implementação.

Muitos elementos de modelo e relacionamentos são usados em mais de um diagrama.

◆ DIAGRAMAS

A UML define em sua versão 2.0 treze tipos de diagramas, divididos em duas categorias: diagramas estruturais ou estáticos (Structural Diagrams) e diagramas dinâmicos (Behavioral Diagrams), conforme divisão mostrada na Tabela 2.1.

A função dos diagramas estruturais é a de mostrar as características do seu sistema que não mudam com o tempo. Já os Diagramas Dinâmicos mostram como o sistema responde às requisições ou como o mesmo evolui durante o tempo.

Diagramas Estruturais Diagrama de Classes Diagrama de Objetos Diagrama de Componentes Diagrama de Pacotes" Diagrama de Implantação Diaarama de Estrutura Composta" **Diagramas Dinâmicos** Diagrama de Casos de Uso Diagramas de Interação: Diagrama de Visão Geral" Diagrama de Seqüências Diagrama Temporal" Diagrama de Comunicação" Diagrama de Atividades

Tabela 2.1 - Diagramas existentes na UML

Os diagramas marcados com asterisco (*) são novos na versão 2.0.

Vamos conhecer um pouquinho sobre cada diagrama:

Diagrama de classes (Class Diagram) - apresenta elementos conectados por relacionamentos. Usado para exibir entidades do mundo real, além de elementos de análise e projeto.

Diagrama de objetos (Object Diagram) - apresenta objetos e valores de dados. Corresponde a uma instância do diagrama de classes, mostrando o estado de um sistema em um determinado ponto do tempo.

Diagrama de componentes (Component Diagram) - mostra as dependências entre componentes de software, apresentando suas interfaces.

Diagrama de implantação (Deployment Diagram) - mostra a arquitetura do sistema em tempo de execução, as plataformas de hardware, artefatos de software e ambientes de software (como sistemas operacionais e máquinas virtuais);

Diagrama de pacotes (Package Diagram) - usado para organizar elementos de modelo e mostrar dependências entre eles;

Diagrama de estrutura composta (Composite Structure Diagram) - usado para mostrar a composição de uma estrutura. Útil em estruturas compostas de estruturas complexas ou em projetos baseados em componentes;

Diagrama de casos de uso (Use Case Diagram) - mostra os casos de uso, atores e seus relacionamentos que expressam a funcionalidade de um sistema.

Diagrama de visão geral (Interaction-Overview Diagram) - uma variação do diagrama de atividades que mostra de uma forma geral o fluxo de controle dentro de um sistema ou processo de negócios. Cada nó ou atividade dentro do diagrama pode representar outro diagrama de interação;

Diagrama de seqüências (Sequence Diagram) - mostra as interações que correspondem a um conjunto de mensagens trocadas entre objetos e a ordem que essas mensagens acontecem.

Diagrama temporal (Timing Diagram) - mostra a mudança de estado de um objeto numa passagem de tempo, em resposta a eventos externos;

Diagrama de comunicação (Communication Diagram) - é o antigo diagrama de colaboração, que mostra objetos, seus interrelacionamentos e o fluxo de mensagens entre eles;

Diagrama de atividades (Activity Diagram) - representa a execução de ações ou atividades e os fluxos que são disparados pela conclusão de outras ações ou atividades.

Diagrama de máquina de estados (Statechart Diagram) - representa as ações ocorridas em resposta ao recebimento de eventos.

2.3.2 Regras de Formação

As regras de formação nos fornecem o caminho para construirmos um modelo coerente, bem-formado, que respeite as regras de sintaxe e semântica a ele aplicado, permitindo uma correta interação entre os blocos de construção.

Toda linguagem possui a sua **sintaxe e semântica**. Na UML, através da sintaxe, temos as regras que definem como os elementos da linguagem são dispostos dentro de expressões, e estas são combinadas. A semântica se refere ao significado dos elementos da linguagem. Ela define como as expressões sintáticas são associadas a um significado. E esses conceitos não se distanciam em nada de outras aplicações de nosso cotidiano.

Veja que por exemplo, na Língua Portuguesa, temos a sintaxe como a parte da Gramática que cuida das relações existentes entre as palavras. Nesse conceito, estuda-se, entre outros, as concordâncias nominal e verbal.

Concordância Nominal:

As palavras menos, alerta, pseudo, salvo e exceto são sempre invariáveis.

Exemplo: Maria convidou menos colegas esse ano. Figura 2.6 - Exemplo de regra de sintaxe na Língua Portuguesa

Numa linguagem de programação como o Pascal, todos os comandos possuem uma sintaxe própria. Veja na Figura 2.7, um exemplo da sintaxe do comando **If ... then ... else**.

```
IF condição THEN
declaração-verdadeira
```

[ELSE
declaração-falsa]

condição indica uma condição que pode ser expressa através de uma expressão simples ou composta.

declaração-verdadeira indica um ou mais comandos que devem ser executados se a condição for verdadeira.

declaração ELSE é opcional.

declaração-falsa indica um ou mais comandos que devem ser executados se a condição for falsa.

Figura 2.7 - Exemplo de regra de sintaxe na linguagem Pascal

Na UML, as regras semânticas especificam o que são e como podem ser aplicados seus diversos elementos. As regras sintáticas mostram notações e representações. Veja exemplos de regra de semântica e sintaxe nas Figuras 2.8 e 2.9.

"Uma restrição não pode ser aplicada a si mesma."

"Herança circular não é permitida"

Figura 2.8 - Exemplo de regras semânticas

Notação:

" ... Um objeto é mostrado como um retângulo com dois compartimentos ... "

Representação:

" ... O nome de um objeto pode ser omitido ... "Figura 2.9 - Exemplo de regras sintáticas

As regras de formação abrangem elementos como:

- nomes (o que identifica elementos, relacionamentos e diagramas)
- escopo (o contexto que determina o significado específico para um nome)
- visibilidade (como os elementos podem ser vistos e utilizados)
- integridade (como os itens se relacionam adequada e consistentemente)
- execução (significado de executar ou simular um modelo)

Os modelos criados durante o processo de desenvolvimento passam por uma evolução natural. Essa evolução frequentemente leva a modelos incompletos e inconsistentes, ou seja, é comum que os desenvolvedores criem seus modelos de forma gradativa.

2.3.3 Normas de Estilo

A UML apresenta em sua documentação sugestões para uso de fontes, atribuição de nomes, simbologia, entre outros. Essas sugestões não fazem parte da notação da UML, mas ajudam a construir modelos mais legíveis. Por exemplo: nomes de atributos tipicamente iniciam com letras minúsculas.

As normas de estilo serão apresentadas aos diversos elementos dos diagramas, no decorrer dos capítulos.

TEXTO 2:

Diagrama de Classes

Se o protagonista de um sistema desenvolvido sob a análise orientada a objetos é um objeto, nada mais justo do que termos onde documentar os objetos encontrados nos requisitos do sistema.

Após extrairmos dos requisitos os objetos da aplicação, precisaremos separar e classificar suas características, modelando, por conseguinte, as classes do sistema. Entretanto, a essência de um sistema não está apenas em suas classes, mas principalmente nos seus relacionamentos. Já diz o ditado que "uma andorinha não faz verão", portanto não conseguiremos um sistema com classes isoladas.

Iniciaremos com alguns conceitos gerais que se aplicam a vários elementos do diagrama de classes. Em seguida, veremos como modelar um diagrama com seus diversos elementos e por último falaremos sobre alguns conceitos mais avançados.

Lembrando que não é minha intenção esgotar neste capítulo todos os elementos que compõem um diagrama de classes (apesar de abordarmos vários itens mais avançados), visto que os conceitos aqui apresentados fornecem ao desenvolvedor um poderoso conjunto de ferramentas para a modelagem de diversos tipos de aplicação.

4.1 CONCEITOS GERAIS

Existem alguns conceitos que são de uso geral dentro de diagramas de classes⁷ e fornecem informações adicionais a diversos elementos estruturais da UML, como por exemplo: relacionamentos, atributos, operações, etc. Essas informações constituem-se de detalhes preciosos para o projeto do sistema.

4.1.1 Visibilidade

A visibilidade identifica por quem uma propriedade (atributo ou operação) pode ser utilizada.

O conceito não é novo, visto estarmos acostumados a definir visibilidades de variáveis (pública, local, privada, etc.) e de outros elementos dentro das diversas linguagens de programação. O que muda efetivamente é o conjunto de visibilidades possíveis e a abrangência dessa visão. Se antes lidávamos com escopo dentro de procedimentos, módulos, sistemas, entre outros, agora estamos falando de classes, objetos, pacotes, etc.

Minha abordagem de visibilidade restringir-se-á às propriedades das classes, mas esse conceito se aplica a outros elementos, inclusive à própria classe.

A visibilidade pode ser suprimida, mas esta decisão não indica que a mesma corresponde a um valor default, pelo contrário, apenas indica que a mesma está omitida.

Definimos a visibilidade de uma propriedade (atributo ou operação) por palavras-chaves ou ícones, conforme descrito a seguir⁸:

+ ou public (público)	A propriedade será vista e usada dentro da classe na qual foi declarada, em qualquer elemento externo (incluindo objetos instanciados a partir desta classe) e nas classes descendentes.
# ou protected (protegido)	A propriedade será vista e usada apenas dentro da classe na qual foi declarada e pelas classes descendentes.
- ou private (privado)	A propriedade será vista e usada apenas dentro da classe na qual foi declarada.

⁷

Em alguns casos estão presentes em outros diagramas da UML.

⁸

É comum desenvolvedores trabalharem com todos os conceitos UML traduzidos (ex: public-'público'). Entretanto, lidamos com linguagens cujos comandos não podem ser traduzidos. Assim, considero mais claro, em certas ocasiões, manter os conceitos no idioma original, a fim de compatibilização com as ferramentas de trabalho (como linguagens, gerenciadores de bancos de dados, ferramentas Case, etc.).

~ ou package (pacote)	A propriedade, poderá ser vista e usada por elementos que estejam declarados dentro do mesmo pacote no qual está inserida a classe que a declarou. É como a visibilidade pública, só que não generalizada a qualquer elemento externo, mas apenas aos elementos externos localizados no mesmo pacote.
-----------------------	---

Para não restar dúvidas, vamos ao exemplo a seguir:

Vamos considerar os seguintes elementos:

- (1) classe na qual o atributo ou método foi declarado
- (2) subclasses da classe (1)
- (3) objeto (instância) da classe (1)
- (4) pacote no qual reside a classe (1)
- (5) pacote externo que não contém nenhum dos elementos anteriores

Vamos considerar o **elemento_A** que pode ser tanto um atributo como uma operação declarada na classe (1).

Vejamos onde esse **elemento_A** será visto e usado se associarmos as visibilidades a seguir:

Público (+): O elemento_A pode ser acessado por todas as entidades do sistema. Visibilidade: (1) (2) (3) (4) (5). Obs: A entidade (5) terá visibilidade ao elemento_A se tiver visibilidade ao pacote (4) e à própria classe (1).

Privado (-): O acesso é permitido somente aos métodos da própria classe. Visibilidade: (1).

Protegido (#): O acesso é permitido à própria classe e às classes descendentes (herança). Visibilidade: (1) (2)

Pacote (-): O acesso é permitido por todas as entidades dentro do pacote que hospeda a classe (1). Visibilidade: (1) (2) (3) (4)

É possível termos alguns tipos adicionais de visibilidade definidos por algumas linguagens de programação, assim como pequenas variações no escopo da visibilidade implantadas pelas linguagens.

4.1.2 Multiplicidade

Indica uma faixa de cardinalidade permitida a um elemento, isto é, a quantidade de instâncias possíveis em um relacionamento. Por exemplo:

- Numa classe Pessoa com o atributo cônjuge podemos afirmar (na nossa atual legislação) que sua multiplicidade é de no mínimo zero e no máximo um cônjuge (levando em conta o cônjuge atual).
- Considerando uma classe Disciplina que se relacione com uma classe Aluno, podemos afirmar que para cada instância de Disciplina há um relacionamento com no mínimo zero e no máximo vários alunos; e para cada instância de Aluno há um relacionamento com no mínimo zero (aluno está com a matrícula trancada) e no máximo várias disciplinas.

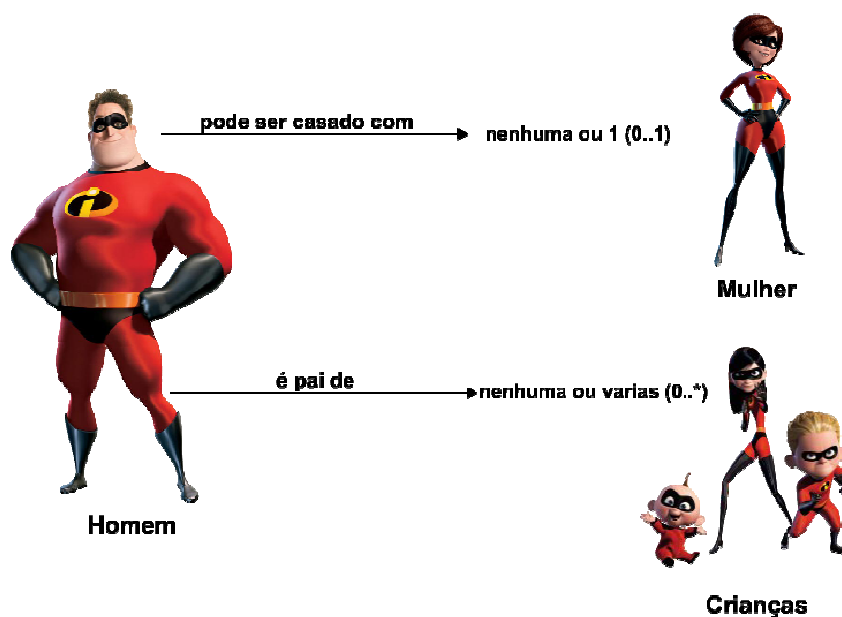


Figura 4.1 - Ilustração de Multiplicidade

A multiplicidade consiste num subconjunto de um conjunto infinito de números inteiros não-negativos. A especificação da multiplicidade é mostrada como uma string compreendida numa sequência de intervalos inteiros separados por vírgula, no qual um intervalo representa uma faixa de inteiros no formato:

limite-inferior .. limite-superior

Se a multiplicidade contiver um asterisco (*), significa que temos uma faixa infinita de números inteiros não-negativos. É equivalente a $0 \dots *$ (zero ou mais). As multiplicidades mais comuns são:

- 0..1 (exatamente um)
- 1 ou 1..1 (exatamente um)
- ou $0 \dots *$ (qualquer valor inteiro não-negativo)
- 1..* (qualquer valor inteiro positivo)

Exemplos de multiplicidades:

"1", "*", "0 .. 1", "1..*", "1..5", "3, 5 .. 7, 10 .. 12"

Algumas normas de estilo devem ser usadas para multiplicidade, como:

- os intervalos devem ser mostrados em ordem crescente. Exemplo:

Adequado: 2 .. 5, 8, 10 .. 15

Inadequado: 10 .. 15, 2 .. 5, 8

- Dois intervalos contíguos devem ser combinados num intervalo simples. Exemplo:

Adequado: 3 .. 8

Inadequado: 3 .. 5, 6 .. 8

4.1.4 Estereótipo

Um estereótipo é um mecanismo de extensibilidade da UML que representa uma subclasse de um elemento já existente com o mesmo formato porém com objetivos diferentes e bem definidos. Geralmente, quando precisamos distinguir um uso específico para um elemento, utilizamos estereótipos.

O estereótipo possui a mesma representação gráfica do elemento, sendo colocado seu nome entre guillemets (<<») acima ou em frente do nome do elemento que está sendo descrito pelo estereótipo. Se múltiplos estereótipos são definidos para o mesmo elemento de modelo, então, eles são colocados verticalmente um abaixo do outro. Por exemplo: «utility»

4.1.5 Notas

É, um símbolo gráfico contendo informação textual. É utilizado para especificar vários tipos de informações sobre os modelos, como: restrições, comentários, corpos de métodos e valores de etiquetas.

Diagrama criado em 10/05/2001

Por Ana Melo

Revisado em 18/05/2001 - Gustavo

Figura 4.2 - Representação de uma nota

4.1.6 Restrições (Constraints)

As restrições podem aparecer em diversos elementos da UML. Uma restrição é uma condição ou restrição propriamente dita expressa textualmente (em linguagem natural ou numa linguagem inteligível pelo computador), com o propósito de declarar alguma semântica de um elemento. O elemento que possuir uma restrição é que determinará o momento em que esta será avaliada.

Uma restrição é mostrada como um texto entre chaves.

Por exemplo: {valor > 1000}

No Diagrama de Classes, podemos incluir restrições ligadas a vários elementos, como associações e atributos. A restrição pode ser apresentada como um texto ao lado de um elemento ou ligado a este por uma linha tracejada. Neste caso, o texto estará num símbolo de nota. Exemplo de restrição ligada a um atributo:

qtdSolicitada: integer {qtdSolicitada > 0}

4.2 CRIANDO DIAGRAMAS DE CLASSE

O diagrama de classes é a estrela principal de um sistema orientado a objetos. Nesse diagrama é possível modelar detalhes das classes e seus relacionamentos. Também são visíveis outros elementos como interfaces e pacotes.

Diagramas de classes podem ser organizados dentro de pacotes, assim como um pacote pode ser representado por um ou mais diagramas de classes.

As classes são declaradas no diagrama de classes mas são usadas em muitos outros diagramas. Uma **classe** é representada como um retângulo subdividido em três compartimentos, separados por linhas horizontais que nessa ordem armazenam:

- O nome da classe e outras propriedades gerais;
- Lista de atributos;

- Lista de operações.

Essa divisão corresponde à notação básica dos diagramas de classes. Entretanto, compartimentos adicionais (não definidos pela UML) podem ser incluídos e usados como extensões das ferramentas de modelagem, com o intuito de exibir outras informações do modelo, como por exemplo: regras de negócio, responsabilidades, exceções, etc. A maior parte desses compartimentos exibem apenas uma lista de strings, entretanto são possíveis outros formatos definidos pelas ferramentas.

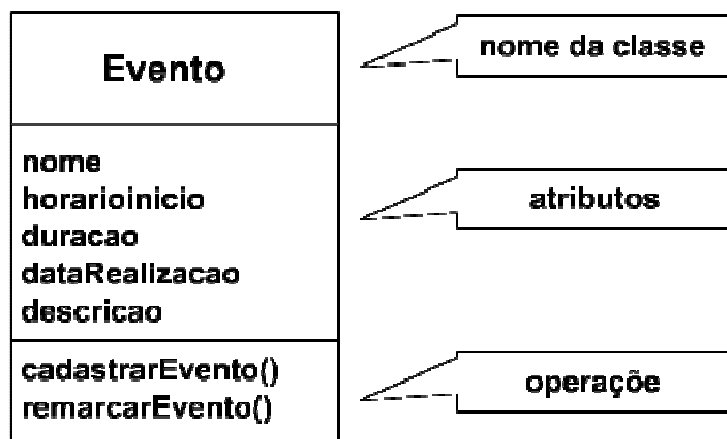


Figura 4.3 - Representação de uma classe

As **normas de estilo** da UML determinam que:

- O nome da classe seja centralizado e negritoado
- Para todas as linguagens que distinguem entre caracteres minúsculos e maiúsculos, escrever as iniciais dos nomes das classes em maiúsculas, inclusive as primeiras letras de nomes compostos

Exemplo: AlunoUniversitario, PessoaFisica, Funcionario

- Os atributos e operações devem ser escritos com formatação normal e alinhados à esquerda
- Os nomes de atributos e operações devem iniciar com letra minúscula, entretanto as iniciais das palavras compostas seguintes devem iniciar com letra maiúscula

Exemplo: reajustarSalario, matricular, dataNascimento, salario

Como extensão de ferramentas, o negrito pode ser usado para marcar listas especiais de elementos.

As normas de estilo da UML são importantes para a padronização da notação da linguagem. Entretanto, às vezes, é comum compatibilizarmos as normas de estilo da UML com as normas da linguagem com a qual estamos trabalhando. Em Delphi, as propriedades e métodos têm o mesmo padrão do nome da classe - as iniciais de cada palavra composta em maiúscula.

A UML permite que em algumas visualizações os detalhes sejam omitidos, isto é, é possível mostrar:

- apenas o nome da classe ou
- o nome da classe e seus atributos ou

- o nome da classe e suas operações

(Repare que o compartimento do nome é obrigatório).

A grande maioria dos processos de modelagem recomendará que, no primeiro diagrama de classes desenhado, não façamos o registro de todos os detalhes das classes. Na realidade, a identificação das classes e seu detalhamento ocorre gradativamente. No processo de desenvolvimento, deve-se levar em conta que a versão do diagrama usada na fase de projeto estará mais completa.

4.2.2 Atributos e Operações

Ao definirmos atributos não informamos apenas seu nome ou tipo de dados. Podemos determinar, também, seu valor inicial, visibilidade e outras características. A única informação obrigatória é o próprio nome do atributo, mas conforme a modelagem é refinada, outras características do atributo tomam-se importantes para o projeto.

A sintaxe default para a definição de atributos é:

visibilidade / nome: tipo [multiplicidade] = valor-default {string-propriedade}

A visibilidade⁹ pode ser representada pelas palavras-chaves public, protected, priVate ou package (ou por seus ícones +, #, -, -). Por exemplo:

private Senha: string
+ tamanho: integer

O tipo do atributo pode corresponder ao nome de uma classe ou ser um tipo dependente da linguagem de programação a ser adotada. Na maior parte das vezes Os atributos corresponderão a tipos básicos (inteiro, real, string, booleano). Mas, também podemos associar tipos definidos pela linguagem de implementação, como por exemplo TDateTime em Delphi.

Outra forma de definir o tipo de um atributo é associá-lo a tipos não-básicos, como uma classe. Por exemplo: um livro pertence a uma editora. Este relacionamento pode aparecer de duas formas: a primeira consiste numa associação entre as classes Livro e Editora; a segunda seria colocar um atributo Editora do tipo Classe Editora, na classe Livro. Esta última forma permite um mapeamento direto para as bases de dados, ou seja, é mais usada na fase de projeto. Por exemplo:

Editora: TEditora¹⁰

Não podemos nos esquecer dos tipos enumerados (ver item 4.3.1 na página 119). Por exemplo:

estadoCivil: TEstadoCivil

A multiplicidade pode ser omitida se corresponder a exatamente 1 (1..1). Em outros casos pode ser representada dentro de colchetes. Por exemplo:

hobby: string **[0 .. 5]**
angulos: integer **[3]**

⁹ Conforme visto no item 4.1.1 na página 94.

¹⁰ Representação de tipos em Delphi.

O primeiro exemplo nos diz que uma determinada classe não tem valor para o atributo hobby ou tem até cinco valores. No segundo exemplo, temos que uma determinada classe tem de um a três ângulos.

O valor-default¹¹ determina o valor inicial do atributo no momento em que o objeto é instanciado. Por exemplo:

coordenadaSuperiorEsquerda: Coordenada = (0,0)
nota1: real = 0

Funcionarionome: strigSalário: realpisoSalarial: real Figura 4.6 - Representação de classe

Temos, ainda, a representação de atributo derivado. Um atributo derivado é aquele cujo valor é computado a partir de outro(s) atributo(s). Por ser um valor sempre gerado não teria necessidade de aparecer em forma de atributo. Todavia sua exibição no diagrama é feita com o intuito de melhorar o entendimento do modelo ou Por propósitos de projeto.

Indica-se graficamente que um atributo é derivado inserindo-se uma barra (/) à frente do nome do atributo.

Veja por exemplo: a classe BoletimEscolar possui, entre outros, os atributos: nota1 e nota2. Sabemos que a média é calculada pela divisão por dois do resultado da soma das notas 1 e 2. Então, para obter a média, basta que tenhamos as notas 1 e 2. Todavia, deixar essa informação implícita pode levar a falhas no entendimento do modelo. Assim, modela-se o atributo média, mas determinando que sua atualização é feita por uma operação de cálculo e não por uma atribuição, como ocorre com os outros atributos. Veja o exemplo na Figura 4.7.

BoletimEscolarnota1:realnota2:real/media:real Figura 4.7 - Exemplo de atributo derivado

Podemos especificar a forma de cálculo de um atributo derivado, fazendo uso de uma nota ligada ao nome do atributo. Veja o exemplo na Figura 4.8.

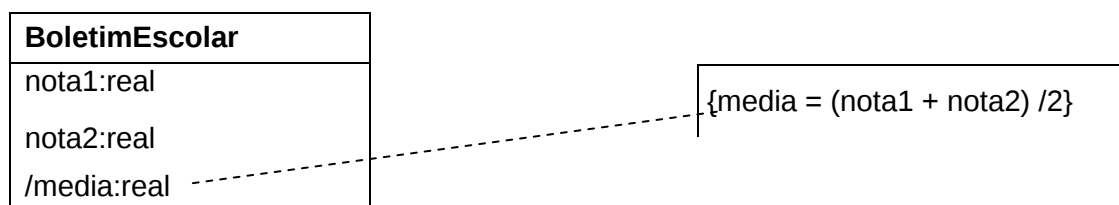


Figura 4.8 - Exemplo da especificação de um atributo derivado

Da mesma forma que ocorre com os atributos, a modelagem das operações não se limita a seu nome e parâmetros. A única informação obrigatória é o próprio nome da operação, mas conforme a modelagem é refinada, outras características da operação tornam-se importantes para o projeto.

A sintaxe default para a definição de operações é:

Visibilidade nome (lista-de-parâmetros): tipo-de-retorno {string-propriedade}

¹¹

Na UML 1.4, o valor associado a um atributo quando um objeto é criado era chamado de valor inicial. Na UML 2.0, esse nome foi alterado para valor default.

A visibilidade¹² pode ser representada pelas palavras-chaves public,protected, private ou package (e ou pelos ícones +,#,-,~). Por exemplo:

```
private obterSenha:string
+modificarTamanho(iTamanho:integer)
```

A lista-de-parâmetro corresponde a uma lista de parâmetros separada por virgula no qual cada parâmetro deve obedecer a seguinte sintaxe:

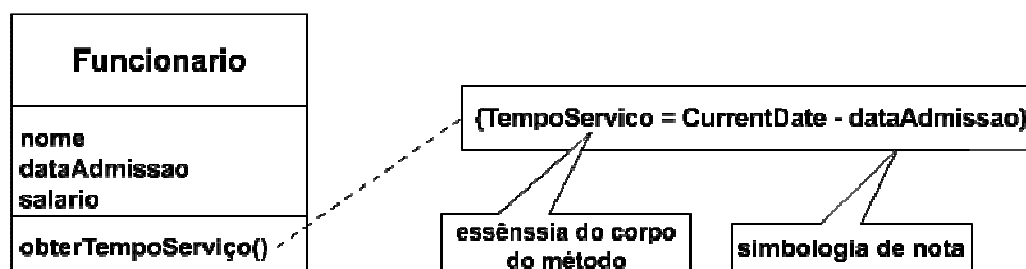


Figura 4.9 - Exemplo da especificação do corpo de um método

Nas listas de atributos e operações, se colocarmos as reticências (...) como o último elemento da lista, estaremos indicando que existem outros elementos no modelo que não foram mostrados.

4.2.3 Relacionamentos

As classes dentro do contexto da modelagem de um sistema, na sua maioria, não trabalham sozinhas. Pelo contrário, elas colaboram umas com as outras por meio de relacionamentos.

No diagrama de classes temos os relacionamentos de **associação**, **generalização** e **dependência**. Como variação do relacionamento de associação, ainda é possível modelar relacionamentos de agregação e composição.

♦ ASSOCIAÇÃO

A associação é um relacionamento que conecta duas (associação binária) ou mais classes (associação ternária ou de ordem-n), demonstrando a colaboração entre as instâncias de classe.

A associação binária conecta exatamente duas classes, sendo possível a associação de uma classe com ela própria. Por exemplo: uma empresa possui diversos funcionários. Essa é uma associação entre as classes Empresa e Funcionário. De outra forma, um funcionário tem um chefe que também é um funcionário, portanto temos uma associação da classe Funcionário com ela própria.

A associação **binária** é representada graficamente por uma linha sólida que liga uma classe a outra ou uma classe a ela própria.

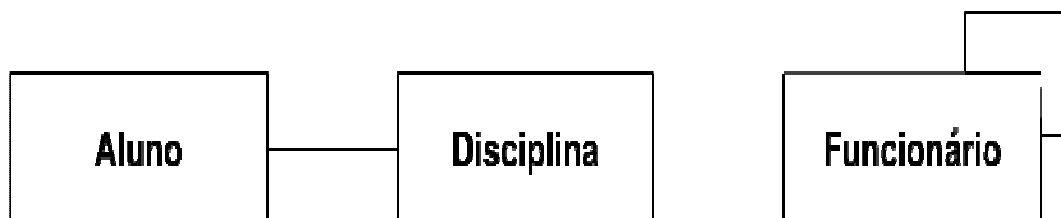


Figura 4.10 - Exemplo da representação de associação

A associação **n-ária** possui mais de duas classes ligadas pelo relacionamento, por meio de um diamante que realiza a conexão entre elas. Veja no exemplo da Figura 4.11, um jogador joga por uma equipe em uma determinada temporada (ano). De outra visão, uma equipe possui jogadores diferentes em cada temporada, e assim por diante.

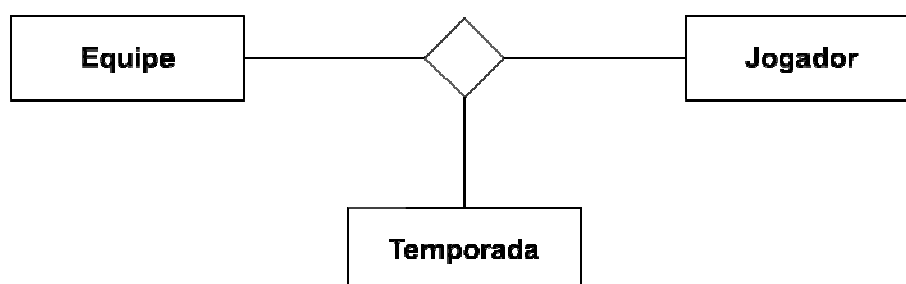


Figura 4.11 - Exemplo da representação de uma associação ternária

Em UML, a associação pode ser de três tipos diferentes: associação simples; agregação por composição; agregação compartilhada. Visto que a construção de agregação pode ter diferentes significados dependendo da área de aplicação, a UML fornece um significado mais preciso para duas dessas construções (associação e agregação por composição) e deixa a agregação compartilhada mais livremente definida no meio das outras. Mais adiante veremos os conceitos de agregação e composição.

Uma associação pode conter adornos que melhoram, em alguns casos, a compreensão do modelo. Todos esses adornos são opcionais e só devem ser usados quando sua função for plenamente atendida, para não poluir visualmente o diagrama de classes. Eles devem ajudar a modelagem, não complicá-la!

Nome da associação: é mostrado próximo à linha do relacionamento, todavia não se deve colocá-lo próximo às extremidades, pois estas posições pertencem aos nomes de papéis.

O nome da associação pode ser acompanhado de um pequeno triângulo preenchido indicando a direção na qual o nome deve ser lido. No exemplo da Figura 4.12 o triângulo expressa a leitura da associação como "aluno cursa disciplina".



Figura 4.12 - Exemplo de nome de associação e direção

Para associações binárias, na sua maioria, não é preciso nomear as associações, pois elas já são auto-explicativas. O nome de associação é mais usado quando possa haver dúvidas sobre o tipo de associação ou nas associações que envolvem uma única classe. Já nas associações n-árias seu uso é mais freqüente. Veja na Figura 4.13 que usamos o nome pré-requisito para esclarecer o relacionamento: "Uma disciplina é pré-requisito de outra disciplina".

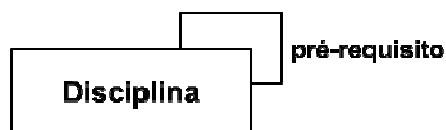


Figura 4.13 - Exemplo de uso do nome de associação

Multiplicidade: Colocada nas extremidades do caminho da associação, identifica o número de instâncias de uma classe que pode-se relacionar com outra. A multiplicidade especificada em uma extremidade determina que essa é a quantidade de instâncias da classe oposta que se relacionará na associação. Apesar de ser opcional, só deve ser omitida nas primeiras visões do diagrama. Uma vez que a modelagem avance, determinadas informações são relevantes para o projeto. Uma delas é a multiplicidade.

Por exemplo: Vamos considerar um estado qualquer da Federação que tenha oito emissoras de TV. Se quiséssemos representar um relacionamento da classe EmissoraTV com uma classe Comercial, poderíamos dizer:

"uma emissora transmite nenhum ou vários comerciais"

"um comercial pode ser transmitido por nenhuma ou até oito emissoras"

Você pode observar na Figura 4.14 como ficaria a representação gráfica dessas multiplicidades.

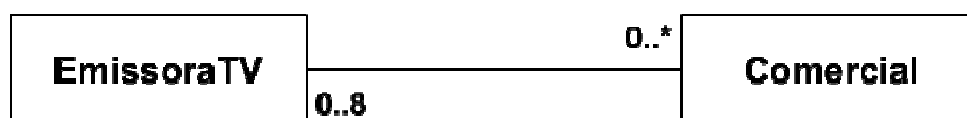


Figura 4.14 - Exemplo de multiplicidade numa associação

Se a multiplicidade da associação for superior a um, então os elementos relacionados podem ser ordenados ou não. Se não incluirmos nenhum adorno na associação, estaremos determinando que os elementos não são ordenados. Para determinar a ordenação deve-se incluir a string {ordered}. Essa declaração não especifica como a ordenação é realizada ou mantida. Essa é uma responsabilidade das operações que agem sobre as instâncias. Geralmente essa especificação é feita no projeto.

Navegabilidade: uma seta pode ser colocada na extremidade de uma associação indicando que a navegação é determinada na direção para onde partiu a seta. As Setas podem ser omitidas, colocadas em apenas uma das extremidades ou em ambas. A navegabilidade quando é omitida indica que a mesma é desconhecida ou bidirecional

Se temos a classe Aluno e Curso sem navegabilidade, podemos dizer que a partir de aluno podemos saber o curso dele, e por outro lado, a partir de um determinado curso podemos saber quais são os alunos matriculados. Se indicássemos uma navegabilidade direcionada ao aluno, estaríamos dizendo que a partir de um aluno não deve ser implementada uma forma de saber seu curso. Claro que essa não é uma situação perfeita, pois é claro que precisamos saber o curso de um aluno. A "via de mão dupla" é mais comum na grande maioria dos relacionamentos de associação.

Temos na Figura 4.16 um exemplo de navegabilidade, na qual um eleitor deve saber em quem votou, mas o candidato não deve saber quem votou nele.

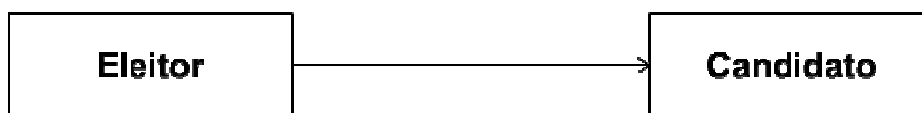
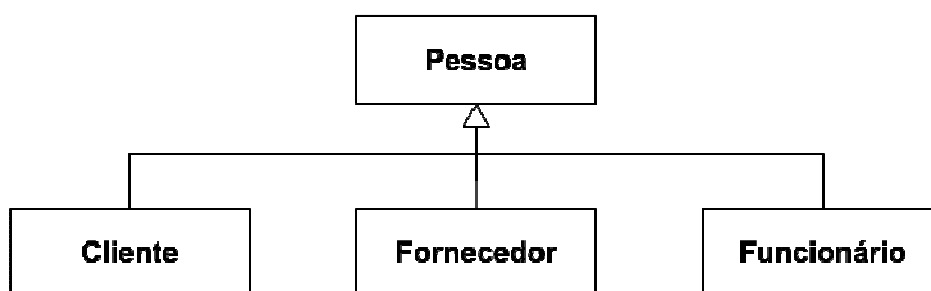


Figura 4.16 - Exemplo de navegabilidade

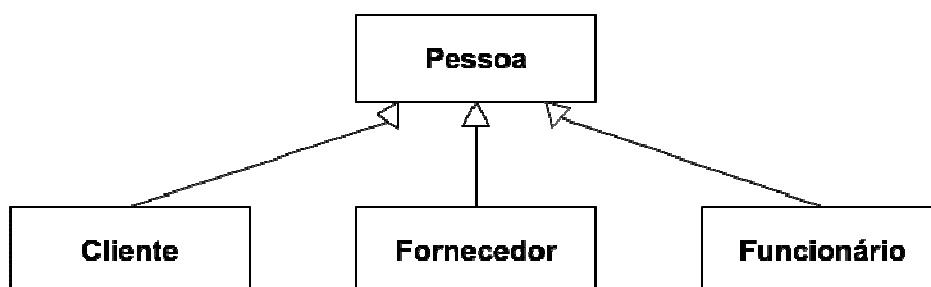
♦ GENERALIZAÇÃO

Vamos recordar: a generalização entre classes envolve elementos mais genéricos e outros mais específicos.

O relacionamento de generalização entre classes é mostrado graficamente como uma seta fechada e vazada, com seu corpo sólido, que parte da classe específica em direção à classe mais genérica.



Primeira forma de representação



Segunda forma de representação

Figura 4.18 - Exemplo de representação de generalização

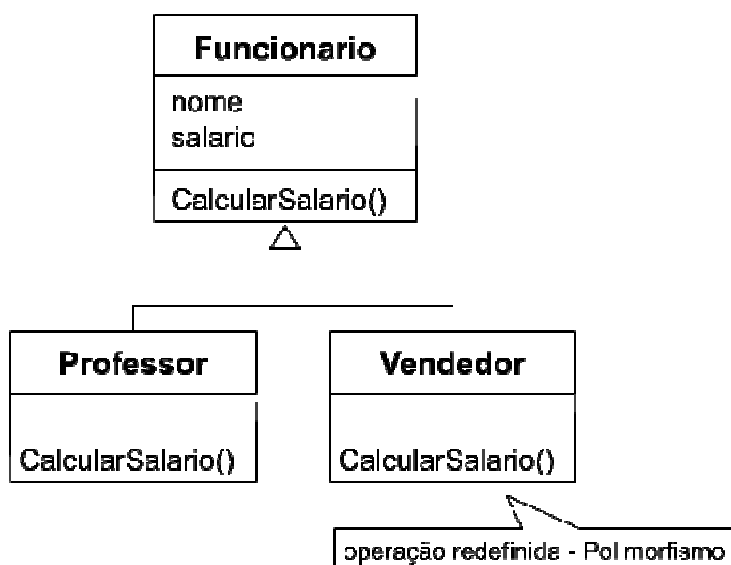


Figura 4.21 - Exemplo da representação de Polimorfismo

◆ DEPENDÊNCIA

O relacionamento de dependência entre duas classes indica que uma mudança na interface de uma delas pode causar mudanças na outra.

Existem vários fatores que levam à dependência entre classes. Vejamos alguns deles:

- troca de mensagens entre classes

Por exemplo: a classe AgendaConsultas necessita da classe HorarioMedico. Mudanças na interface dessa classe pode afetar a classe AgendaConsultas. Assim, dizemos que a classe AgendaConsultas é dependente da classe HorarioMedico.

- uma classe tem como atributo outra classe

Por exemplo: uma classe Imovel possui o atributo Proprietario que é uma instância da classe Proprietario.

- Na assinatura de uma operação, uma classe aparece como parâmetro.

Por exemplo: Na classe Gôndola de um supermercado, vamos supor a existência da operação ReporProduto (produto_Reposicao: Produto). Essa operação recebe por parâmetro uma instância da classe Produto.



Figura 4.22 - Exemplo de dependência

Não representamos sempre todas as dependências, pois essas características citadas anteriormente já expressam o relacionamento. Representamos o relacionamento de dependência apenas em casos que a mesma é implícita.

◆ AGREGAÇÃO

A agregação corresponde a um caso particular da associação (apenas associações binárias), utilizada para expressar um relacionamento "todo-parte". A agregação representa uma propriedade fraca, pois uma classe "parte" pode estar contida em outras agregações.

Deixe-me ilustrar a agregação: quando definimos um relacionamento de associação, estabelecemos uma ligação entre as classes, mantendo a independência de vida das mesmas. No contexto de um Sistema Acadêmico, ao acabar com uma associação entre as classes Curso e Professor, as classes envolvidas continuam a ter vida própria, com pleno significado. Eu pergunto "o que é um curso dentro de um Sistema Acadêmico?" ou pergunto "O que é um professor dentro de um Sistema Acadêmico?". As respostas provavelmente serão objetivas e explicativas. Mais do que isso, curso pode ser definido sem depender da existência do professor e da mesma forma professor pode ser conceituado sem depender da existência do curso.

Agora, vamos pensar em uma página da Web. Uma página é composta (ou não) de imagens. Uma imagem também é parte de um diretório do Site. E se eu perguntasse "O que é uma imagem dentro de um Site?". Você poderia começar respondendo "é uma Figura que ilustra uma página" ou "é uma Figura que serve de link em uma página", etc.

Veja que a Figura tem um significado que para ser completo está dependente de outro elemento. É como um verbo transitivo direto ou indireto. Se eu digo "você acorda", estou me referindo a um verbo intransitivo (que não pede complemento). Quem acorda, acorda e ponto final. Já se digo: "eu chamo", fica faltando alguma coisa. Quem chama, chama alguém. Paralelamente, uma imagem é uma imagem de alguém (ou algo). A página Web representa o todo enquanto que a imagem representa a parte. Portanto, nesse caso, podemos modelar o relacionamento entre Página Web e Imagem como uma agregação. Lembrem-se sempre de que o contexto interfere diretamente no tipo de relacionamento. Uma imagem não fará sempre parte de uma agregação - depende do contexto no qual ela está envolvida.



Figura 4.23 - Representação da agregação envolvendo a classe Imagem

A representação gráfica da agregação consiste em se colocar um diamante aberto junto à classe agregadora.

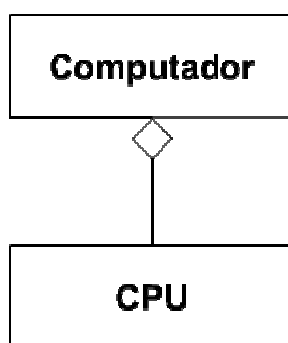


Figura 4.24 - Exemplo de uma agregação

♦ COMPOSIÇÃO

A composição (ou agregação por composição) é uma variação mais poderosa da agregação. Passamos a interpretar o relacionamento como um objeto composto de partes. A diferença consiste no fato de que a classe parte pertence só e somente só à classe todo, em um determinado momento, não podendo fazer parte de outro relacionamento de composição. A classe composta é responsável pela criação e destruição de suas partes. Entende-se, assim, que uma vez que a classe composta deixe de existir, todas as suas partes morrem juntas.

Ao definirmos uma classe como parte de uma composição, indicamos que no relacionamento essa classe perdeu sua identidade, pois é parte incorporada da outra classe.

Seguindo o exemplo do relacionamento de agregação, vamos pensar numa tabela dentro de uma página Web. Não posso usar essa tabela ao mesmo tempo em duas páginas ou dois elementos quaisquer. A tabela só tem sentido de ser, se estiver dentro da página. Como você se refere a uma tabela sem se referir ao seu local físico (a página)? Temos, portanto, uma agregação mais forte: uma composição.

Graficamente, é representada por um diamante preenchido junto à classe composta.

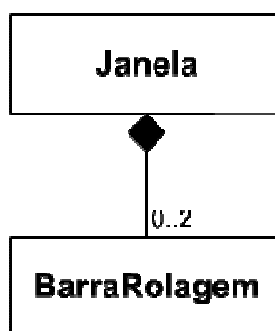


Figura 4.25 - Exemplo de uma composição

TEXTO 3

A história de UML e seus diagramas*

*Adaptado do artigo de Thânia Clair de Souza Vargas

Departamento de Informática e Estatística

Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brazil

thania@inf.ufsc.br

1. Introdução

Modelagem de software é a atividade de construir modelos que expliquem as características ou o comportamento de um software ou de um sistema de software. Na construção do software os modelos podem ser usados na identificação das características e funcionalidades que o software deverá prover (análise de requisitos), e no planejamento de sua construção. Frequentemente a modelagem de software usa algum tipo de notação gráfica e são apoiados pelo uso de ferramentas.

A modelagem de software normalmente implica a construção de modelos gráficos que simbolizam os artefatos dos componentes de software utilizados e os seus inter-relacionamentos. Uma forma comum de modelagem de programas orientados a objeto é através da linguagem unificada UML. A UML (Unified Modeling Language) é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos.

Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos. Por meio de seus diagramas é possível representar sistemas de softwares sob diversas perspectivas de visualização. Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores - por apresentar um vocabulário de fácil entendimento (OMG, 2005a) (OMG, 2005b) (OMG, 2005c) (OMG, 2006).

2. História de UML

No início da utilização do paradigma de orientação a objetos, diversos métodos foram apresentados para a comunidade. Chegaram a mais de cinquenta entre os anos de 1989 a 1994, porém a maioria deles cometeu o erro de tentar estender os métodos estruturados da época. Com isso os maiores prejudicados foram os usuários que não conseguiam encontrar uma maneira satisfatória de modelar seus sistemas. Foi a partir da década de 90 que começaram a surgir teorias que procuravam trabalhar de forma mais ativa com o paradigma da orientação a objetos. Diversos autores famosos contribuíram com publicações de seus respectivos métodos.

Por volta de 1993 existiam três métodos que mais cresciam no mercado, eram eles: Booch'93 de Grady Booch, OMT-2 de James Rumbaugh e OOSE de Ivar Jacobson. Cada um deles possuía pontos fortes em algum aspecto. O OOSE possuía foco em casos de uso (*use cases*), OMT-2 se destaca na fase de análise de sistemas de informação e Booch'93 era mais forte na fase de projeto. O sucesso desses métodos foi, principalmente, devido ao fato de não terem tentado estender os métodos já existentes.

Seus métodos já convergiam de maneira independente, então seria mais produtivo continuar de forma conjunta (SAMPAIO, 2007). Em outubro de 1994, começaram os esforços para unificação dos métodos. Já em outubro de 1995, Booch e Rumbaugh lançaram um rascunho do “Método Unificado” unificando o Booch'93 e o OMT-2. Após isso, Jacobson se juntou a equipe do projeto e o “Método Unificado” passou a incorporar o OOSE. Em junho de 1996, os três amigos, como já eram conhecidos, lançaram a primeira versão com os três métodos - a versão 0.9 que foi batizada como UML (FOWLER, 2003). Posteriormente, foram lançadas várias novas versões.

OMG3 lançou uma RFP (Request for Proposals) para que outras empresas pudessem contribuir com a evolução da UML, chegando à versão 1.1. Após alcançar esta versão, a OMG3 passou a adotá-la como padrão e a se responsabilizar (através da RTF – Revision Task Force) pelas revisões. Essas revisões são, de certa forma, “controladas” a não provocar uma grande mudança no escopo original. Se observarmos as diferenças entre as versões atualmente, veremos que de uma para a outra não houve grande impacto, o que facilitou sua disseminação pelo mundo.

3. Estrutura da Especificação

A especificação de UML é composta por quatro documentos: infra-estrutura de UML(OMG, 2006),

superestrutura de UML (OMG, 2005c), *Object Constraint Language* (OCL) (OMG, 2005a) e Intercâmbio de Diagramas (OMG, 2005b).

□ *Infra-estrutura de UML*: O conjunto de diagramas de UML constitui uma linguagem definida a partir de outra linguagem que define os elementos construtivos fundamentais. Esta linguagem que suporta a definição dos diagramas é apresentada no documento infra-estrutura de UML.

□ *Superestrutura de UML*: Documento que complementa o documento de infra-estrutura e que define os elementos da linguagem no nível do usuário.

□ *Linguagem para Restrições de Objetos (OCL)*: Documento que apresenta a linguagem usada para descrever expressões em modelos UML, com pré-condições, pós-condições e invariantes.

□ *Intercâmbio de diagramas de UML*: Apresenta uma extensão do meta-modelo voltado a informações gráficas. A extensão permite a geração de uma descrição no estilo XMI orientada a aspectos gráficos que, em conjunto com o XMI original permite produzir representações portáteis de especificações UML.

4. Organização dos diagramas de UML 2

A linguagem UML 2 é composta por treze diagramas, classificados em diagramas estruturais e diagramas de comportamento. Os diagramas estruturais tratam o aspecto estrutural tanto do ponto de vista do sistema quanto das classes. Existem para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema, ou seja, a representação de seu esqueleto e estruturas “relativamente estáveis”. Os aspectos estáticos de um sistema de software abrangem a existência e a colocação de itens como classes, interfaces, colaborações, componentes.

Os diagramas de comportamento são voltados a descrever o sistema computacional modelado quando em execução, isto é, como a modelagem dinâmica do sistema. São usados para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema que é a representação das partes que “sofrem alterações”, como por exemplo o fluxo de mensagens ao longo do tempo e a movimentação física de componentes em uma rede.

5. Diagramas de UML 2

Um diagrama é uma representação gráfica de um conjunto de elementos (classes, interfaces, colaborações, componentes, nós, etc) e são usados para visualizar o sistema sob diferentes perspectivas. A UML define um número de diagramas que permite dirigir o foco para aspectos

diferentes do sistema de maneira independente. Se bem usados, os diagramas facilitam a compreensão do sistema que está sendo desenvolvido.

Nas próximas seções, serão sintetizadamente apresentados os diagramas que compõem a linguagem UML 2.

5.1. Diagrama de Classes

Um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. Classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (SILVA, 2007).

5.2. Diagrama de Objetos

O diagrama de objetos consiste em uma variação do diagrama de classes em que, em vez de classes, são representadas instâncias e ligações entre instâncias. A finalidade é descrever um conjunto de objetos e seus relacionamentos em um ponto no tempo.

5.3. Diagrama de Pacotes

O pacote é um elemento sintático voltado a conter elementos sintáticos de uma especificação orientada a objetos. Esse elemento foi definido na primeira versão de UML para ser usado nos diagramas então existentes, como diagrama de classes, por exemplo. Na segunda versão da linguagem, foi introduzido um novo diagrama, o diagrama de pacotes, voltado a conter exclusivamente pacotes e relacionamentos entre pacotes (SILVA, 2007). Sua finalidade é tratar a modelagem estrutural do sistema dividindo o modelo em divisões lógicas e descrevendo as interações entre ele em alto nível.

5.4. Diagrama de Estrutura Composta

O diagrama de estrutura composta fornece meios de definir a estrutura de um elemento e de focalizá-la no detalhe, na construção e em relacionamentos internos. É um dos novos diagramas propostos na segunda versão de UML, voltado a detalhar elementos de modelagem estrutural, como classes, pacotes e componentes, descrevendo sua estrutura interna.

O diagrama de estrutura composta introduz a noção de “porto”, um ponto de conexão do elemento modelado, a quem podem ser associadas interfaces. Também utiliza a noção de “colaboração”, que consiste em um conjunto de elementos interligados através de seus portos para a execução de uma funcionalidade específica –recurso útil para a modelagem de padrões de projeto (SILVA,

2007).

5.5. Diagrama de Componentes

O diagrama de componentes é um dos dois diagramas de UML voltados a modelar software baseado em componentes. Tem por finalidade indicar os componentes do software e seus relacionamentos. Este diagrama mostra os artefatos de que os componentes são feitos, como arquivos de código fonte, bibliotecas de programação ou tabelas de bancos de dados. As interfaces é que possibilitam as associações entre os componentes.

5.6. Diagrama de Implantação

O diagrama de utilização, também denominado diagrama de implantação, consiste na organização do conjunto de elementos de um sistema para a sua execução. O principal elemento deste diagrama é o nodo, que representa um recurso computacional. Podem ser representados em um diagrama tanto os nodos como instâncias de nodos. O diagrama de implantação é útil em projetos onde há muita interdependência entre pedaços de hardware e software.

5.7. Diagrama de Casos de Uso

O diagrama de casos de uso especifica um conjunto de funcionalidades, através do elemento sintático “casos de uso”, e os elementos externos que interagem com o sistema, através do elemento sintático “ator” (SILVA, 2007). Além de casos de uso e atores, este diagrama contém relacionamentos de dependência, generalização e associação e são basicamente usados para fazer a modelagem de visão estática do caso de uso do sistema. Essa visão proporciona suporte principalmente para o comportamento de um sistema, ou seja, os serviços externamente visíveis que o sistema fornece no contexto de seu ambiente. Neste caso os diagramas de caso de uso são usados para fazer a modelagem do contexto de um sistema e fazer a modelagem dos requisitos de um sistema.

5.8. Diagrama de Seqüência

O diagrama de seqüência mostra a troca de mensagens entre diversos objetos, em uma situação específica e delimitada no tempo. Coloca ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas. Em diagramas de seqüência, objetos são representados através de linhas verticais tracejadas (denominadas como linha de existência), com o nome do objeto no topo. O eixo do tempo é também vertical, aumentando para baixo, de modo que as mensagens são enviadas de um objeto para outro na forma de setas com a operação e os nomes dos

parâmetros.

5.8. Diagrama de Máquina de Estados

O diagrama de máquina de estados tem como elementos principais o estado, que modela uma situação em que o elemento modelado pode estar ao longo de sua existência, e a transição, que leva o elemento modelado de um estado para o outro. O diagrama de máquina de estados vê os objetos como máquinas de estados ou autômatos finitos que poderão estar em um estado pertencente a uma lista de estados finita e que poderão mudar o seu estado através de um estímulo pertencente a um conjunto finito de estímulos.

5.8. Diagrama de Comunicação

Os elementos de um sistema trabalham em conjunto para cumprir os objetos do sistema e uma linguagem de modelagem precisa poder representar esta característica. O diagrama de comunicação é voltado a descrever objetos interagindo e seus principais elementos sintáticos são “objeto” e “mensagem”. Corresponde a um formato alternativo para descrever interação entre objetos. Ao contrário do diagrama de seqüência, o tempo não é modelado explicitamente, uma vez que a ordem das mensagens é definida através de enumeração. Vale ressaltar que tanto o diagrama de comunicação como o diagrama de seqüência são diagramas de interação.

5.8. Diagrama de Atividades

O diagrama de atividades representa a execução das ações e as transições que são acionadas pela conclusão de outras ações ou atividades. Uma atividade pode ser descrita como um conjunto de ações e um conjunto de atividades. A diferença básica entre os dois conceitos que descrevem comportamento é que a ação é atômica, admitindo particionamento, o que não se aplica a atividade, que pode ser detalhada em atividades e ações (SILVA, 2007).

5.8. Diagrama de Visão Geral de Integração

O diagrama de visão geral de interação é uma variação do diagrama de atividades, proposto na versão atual de UML. Seus elementos sintáticos são os mesmos do diagrama de atividades. As interações que fazem parte do diagrama de visão geral de interação podem ser referências a diagramas de interação existentes na especificação tratada.

5.8. Diagrama de Temporização

O diagrama de temporização consiste na modelagem de restrições temporais do sistema. É um diagrama introduzido na segunda versão de UML, classificado como diagrama de interação. Este diagrama modela interação e evolução de estados.

6. Conclusão

Embora a UML defina uma linguagem precisa, ela não é uma barreira para futuros aperfeiçoamentos nos conceitos de modelagem. O desenvolvimento da UML foi baseado em técnicas antigas e marcantes da orientação a objetos, mas muitas outras influenciarão a linguagem em suas próximas versões. Muitas técnicas avançadas de modelagem podem ser definidas usando UML como base, podendo ser estendida sem se fazer necessário redefinir a sua estrutura interna. A UML está sendo a base para muitas ferramentas de desenvolvimento, incluindo modelagem visual, simulações e ambientes de desenvolvimento. Em breve, ferramentas de integração e padrões de implementação baseados em UML estarão disponíveis para qualquer um.

A UML integrou muitas idéias adversas, e esta integração acelera o uso do desenvolvimento de softwares orientados a objetos.

References

- FOWLER, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. [S.l.: s.n.], 2003.
- OMG. OCL 2.0 Specification. 2005.
- OMG. Unified Modeling Language: diagram interchange. 2005.
- OMG. Unified Modeling Language: superstructure. 2005.
- OMG. Unified Modeling Language: infrastructure. 2006.
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. Qualidade de Software. São Paulo: a Pretince Hall, 2001.
- SAMPAIO, M. C. História de UML. <http://www.dsc.ufcg.edu.br/sampaio>: [s.n.], 2007.
- SILVA, R. P. e. UML 2 em Modelagem Orientada a Objetos. Florianópolis: Visual Books, 2007.

ANEXO III - TELAS DO SISTEMA WEB DO EXPERIMENTO I E II

- Telas do Sistema WEB do primeiro texto do experimento I

The screenshot shows the Proling web system interface. At the top, there is a green header bar with the Proling logo on the left and a navigation bar with 'Início' and 'Aula Experimental'. Below the header, on the left, is a sidebar with 'Aula' and 'Bem vindo' links. The main content area is titled 'Aula Experimental' and contains the text 'Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística' and 'Faça login no sistema.' Below this is a login form with fields for 'Login' (containing 'kikopalmas') and 'Senha' (masked with dots), and an 'Entrar' button. At the bottom, a green footer bar contains the text 'Miniter em Linguística UFFBI/FTO - Sistema para aulas em hipertexto experimentais - 1.0'.

The screenshot shows the Proling web system interface for the second screen of Experiment I. The layout is similar to the first screen, but the main content area is titled 'Aula Experimental' and contains the text 'Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística' and 'Escolha no Menu Abaixo qual a aula que irá fazer.' Below this is a menu with three buttons: 'Visão Geral da UML', 'Diagramas de Classes', and 'Questionário de Memorização'. At the bottom, a green footer bar contains the text 'Miniter em Linguística UFFBI/FTO - Sistema para aulas em hipertexto experimentais - 1.0'.

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Visão Geral da UML - Aula 1.

Já diz o ditado que "uma imagem vale por mil palavras" (ou algo parecido!). Duvida? Então vamos fazer o teste! O que a Figura 2.1 representa?



Figura 2.1 - Representação de uma casa

É uma casa. Melhor: é uma casa com dois andares, três janelas (sendo uma no andar inferior), uma porta, chaminé e telhado.

Reparou como uma descrição pode ser extensa, enquanto a imagem pode ser muito mais simples. Todavia, a imagem deve ser simples sem perder a expressividade, pois senão o que seria uma ajuda se transforma em pesadelo. Assim, todas as boas características de uma representação gráfica, ou no nosso caso, de uma modelagem gráfica, estão presentes nos diagramas da UML.

Por exemplo, leia a descrição a seguir:

"Classe Aluno possui os atributos: matrícula (string), nome (string), data de nascimento (date), data de matrícula (date), valor da mensalidade (real). Possui, também, as operações: Matricular Aluno, Reajustar Mensalidade (recebendo por parâmetro o percentual de reajuste), Trancar Matrícula, Emitir Boleto de Pagamento (recebendo por parâmetro o mês/ano de referência)."

Agora vamos observar a representação gráfica desta descrição, na Figura 2.2:



Figura 2.2 - Representação da Classe Aluno

É essa facilidade de entendimento e absorção de idéias, não só pelo desenvolvedor mas também e principalmente pelo usuário, que faz com que a UML seja cada vez mais adotada.

Próximo:

[Histórico da UML](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
HISTÓRICO DA UML

Visão Geral da UML - Aula 1.

Desde os primeiros conceitos de orientação a objetos, diversos métodos foram apresentados à comunidade (chegaram a mais de 50 no período compreendido entre 1989 e 1994). Todavia, a grande maioria pecou por tentar estender os métodos estruturados. Os maiores prejudicados eram os usuários que não conseguiam encontrar uma completa satisfação no uso do que havia disponível. Essa situação gerou o que se conheceu como "a guerra dos métodos". Aproximadamente, a partir de meados da década de 90, passamos a conviver com novas abordagens que procuravam trabalhar mais ativamente a idéia do novo paradigma. Nessa época, vários nomes consagrados de nossa área, por meio de suas publicações e métodos, deram grandes contribuições à orientação a objetos, como: Ward Cunningham e Kent Beck (cartões CRC - Class-Responsibility Collaboration1); Sally Shlaer e Steve Mellor, Jim Odell e James Martin; Peter Coad e Ed Yourdon; James Rumbaugh (OMT - Object Modeling Technique); Grady Booch (método Booch); Ivar Jacobson (OOSE - Object-Oriented Software Engineering).

Com o passar do tempo, cada método ganhava uma fatia diferente do mercado. Juntamente com a divisão do mercado, pairava a acirrada competição. Tentativas de padronização foram propostas, mas não obtiveram sucesso. Por volta de 1993, os métodos que mais cresciam no mercado eram: Booch93, OMT-2 e OOSE. Todavia, apesar das semelhanças, existiam pontos significativos e fortes em cada método. Resumidamente, o OOSE possuía seu foco em casos de uso (use cases), provendo excelente suporte à engenharia de negócios e análise de requisitos. OMT-2 era expressivo na fase de análise dos sistemas de informação. Booch93 já se destacava na fase de projeto. Ao invés de seguir a linha dos primeiros autores (que procuravam redefinir ou estender os métodos já existentes), Booch, Rumbaugh e Jacobson decidiram unir forças e criar um método único. Seus métodos já estavam evoluindo um em direção ao outro, de maneira independente. Seria mais sensato essa evolução prosseguir de forma conjunta do que individualmente. Estava terminando a "guerra dos métodos".

Os esforços para essa unificação tiveram início em outubro de 1994. James Rumbaugh deixou a General Electric e se juntou à Grady Booch na Rational Software, no intuito de unir seus métodos (Booch e OMT). Em outubro de 95 eles lançaram publicamente o rascunho de seu Método Unificado (assim chamado na época) na versão 0.8. Nesta época, Jacobson (da Objectory) se juntou à equipe e o projeto inicial passou a incorporar o método OOSE. Em junho de 1996, os três autores (que já passaram a ser conhecidos como os três amigos) lançaram uma nova versão. O Método Unificado passou a se chamar UML - Unified Modeling Language. A nova versão foi lançada como UML 0.9. Em outubro de 1996, mais uma versão: a UML 0.91.

Durante todo o ano de 1996, a UML já era vista pelas organizações como uma ótima estratégia para seus negócios. Um requerimento de proposta de padronização denominado RFP (Request for Proposals) emitido pela OMG, veio gerar uma união de forças em prol de produzir uma resposta a este RFP. Houve uma participação ativa da comunidade de engenharia de software, além de várias empresas de software2 que passaram a contribuir no intuito de fortalecer a UML. Essas colaborações muito contribuíram para a UML. Conseguiu-se uma linguagem bem definida, expressiva, poderosa e genericamente aplicável. Uma força de trabalho foi formada para fazer a padronização da UML na área de metodologias. Alguns autores já apresentavam sinais de abrir mão de seus métodos em favor desse padrão.

Em janeiro de 1997 a Rational lançou a versão 1.0 da UML como proposta para a padronização no OMG (Object Management Group). Entre janeiro e julho de 1997, o grupo original de parceiros se expandiu, passando a incluir outros participantes e colaboradores3, que separadamente haviam submetido respostas ao RFP (OMG). Juntos produziram a versão 1.1 revisada da UML, que tinha por foco principal melhorar a limpidez das semânticas da UML 1.0 e incorporar todas as contribuições dos novos parceiros. Esta versão foi oferecida para padronização ao OMG em julho de 1997 e em setembro do mesmo ano foi aceita. Em 14 de novembro de 1997, a UML 1.1 foi adotada como padrão pelo OMG.

A manutenção da UML passou a ser responsabilidade da RTF (Revision Task Force), pertencente à OMG e dirigida por Chris Kobryn. O objetivo da RTF é aceitar comentários da comunidade em geral a fim de realizar revisões nas especificações, referentes a erros, inconsistências, ambiguidades e pequenas omissões. Essas revisões são guiadas de forma a não provocar uma grande mudança no escopo original da proposta de padronização. Nestes últimos anos novas revisões foram editadas e apresentadas à comunidade: em julho de 1998, a UML 1.2; no final de 1998, a UML 1.3; em maio de 2001, a UML 1.4. Em agosto de 2001, a RTF submeteu ao OMG um relatório provisório da UML 1.5, publicada em março de 2003. Entretanto, a grande mudança está na versão seguinte - a UML 2.0, que hoje é nossa versão atual.

[Anterior](#)

[Próximo](#)

[Visão Geral](#)

[Conhecendo a UML](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
CONHECENDO A UML (UNIFIED MODELING LANGUAGE)

Visão Geral da UML - Aula 1.

Os métodos, na sua maioria, são compostos de uma linguagem de modelagem (notação gráfica) e de um processo (passos para elaboração de um projeto). A UML é uma linguagem de modelagem; não é um método.

A UML, através de sua estrutura, conduz à criação e leitura de seus modelos, mas não determina quais e nem quando esses modelos precisam ser criados. Essa é uma responsabilidade do processo de desenvolvimento. Desta forma, pode-se usar qualquer processo com a UML, pois esta é independente de processo.

UML (Unified Modeling Language - Linguagem de Modelagem Unificada) é uma linguagem para especificação, visualização, construção e documentação de artefatos de sistemas de software.

Booch, Rumbaugh e Jacobson acrescentam:

"A UML proporciona uma forma padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis".

Quando falamos em especificação, dizemos que a UML através de modelos precisos, completos e sem ambiguidades, atende às decisões que necessitam ser tomadas para o desenvolvimento e implantação de sistemas.

No desenvolvimento da maioria dos sistemas precisamos de um padrão para modelagem gráfica, a fim de que um desenvolvedor possa escrever seu modelo e qualquer outro possa interpretá-lo sem ambiguidades. Imagine se um analista modelasse uma classe como um retângulo e outro resolvesse representar essa mesma classe como um cubo. Seria o caos! A UML, através de sua semântica bem definida, atende plenamente a essa necessidade. Essa visualização precisa dá aos desenvolvedores e às ferramentas condições de interpretar os modelos sem ambiguidades.

A UML auxilia na construção de software pois permite um mapeamento de seus modelos para as linguagens de programação, ou vice-versa. Assim, é possível gerar código a partir de modelos da UML, ou com um suporte adequado realizar a engenharia reversa, reconstruindo um modelo a partir de sua implementação.

A UML alcançou dois aspectos muito importantes. Primeiramente, ela terminou com as diversas diferenças existentes entre os métodos de modelagem anteriores. Em segundo lugar, unificou as perspectivas entre muitos sistemas de tipos diferentes (negócios x software), fases de desenvolvimento (análise de requisitos, projeto e implementação) e conceitos internos.

O projeto da UML procurou desenvolver uma linguagem de modelagem que atingisse as seguintes metas:

- * Prover à comunidade uma linguagem de modelagem visual pronta para o uso e expressiva, possibilitando desenvolver e intercambiar modelos significativos.
- * Fornecer extensibilidade e mecanismos de especialização para estender os conceitos centrais.
- * Suportar especificações que são independentes de processos de desenvolvimento e linguagens de programação particulares.
- * Prover uma base formal para entendimento da linguagem de modelagem.
- * Encorajar o crescimento do mercado de ferramentas de objetos.
- * Suportar alto nível de conceitos de desenvolvimento como componentes, colaborações, estruturas e padrões.
- * Integrar melhores práticas.

[Anterior](#)

[Próximo](#)

[Histórico da UML](#)

[Modelando a UML](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
CONHECENDO A UML (UNIFIED MODELING LANGUAGE)

Visão Geral da UML - Aula 1.

Os métodos, na sua maioria, são compostos de uma linguagem de modelagem (notação gráfica) e de um processo (passos para elaboração de um projeto). A UML é uma linguagem de modelagem; não é um método.

A UML, através de sua estrutura, conduz à criação e leitura de seus modelos, mas não determina quais e nem quando esses modelos precisam ser criados. Essa é uma responsabilidade do processo de desenvolvimento. Desta forma, pode-se usar qualquer processo com a UML, pois esta é independente de processo.

UML (Unified Modeling Language - Linguagem de Modelagem Unificada) é uma linguagem para especificação, visualização, construção e documentação de artefatos de sistemas de software.

Booch, Rumbaugh e Jacobson acrescentam:

"A UML proporciona uma forma padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis".

Quando falamos em especificação, dizemos que a UML através de modelos precisos, completos e sem ambiguidades, atende às decisões que necessitam ser tomadas para o desenvolvimento e implantação de sistemas.

No desenvolvimento da maioria dos sistemas precisamos de um padrão para modelagem gráfica, a fim de que um desenvolvedor possa escrever seu modelo e qualquer outro possa interpretá-lo sem ambiguidades. Imagine se um analista modelasse uma classe como um retângulo e outro resolvesse representar essa mesma classe como um cubo. Seria o caos! A UML, através de sua semântica bem definida, atende plenamente a essa necessidade. Essa visualização precisa dá aos desenvolvedores e às ferramentas condições de interpretar os modelos sem ambiguidades.

A UML auxilia na construção de software pois permite um mapeamento de seus modelos para as linguagens de programação, ou vice-versa. Assim, é possível gerar código a partir de modelos da UML, ou com um suporte adequado realizar a engenharia reversa, reconstruindo um modelo a partir de sua implementação.

A UML alcançou dois aspectos muito importantes. Primeiramente, ela terminou com as diversas diferenças existentes entre os métodos de modelagem anteriores. Em segundo lugar, unificou as perspectivas entre muitos sistemas de tipos diferentes (negócios x software), fases de desenvolvimento (análise de requisitos, projeto e implementação) e conceitos internos.

O projeto da UML procurou desenvolver uma linguagem de modelagem que atingisse as seguintes metas:

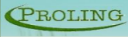
- * Prover à comunidade uma linguagem de modelagem visual pronta para o uso e expressiva, possibilitando desenvolver e intercambiar modelos significativos.
- * Fornecer extensibilidade e mecanismos de especialização para estender os conceitos centrais.
- * Suportar especificações que são independentes de processos de desenvolvimento e linguagens de programação particulares.
- * Prover uma base formal para entendimento da linguagem de modelagem.
- * Encorajar o crescimento do mercado de ferramentas de objetos.
- * Suportar alto nível de conceitos de desenvolvimento como componentes, colaborações, estruturas e padrões.
- * Integrar melhores práticas.

[Anterior](#)

[Próximo](#)

[Histórico da UML](#)

[Modelando a UML](#)



Início

Aula Experimental

Aula

Bem vindo

[Início](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
MODELANDO COM A UML
Visão Geral da UML - Aula 1.

Para modelarmos sistemas com a UML, trabalhamos com elementos básicos do modelo, relacionamentos, diagramas e regras de formação.

Elementos básicos do modelo

Relacionamentos

Diagramas

Regras de formação

Anterior

Próximo

Conhecendo a UML

Finalizar

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Elementos básicos do modelo

Visão Geral da UML - Aula 1.

Podemos citar como os principais elementos básicos para utilização na modelagem de qualquer sistema (em ordem alfabética):

Ações (Action)* - uma unidade básica de especificação de comportamento que representa alguma transformação ou processamento em um sistema. Ações estão contidas em atividades, que provêm seu contexto.

Artefatos (Artifact)* - um pedaço físico de informação que é usado ou produzido por um processo de desenvolvimento. São exemplos de artefatos: modelos, arquivos fontes, scripts e arquivos executáveis.

Atividades (Activity) - é a especificação de um comportamento parametrizado que é expresso como um fluxo de execução a partir de elementos sequenciados, cujos elementos primitivos são ações individuais.

Caso de Uso (Use Case) - representa a funcionalidade provida por um sistema. Consiste na descrição de um conjunto de ações organizadas sequencialmente e que são executadas pelo sistema, interagindo com os atores do mesmo.

Classe (Class) - representa a descrição de um conjunto de objetos que dividem os mesmos atributos, operações (com suas implementações em métodos), relacionamentos e semântica.

Classes Ativas (Active Class) - representam atividades de controle, como threads.

Colaboração (Collaboration) - indica as instâncias e cooperações de elementos que estão envolvidos na realização de alguma tarefa.

Componente (Component) - corresponde a uma parte significativa de um sistema, que é representado modularmente, de forma totalmente substituível.

Estado (State) - é uma condição durante a vida de um objeto ou uma interação durante a qual alguma condição executa alguma ação ou espera por algum evento.

Interação (Interaction) - é um padrão de comunicação com o objetivo de realizar algum propósito.

Interface - especifica as operações externamente visíveis de uma classe ou componente. Interfaces não possuem implementação.

Nó (Node) - é um objeto físico existente em tempo de execução que representa um recurso computacional.

Nota (Note) - é um símbolo gráfico contendo informação textual que possibilita a representação de informações do tipo restrições, comentários e descrição de métodos.

Pacote (Package) - é o agrupamento de elementos de modelo. Pacotes podem ser aninhados a outros pacotes.

Partes (Part)* - um elemento que representa um conjunto de instâncias que são pertencentes à instância de um classificador ou ao papel de um classificador.

Portas (Port)* - uma característica de um classificador que especifica um ponto de interação distinto entre o classificador e seu ambiente ou entre o ambiente do classificador e suas partes internas.

Os elementos marcados com asterisco (*) são novos na versão 2.0.

A UML possui ainda mecanismos de extensão que são os estereótipos, valores de etiqueta e restrições.

Estereótipo (Stereotype) - permite que sejam introduzidos novos elementos no modelo, derivados dos elementos básicos da UML, com o objetivo de atender a Situações não descritas explicitamente na linguagem.

Valores de Etiqueta (tag definition) - definem novos tipos de propriedades que podem ser atribuídos a elementos de modelo. Um exemplo de uso é a representação de propriedades de gerenciamento de informações, como autor e data de confecção.

Restrições (constraints) - podem ser anexados a qualquer elemento de modelo com o objetivo de refinar sua semântica, através de novas ou modificadas definições.

[Anterior](#)

Modelando com a UML

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

RELACIONAMENTOS

Visão Geral da UML - Aula 1.

Realizam a ligação, entre si, dos elementos do modelo. São eles: dependência, associação, generalização e realização.

Dependência (dependency) - é um relacionamento entre dois elementos de modelagem, que indica que a mudança em um elemento afetará o outro.

Associação (association) - é um relacionamento entre dois ou mais classificadores que envolve conexões entre suas instâncias.

Generalização (generalization) - é um relacionamento entre um elemento mais, genérico e outro mais específico.

Realização (realization) - é um relacionamento entre uma especificação e sua implementação.

Muitos elementos de modelo e relacionamentos são usados em mais de um diagrama.

Anterior

Modelando com a UML

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

RELACIONAMENTOS

Visão Geral da UML - Aula 1.

A UML define em sua versão 2.0 treze tipos de diagramas, divididos em duas categorias: diagramas estruturais ou estáticos (Structural Diagrams) e diagramas dinâmicos (Behavioral Diagrams), conforme divisão mostrada na Tabela 2.1.

A função dos diagramas estruturais é a de mostrar as características do seu sistema que não mudam com o tempo. Já os Diagramas Dinâmicos mostram como o sistema responde às requisições ou como o mesmo evolui durante o tempo.

Diagramas Estruturais	Diagrama de Classes
	Diagrama de Objetos
	Diagrama de Componentes
	Diagrama de Pacotes
	Diagrama de Implantação
Diagramas Dinâmicos	Diagrama de Estrutura Composta
	Diagrama de Caso de Uso
	Diagrama de Interação
	Diagrama de Visão Geral
	Diagrama de Sequências
	Diagrama Temporal
	Diagrama de Comunicação
	Diagrama de Atividades
	Diagrama de Máquina de Estados

Tabela 2.1 – Diagramas existentes na UML.

Vamos conhecer um pouquinho sobre cada diagrama:

Diagrama de classes (Class Diagram) - apresenta elementos conectados por relacionamentos. Usado para exibir entidades do mundo real, além de elementos de análise e projeto.

Diagrama de objetos (Object Diagram) - apresenta objetos e valores de dados. Corresponde a uma instância do diagrama de classes, mostrando o estado de um sistema em um determinado ponto do tempo.

Diagrama de componentes (Component Diagram) - mostra as dependências entre componentes de software, apresentando suas interfaces.

Diagrama de implantação (Deployment Diagram) - mostra a arquitetura do sistema em tempo de execução, as plataformas de hardware, artefatos de software e ambientes de software (como sistemas operacionais e máquinas virtuais);

Diagrama de pacotes (Package Diagram) - usado para organizar elementos de modelo e mostrar dependências entre eles;

Diagrama de estrutura composta (Composite Structure Diagram) - usado para mostrar a composição de uma estrutura. Útil em estruturas compostas de estruturas complexas ou em projetos baseados em componentes;

Diagrama de casos de uso (Use Case Diagram) - mostra os casos de uso, atores e seus relacionamentos que expressam a funcionalidade de um sistema.

Diagrama de visão geral (Interaction-Overview Diagram) - uma variação do diagrama de atividades que mostra de uma forma geral o fluxo de controle dentro de um sistema ou processo de negócios. Cada nó ou atividade dentro do diagrama pode representar outro diagrama de interação;

Diagrama de sequências (Sequence Diagram) - mostra as interações que correspondem a um conjunto de mensagens trocadas entre objetos e a ordem que essas mensagens acontecem.

Diagrama temporal (Timing Diagram) - mostra a mudança de estado de um objeto numa passagem de tempo, em resposta a eventos externos;

Diagrama de comunicação (Communication Diagram) - é o antigo diagrama de colaboração, que mostra objetos, seus interrelacionamentos e o fluxo de mensagens entre eles;

Diagrama de atividades (Activity Diagram) - representa a execução de ações ou atividades e os fluxos que são disparados pela conclusão de outras ações ou atividades.

Diagrama de máquina de estados (Statechart Diagram) - representa as ações ocorridas em resposta ao recebimento de eventos.

[Anterior](#)

Modelando com a UML

PROLING

Início

Aula Experimental

Aula

Bem vindo

Inicial

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Regras de Formação

Visão Geral da UML - Aula 1.

As regras de formação nos fornecem o caminho para construirmos um modelo coerente, bem-formado, que respeite as regras de sintaxe e semântica a ele aplica-do, permitindo uma correta interação entre os blocos de construção.

Toda linguagem possui a sua sintaxe e semântica. Na UML, através da sintaxe, temos as regras que definem como os elementos da linguagem são dispostos dentro de expressões, e estas são combinadas. A semântica se refere ao significado dos elementos da linguagem. Ela define como as expressões sintáticas são associadas a um significado. E esses conceitos não se distanciam em nada de outras aplicações de nosso cotidiano.

Veja que por exemplo, na Língua Portuguesa, temos a sintaxe como a parte da Gramática que cuida das relações existentes entre as palavras. Nesse conceito, es-tuda-se, entre outros, as concordâncias nominal e verbal.

Concordância Nominal:
As palavras menos, alerta, pseudo, salvo e exceto são sempre **invariáveis**.
Exemplo: **Maria** convidou **menos** colegas **esse** ano.

Figura 2.6 - Exemplo de regra de sintaxe na Língua Portuguesa

Numa linguagem de programação como o Pascal, todos os comandos possuem uma sintaxe própria. Veja na Figura 2.7, um exemplo da sintaxe do comando If ... then ... else.

IF condição THEN N
declaração-verdadeira
ELSE
declaração-falsa
condição indica uma condição que pode ser expressa através de uma expressão simples ou composta.
declaração-verdadeira indica um ou mais comandos que devem ser executados se a condição for verdadeira.
declaração ELSE é opcional.
declaração-falsa indica um ou mais comandos que devem ser executados se a condição for falsa.

Figura 2.7 - Exemplo de regra de sintaxe na linguagem Pascal

Na UML, as regras semânticas especificam o que são e como podem ser aplicados seus diversos elementos. As regras sintáticas mostram notações e representações. Veja exemplos de regra de semântica e sintaxe nas Figuras 2.8 e 2.9.

Uma restrição não pode ser aplicada a si mesma.
"herança circular não é permitida"

Figura 2.8 - Exemplo de regras semânticas

Notação:
" ... Um objeto é mostrado como um retângulo com dois compartimentos ... "
Representação:
" ... O nome de um objeto pode ser omitido ... "

Figura 2.9 - Exemplo de regras sintáticas

As regras de formação abrangem elementos como:

- nomes (o que identifica elementos, relacionamentos e diagramas)
- escopo (o contexto que determina o significado específico para um nome)
- visibilidade (como os elementos podem ser vistos e utilizados)
- integridade (como os itens se relacionam adequada e consistentemente)
- execução (significado de executar ou simular um modelo)

Os modelos criados durante o processo de desenvolvimento passam por uma evolução natural. Essa evolução frequentemente leva a modelos incompletos e inconsistentes, ou seja, é comum que os desenvolvedores criem seus modelos de forma gradativa.

Anterior

Modelando com a UML

Minter em Linguística UFFBIFTO - Sistema para aulas em hipertexto experimentais - 1.0

PROLING

Início

Aula Experimental

Aula

Bem vindo

Inicial

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Fim da Aula 1. Responda agora os questionários.

Minter em Linguística UFFBIFTO - Sistema para aulas em hipertexto experimentais - 1.0

- Telas do Sistema WEB do segundo texto do experimento II.



Início

Aula Experimental

Aula

Bem vindo

Inicial

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Classes - Aula 2.

Diagrama de Classes

Se o protagonista de um sistema desenvolvido sob a análise orientada a objetos é um objeto, nada mais justo do que termos onde documentar os objetos encontrados nos requisitos do sistema.

Após extrairmos dos requisitos os objetos da aplicação, precisaremos separar e classificar suas características, modelando, por conseguinte, as classes do sistema. Entretanto, a essência de um sistema não está apenas em suas classes, mas principalmente nos seus relacionamentos. Já diz o ditado que "uma andorinha não faz verão", portanto não conseguiremos um sistema com classes isoladas.

Iniciaremos com alguns conceitos gerais que se aplicam a vários elementos do diagrama de classes. Em seguida, veremos como modelar um diagrama com seus diversos elementos e por último falaremos sobre alguns conceitos mais avançados.

Lembrando que não é minha intenção esgotar neste capítulo todos os elementos que compõem um diagrama de classes (apesar de abordarmos vários itens mais avançados), visto que os conceitos aqui apresentados fornecem ao desenvolvedor um poderoso conjunto de ferramentas para a modelagem de diversos tipos de aplicação.

Próximo

Conceitos Gerais

Miniter em Linguística UFFB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

Inicial

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Classes - Aula 2.

CONCEITOS GERAIS

Existem alguns conceitos que são de uso geral dentro de diagramas de classes e fornecem informações adicionais a diversos elementos estruturais da UML, como por exemplo: relacionamentos, atributos, operações, etc. Essas informações constituem-se de detalhes preciosos para o projeto do sistema.

Clique nos botões e veja os conceitos abaixo:

Visibilidade

Multiplicidade

Oriento Diagrama de Classe

Anterior Próximo

Início Aula 2 Finalizar

Miniter em Linguística UFFB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2.

Visibilidade

A visibilidade identifica por quem uma propriedade (atributo ou operação) pode ser utilizada.

O conceito não é novo, visto estamos acostumados a definir visibilidades de variáveis (pública, local, privada, etc.) e de outros elementos dentro das diversas linguagens de programação. O que muda efetivamente é o conjunto de visibilidades possíveis e a abrangência dessa visão. Se antes lidávamos com escopo dentro de procedures, módulos, sistemas, entre outros, agora estamos falando de classes, objetos, pacotes, etc.

Minha abordagem de visibilidade restringir-se-á às propriedades das classes, mas esse conceito se aplica a outros elementos, inclusive à própria classe.

A visibilidade pode ser suprimida, mas esta decisão não indica que a mesma corresponde a um valor default, pelo contrário, apenas indica que a mesma está omitida.

Definimos a visibilidade de uma propriedade (atributo ou operação) por palavras-chaves ou ícones, conforme descrito a seguir:

+ ou public (público)	A propriedade será vista e usada dentro da classe na qual foi declarada, em qualquer elemento externo (incluindo objetos instanciados a partir desta classe) e nas classes descendentes.
# ou protected (protegido)	A propriedade será vista e usada apenas dentro da classe na qual foi declarada e pelas classes descendentes.
- ou private (privado)	A propriedade será vista e usada apenas dentro da classe na qual foi declarada.
~ ou package (pacote)	A propriedade, poderá ser vista e usada por elementos que estejam declarados dentro do mesmo pacote no qual está inserida a classe que a declarou. É como a visibilidade pública, só que não generalizada a qualquer elemento externo, mas apenas aos elementos externos localizados no mesmo pacote.

Para não restar dúvidas, vamos ao exemplo a seguir:

Vamos considerar os seguintes elementos:

(1) classe na qual o atributo ou método foi declarado

(2) subclasses da classe (1)

(3) objeto (instância) da classe (1)

(4) pacote no qual reside a classe (1)

(5) pacote externo que não contém nenhum dos elementos anteriores

Vamos considerar o elemento_A que pode ser tanto um atributo como uma operação declarada na classe (1).

Vejamos onde esse elemento_A será visto e usado se associarmos as visibilidades a seguir:

Público (+): O elemento_A pode ser acessado por todas as entidades do sistema. Visibilidade: (1) (2) (3) (4) (5). Obs: A entidade (5) terá visibilidade ao elemento_A se tiver visibilidade ao pacote (4) e à própria classe (1).

Privado (-): O acesso é permitido somente aos métodos da própria classe. Visibilidade: (1).

Protegido (#): O acesso é permitido à própria classe e às classes descendentes (herança). Visibilidade: (1) (2)

Pacote (~): O acesso é permitido por todas as entidades dentro do pacote que hospeda a classe (1). Visibilidade: (1) (2) (3) (4)

É possível termos alguns tipos adicionais de visibilidade definidos por algumas linguagens de programação, assim como pequenas variações no escopo da visibilidade implantadas pelas linguagens.

Anterior

Conceitos Gerais

Aula Experimental

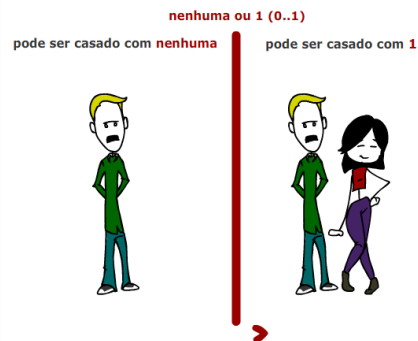
Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2.

Multiplicidade

Indica uma faixa de cardinalidade permitida a um elemento, isto é, a quantidade de instâncias possíveis em um relacionamento. Por exemplo:

* Numa classe Pessoa com o atributo cônjuge podemos afirmar (na nossa atual legislação) que sua multiplicidade é de no mínimo zero e no máximo um cônjuge (levando em conta o cônjuge atual).

* Considerando uma classe Disciplina que se relacione com uma classe Aluno, podemos afirmar que para cada instância de Disciplina há um relacionamento com no mínimo zero e no máximo vários alunos; e para cada instância de Aluno há um relacionamento com no mínimo zero (a-luno está com a matrícula trancada) e no máximo várias disciplinas.



A multiplicidade consiste num subconjunto de um conjunto infinito de números inteiros não-negativos. A especificação da multiplicidade é mostrada como uma string compreendida numa sequência de intervalos inteiros separados por vírgula, no qual um intervalo representa uma faixa de inteiros no formato:

limite-inferior .. limite-superior

Se a multiplicidade contiver um asterisco (*), significa que temos uma faixa infinita de números inteiros não-negativos. É equivalente a 0 .. * (zero ou mais). As multiplicidades mais comuns são:

- 0..1 (exatamente um)

- 1 ou 1..1 (exatamente um)

- 0 ou 0..* (qualquer valor inteiro não-negativo)

- 1..* (qualquer valor inteiro positivo)

Exemplos de multiplicidades:

"1", "", "0..1", "1..*", "1..5", "3..5", "7..10", "12"

Algumas normas de estilo devem ser usadas para multiplicidade, como:

Exemplo 1: Os intervalos devem ser mostrados em ordem crescente. Exemplo:

Adequado: 2..5, 8, 10..15 Inadequado: 10..15, 2..5, 8

Exemplo 2: Dois intervalos contíguos devem ser combinados num intervalo simples. Exemplo:

Adequado: 3..8 Inadequado: 3..5, 6..8

Anterior

[Conceitos Gerais](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2.

CRIANDO DIAGRAMAS DE CLASSE

O diagrama de classes é a estrela principal de um sistema orientado a objetos. Nesse diagrama é possível modelar detalhes das classes e seus relacionamentos. Também são visíveis outros elementos como interfaces e pacotes.

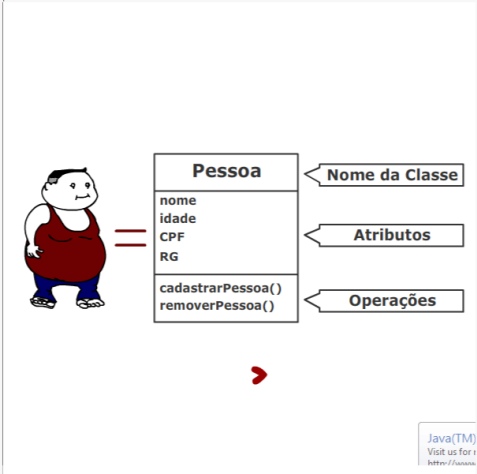
Diagramas de classes podem ser organizados dentro de pacotes, assim como um pacote pode ser representado por um ou mais diagramas de classes.

As classes são declaradas no diagrama de classes mas são usadas em muitos outros diagramas. Uma classe é representada como um retângulo subdividido em três compartimentos, separados por linhas horizontais que nessa ordem armazenam:

O nome da classe e outras propriedades gerais;

- 1 - Lista de atributos;
- 2 - Lista de operações.

Essa divisão corresponde à notação básica dos diagramas de classes. Entretanto, compartimentos adicionais (não definidos pela UML) podem ser incluídos e usados como extensões das ferramentas de modelagem, com o intuito de exibir outras informações do modelo, como por exemplo: regras de negócio, responsabilidades, exceções, etc. A maior parte desses compartimentos exibem apenas uma lista de strings, entretanto são possíveis outros formatos definidos pelas ferramentas.



As normas de estilo da UML, determinam que:

- 1 - O nome da classe seja centralizado e negritoado
- 2 - Para todas as linguagens que distinguem entre caracteres minúsculos e maiúsculos, escrever as iniciais dos nomes das classes em maiúsculas, inclusive as primeiras letras de nomes compostos
Exemplo: AlunoUniversitário, PessoaFísica, Funcionário
- 3 - Os atributos e operações devem ser escritos com formatação normal e alinhados à esquerda
- 4 - Os nomes de atributos e operações devem iniciar com letra minúscula, entretanto as iniciais das palavras compostas seguintes devem iniciar com letra maiúscula
Exemplo: reajustarSalário, matricular, dataNascimento, salário

Como extensão de ferramentas, o negrito pode ser usado para marcar listas especiais de elementos.

Nota do autor

A UML permite que em algumas visualizações os detalhes sejam omitidos, isto é, é possível mostrar:

- 1 - apenas o nome da classe ou
- 2 - o nome da classe e seus atributos ou
- 3 - o nome da classe e suas operações
(Repare que o compartimento do nome é obrigatório).

Nota do autor

Leia mais clicando abaixo.

Atributos e Operações

Leia mais clicando abaixo.

Relacionamentos

Anterior

Conceitos Gerais

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2

Atributos e Operações

Ao definirmos atributos não informamos apenas seu nome ou tipo de dados. Podemos determinar, também, seu valor inicial, visibilidade e outras características. A única informação obrigatória é o próprio nome do atributo, mas conforme a modelagem é refinada, outras características do atributo tomam-se importantes para o projeto.

A sintaxe default (padrão) para a definição de atributos é:

visibilidade / nome: tipo [multiplicidade] = valor-default {string-propriedade}

A visibilidade pode ser representada pelas palavras-chaves public, protected, private ou package (ou por seus ícones +, #, -). Por exemplo:

```
private Senha: string
+ tamanho: integer
```

O tipo do atributo pode corresponder ao nome de uma classe ou ser um tipo dependente da linguagem de programação a ser adotada. Na maior parte das vezes Os atributos corresponderão a tipos básicos (inteiro, real, string, booleano). Mas, também podemos associar tipos definidos pela linguagem de implementação, como no exemplo `TDate` em

Delphi.

Outra forma de definir o tipo de um atributo é associá-lo a tipos não-básicos, como uma classe. Por exemplo: um livro pertence a uma editora. Este relacionamento pode aparecer de duas formas: a primeira consiste numa associação entre as classes Livro e Editora; a segunda seria colocar um atributo Editora do tipo Classe Editora, na classe Livro. Esta última forma permite um mapeamento direto para as bases de dados, ou seja, é mais usada na fase de projeto. Por exemplo:

Editora: TEditora

Não podemos nos esquecer dos tipos enumerados. Por exemplo:

estadoCivil: TEstadoCivil

A multiplicidade pode ser omitida se corresponder a exatamente 1 (1.1). Em outros casos pode ser representada dentro de colchetes. Por exemplo:

```
hobby: string [0 .. 5]
angulos: integer [3]
```

O primeiro exemplo nos diz que uma determinada classe não tem valor para o atributo hobby ou tem até cinco valores. No segundo exemplo, temos que uma determinada classe tem de um a três ângulos.

O valor-default determina o valor inicial do atributo no momento em que o objeto é instanciado. Por exemplo:

```

nota1: real = 0

```

Funcionario nome: string Salário: real pisoSalarial: real

Figura 4.6 - Representação de classe

Temos, ainda, a representação de atributo derivado. Um atributo derivado é aquele cujo valor é computado a partir de outros atributos. Por ser um valor sempre gerado não tem necessidade de aparecer em forma de atributo. Todavia, sua existência é indicada no modelo de dados por uma seta (Figura 4.7). Por exemplo, no projeto de banco de dados, indica-se graficamente que um atributo é derivado inserindo-se uma barra (|) à frente do nome do atributo. Veja por exemplo: a classe *BoleimEmissao* possui, entre outros, os atributos: *nota1* e *nota2*. Sabemos que a média é calculada pela divisão por dois do resultado da soma das notas: *nota1 + nota2*. Então, para obter a média, basta que tenhamos as notas 1 e 2. Todavia, dessa informação implícita pode levar a falhas no entendimento do *10* modelo. Assim, modela-se o atributo *media*, não determinando que sua atribuição é feita por uma operação de cálculo e não por uma atribuição, como ocorre com os outros atributos. Veja o exemplo na Figura 4.7.

BoletimEscolar:nota1:real;nota2:real;media:rea

Figura 4.7 - Exemplo de atributo derivado

Podemos especificar a forma de cálculo de um atributo derivado, fazendo uso de uma nota ligada ao nome do atributo. Veja o exemplo na Figura 4.8.

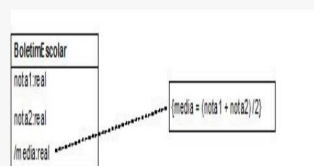


Figura 4.8 - Exemplo da especificação de um atributo derivado

Da mesma forma que ocorre com os atributos, a modelagem das operações não se limita a seu nome e parâmetros. A única informação obrigatória é o próprio nome da operação, mas conforme a modelagem é refinada, outras características da operação tornam-se importantes para o projeto.

A sintaxe default para a definição de operações é

Visibilidade nome (lista-de-parâmetros): tipo-de-retorno {string-propriedade}

A visibilidade pode ser representada pelas palavras-chaves `public`, `protected`, `private` ou `package` (e ou pelos ícones `+`, `#`, `~`). Por exemplo:

```
private obterSenha:string
+modificarTamanho(iTamanho:integer)
```

A lista-de-parâmetro corresponde a uma lista de parâmetros separada por vírgula no qual cada parâmetro deve obedecer a seguinte sintaxe:

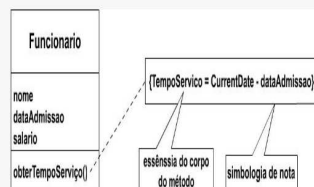


Figura 4.9 - Exemplo da especificação do corpo de um método

Nas listas de atributos e operações, se colocarmos as reticências (...) como o último elemento da lista, estaremos indicando que existem outros elementos no modelo que não foram mostrados.

Anterior

Criando Diagramas de Classe

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2.

Relacionamentos

As classes dentro do contexto da modelagem de um sistema, na sua maioria, não trabalham sozinhas. Pelo contrário, elas colaboram umas com as outras por meio de relacionamentos.

No diagrama de classes temos os relacionamentos de associação, generalização e dependência. Como variação do relacionamento de associação, ainda é possível modelar relacionamentos de agregação e composição.

Delphi.

ASSOCIAÇÃO

GENERALIZAÇÃO

DEPENDÊNCIA

AGREGAÇÃO

COMPOSIÇÃO

Anterior

[Criando Diagramas de Classe](#)

PROLING

Início

Aula

Bem vindo

Inicial

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Classes - Aula 2.

ASSOCIAÇÃO

A associação é um relacionamento que conecta duas (associação binária) ou mais classes (associação ternária ou de ordem n), demonstrando a coexistência entre as instâncias de classe.

A associação binária conecta exatamente duas classes, sendo possível a associação de uma classe com ela própria. Por exemplo, uma empresa possui diversos funcionários. Essa é uma associação entre as classes Empresa e Funcionário. De outra forma, um funcionário tem um chefe que também é um funcionário, portanto temos uma associação de classe Funcionário com ela própria.

A associação binária é representada graficamente por uma linha sólida que liga uma classe a outra ou uma classe a ela própria.



Figura 4.10 - Exemplo da representação de associação

A associação ternária possui mais de duas classes ligadas pelo relacionamento, por meio de um diamante que realiza a conexão entre elas. Veja no exemplo da Figura 4.11, um jogador joga por uma equipe em uma determinada temporada (ano). De outra visão, uma equipe possui jogadores diferentes em cada temporada, e assim por diante.

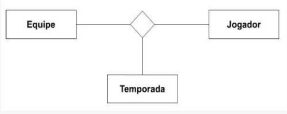


Figura 4.11 - Exemplo da representação de uma associação ternária

Em UML, a associação pode ser de três tipos diferentes: associação simples; agregação por composição; agregação compartilhada. Visto que a construção de agregação pode ter diferentes significados dependendo da área de aplicação, a UML fornece um significado mais preciso para duas dessas construções (associação e agregação por composição) e deixa a agregação compartilhada mais livremente definida no meio das outras. Mais adiante veremos os conceitos de agregação e composição.

Uma associação pode conter adomos que melhoram, em alguns casos, a compreensão do modelo. Todos esses adomos são opcionais e só devem ser usados quando sua função for plenamente atendida, para não poluir visualmente o diagrama de classes. Eles devem ajudar a modelagem, não complicá-la!

Nome da associação: é mostrado próximo à linha do relacionamento, todavia não se deve colocá-lo próximo às extremidades, pois estas posições pertencem aos nomes de papéis.

O nome da associação pode ser acompanhado de um pequeno triângulo preenchido indicando a direção na qual o nome deve ser lido. No exemplo da Figura 4.12, o triângulo expressa a leitura da associação como "aluno cursa disciplina".

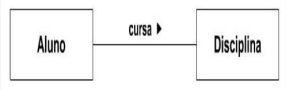


Figura 4.12 - Exemplo de nome de associação e direção

Para associações binárias, na sua maioria, não é preciso nomear as associações, pois elas já são auto-explicativas. O nome de associação é mais usado quando possa haver dúvidas sobre o tipo de associação ou nas associações que envolvem uma única classe. Já nas associações n-árias seu uso é mais frequente. Veja na Figura 4.13 que usamos o nome pré-requisito para escalar o relacionamento. Uma disciplina é pré-requisito de outra disciplina.

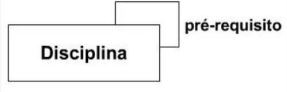


Figura 4.13 - Exemplo de uso do nome da associação

Multiplicidade: Colocado nas extremidades do caminho da associação, identifica o número de instâncias de uma classe que pode se relacionar com outra. A multiplicidade especificada em uma extremidade determina que essa é a quantidade de instâncias da classe ocosta que se relacionará na associação. Apesar de ser opcional, só deve ser omitida nas primeiras vezes do diagrama. Uma vez que a modelagem avanço, determinadas informações são relevantes para o projeto. Uma delas é a multiplicidade.

Por exemplo: Vamos considerar um estado qualquer da Federação que tenha oito emissoras de TV. Se quiséssemos representar um relacionamento da classe EmissoraTV com uma classe Comercial, poderíamos dizer:

"uma emissora transmite nenhum ou vários comerciais"

"um comercial pode ser transmitido por nenhuma ou até oito emissoras"

Você pode observar na Figura 4.14 como ficaria a representação gráfica dessas multiplicidades.

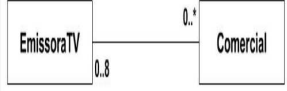


Figura 4.14 - Exemplo de multiplicidade numa associação

Se a multiplicidade da associação for superior a um, então os elemento rela-cionados podem ser ordenados ou não. Se não indicarmos nenhum adomo na associação, estamos determinando que os elementos não são ordenados. Para determinar a ordenação deve-se incluir a string {ordered}. Essa declaração não especí-fica como a ordenação é realizada ou mantida. Essa é uma responsabilidade das operações que agem sobre as instâncias. Geralmente essa especificação é feita no projeto.

Navegabilidade: uma seta pode ser colocada na extremidade de uma associação indicando que a navegação é determinada na direção para onde aponta a seta. As Setas podem ser omitidas, colocadas em apenas uma das extremidades ou em ambas. A navegabilidade quando é omitida indica que a mesma é desconhecida ou bidirecional.

Se temos a classe Aluno e Curso sem navegabilidade, podemos dizer que a partir de aluno podemos saber o curso dele, e por outro lado, a partir de um determinado curso podemos saber quais são os alunos matriculados. Se indicássemos uma nave-gabilidade direcionada ao aluno, estaríamos dizendo que a partir de um aluno não deve ser implementada uma forma de saber seu curso. Claro que essa não é uma situação perfeita, pois é claro que precisamos saber o curso de um aluno. A via de mão dupla é mais comum na grande maioria dos relacionamentos de associação.

Temos na Figura 4.15 um exemplo de navegabilidade, na qual um eleitor deve saber em quem votou, mas o candidato não deve saber quem votou nele.

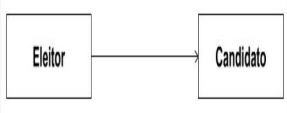


Figura 4.15 - Exemplo de navegabilidade

Anterior

Relacionamentos

Mestr em Linguística UFPE/ITO - Sistema para aulas em hipertexto experimentais - 1.0

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2.

GENERALIZAÇÃO

Vamos recordar: a generalização entre classes envolve elementos mais genéricos e outros mais específicos.

O relacionamento de generalização entre classes é mostrado graficamente como uma seta fechada e vazada, com seu corpo sólido, que parte da classe específica em direção à classe mais genérica.

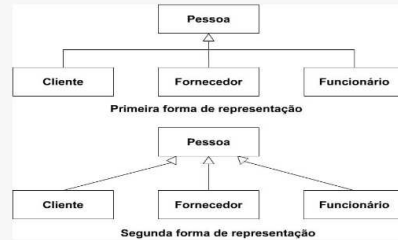


Figura 4.18 - Exemplo de representação de generalização

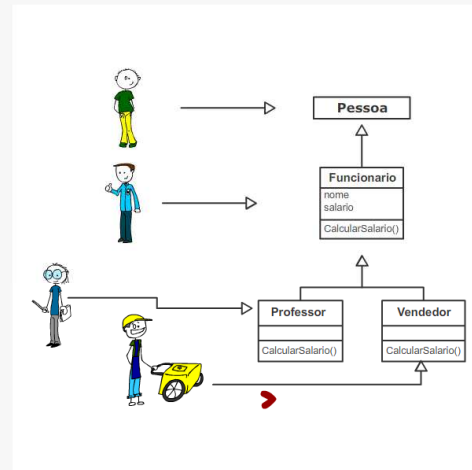


Figura 4.21 - Exemplo da representação de Polimorfismo na função CalcularSalario().

Anterior

Relacionamentos

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes - Aula 2.

AGREGAÇÃO

A agregação corresponde a um caso particular da associação (apenas associações binárias), utilizada para expressar um relacionamento "todo-parte". A agregação representa uma propriedade fraca, pois uma classe "parte" pode estar contida em outras agregações. Deixei-me ilustrar a agregação: quando definimos um relacionamento de associação, estabelecemos uma ligação entre as classes, mantendo a independência de vida das mesmas. No contexto de um Sistema Acadêmico, ao acabar com uma associação entre as classes Curso e Professor, as classes envolvidas continuam a ter vida própria, com pleno significado. Eu pergunto "o que é um curso dentro de um Sistema Acadêmico?" ou pergunto "O que é um professor dentro de um Sistema Acadêmico?". As respostas provavelmente serão objetivas e explicativas. Mais do que isso, curso pode ser definido sem depender da existência do professor e da mesma forma professor pode ser conceituado sem depender da existência do curso.

Agora, vamos pensar em uma página da Web. Uma página é composta (ou não) de imagens. Uma imagem também é parte de um diretório do Site. E se eu perguntasse "O que é uma imagem dentro de um Site?". Você poderia começar respondendo "é uma Figura que ilustra uma página" ou "é uma Figura que serve de link em uma página", etc. Veja que a Figura tem um significado que para ser completo está dependente de outro elemento. É como um verbo transitivo direto ou indireto. Se eu digo "você acorda", estou me referindo a um verbo intransitivo (que não pede complemento). Quem acorda, acorda e ponto final. Já se digo "eu chamo", fica faltando alguma coisa. Quem chama, chama alguém. Paralelamente, uma imagem é uma imagem de alguém (ou algo). A página Web representa o todo enquanto que a imagem representa a parte. Portanto, nesse caso, podemos modelar o relacionamento entre Página Web e Imagem como uma agregação. Lembrem-se sempre de que o contexto interfere diretamente no tipo de relacionamento. Uma imagem não fará sempre parte de uma agregação - depende do contexto no qual ela está envolvida.

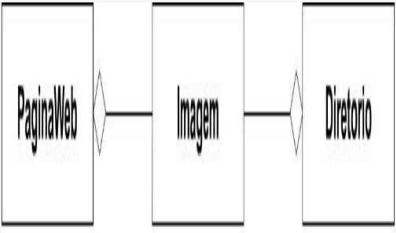


Figura 4.23 - Representação da agregação envolvendo a classe Imagem

A representação gráfica da agregação consiste em se colocar um diamante aberto junto à classe agregadora.

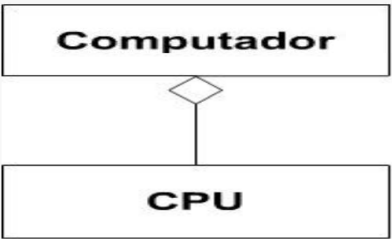


Figura 4.24 - Exemplo de uma agregação

Anterior

[Relacionamentos](#)

PROLING

Início

Aula Experimental

Aula

Bem vindo

Inicial

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Classes - Aula 2.

COMPOSIÇÃO

A composição (ou agregação por composição) é uma variação mais poderosa da agregação. Passamos a interpretar o relacionamento como um objeto composto de partes. A diferença consiste no fato de que a classe parte pertence só e somente só à classe todo, em um determinado momento, não podendo fazer parte de outro relacionamento de composição. A classe composta é responsável pela criação e destruição de suas partes. Entende-se, assim, que uma vez que a classe composta deive de existir, todas as suas partes morrem juntas.

Ao definirmos uma classe como parte de uma composição, indicamos que no relacionamento essa classe perdeu sua identidade, pois é parte incorporada da outra classe.

Seguindo o exemplo do relacionamento de agregação, vamos pensar numa tabela dentro de uma página Web. Não posso usar essa tabela ao mesmo tempo em duas páginas ou dois elementos quaisquer. A tabela só tem sentido de ser, se estiver dentro da página. Como você se refere a uma tabela sem se referir ao seu local físico (a página)? Temos, portanto, uma agregação mais forte: uma composição.

Gráficamente, é representada por um diamante preenchido junto à classe composta.

```

classDiagram
    class Janela
    class BarraRolagem
    Janela "1" *-- "0..2" BarraRolagem
  
```

Figura 4.25 - Exemplo de uma composição

Anterior

Relacionamentos

PROLING

Início

Aula Experimental

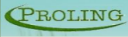
Aula

Bem vindo

Inicial

Obrigado por participar do nosso experimento, agora responda os questionários.

- Telas do Sistema WEB do primeiro dia de experimento



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

Qual é a versão atual utilizada na UML?

Selecione uma opção:

☐ a) 1.1

☐ b) 2.0

☐ c) 3.0

☐ d) 2009

☐ e) 1.2

Confirma Resposta

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

Representa a funcionalidade provida por um sistema. Consiste na descrição de um conjunto de ações organizadas sequencialmente e que são executadas pelo sistema, interagindo com os atores do mesmo, é a definição de:

Selecione uma opção:

☒ a) Diagrama de Classe

☐ b) Diagrama de Sequencia

☐ c) de Toda UML

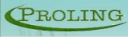
☐ d) Diagrama de Caso de Uso (Use Case)

☐ e) Diagrama de Componentes

Confirma Resposta

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

Relacionamentos: realizam a ligação, entre si, dos elementos do modelo. São eles:

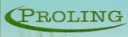
Selecione uma opção:

- ☐ a) dependência, associação, generalização e realização.
- ☐ b) dependência, associação, generalização e maturação.
- ☐ c) dependência, associação, maturação e realização.
- ☐ d) dependência, associação, maturação e independência.
- ☐ e) dependência, associação e desassociação

Confirma Resposta

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

A propriedade será vista e usada dentro da classe na qual foi declarada, em qualquer elemento externo (incluindo objetos instanciados a partir desta classe) e nas classes descendentes. É um tipo de visibilidade:

Selecione uma opção:

- ☐ a) Privada
- ☐ b) Pública
- ☐ c) Protegida
- ☐ d) Explícita
- ☐ e) Pacote

Confirma Resposta

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

A propriedade será vista e usada apenas dentro da classe na qual foi declarada. É um tipo de visibilidade:

Selecione uma opção:

☐

a) Privada

☐

b) Publica

☐

c) Protegida

☐

d) Explícita

☐

e) Pacote

[Confirma Resposta](#)

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Minter em Linguística UFFB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

Sobre multiplicidade:

Sabendo que um homem pode se casar apenas com uma mulher ou ser solteiro, porém o homem pode ter nenhum, um ou mais filhos.

Como representa essa multiplicidade respectivamente:

Selecione uma opção:

☐

a) 1..1 e 1..*

☐

b) 1..1 e 0..0

☐

c) 0..1 e 0..*

☐

d) 0..n e 0..1

☐

e) 0..* e *.*

[Confirma Resposta](#)

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Minter em Linguística UFFB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0



Início

Aula Experimental

Aula

Bem vindo

[Início](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Escolha no Menu Abaixo qual a aula que irá fazer:

Observe a figura

Aluno
matricula: string
nome: string
dataNascimento: date
dataMatricula: date
valorTensalidade: real
MatriculaAluno()
ReajustaTensalidade (percentual: real)
TransfereMatricula()
EmiteBoletoPagto (mesAno: string)

Figura 2.2 - Representação da Classe Aluno

Qual o nome da Classe representada?

Selecione uma opção:

☐ a) Data matricula

☐ b) Matricula

☐ c) nome

☐ d) Data nascimento

☐ e) Aluno

[Confirma Resposta](#)

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Miner em Linguística UFFB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0

PROLING

Início

Aula Experimental

Aula

Bem vindo

[Início](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Escolha no Menu Abaixo qual a aula que irá fazer:

Observe a figura

Aluno
matricula: string
nome: string
dataNascimento: date
dataMatricula: date
valorTensalidade: real
MatriculaAluno()
ReajustaTensalidade (percentual: real)
TransfereMatricula()
EmiteBoletoPago (mesAno: string)

Figura 2.2 - Representação da Classe Aluno

Qual das propriedades abaixo não pertencem a classe aluno?

Selecione uma opção:

☐ a) Data matricula

☐ b) Matricula

☐ c) nome

☐ d) Data nascimento

☐ e) Aluno

[Confirma Resposta](#)

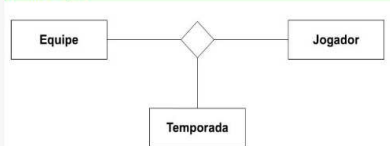
Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Miner em Linguística UFFB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer:

Observe a figura



A figura é uma representação de que tipo de associação?

Selecione uma opção:

- ☐ a) Binária
- ☐ b) Direta
- ☐ c) pré-requisito
- ☐ d) Ternária
- ☐ e) auto associação

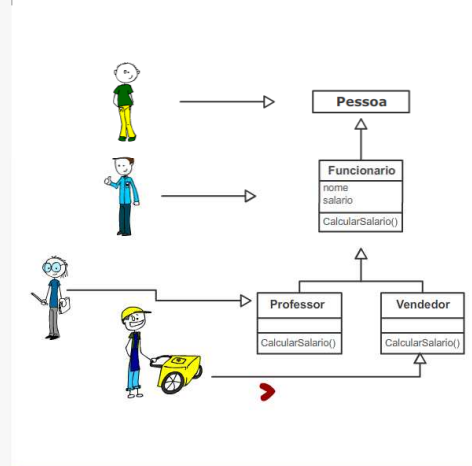
Confirma Resposta

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Escolha no Menu Abaixo qual a aula que irá fazer.

Observe a figura



É qual tipo de relacionamento?

Selecione uma opção:

- ☐ a) Associação
- ☐ b) Dependência
- ☐ c) Agregação
- ☐ d) Generalização
- ☐ e) Composição

Confirme Resposta

Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
MODELANDO COM A UML
Visão Geral da UML - Aula 1.

Digite seu Nome Completo

Sua data de nascimento

Fale sobre sua opinião sobre o experimento

Confirme Resposta

- Tela do sistema web do texto do experimento II.



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

A história de UML e seus diagramas - Experimento do Mestrado em Linguística

Introdução

Modelagem de software é a atividade de construir modelos que expliquem as características ou o comportamento de um software ou de um sistema de software. Na construção do software os modelos podem ser usados na identificação das características e funcionalidades que o software deverá prover (análise de requisitos), e no planejamento de sua construção. Frequentemente a modelagem de software usa algum tipo de notação gráfica e são apoiados pelo uso de ferramentas.

A modelagem de software normalmente implica a construção de modelos gráficos que simbolizam os artefatos dos componentes de software utilizados e os seus interrelacionamentos. Uma forma comum de modelagem de programas orientados a objeto é através da linguagem unificada UML.

A UML (Unified Modeling Language) é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos. Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos. Por meio de seus diagramas é possível representar sistemas de softwares sob diversas perspectivas de visualização. Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores - por apresentar um vocabulário de fácil entendimento (OMG, 2005a) (OMG,2005b) (OMG, 2005c) (OMG, 2006).

Leia Mais sobre UML

Próximo

Estrutura de Especificação

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
HISTÓRICO DA UML

No início da utilização do paradigma de orientação a objetos, diversos métodos foram apresentados para a comunidade. Chegaram a mais de cinquenta entre os anos de 1989 a 1994, porém a maioria deles cometeu o erro de tentar estender os métodos estruturados da época. Com isso os maiores prejudicados foram os usuários que não conseguiam encontrar uma maneira satisfatória de modelar seus sistemas. Foi a partir da década de 90 que começaram a surgir teorias que procuravam trabalhar de forma mais ativa com o paradigma da orientação a objetos. Diversos autores famosos contribuíram com publicações de seus respectivos métodos.

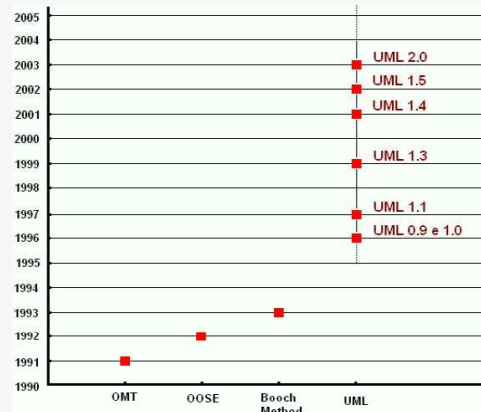


Figura 1. Linha do Tempo de UML

Por volta de 1993 existiam três métodos que mais cresciam no mercado, eram eles: Booch(93) de Grady Booch, OMT-2 de James Rumbaugh e OOSE de Ivar Jacobson. Cada um deles possuía pontos fortes em algum aspecto. O OOSE possuía foco em casos de uso (use cases), OMT-2 se destaca na fase de análise de sistemas de informação e Booch(93) era mais forte na fase de projeto. O sucesso desses métodos foi, principalmente, devido ao fato de não terem tentado estender os métodos já existentes.

Seus métodos já convergiam de maneira independente, então seria mais produtivo continuar de forma conjunta (SAMPAIO, 2007).

Em outubro de 1994, começaram os esforços para unificação dos métodos. Já em outubro de 1995, Booch e Rumbaugh lançaram um rascunho do *Método Unificado* unificando o Booch(93) e o OMT-2. Após isso, Jacobson se juntou a equipe do projeto e o *Método Unificado* passou a incorporar o OOSE. Em junho de 1996, os três amigos, como já eram conhecidos, lançaram a primeira versão com os três métodos a versão 0.9 que foi batizada como UML (FOWLER, 2003). Posteriormente, foram lançadas várias novas versões na qual podemos acompanhar através do gráfico na figura 1.

OMG3 lançou uma RFP (Request for Proposal) para que outras empresas pudessem contribuir com a evolução da UML, chegando à versão 1.1. Após alcançar esta versão, a OMG3 passou a adotá-la como padrão e a se responsabilizar (através da RTF - Revision Task Force) pelas revisões. Essas revisões são, de certa forma, *controladas* a não provocar uma grande mudança no escopo original. Se observarmos as diferenças entre as versões atualmente, veremos que de uma para a outra não houve grande impacto, o que facilitou sua disseminação pelo mundo.

[Leia Mais sobre Histórico da UML](#)

[Anterior](#)

[Próximo](#)

[Introdução](#)

[Estrutura de Especificação](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Estrutura da UML

A especificação de UML é composta por quatro documentos: infra-estrutura de UML (OMG, 2006), superestrutura de UML (OMG, 2005c), Object Constraint Language (OCL) (OMG, 2005a) e Intercâmbio de Diagramas (OMG, 2005b).

Infra-estrutura de UML: O conjunto de diagramas de UML constitui uma linguagem definida a partir de outra linguagem que define os elementos construtivos fundamentais. Esta linguagem que suporta a definição dos diagramas é apresentada no documento infra-estrutura de UML.

Superestrutura de UML: Documento que complementa o documento de infra-estrutura e que define os elementos da linguagem no nível do usuário.

Linguagem para Restrições de Objetos (OCL): Documento que apresenta a linguagem usada para descrever expressões em modelos UML, com pré-condições, pós-condições e invariantes.

Intercâmbio de diagramas de UML: Apresenta uma extensão do meta-modelo voltado a informações gráficas. A extensão permite a geração de uma descrição no estilo XML orientada a aspectos gráficos que, em conjunto com o XML original permite produzir representações portáteis de especificações UML.

Leia Mais sobre a Estrutura da UML

[Anterior](#)

[Próximo](#)

[Histórico](#)

[Organização dos diagramas de UML 2](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagramas DA UML

A linguagem UML 2 é composta por treze diagramas, classificados em diagramas estruturais e diagramas de comportamento. A figura 2 apresenta a estrutura das categorias utilizando a notação de diagramas de classes (OMG, 2006).

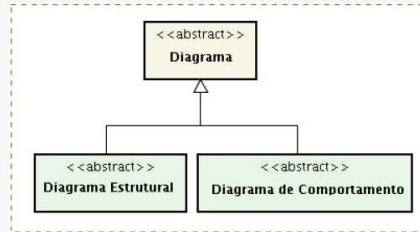


Figura 2. Organização Geral dos Diagramas de UML 2

Os diagramas estruturais, ilustrados na imagem 3 conforme a especificação OMG (OMG, 2006), tratam o aspecto estrutural tanto do ponto de vista do sistema quanto das classes. Existem para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema, ou seja, a representação de seu esqueleto e estruturas "relativamente estáveis". Os aspectos estáticos de um sistema de software abrangem a existência e a colocação de itens como classes, interfaces, colaborações, componentes.

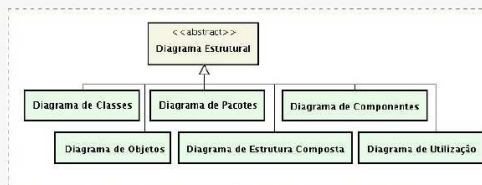


Figura 3. Diagramas Estruturais

Os diagramas de comportamento, ilustrados na imagem 4 conforme a especificação OMG (OMG, 2006), são voltados a descrever o sistema computacional modelado quando em execução, isto é, como a modelagem dinâmica do sistema. São usados para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema que é a representação das partes que "sofrem alterações", como por exemplo o fluxo de mensagens ao longo do tempo e a movimentação física de componentes em uma rede.

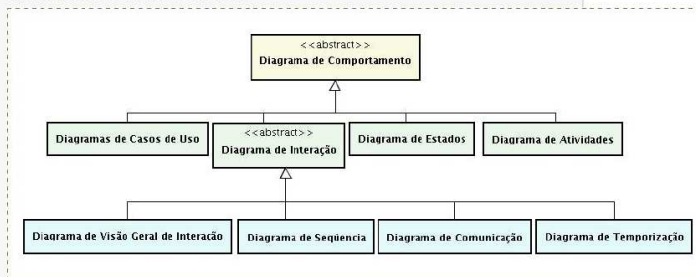


Figura 4. Diagramas Comportamentais

[Leia Mais sobre os Diagramas da UML](#)

[Anterior](#)

[Próximo](#)

[Histórico](#)

[Diagrama de Classe](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes

Diagrama de Classes

Um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. Classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (SILVA, 2007).

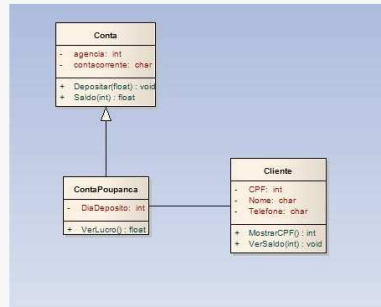


Figura 5: Exemplo de Diagrama de Classes

Diagrama de Objetos

O diagrama de objetos consiste em uma variação do diagrama de classes em que, em vez de classes, são representadas instâncias e ligações entre instâncias. A finalidade é descrever um conjunto de objetos e seus relacionamentos em um ponto no tempo. Contém objetos e vínculos e são usados para fazer a modelagem da visão de projeto estática de um sistema a partir da perspectiva de instâncias reais ou prototípicas.



Figura 6: Exemplo de Diagrama de Objetos

[Anterior](#)

[Próximo](#)

[Diagramas UML 2](#)

[Diagrama de Pacotes](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes

Diagrama de Classes

Um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. Classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (SILVA, 2007).

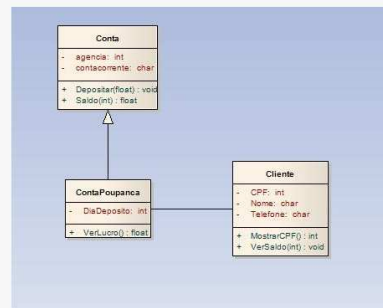


Figura 5. Exemplo de Diagrama de Classes

Diagrama de Objetos

O diagrama de objetos consiste em uma variação do diagrama de classes em que, em vez de classes, são representadas instâncias e ligações entre instâncias. A finalidade é descrever um conjunto de objetos e seus relacionamentos em um ponto no tempo. Contém objetos e vínculos e são usados para fazer a modelagem da visão de projeto estática de um sistema a partir da perspectiva de instâncias reais ou prototípicas.



Figura 6. Exemplo de Diagrama de Objetos

[Anterior](#)

[Próximo](#)

[Diagramas UML: 2](#)

[Diagrama de Pacotes](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Classes

Diagrama de Classes

Um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. Classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (SILVA, 2007).

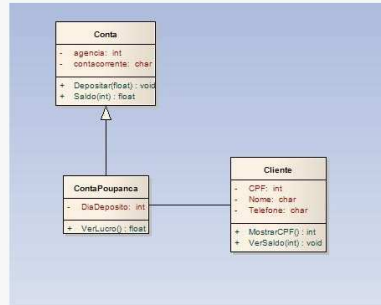


Figure 5: Exemplo de Diagrama de Classes

Diagrama de Objetos

O diagrama de objetos consiste em uma variação do diagrama de classes em que, em vez de classes, são representadas instâncias e ligações entre instâncias. A finalidade é descrever um conjunto de objetos e seus relacionamentos em um ponto no tempo. Contém objetos e vínculos e são usados para fazer a modelagem da visão de projeto estática de um sistema a partir da perspectiva de instâncias reais ou prototípicas.



Figure 6: Exemplo de Diagrama de Objetos

[Anterior](#)

[Próximo](#)

[Diagramas UML 2](#)

[Diagrama de Pacotes](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagramas de Pacotes

Diagrama de Pacotes

O pacote é um elemento sintático voltado a conter elementos sintáticos de uma especificação orientada a objetos. Esse elemento foi definido na primeira versão de UML, para ser usado nos diagramas então existentes, como diagrama de classes, por exemplo. Na segunda versão da linguagem, foi introduzido um novo diagrama, o diagrama de pacotes, voltado a conter exclusivamente pacotes e relacionamentos entre pacotes (SILVA, 2007). Sua finalidade é tratar a modelagem estrutural do sistema dividindo o modelo em divisões lógicas e descrevendo as interações entre ele em alto nível.

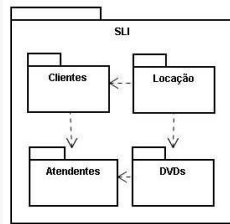


Figure 7. Exemplo de Diagrama de Pacotes

Diagrama de Estrutura Composta

O diagrama de estrutura composta fornece meios de definir a estrutura de um elemento e de focalizá-la no detalhe, na construção e em relacionamentos internos. É um dos novos diagramas propostos na segunda versão de UML, voltado a detalhar elementos de modelagem estrutural, como classes, pacotes e componentes, descrevendo sua estrutura interna. O diagrama de estrutura composta introduz a noção de "porto", um ponto de conexão do elemento modelado, a quem podem ser associadas interfaces. Também utiliza a noção de "colaboração", que consiste em um conjunto de elementos interligados através de seus portos para a execução de uma funcionalidade específica - recurso útil para a modelagem de padrões de projeto (SILVA, 2007).

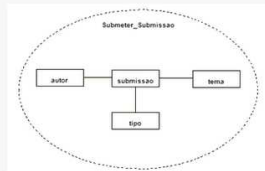


Figure 8 - Diagrama de Estrutura Composta

Anterior: [Diagrama de Classes](#)

Próximo: [Diagrama de Componentes](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Componentes

Visão Geral da UML - Aula 1.

Diagrama de Componentes

O diagrama de componentes é um dos dois diagramas de UML voltados a modelar software baseado em componentes. Tem por finalidade indicar os componentes do software e seus relacionamentos. Este diagrama mostra os artefatos de que os componentes são feitos, como arquivos de código fonte, bibliotecas de programação ou tabelas de bancos de dados. As interfaces é que possibilitam as associações entre os componentes.

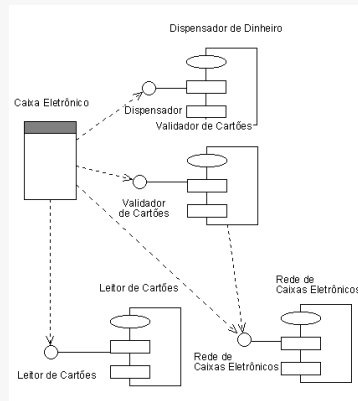


Figura 9 - Diagrama de Componentes

Diagrama de Implantação

O diagrama de utilização, também denominado diagrama de implantação, consiste na organização do conjunto de elementos de um sistema para a sua execução. O principal elemento deste diagrama é o nó, que representa um recurso computacional. Podem ser representados em um diagrama tantos os nós como instâncias de nós. O diagrama de implantação é útil em projetos onde há muita interdependência entre pedaços de hardware e software.

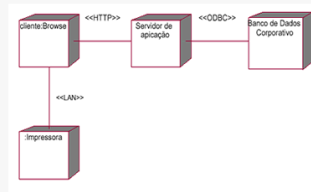


Figura 10 - Diagrama de Implantação

Anterior: [Diagrama de Pacotes](#)

Próximo: [Diagrama de Casos de Uso](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Casos de Uso

Visão Geral da UML - Aula 1.

Diagrama de Casos de Uso

O diagrama de casos de uso especifica um conjunto de funcionalidades, através do elemento sintático "casos de uso", e os elementos externos que interagem com o sistema, através do elemento sintático "ator" (SILVA, 2007). Além de casos de uso e atores, este diagrama contém relacionamentos de dependência, generalização e associação e são basicamente usados para fazer a modelagem de visão estática do caso de uso do sistema. Essa visão proporciona suporte principalmente para o comportamento de um sistema, ou seja, os serviços externamente visíveis que o sistema fornece no contexto de seu ambiente. Neste caso os diagramas de caso de uso são usados para fazer a modelagem do contexto de um sistema e fazer a modelagem dos requisitos de um sistema.

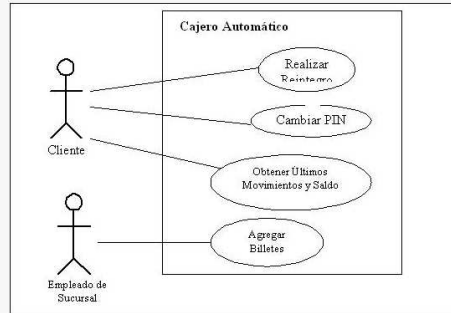


Figura 11 - Diagrama de Caso de Uso

Diagrama de Sequência

O diagrama de sequência mostra a troca de mensagens entre diversos objetos, em uma situação específica e delimitada no tempo. Coloca ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas. Em diagramas de sequência, objetos são representados através de linhas verticais tracejadas (denominadas como linha de existência), com o nome do objeto no topo. O eixo do tempo é também vertical, aumentando para baixo, de modo que as mensagens são enviadas de um objeto para outro na forma de setas com a operação e os nomes dos parâmetros.

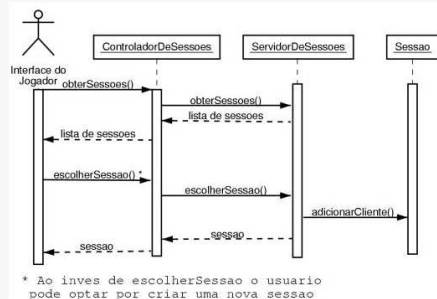


Figura 12 - Diagrama de Sequência

Anterior [Diagrama de Componentes](#)

Próximo [Diagrama de Máquina de Estados](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Máquina de Estados

Diagrama de Máquina de Estados

O diagrama de máquina de estados tem como elementos principais o estado, que modela uma situação em que o elemento modelado pode estar ao longo de sua existência, e a transição, que leva o elemento modelado de um estado para o outro. O diagrama de máquina de estados vê os objetos como máquinas de estados ou autômatos finitos que poderão estar em um estado pertencente a uma lista de estados finita e que poderão mudar o seu estado através de um estímulo pertencente a um conjunto finito de estímulos.

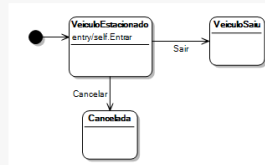


Figura 13 - Diagrama de Máquina de Estados

Diagrama de Comunicação

Os elementos de um sistema trabalham em conjunto para cumprir os objetos do sistema e uma linguagem de modelagem precisa poder representar esta característica. O diagrama de comunicação é voltado a descrever objetos interagindo e seus principais elementos sintáticos são "objeto" e "mensagem". Corresponde a um formato alternativo para descrever interação entre objetos. Ao contrário do diagrama de sequência, o tempo não é modelado explicitamente, uma vez que a ordem das mensagens é definida através de enumeração. Vale ressaltar que tanto o diagrama de comunicação como o diagrama de sequência são diagramas de interação.

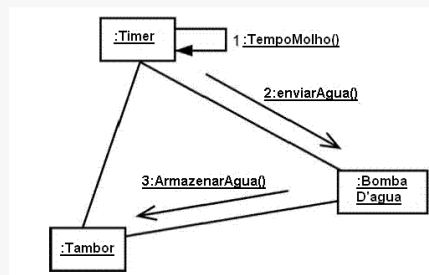


Figura 14 - Diagrama de Comunicação

Anterior: [Diagrama de Gestão de UML](#)

Próximo: [Diagrama de Atividades](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Diagrama de Componentes

Diagrama de Atividades

O diagrama de atividades representa a execução das ações e as transições que são acionadas pela conclusão de outras ações ou atividades. Uma atividade pode ser descrita como um conjunto de ações e um conjunto de atividades. A diferença básica entre os dois conceitos que descrevem comportamento é que a ação é atômica, admitindo particionamento, o que não se aplica a atividade, que pode ser detalhada em atividades e ações (SILVA, 2007).

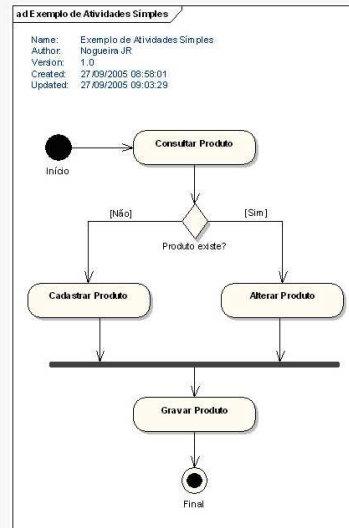


Figura 15 - Diagrama de Atividades

Diagrama de Temporização

O diagrama de temporização consiste na modelagem de restrições temporais do sistema. É um diagrama introduzido na segunda versão de UML, classificado como diagrama de interação. Este diagrama modela interação e evolução de estados.

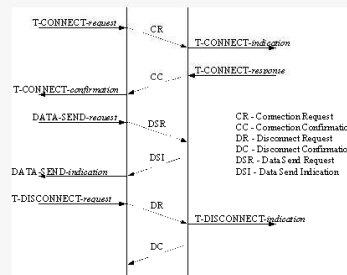


Figura 16 - Diagrama de Temporização

Anterior: [Diagrama de Máquina de Estados](#)

Próximo: [Conclusão](#)



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Diagrama de Componentes

Conclusão

Embora a UML defina uma linguagem precisa, ela não é uma barreira para futuros aperfeiçoamentos nos conceitos de modelagem. O desenvolvimento da UML foi baseado em técnicas antigas e marcantes da orientação a objetos, mas muitas outras influenciaram a linguagem em suas próximas versões. Muitas técnicas avançadas de modelagem podem ser definidas usando UML como base, podendo ser estendida sem se fazer necessário redefinir a sua estrutura interna.

A UML está sendo a base para muitas ferramentas de desenvolvimento, incluindo modelagem visual, simulações e ambientes de desenvolvimento. Em breve, ferramentas de integração e padrões de implementação baseados em UML estarão disponíveis para qualquer um. A UML integrou muitas idéias adversas, e esta integração acelera o uso do desenvolvimento de softwares orientados a objetos.

Anterior: [Diagrama de Componentes](#)

Próximo: [Finalizar](#)



Início

Aula Experimental

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística

Final da leitura

Agora responda o questionário

Minter em Linguística UFPB/FTO - Sistema para aulas em hipertexto experimentais - 1.0

ANEXO IV – EXPERIMENTO I E II

QUESTIONÁRIO DO EXPERIMENTO I

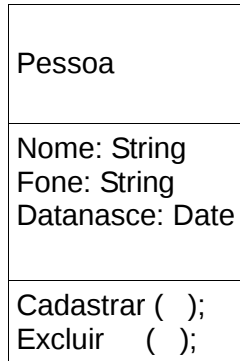
Testando a compreensão

- Através do texto, observa-se que a UML é extremamente adotada em desenvolvimento de sistemas por propor um modelo único que externa de forma gráfica os artefatos do mundo real, conforme o exemplo da classe ALUNO. De posse da leitura do texto fale de forma sucinta o que você julga que favorece essa adoção.
- A UML trouxe uma proposta metodológica de padronização da arquitetura de projetos de sistemas em resposta a tantos problemas vividos nesta área ocasionados pela quantidade de métodos diferentes adotados e desenvolvido por entidades dessa natureza. Pensando assim, e tendo em vista a analogia de um projeto arquitetônico de uma casa (onde qualquer pessoa, sendo ou não profissional da engenharia civil pode entender) fale sobre a importância do padrão no desenvolvimento de sistemas.
- Quais são os diagramas estruturais da UML?
- (Concurso Serpro-2001 - Técnico/Programador de Computador) A respeito da linguagem UML, é correto afirmar que:
 - não se trata de uma linguagem de documentação
 - é voltada para a representação conceitual e física de um sistema (c) não abrange a documentação para a realização de testes
 - não deve ser empregada para a documentação de artefatos que façam uso de sistemas complexos de software
 - é uma linguagem utilizada para a realização de testes de programas.
- (Concurso Serpro-2001 - Técnico/Programador de Computador) Entre outros, a UML inclui os diagramas de
 - classes, de objetos, de fluxo de dados e de atividades
 - classes, de implantação, de gráficos de estados e de seqüências
 - objetos, de classes, de contexto e de implantação
 - classes, de objetos, testes e de implantação
 - objetos, de casos de uso, de contexto e de implantação

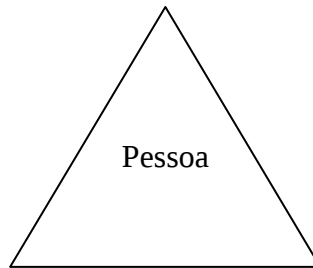
TESTE DE RESOLUÇÃO DE PROBLEMAS EXPERIMENTO I

Testes de compreensão

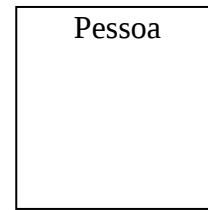
1- Identifique a representação conforme a notação da UML de uma classe. (pode haver mais de uma resposta certa).



()



()



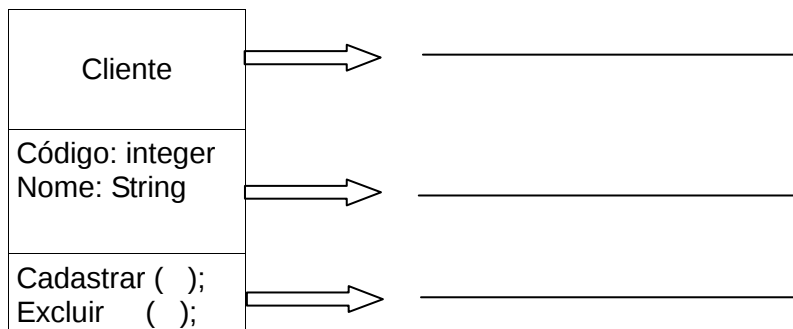
()

2- A elaboração entre duas classes se dá por meio de:

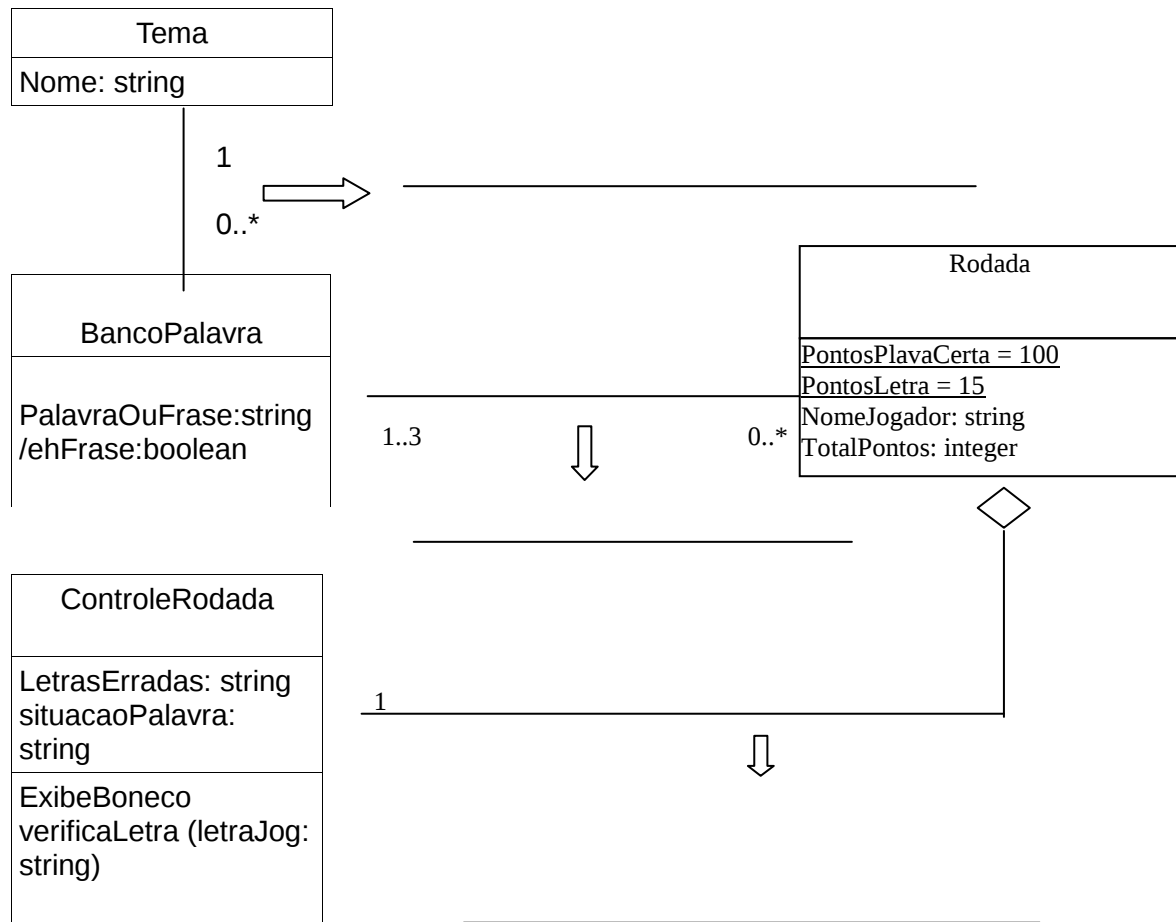
- () Multiplicidade
- () Alternância
- () Relacionamento
- () Visibilidade

3- Desenhe uma classe conforme a notação de UML com seus atributos e operações.

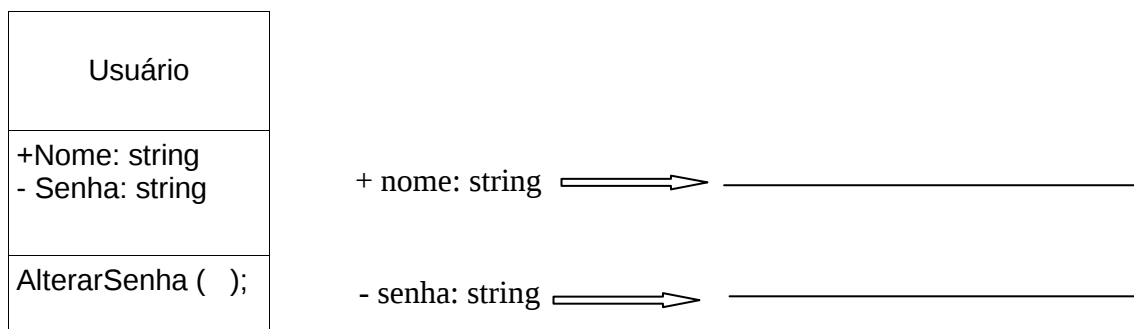
4- No desenho de uma classe abaixo identifique os compartimentos de nome, atributos e operações



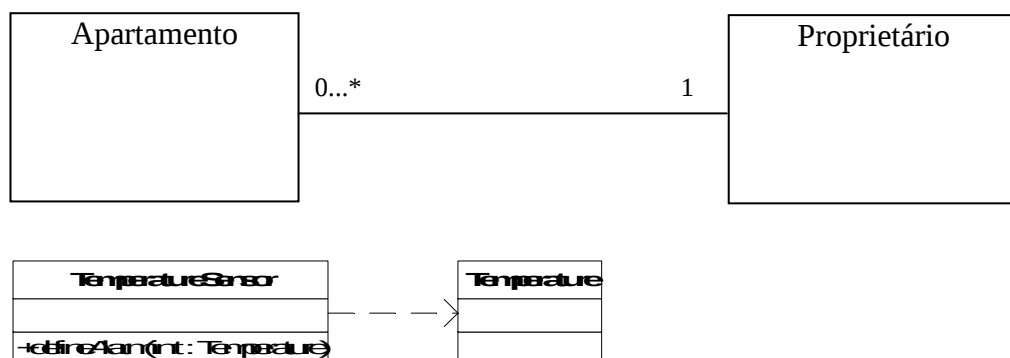
5- Identifique os relacionamentos a seguir:



6- Na classe abaixo, observe que os atributos possuem visibilidade explícita. Com base na leitura do texto descreva a visibilidade que pode ser alcançada pelos atributos:



7- Identifique os relacionamentos abaixo conforme a notação da UML.



8- Leia o cenário seguinte :

¹³**Coleção de CD's** : Adriano tem uma coleção de CD's e gostaria de cadastrar no seu Palm a lista desses CD's, pois às vezes nem sabe o que tem.

Ele pensou em cadastrar o nome do cantor(a) ou conjunto, o título do CD e o ano de lançamento. Adriano notou que alguns CD's são de coletâneas. Sendo assim, não têm apenas um cantor (ou conjunto) e sim vários. Ele quer cadastrar essa lista de músicos, sem relacioná-los às músicas. Deseja controlar também se o CD é de coletânea ou se é duplo. Adriano gostaria de ter cadastrada a lista das músicas de cada CD, com o tempo de duração de cada faixa.

Analise os requisitos citados por Adriano e represente a solução do problema através de classes e relacionamento entre elas, siga as instruções:

- Identifique as classes
- Identifique os atributos
- Identifique ações das classes (relacionamentos)
- Desenhe as classes
- Identifique o relacionamento entre as classes, se possível coloque também a multiplicidade e ainda verifique se houve relacionamento de composição e ou agregação, herança e dependência.

QUESTIONÁRIO DO EXPERIMENTO II

Testando a Compreensão

1. Tendo em vista a leitura do artigo “A história de UML e seus diagramas”, fale o que você entendeu sobre modelagem de software e o papel da UML dentro desse trabalho.
2. Para fabricar uma bicicleta é necessário que se tenha o modelo da estrutura com cada elemento que a compõe, e ainda são necessários estudos que comprovem a eficiência no funcionamento dos componentes da bicicleta quando esta está em movimento (a relação da catraca com a corrente e destes com a aro, o acionamento dos freios e a repercussão dessa ação no movimento da bicicleta, entre outros). Para modelar um sistema orientado a objetos, a UML utiliza treze diagramas. Parte deles, tem por finalidade representar aspectos de estrutura e composição do software e a outra parte representa o comportamento dos componentes do sistema quando em execução. Sabendo disso e de posse da leitura do artigo, fale sobre o que você entende por modelar um sistema computacional de forma estrutural e comportamental.

¹³ Adaptado do livro Exercitando Modelagem em UML de Ana Cristina Melo.