

FRANCISCO WILLIANS MAKOTO PLÁCIDO HIRANO

Memória e tempo: Análise Linguístico-Cognitiva dos fatores que influenciam a compreensão de textos multimodais em suporte digitais e impressos.

João Pessoa-PB, 2010

FRANCISCO WILLIANS MAKOTO PLÁCIDO HIRANO

Memória e tempo: Análise Linguístico-Cognitiva dos fatores que influenciam a compreensão de textos multimodais em suporte digitais e impressos.

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Lingüística da Universidade Federal de João Pessoa - PB como requisito parcial para a obtenção do título de Mestre em Lingüística, sob a orientação do Prof. Dr. Jan Edson Rodrigues Leite.

João Pessoa-PB, 2010

H668m Hirano, Francisco Willians Makoto Plácido

Memória e tempo: análise linguístico-cognitiva dos fatores que influenciam a compreensão de textos multimodais em suporte digitais e impressos / Francisco Willians Makoto Plácido Hirano. – – João Pessoa, 2010.

144f. : 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. Compreensão de textos. 2. Textos multimodais. 3. Linguística. I. Universidade Federal da Paraíba. II. Título

CDD 418.4

FRANCISCO WILLIANS MAKOTO PLÁCIDO HIRANO

Memória e tempo: Análise Linguístico-Cognitiva dos fatores que influenciam a compreensão de textos multimodais em suporte digitais e impressos.

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 pela Banca Examinadora composta pelos membros:

Data: 30/10/2010

Conceito: _____

Prof. Dr. Jan Edson Rodrigues Leite - Orientador

Professora Doutora Marianne C. Bezerra Cavalcante

Professora Doutora Karina Falcone – Avaliadora (UFPE)

DEDICATÓRIA

Ao senhor Ademar Hiroshi Hirano, meu pai,
que sempre me deu conselhos sábios.

À senhora Zélia Plácido Hirano, minha mãe e
meu guia.

À Francisca Fernandes de Sousa Neta, mulher
e companheira.

AGRADECIMENTOS

Em especial ao professor Dr. Jan Edson Rodrigues Leite, pela dedicação e seriedade com que conduziu a elaboração deste trabalho, bem como a relevante contribuição na minha formação.

À Francisca Fernandes de Sousa Neta, companheira que teve paciência nas minhas ausências, que independente do futuro sempre fará parte desta obra.

Aos meus pais e minha irmã que com humildade e simplicidade, me conduziram numa formação que foi basilar para minhas conquistas.

Aos amigos do Minter, Liliane, Silmara, Francisco Chagas, Quenizia, Débora, Amilcar, Adriano e Márcia que compartilharam essa luta comigo.

LISTA DE TABELAS

TABELA 01 - HIPÓTESES E VARIÁVEIS INVESTIGADAS DE CADA EXPERIMENTO.	16
QUADRO 1 - EXEMPLOS DE NÍVEIS DE HIERARQUIZAÇÃO DE CONCEITOS SEGUNDO A ABORDAGEM ROSCHIANA.	19
TABELA 02 – TEMPOS DE LEITURA DO PRIMEIRO DIA DE LEITURA DO EXPERIMENTO I.	48
TABELA 03 – LISTA DE BLOCOS DO HIPERTEXTO 1	50
TABELA 04 – NAVEGABILIDADE DO ALUNO 1	51
TABELA 05 – NAVEGABILIDADE ALUNO 2	52
TABELA 06 – NAVEGABILIDADE DO ALUNO 3	52
TABELA 07 – TEMPOS DE LEITURA DO SEGUNDO DIA DE LEITURA DO EXPERIMENTO I.	53
TABELA 08 – LISTA DE BLOCOS DO HIPERTEXTO 2	54
TABELA 09 – LISTA DE JANELAS POPUP DO HIPERTEXTO 2	54
TABELA 10 – TABULAÇÃO DAS RESPOSTAS DO QUESTIONÁRIO DE MEMORIZAÇÃO.	57
TABELA 11 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 1.	59
TABELA 12 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 2.	59
TABELA 13 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 3.	60
TABELA 14 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 4.	60
TABELA 15 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 5.	62
TABELA 16 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 6.	63
TABELA 17 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 7.	65
TABELA 18 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 8.	65
TABELA 19 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 9.	66
TABELA 20 – RESULTADOS DO QUESTIONÁRIO DE MEMORIZAÇÃO PARA QUESTÃO 10.	66
TABELA 21 – TABELA DE TEMPOS DE LEITURA DO EXPERIMENTO II	67
TABELA 22 – LISTA DE BLOCOS DO HIPERTEXTO DO EXPERIMENTO II	68
TABELA 23 – LISTA DE POPUP DO HIPERTEXTO DO EXPERIMENTO II	68
TABELA 24 – RESULTADO DO QUESTIONÁRIO DE MEMORIZAÇÃO	71

LISTA DE FIGURAS

FIGURA 01 – ILUSTRAÇÃO DE UM PROCESSO DE MESCLAGEM SIMPLES	27
FIGURA 02 – MAPA DO CÉREBRO HUMANO	31
FIGURA 03 – TEXTO IMPRESSO UTILIZADO.	42
FIGURA 04 – TELA DE ACESSO AO SISTEMA WEB.	43
FIGURA 05 – TELA DO INÍCIO DA AULA DO PRIMEIRO DIA.	43
FIGURA 06 – TEXTO UTILIZADO NO EXPERIMENTO II	45
FIGURA 07 – GRÁFICO DA DISTRIBUIÇÃO DA MÉDIA, MODA E MEDIANA NO PRIMEIRO DIA DO EXPERIMENTO I	49
FIGURA 08 – LEGENDAS PARA REPRESENTAÇÃO DA NAVEGABILIDADE NO HIPERTEXTO.	50
FIGURA 09 – NAVEGABILIDADE SUGERIDA DO HIPERTEXTO UTILIZADO NO EXPERIMENTO I.	51
FIGURA 10 – GRÁFICO DE NAVEGABILIDADE DO ALUNO 1 X NAVEGABILIDADE SUGERIDA.	51
FIGURA 11 – GRÁFICO DE NAVEGABILIDADE DO ALUNO 2 X NAVEGABILIDADE SUGERIDA.	52
FIGURA 12 – GRÁFICO DE NAVEGABILIDADE DO ALUNO 2 X NAVEGABILIDADE SUGERIDA.	52
FIGURA 13 – GRÁFICO DA DISTRIBUIÇÃO DA MÉDIA, MODA E MEDIANA NO SEGUNDO DIA DO EXPERIMENTO I.	53
FIGURA 14 – NAVEGABILIDADE SUGERIDA PARA O SEGUNDO DIA DO EXPERIMENTO I.	55
FIGURA 15 – GRÁFICO DA NAVEGABILIDADE DO ALUNO 1 X NAVEGABILIDADE SUGERIDA.	55
FIGURA 16 – GRÁFICO DA NAVEGABILIDADE DO ALUNO 2 X NAVEGABILIDADE SUGERIDA.	56
FIGURA 17 – GRÁFICO DA NAVEGABILIDADE DO ALUNO 3 X NAVEGABILIDADE SUGERIDA	56
FIGURA 18 – QUESTÃO DO QUESTIONÁRIO DE MEMORIZAÇÃO.	57
FIGURA 19 – GRÁFICO DA QUANTIDADE DE QUESTÕES ACERTADAS POR CADA GRUPO DO EXPERIMENTO I.	58
FIGURA 20 – RESULTADOS DOS TEMPOS DE LEITURA DO EXPERIMENTO II – GRUPO EXPERIMENTAL A.	67
FIGURA 21 – RESULTADO DOS TEMPOS DE LEITURA DO EXPERIMENTO II – GRUPO EXPERIMENTAL B.	67
FIGURA 22 – NAVEGABILIDADE SUGERIDA DO EXPERIMENTO II.	69
FIGURA 23 – GRÁFICO DE NAVEGABILIDADE ALUNO 1 NO EXPERIMENTO II.	69
FIGURA 24 – GRÁFICO DE NAVEGABILIDADE ALUNO 2 NO EXPERIMENTO II.	70
FIGURA 25 – GRÁFICO DE NAVEGABILIDADE ALUNO 3 NO EXPERIMENTO II.	70

RESUMO

O objetivo de estudo desta pesquisa é descrever como o tempo e a memória, utilizados no processo de leitura, implicam diretamente na compreensão de texto com alto grau de multimodalidade em suportes diferentes (digital e impresso), como também o que essas variáveis (tempo e memória) influenciam diretamente na compreensão em texto com baixo grau de multimodalidade. Admitindo que com os avanços tecnológicos nas áreas de ensino, a cognição deve ser estudada como ela acontece na utilização de novos elementos tecnológicos, na nossa pesquisa escolhemos o hipertexto. Nossa pesquisa é classificada como uma pesquisa experimental que através de dois experimentos realizados com alunos do Instituto Federal do Tocantins Campus de Palmas-TO, que podemos chegar a conclusões sobre a utilização de suporte digital (hipertexto) comparado ao suporte impresso tem alterações mínimas quando analisado os aspectos cognitivos. Demonstrando, assim, que a mudança de suporte apesar de não ajudar nos processos cognitivos também não prejudica sendo isso considerado um grande ganho uma vez que a utilização do suporte digital diminui os custos com papel e deslocamento do material didático para as instituições de ensino.

Palavras-chave: Cognição-Linguística, Memória, Formação do Conhecimento

ABSTRACT

The aim of this study is to describe how time and memory, used in the reading process, directly affect the understanding of text with a high degree of multimodality in different media (digital and printed), but also that these variables (time and memory) directly influence the understanding text with a low degree of multimodality. Assuming that with technological advances in the areas of education, cognition must be studied as it occurs in the use of new technological elements in our research we chose the hypertext. Our research is classified as an experimental research by means of two experiments that with students of the Federal Institute of Tocantins Campus de Palmas-TO, we can draw conclusions about the use of digital media (hypertext) compared to printed support has minimal changes when analyzed cognitive aspects. Thereby demonstrating that the change of support despite do not help in cognitive processes does not preclude it being considered a great gain since the use of digital media reduces papers costs and displacement of didacti material for educational institutions.

Keywords: Linguistics-Cognition, Memory, Knowledge

SUMÁRIO

INTRODUÇÃO	13
2 ENTENDENDO O CONHECER: CATEGORIZAÇÃO E PERSPECTIVAS TEÓRICAS COGNITIVAS	17
2.1 Perspectiva simbólica	18
2.2 Perspectiva Conexionista	19
2.3 Perspectiva Enatista	21
2.4 Modelos Cognitivos Idealizados	22
2.5 Espaços Mentais	24
3 ARMAZENAMENTO DO CONHECIMENTO: CONHECENDO A MEMÓRIA	29
3.1 O que é memória	29
3.2 Os Mecanismos da Formação das Memórias	31
3.3 O aprendizado, como acontece no cérebro.	32
4 DA LEITURA ATÉ A COMPREENSÃO: O CAMINHO A SER PERCORRIDO	33
4.1 Leitura	33
4.2 Compreensão	34
4.3 texto	35
4.4 Hipertexto	35
4.5 Multimodalidade	36
4.6 Processamento da Leitura	38
4.7 Metacognição	38
5 METODOLOGIA	40
5.1 Objetivo da Pesquisa	40
5.2 Descrição do Experimento	40
5.3.1 Experimento I	41
5.3.1.1 Sujeitos da Experiência I	41
5.3.1.2 Procedimentos da Experiência I	41
5.3.1.3 Ferramentas Experimentais Utilizadas	42

5.3.2 Experimento II	44
5.3.2.1 Corpus da Experiência	44
5.3.2.2 Procedimentos da Experiência	44
5.3.2.3 Ferramentas Experimentais Utilizadas	45
5.4 Corpus de Análise – Critério de Exclusão	46
5.5 Procedimento de Análise (Verificação das variáveis)	46
5.5.1 Procedimento de análise da variável tempo	46
5.5.2 Procedimento de análise da variável de memorização	47
5.5.4 Procedimento de análise global	47
6 ANALISE DOS DADOS	48
6.1 Experimento I	48
6.1.1 Investigando a variável tempo	48
6.1.2 Investigando a variável grau de memorização	56
6.2 Experimento II	67
6.2.1 Investigando a variável tempo	67
6.2.2 Investigando a variável memória	70
CONSIDERAÇÕES FINAIS	72
REFERÊNCIAS	74
ANEXO I – DESCRIÇÃO DA TECNOLOGIA UTILIZADA NO SISTEMA WEB DOS EXPERIMENTOS.	77
ANEXO II – TEXTOS DO SUPORTE IMPRESSO EXPERIMENTO I E II	78
ANEXO III - TELAS DO SISTEMA WEB DO EXPERIMENTO I E II	109

INTRODUÇÃO

O conceito de cognição diz respeito ao conhecimento e as suas formas de produção e processamento, que são hoje tratadas por uma área chamada de Ciência Cognitiva.

A Ciência Cognitiva reporta-se à natureza e aos tipos de operações mentais que realizadas no ato de conhecer ou de dar a conhecer. E para os nossos estudos o que interessam são as operações mentais que relacionadas à linguagem.

O avanço tecnológico nas últimas duas décadas passou a ser uma preocupação para os estudiosos em Ciências Cognitivas devido à facilidade de fazer o conhecimento trafegar por milhares de quilômetros sem sair de casa. Porém estudar “sem sair de casa” nunca foi algo novo, já existiam estudos à distância via correspondências, que apesar de demorados, também levam o conhecimento ao leitor.

Estudiosos em Ciências Cognitivas, como Marcuschi (2007), começam a se perguntar se essas novas tecnologias são favoráveis à diminuição do esforço cognitivo para o leitor compreender o conteúdo e também se existe ganho no tempo da aquisição do conhecimento e no esforço cognitivo para o leitor.

Na Era da informação, também conhecida como Era digital, a informação veicula de diferentes modos, principalmente após a invenção de microcomputadores e redes de computadores (internet, por exemplo), os textos já não se apresentam sempre da mesma forma, é muito comum termos os textos com suporte digital, ou seja, no formato para uso dos computadores.

Hoje, ainda não são abundantes os estudos da ciência cognitiva ou da linguística que analisam a formação de conhecimento através destes novos suportes trazidos pelas inovações tecnológicas, trabalhos feitos e orientados por MARCUSCHI na UFPE (Universidade Federal de Pernambuco) dão os primeiros passos para esse estudo.

Dentro desse aspecto, o objetivo geral desta pesquisa é descrever como o tempo e a memória utilizada no processo de leitura influência diretamente a compreensão de texto com alto grau de multimodalidade em suportes diferentes (impresso e digital), como também o que essas variáveis (tempo e memória) influenciam diretamente a compreensão em texto com baixo grau de multimodalidade em comparação a um texto com alto grau de multimodalidade (hipertexto). Utilizamos o hipertexto como ferramenta de maior grau de multimodalidade, no suporte digital, o hipertexto que foi cunhado por Theodor Holm Nelson em 1964, para referir uma escritura eletrônica não-sequencial e não-linear. E utilizaremos como ferramenta multimodal impressa em textos com formato de apostilas, já que o fato de conter textos, figuras, animações, efeitos sobre o texto (itálico, negrito, sublinhado e etc.) caracteriza a multimodalidade.

Os avanços tecnológicos fazem do hipertexto um novo espaço para escrita, caracterizado pela escrita eletrônica utilizando como base a dinâmica oferecida pela tecnologia. O hipertexto tem inúmeras características que podem facilitar a compreensão e a memória, tais como, segundo a classificação proposta por Marcuschi (2007, pag. 150):

- a) Não-linearidade: cada texto aponta pro melhor caminho que o leitor deseja seguir, facilitando o entendimento;
- b) Volatilidade: o hipertexto não tem a mesma estabilidade dos textos, tais como livros e apostilas, e todas as escolhas são tão passageiras quanto às conexões estabelecidas por seus leitores;
- c) Topografia: não é um texto hierárquico ou em tópicos, é um lugar onde não se tem limites para escrever e desenvolver;
- d) Fragmentariedade: consiste na constante ligação de porções, em geral, breves, com sempre possíveis retornos ou fugas;
- e) Acessibilidade ilimitada: o hipertexto acessa todo tipo de fonte, sejam elas dicionários, enciclopédias, museus, obras científicas, literárias, arquitetônicas, etc. E, a princípio, não experimenta limites quanto às ligações que queria fazer;
- f) Multisemiose: este traço caracteriza-se pela possibilidade de interconectar simultaneamente a linguagem verbal e não-verbal;
- g) Interatividade: refere-se à interconexão interativa que por um lado, é propiciada pela multisemiose e pela acessibilidade ilimitada e, por outro lado, pela contínua relação de um leitor-navegador com múltiplos autores em quase sobreposição em tempo real, chegando a simular uma interação verbal face a face.

De um modo geral essas características do hipertexto fazem dele um fenômeno essencialmente virtual e descentralizado. Para a ciência cognitiva estudar os possíveis avanços para a compreensão e a memória que esse fenômeno possa trazer, passa a ser fundamental para avanço das pesquisas cognitivas.

Para estudar o hipertexto como fenômenos para auxílio da compreensão farão esse projeto no qual observaremos a variável “tempo” e a variável “memorização” em leituras comparativas dos grupos dos experimentos, um com hipertexto (suporte digital) e outro com o texto (suporte impresso). Descobriremos o potencial da leitura do hipertexto para o tempo gasto na leitura e na memorização obtida do conteúdo.

Como objetivos específicos temos:

- 1) analisar o grau de memorização em suportes diferentes e em graus de multimodalidades textuais diferentes;
- 2) investigar o tempo de leitura em suportes diferentes e em graus multimodalidades textuais diferentes;
- 3) entender caminhos utilizados para a formação do conhecimento obtido pelos alunos;
- 4) determinar as potencialidades da leitura para o trabalho de compreensão e da memória em diferentes suportes (digital e impresso);
- 5) determinar as potencialidades da leitura para o trabalho de compreensão e da memória em textos com alto grau de multimodalidade com suporte digital a textos impressos lineares.

A importância desta pesquisa, para a linguística cognitiva, se faz por acreditar que atualmente a utilização de microcomputadores como suporte para leitura é uma prática cada vez mais comum, independentemente da idade do leitor. Esta pesquisa estudará que a formação do conhecimento através deste suporte digital e do hipertexto em comparação a formação do conhecimento através do suporte impresso em textos comuns.

O hipertexto deixa que o leitor tenha seu próprio rumo da leitura no qual será diferente dos impressos onde a linearidade é predominante no texto. O hipertexto, com sua característica de multisemiose, utiliza-se de cores, dos sons, das formas plásticas, dos desenhos, dos links que definem os rumos da leitura.

Assim, acredita-se que ao pesquisar este tema é possível demonstra que ganho cognitivo na compreensão de textos multimodais com suporte digital (hipertextos) em relação à compreensão de textos multimodais impressos, e observar a variável “tempo de leitura” pode demonstrar o ganho cognitivo que se possa ter.

Nossa pesquisa tem como hipóteses que:

- 1) a compreensão leitora em textos multimodais através do suporte digital demanda menos tempo em atividades que exigem esforço cognitivo igual ao utilizado na leitura de textos multimodais em suporte impresso;
- 2) a compreensão leitora em textos multimodais através do suporte digital demanda menos tempo em atividades que exigem esforço cognitivo igual ao utilizado na leitura de textos monomodal em suporte impresso;
- 3) a memorização de informações de textos multimodais com suporte digital é maior que o grau memorização através de textos multimodais com suporte impresso;
- 4) a memorização de informações de textos multimodais com suporte digital é maior que o grau memorização através de textos monomodal com suporte impresso;

Veja na tabela 1 a representação das hipóteses.

Tabela 01 - Hipóteses e variáveis investigadas de cada experimento.

Experimento	Variável investigada	Variável controlada	Resultado esperado
I	Tempo de leitura	Suporte (digital e impresso)	Menor tempo de leitura no suporte digital
II	Tempo de leitura	Suporte e Modalidade do texto	Menor tempo de leitura no suporte digital e textos multimodais
I	Memorização de informações	Suporte	Maior grau de memorização no suporte digital
II	Memorização de informações	Suporte e Modalidade do texto	Maior grau de memorização no suporte digital e textos multimodais

Far-se-á tal pesquisa através de um experimento cujas variáveis serão observadas e controladas para que se tenha a confirmação ou não das hipóteses citadas.

Nossa pesquisa, descrita aqui neste trabalho, é apresentada em 6 capítulos além da introdução, sendo aqui numerada como capítulo 1, o segundo capítulo é uma descrição das perspectivas teóricas cognitivas, fazendo um apanhado dos principais modelos cognitivos, o terceiro capítulo demonstra como é o funcionamento da memória do ser humano, o quarto capítulo é dedicado a teorias de leitura e compreensão, em seguida o quinto capítulo dedicado a metodologia, define e classifica a pesquisa e expressa como será executada a pesquisa, por fim sexto capítulo que apresentará as análises dos dados levantados no experimento descrito no quinto capítulo, e neste capítulo utilizará as teorias apresentadas nos capítulos 2, 3 e 4 para esclarecer os resultados obtidos.

2 ENTENDENDO O CONHECER: CATEGORIZAÇÃO E PERSPECTIVAS TEÓRICAS COGNITIVAS

“... Havendo, pois, o SENHOR Deus formado da terra todo o animal do campo, e toda a ave dos céus, os trouxe a Adão, para este ver como lhes chamaria; e tudo o que Adão chamou a toda a alma vivente, isso foi o seu nome. E Adão pôs os nomes a todo o gado, e às aves dos céus, e a todo o animal do campo...” (Gênesis, cap. 2 versículos 19 e 20).

Visando a um melhor entendimento dos pressupostos das ciências cognitivas é importante fazer uma reflexão histórica entre os caminhos que no entendimento e definições de categorização.

Segundo Saraiva (2009, PAG. 39) categorizar significa ordenar um número infinito de estímulos encontrados nas nossas experiências cotidianas em categorias finitas e de fácil acesso. Assim como Adão deu os nomes as coisas para identificá-las pela língua, criando para cada nome uma representação mental das animais e das coisas, o ser humano tem a habilidade cognitiva de organizar as coisas e criaturas ao seu redor, e tratar estímulos diferentes como equivalentes. Assim como associar um ônibus e um fusca como automóveis. Categorizar é um processo cognitivo de divisão das experiências de mundo em entidades (objetos, idéias, ações, etc.), resultando na construção de um ordem física e social do mundo. Ainda segundo Saraiva:

“... categorizar como processo cognitivo, proporciona, de forma extremamente eficaz, uma economia cognitiva ao cérebro humano, que simplifica nossa vida, na medida em que nos permite dividir os estímulos em classes de coisas e eventos para compactar a quantidade de informação que precisamos assimilar, perceber, lembrar e reconhecer ao longo da nossa vida.” (SARAIVA, 2009, pag. 30).

A economia cognitiva que nos fornece a categorização agiliza o reconhecimento e identificação das coisas, através das funções básicas da categorização: classificação, apoio as explicações e sustentabilidade a mente.

Lakoff enfatiza a importância de se estudar a categorização:

“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 que nós vemos algo como um “tipo de coisa”, por exemplo, uma árvore, nós estamos categorizando.” (LAKOFF, 1987, p. 5).

Assim podemos afirmar que sem categorização não entenderemos nossos pensamentos, ou seja, categorização é o ponto central para a compreensão daquilo que está ao nosso redor.

“Entendemos, assim, as categorias como nossas versões públicas do mundo, que constituem e são constituídas pelo conhecimento como formas palpáveis que utilizamos para manusear a fluidez da realidade e como meios de organizarmos o conhecimento sobre algo. É por isso que são itens priorizados nos diversos sistemas de ensino, e, em geral, constituem grades

Categorizar é ver e reconhecer o mundo, ou seja, é a representação de mundo que fazemos cognitivamente. Para melhor entendimento de categorização vamos fazer uma viagem epistemológica das discussões sobre a categorização enquanto processo cognitivo, nos principais paradigmas no âmbito das Ciências da Cognição: 1) Perspectiva simbólica, 2) Conexionistas e a 3) Enatista.

2.1 PERSPECTIVA SIMBÓLICA

Esta vertente simbólica considera que a cognição pode ser explicada através de operações sobre símbolos, em termos de teorias computacionais e modelos mentais. Tem-se então que os processos mentais são análogos a processamento computacionais, em suas características essenciais.

“(...) Isso porque as atividades do computador em si pareciam, em alguns aspectos, semelhantes aos processos cognitivos. Os computadores recebem informação, manipulam símbolos, armazenam itens na “memória” e buscam-nos novamente, classificam inputs, reconhecem padrões e assim por diante. Na verdade, os pressupostos que servem de base à maior parte dos trabalhos contemporâneos sobre processamento de informação são surpreendentemente parecidos com os da psicologia introspeccionista do século XIX, embora sem a introspecção propriamente dita.” (NEISSER, 1976, p.5-7).

A partir dessa premissa, os cognitivistas dessa vertente têm tradicionalmente pensado o conhecimento como estando representado quer sob uma forma simples, quer sob uma forma complexa. A pesquisa sobre a organização simples investiga o modo como diferentes categorias semânticas podem ser agrupadas sob um conceito comum e como essas entidades se relacionam umas com as outras. Consideraremos aqui a Teoria dos Protótipos proposta por Rosch (1973) e colabores como ponto central para a discussão dessa linha de pesquisa.

As teorias de protótipos surgiram com o intuito de explicar as falhas do ponto de vista de definição de atributos, como, por exemplo, os efeitos de tipicidade e a falta de distinção das concepções. Elas possuem como fundamento básico o fato de que as categorias são organizadas em torno de protótipo (exemplares) principais, sendo que a natureza de tais estruturas é concebida de modo diferente nas diversas teorias dessa corrente.

Um ponto importante nas teorias de protótipo é que elas rejeitam a existência dos atributos de definição, sendo, por esse motivo, intitulada também de teorias do atributo característicos. Argumentam que, na verdade, os conceitos tem uma estrutura de protótipo com seu conjunto de

atributos, dentro desse conceito, quando mais atributos tiver melhor o exemplar representa a categoria (ROSCH, 1973).

Outras implicações da teoria são:

- Os limites das categorias não são bem definidos, podendo alguns membros de uma categoria deslizar para outra categoria (ex. tomates como fruta ou verduras).
- Dentro de uma categoria, os membros variam em graus de tipicidade.
- A pertença a uma categoria é determinada pela semelhança dos atributos de um objeto ao protótipo de uma categoria, seja esse representado por atributos característicos ou por um exemplar da categoria.

Considerando que existe variação em graus de tipicidade, os níveis de hierarquização dos conceitos são explicados a partir de uma estrutura definida em três níveis:

- Nível supra-ordenado, de baixa informatividade e grande economia cognitiva;
- Nível básico, em que os conceitos tem o maior número de atributos distintivos de outros conceitos do mesmo nível, variando em função de diferenças individuais e culturais;
- Nível subordenado, para conceitos específicos.

No estabelecimento dessa hierarquia, o nível básico adquire um caráter privilegiado dentro de uma categoria, por ser aquele em que os conceitos tem maior número de atributos distintivos e o mais econômico cognitivamente, isto é, há menos partilha de atributos de conceitos:

Quadro 1 - Exemplos de níveis de hierarquização de conceitos segundo a abordagem roschiana.

Nível Supra-ordenado	Arma	Fruta	Móvel
Nível básico	Arma de fogo	Banana	Cadeira
Nível Subordenado	Revólver	Banana prata	Poltrona

2.2 PERSPECTIVA CONEXIONISTA

O paradigma Conexionista constitui-se como um modelo de cognição que procura suprir as falhas e lacunas deixadas pelos simbolistas. O conexionismo, como também é denominado, se interpõe entre as duas abordagens, diferindo do primeiro porque procura analisar os processos que ocorrem no cérebro, não somente os dados de entrada e saída. Difere do segundo por romper

definitivamente com a idéia de representação mental e com a dicotomia mente/cérebro, ainda que em alguns aspectos tenha um caráter subsimbólico. Ao se apoiar em bases fisiológicas para explicar o processamento mental, causou uma profunda transformação no entendimento das estruturas mentais.

“O connexionismo foi a alternativa mais forte entre as que disputavam com o processamento serial a primazia como modelo dos processos cognitivos humanos” (KOCH, CUNHA-LIMA, 2007, p. 272).

Nesse modelo, inspirado no sistema nervoso humano, caracteriza-se por uso de redes neurais, rede formada por nódulos de capacidade limitada e interligada. Essas redes são capazes de demonstrar aprendizado quando fornecemos dados, mesmo que não tenhamos anteriormente instruído a rede sobre que tipo de informação procurar.

O aprendizado acontece porque as ligações entre os nós vão mudando à medida que a rede torna contato com os dados, alterando sua configuração interna. A rede é capaz de extrair as informações, desde que elas estejam presentes nos dados de forma relevante, ou seja, desde que estejam armazenadas de forma acessível.

“Ao determinar a forma como as informações serão apresentadas à rede e qual a arquitetura da rede é necessário sermos completamente explícitos a respeito de como acreditamos que um dado processo aconteça. Se a rede em questão tiver sucesso em seu aprendizado, a forma pela qual o seu modelo foi construído é coerente internamente. Essas ferramentas computacionais constituem uma novidade interessante, porque permitem desenvolver modelos precisos de funcionamento de determinado sistema cognitivo e testa minuciosamente as hipóteses” (KOCH, CUNHA-LIMA, 2007, p. 273).

É importante poder estudar as possibilidades de sistemas cognitivos sem representação simbólica por dois motivos: primeiro, porque oferecem uma alternativa séria, com méritos próprios à representação simbólica e já devem ser postos em competição com ela; segundo, porque possibilitam concretamente modelar um sistema cognitivo de forma dinâmica, captando a organização temporal desses sistemas e construindo um modelo da cognição que parece ajustar-se naturalmente às concepções de cognição como fenômeno social. (KOCH, CUNHA-LIMA, 2007, p. 274).

No entanto, as críticas mais contundentes à perspectiva connexionista vem da perspectiva enatista e refutam a idéia de uma dicotomia mente-cérebro-organismo e de uma cognição desincorporada, como veremos a seguir.

2.3 PERSPECTIVA ENATISTA

O conexionismo e o modo pela qual os estímulos são representados e os parâmetros associados com o aprendizado e sua arquitetura de redes de conexões passam a idéia de que o cérebro é uma estrutura central no processo de categorização e, de certo modo, pré-programada, apesar de o aprendizado modificar tais programações.

“O modelo enatista contrapõe-se criticando o conexionismo e sua postura centrada num cérebro dissociado de um corpo, que recebe dados externos (input) e processa-os gerando assim respostas (output).” (SARAIVA, 2009, pag. 63).

O atuacionismo se propõe resolver a questão que separa mente e corpo como entidades não estanques, sensíveis às pressões de um sobre o outro, na teoria da mente corporificada.

“Tal concepção de cognição surge com o objetivo de libertar as ciências cognitivas do modelo realista e, logo, representacional da cognição que se sustém na compreensão do processamento cognitivo a partir do tratamento da informação como procedente de propriedade formais preexistentes no ambiente contextual. Duas posições teóricas contribuíram para a elaboração dessa concepção: a teoria da mente corporificada (VARELA; THOMPSON; ROSCH, 1991), que sustenta a visão de que a cognição depende dos tipos de experiência que derivam do fato de termos um corpo com várias capacidades sensório-motrizas individuais embutidas em um contexto biológico, psicológico e cultural mais fechado; e o realismo experiencialista de Lakoff e Johnson (1980) e Lakoff (1987).” (RODRIGUES-LEITE, 2008, p. 98/99).

O realismo experiencialista Lakoff e Johnson (1980) e Lakoff (1987) defendem que o nosso sistema de conceitualização se assenta na utilização das nossas experiências, como organismos dotados de certa configuração biológica. Essa projeção do corpo na mente é a hipótese da corporificação. Uma noção central a essa teoria é a de imagem schema, modelo cognitivo determinado de forma genérica recorrente na nossa experiência sensorial e motora, que é projetado noutros domínios da experiência, determinando sua configuração (RODRIGUES-LEITE, 2008, p. 71).

A teoria da mente corporificada (VARELA; THOMPSON; ROSCH, 1991) se fundamenta sobre os símbolos significantes e não finitos, que funcionam segundo esquemas imagéticos baseados em processos perceptivo-conceituais elementares e considera a experiência como ativa, funcionando como parte de um ambiente natural e social, motivando o que é significativo no pensamento humano.

2.4 MODELOS COGNITIVOS IDEALIZADOS

“A noção proposta pelo autor, na lingüística cognitiva, relativa aos modelos cognitivos idealizados (MCI), reflete a maneira como organizamos o nosso conhecimento através de estruturas de categorias e efeitos prototípicos que são produtos resultantes da disposição dos MCI.” (RODRIGUES-LEITE, 2004, p. 65).

Os modelos cognitivos estruturam o pensamento e são usados na formação de categorias e do raciocínio. Os conceitos caracterizados pelos modelos cognitivos são compreendidos via a corporificação desses modelos. Utilizamos modelos cognitivos idealizados para tentarmos compreender o mundo. Em geral, qualquer elemento de um modelo cognitivo pode corresponder a uma categoria. Um MCI pode se adequar à compreensão que uma pessoa tem do mundo de forma perfeita ou imperfeita. Quanto menos perfeita é a adequação entre as condições prévias do MCI e nosso conhecimento, menos apropriada para nós é a aplicação de um conceito (RODRIGUES-LEITE, 2004, p. 66).

“A importância dos Modelos Cognitivos Idealizados (MCIs) na construção de categorias que atuam na elaboração dos objetos de discurso é a de produzir certos domínios estáveis, sócio-cognitivamente elaborados a partir da memória social dos indivíduos e partilhados nas atividades de significação como “bases de dados” que orientam a produção coletiva da conceptualização. Apesar de estáveis estes domínios têm natureza flexível, ou seja, como conjuntos de conhecimentos, podem ser alterados, dadas certas contingências sócio-históricas, além de certas necessidades comunicativas.” (RODRIGUES-LEITE, 2008, p. 75).

Os domínios estáveis são os nossos conhecimentos prévios e correspondem às estruturas de memória pessoal ou social (os esquemas ou frames). São, ao mesmo tempo, flexíveis e estáveis, porém não estáticos. Para Salomão em Medrado (2006, pag. 73), os domínios estáveis caracterizam-se pela sua permanência como ordens cognitivas, identificáveis e evocáveis; pela organização interna das informações que os constituem e pela flexibilidade de sua instanciação, conforme as necessidades locais manifestadas. São domínios estáveis os Modelos Cognitivos Idealizados (MCIs).

Os MCI'S têm status cognitivo. São usados para a compreensão do mundo e para a criação de teorias sobre o mundo. Assim, os modelos cognitivos nos permitem fazer sentido de uma variedade de fenômenos semânticos. O desenrolar do discurso exige que sejam efetuadas construções cognitivas que incluem conjuntos de conhecimentos organizados e estruturados de acordo com os domínios estáveis e locais. Os MCI'S são domínios estáveis, que, ao lado dos enquadres comunicativos e esquemas genéricos estruturam o conhecimento socialmente produzido e localmente disponível (RODRIGUES-LEITE, 2008, p. 75-77).

“Nós usamos modelos cognitivos idealizados para tentarmos compreender o mundo. Em geral, qualquer elemento de um modelo cognitivo pode corresponder a uma categoria conceptual. Um MCI pode se adequar à compreensão que uma pessoa tem do mundo de forma perfeita ou imperfeita. Quanto menos perfeita é a adequação entre as condições prévias do MCI e nosso conhecimento menos apropriada para nós é a aplicação de um conceito.” (RODRIGUES-LEITE, 2004, p. 66).

O mapeamento entre esses domínios permite a construção dos significados partindo dos esquemas mais genéricos, de base, para esquemas particulares. O processo de conceitualização, dessa forma, envolve o reconhecimento da integração entre os domínios-fontes e alvo para a construção de um terceiro domínio que utiliza os conteúdos estruturados nos dois níveis anteriores. Demonstramos abaixo com um exemplo de Rodrigues - Leite (2004).

1. P= /../ vocês sabem o que significa reciclagem?
2. AA= ((silêncio))
3. P=que símbolo é este ((a professora mostra o símbolo de reciclar formado).
4. (Por três setas largas interconectadas em um círculo)
5. A = já sei/ reciclar é colocar o lixo na caixa azul /../

Rodrigues - Leite (2004) conclui que sobre o exemplo acima:

“A significação a que chega o falante na linha cinco é contextualmente possível, porque os participantes do evento compartilham um modelo cultural construído localmente. Por tratar-se de um ambiente de escolarização, possivelmente, através de um contrato sociodidático, os participantes negociaram a utilização de um recipiente (uma caixa envolta em papel azul), onde possam ser colocadas as sobras de papel e outros materiais reutilizáveis. Tal recipiente, dada sua função de depósito de materiais recicláveis contém um símbolo internacionalmente reconhecível que indica sua função. Como o evento se trata de uma aula de alfabetização com alunos na faixa etária de cinco a seis anos, a pergunta da professora na linha um não contextualiza conhecimentos possuídos pelos alunos. Trata-se na verdade de um conceito técnico que engloba uma série de possíveis definições. No entanto, quando o símbolo e a categoria introduzida pela professora, o que engatilha uma fusão das duas sendo conceptualizadas na assertiva do aluno (linha cinco). O processo, nesse caso, conta com o mapeamento entre dois domínios: aquele em que um Modelo Cognitivo Idealizado expresso pela categoria conceptual reciclagem ativa, uma expressão na língua, vinculada ao discurso técnico; e aquele em que a expressão aciona uma semiose, - imagética correlacionada às experiências prévias dos participantes.” (RODRIGUES-LEITE, 2004, p. 110)

O MCI formado, no exemplo, quanto ao símbolo de reciclagem, demonstra como a formação do conceito em sala de aula e na vida cotidiana de crianças de 5 e 6 anos de idade.

Medrado (2006) fez considerações sobre os MCI's que são formados a partir da estrutura básica que são as categorias cognitivas.

“Os MCI’s têm um papel importante na cognição humana, visto que permitem que nos lembremos de situações vividas e usemos os conhecimentos que adquirimos. É, também, uma espécie de protótipo virtual, pois existem enquanto esquemas conceituais permanentes, compondo um nível básico de estabilidade que organizam, a partir de categorias, o nosso conhecimento.” (MEDRADO, 2006, p. 73).

Verificamos também a utilização de MCI nesse outro exemplo:

Nesse outro exemplo, colhido em uma observação do dialogo entre um pai e uma filha de 4 anos de idade.

1. Filha: /.. / pai, de onde vem o trovão?
2. Pai: trovão vem lá do céu.
3. Filha: hum, ((a criança pensa por alguns segundos)).
4. Filha: pai se trovão vem do céu por que ele é mau?
5. Pai: ((silêncio))
6. Filha: pai se trovão é mau e tá no céu, por que Jesus não mata ele?

As indagações sobre o significado de “trovão” para a criança demonstram já correção do Modelo Cognitivo Idealizado “céu”, onde para a criança é o lugar que Jesus vive e só tem coisas definidas como coisas do bem.

Nesse caso, o processo conta com dois domínios estáveis, um primeiro em que um Modelo Cognitivo Idealizado expresso pela categoria conceptual científica, e um segundo em que um Modelo Cognitivo Idealizado expresso pela categoria conceptual religiosa.

Semelhantemente a teoria do MCI’s existe a teoria dos espaços Mentais (domínios locais), veremos logo em seguida.

2.5 ESPAÇOS MENTAIS

Constituem operadores do processamento cognitivo, ou seja, tem um caráter dinâmico e sequencial, já que são produzidos na medida em que falamos. Esses espaços mentais são domínios dinâmicos estruturados internamente por domínios estáveis, o que significa que são elementos que suscitam aspectos do conhecimento compartilhado entre elementos que suscitam aspectos do conhecimento compartilhado entre os interlocutores em um determinado ponto do discurso.

Por outro lado, os espaços mentais são ativados através de conectores, chamados construtores de espaços mentais, que no nível gramatical podem assumir estrutura de conectivos, de sintagmas preposicionais ou adverbiais, de orações, entre outras possibilidades, criando diferentes tipos de espaços.

“Assim, de acordo com o modelo de mapeamentos de Fauconnier (1997), as categorias apresentadas e discutidas em aula pelo professor e alunos estruturados pelos modelos

cognitivos idealizados do discurso cotidiano podem ser projetadas entre si, através do princípio de identificação conceptual da teoria dos Espaços Mentais (EM)” (RODRIGUES-LEITE, 2004, p. 67/68).

Este mapeamento permite a construção dos significados partindo dos esquemas mais genéricos, de base, para esquemas particulares.

“O processo de reconceptualização, desta forma, envolve o reconhecimento da integração entre os domínios fonte e alvo para a construção de um domínio mescla (blending) que levam em conta conhecimentos estruturados nos dois níveis anteriores (fonte e alvo)” (RODRIGUES-LEITE, 2004, p. 67/68).

A partir da definição de MCI's, a teoria “espaços mentais” sendo definidos como domínios cognitivos de natureza semântico-pragmática que se configuram no processamento discursivo ativados por certas expressões lingüísticas e por alguns mecanismos de reconhecimento de elementos em diferentes campos (psicológico, cultural, histórico, ficcional etc.). Numa prática comunicativa qualquer, ativamos vários espaços mentais e inter-relacionamos elementos de vários desses espaços, não só numa relação biunívoca, mas estabelecendo uma rede de projeções tal que a linguagem se configura como um complexo emaranhado de elementos, domínios e projeções (SOUZA, 2003, p. 101).

A projeção conceptual são definidos como representações temporárias construídas pelos falantes.

Segundo Rodrigues-Leite (2004, pag. 71-72) estes espaços dependem em larga medida de estruturas cognitivas estáveis (como os Modelos Cognitivos Idealizados), mas diferentemente destes, os EM são representações de curto prazo, cuja função é responder às necessidades de conceptualização, muitas vezes novas e mesmo únicas, dos falantes. Esta teoria postula a existência de quatro (ou mais) espaços mentais envolvidos no processo de projeção conceptual entre domínios: dois espaços de input (espaços influentes correspondentes ao domínio-fonte e ao domínio-alvo), um espaço genérico que comporta a estrutura abstrata partilhada pelos dois espaços anteriores (e eventualmente por muitos outros) e ainda um espaço-mescla (blend), em que se verifica a combinação, a mistura, de representações dos espaços influentes, e por vezes, também de outros espaços mentais cuja informação é mobilizada. É desta mesclagem que resulta uma nova conceptualização, não submissível a uma soma das estruturas dos espaços influentes, nem a um mero conjunto de correspondências previsíveis fora deste processo.

Segundo Miranda (2000), a escolha do termo satisfaz, visto que evoca o tecido conhecido por mescla, cujos fios, embora se misturem para compor a trama, não se fundem, mantendo-se visíveis.

“Devemos entender o processo de integração conceptual ou mesclagem como aquele que traz para a interpretação os traços ou indícios das idéias que foram parcialmente projetadas nos domínios fonte, integrando situações, solucionando problemas e construindo conceitos” (MEDRADO, 2006, p. 92).

Relações associativas como as apresentadas são possíveis pela existência prévia do processo de mesclagem, o qual estabelece uma conexão entre diferentes domínios conceituais, ou seja, diferentes conjuntos de conhecimentos pré-linguísticos que são estruturados por nossas experiências coletivas ou individuais.

O conceito de mesclagem é definido pelo referido autor como o espaço que herda estruturas parciais de espaços denominados fonte e que possui uma estrutura emergente própria. Logo, o espaço-mescla surge a partir da composição de elementos provenientes de um esquema genérico, dos MCI's ativos e dos espaços mentais locais.

Assim, um enunciado ativa domínios conceituais sobre os sentidos que veicula. Dependendo do que é referido, múltiplos domínios podem ser ativados e informações são transferidas dessas estruturas cognitivas para a construção de novos significados na linguagem. Esses novos significados “mesclam” informações dos domínios, dos quais partiram. Logo, o enunciado resultante dessas combinações é pré-organizado na mente pelo processo cognitivo de mesclagem.

Fauconnier formaliza o processo de mesclagem através de diagramas nos quais os espaços mentais (inputs) são representados por círculos; os elementos nos espaços por pontos ou ícones; as conexões entre os elementos, nos diferentes inputs, por linhas, e as projeções entre os elementos dos inputs, o espaço genérico e o espaço mescla, por linhas tracejadas. As linhas representariam, segundo Fauconnier e Turner (2002), as ativações (coactivations and binding) neuronais. A estrutura emergente sendo representada por um retângulo. Uma rede de integração conceptual mínima apresenta quatro espaços: dois inputs, um espaço genérico e um espaço mescla. Outros tipos de redes podem conter vários espaços de input e, até mesmo, espaços mesclas múltiplos. Vejamos, então, a ilustração de um processo de mesclagem simples:

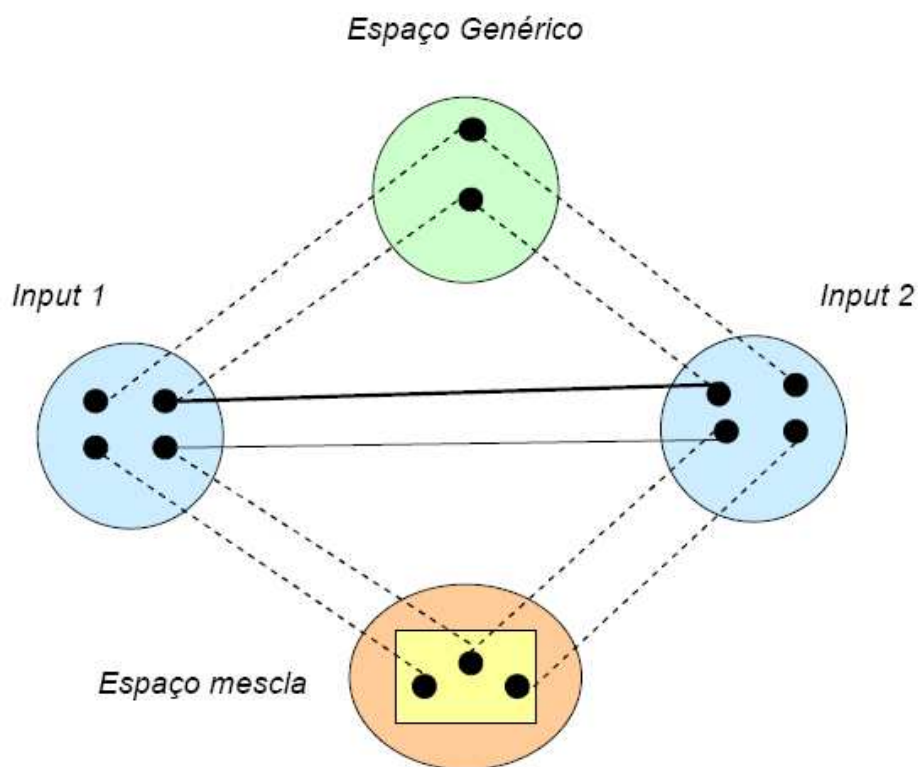


Figura 01 – Ilustração de um processo de mesclagem simples

O processo de mesclagem é construído a partir dos seguintes princípios (MEDRADO, 2006):

- Espaços de entrada (ou input) – domínios independentes que, parcialmente se correspondem, possuindo os elementos que vão resultar no espaço mental da mescla ou fusão.
- Projeção ou correspondência entre espaços de entrada - fenômenos periféricos que operam ligando domínios e servem justamente para estabelecer as conexões entre os espaços mentais. Há vários tipos de projeção:
 - o Projeção entre MCI's que projetam parte de um domínio em outro. As metáforas e analogias são bons exemplos deste tipo de projeção: existe um domínio-fonte com espaços mentais estabilizados do qual partimos para um domínio-alvo (onde produzimos a metáfora, por exemplo);
 - o Projeção de funções pragmáticas, como as metonímias. Segundo esta regra, o princípio de acesso (SWEETSER e FAUCONNIER, 1996. In MEDRADO 2006) sustenta a idéia de que uma expressão tida como gatilho (trigger) pode nomear ou descrever uma entidade e ser usada para acessar e/ou referir-se a uma segunda entidade em um outro domínio. Este último deve ser cognitivamente acessível ao

primeiro e, portanto, precisa existir algum tipo de conexão entre o gatilho e o alvo (target).

- o Projeção das relações vitais - quando um esquema geral é usado para estruturar uma situação no contexto. Construções gramaticais e lexicais evocam tais esquemas bem como construções discursivas.
-
- Espaço genérico – mais amplo e abstrato, contém as estruturas comuns aos espaços de entrada.
 - Espaço mescla – resultado das combinações parciais dos espaços de entrada gera a estrutura emergente.
 - Estrutura emergente – produto último de todo o processo de mesclagem. A constituição dessa estrutura se dá, segundo Fauconnier e Turner (2002, in MEDRADO, 2006), a partir de três elementos:
 - o Composição - A composição dos elementos dos inputs constrói relações que não existiam anteriormente nos inputs de entrada.
 - o Complementação - Acrescenta um conjunto ou estrutura adicional de conhecimento prévio, modelos cognitivos e culturais, que não estava nos inputs sendo a mesma integrada ao espaço-mescla.
 - o Elaboração – Fase dinâmica da fusão – A mescla é posta em funcionamento de forma criativa.

3 ARMAZENAMENTO DO CONHECIMENTO: CONHECENDO A MEMÓRIA

Segundo HENRY e GOFF (1990, pag. 1-2) memória é a única palavra que se refere a um conjunto complexo e fascinante de habilidades que as pessoas e outros animais possuem que lhes permite aprender com a experiência e reter o que aprendem. Uma memória, uma experiência que afeta o sistema nervoso e deixa resíduo ou vestígio e alterações posteriores no comportamento. Tipos de memória são extremamente variadas, assim também são as técnicas utilizadas na ciência cognitiva para investigá-las.

3.1 O QUE É MEMÓRIA

“Memória é a aquisição, a formação, a conservação e a evocação de informações. A aquisição é também chamada de aprendizado: só se grava aquilo que foi aprendido. A evocação é também chamada de recordação, lembrança, recuperação. Só lembramos aquilo que gravamos aquilo que foi aprendido” (IZQUIERDO, 2002, p. 9).

Segundo Izquierdo (2002, p. 9) podemos afirmar que somos aquilo que recordamos, literalmente. Não podemos fazer aquilo que não sabemos como fazer, nem comunicar nada que desconheçamos, isto é, nada que não esteja na nossa memória. Não podemos usar como base para projetar nossos futuros possíveis aquilo que esquecemos ou que nunca aprendemos. Também não estão à nossas memórias faz com que cada um de nós seja o que é, com que sejamos, cada um, um indivíduo, um ser para o qual não existe outro idêntico.

De acordo com Henry e Goff (1999, pag. 3-4) as memórias são divididas em:

- A memória de procedimento refere-se ao conhecimento de como fazer coisas como andar, falar, andar de bicicleta e amarrar o cadarço. Os procedimentos são muitas vezes adquiridos lentamente e só após muita prática e repetição. Os tipos de condicionamentos a que as espécies de animais estão sujeitos são outros exemplos de memória processual. Memória de procedimento é muitas vezes comparada com a memória declarativa, ou com conhecer fatos sobre o mundo e sobre o passado.
- A memória episódica refere-se ao momento de lembrar os episódios da nossa vida e é contextualmente vinculados, ou seja, à hora e o local de ocorrência. Este tipo de memória permite que o tempo de utilização mental em que nos envolvemos seja menor quando voltamos a mesma situação anterior, porque assim, constitui a história pessoal de cada indivíduo, às vezes é chamada memória autobiográfica.

- Memória semântica se refere ao nosso conhecimento geral do mundo. Este conhecimento não está vinculado a um episódio e não precisamos nos referir ao tempo ou ao lugar em que aprendemos esses fatos para saber que eles são verdadeiros.

Esta não é a única forma de distinguir os tipos de memória. Izquierdo (2002) distingue os tipos de memória como memória de curto prazo e memória de longo prazo:

- Memória de trabalho seu principal papel não é o de formar arquivos, mas sim o de analisar as informações que chegam constantemente ao cérebro e compará-las com as existente nas demais memórias (memória de curto prazo e de longo prazo). Sua duração é no máximo de poucos segundos a no máximo 3 minutos.
- Memória de curto prazo (ou memória principal) está relacionada à nossa capacidade de manter em mente uma quantidade relativamente pequena de informações que é rapidamente esquecida quando deixamos de atender a ela. Um bom exemplo é lembrar um número de telefone por um breve período depois de olhá-lo. Esta habilidade é também referida como memória de trabalho porque nos permite realizar o trabalho mental de manipulação de símbolos e de pensar, sua duração vai de poucos segundos até no máximo de 3 a 6 horas.
- Memória de longo prazo (ou memória secundária) é um pouco impreciso o termo que é usado para se referir à retenção de vários tipos durante longos períodos de tempo, dependendo do conteúdo, prazo pode significar em qualquer lugar de 30 segundos para muitos anos. Izquierdo (2002) afirma que a formação de uma memória de longa duração ou de longo prazo envolve uma série de processos metabólicos no hipocampo (veja figura 02) e outras estruturas cerebrais que compreendem diversas fases e que requerem entre três e oito horas. Enquanto esses processos não estiverem concluídos, as memórias de longa duração são instáveis (IZQUIERDO, 2002, p. 35).

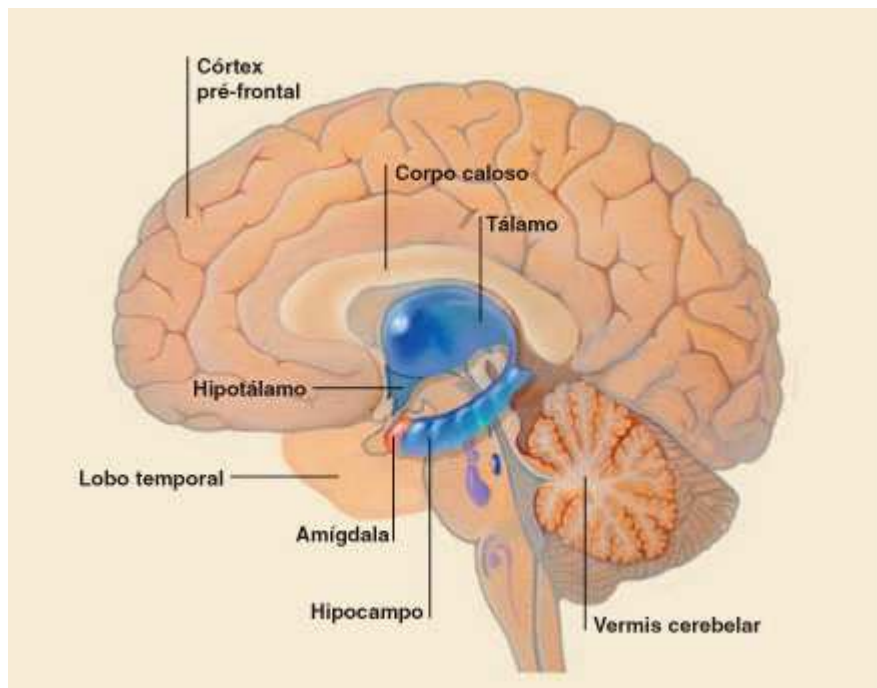


Figura 02 – Mapa do Cérebro Humano

O mapa do cérebro humano representado na figura 02, demonstra a o pequeno pedaço que o hipocampo representa no cérebro todo, sendo responsável pelos processos metabólicos que ocorrem na memória de longa duração.

As memórias de longa duração que duram muitos meses ou anos costumam ser denominadas memórias remotas. Segundo Izquierdo (2002, pag. 27) um rato é capaz de lembrar, um ano depois, que, em um dado compartimento de determinada caixa, recebeu um choque elétrico nas patas.

3.2 OS MECANISMOS DA FORMAÇÃO DAS MEMÓRIAS

As memórias não são adquiridas imediatamente na sua forma final. Durante os primeiros minutos ou horas após sua aquisição, elas são suscetíveis à interferência por outras memórias, por drogas ou por outro tratamentos (Izquierdo, 2002, pag. 35).

Segundo Izquierdo e Medina (1997) em Izquierdo (2002, pag. 36) a formação de uma memória de longa duração envolve uma série de processos metabólicos no hipocampo e outras estruturas cerebrais que compreendem diversas fases e que requerem entre três e outro horas. Enquanto esses processos não estiverem concluídos, as memórias de longa duração são lábeis. O conjunto desses processos e o seu resultado final denominam-se consolidação.

Ainda segundo o mesmo autor, a descoberta da consolidação veio de duas fontes. Uma, da observação popular, já no fim do século XIX, de que após um traumatismo craniano os indivíduos

esquecem seletivamente aquilo que tinha acontecido nos minutos anteriores. Outra, da observação de dois pesquisadores alemães, Müller e Pilzecker, de que muitas memórias interferem com outras, adquiridas pouco tempo antes.

O papel da memória de trabalho e o da memória de longa duração são óbvios, mas o papel da memória de curta duração é, basicamente, o de manter o indivíduo em condições de responder através de uma “cópia” da memória principal, enquanto esta ainda não está formada.

Para isto, a memória de curta duração não sofre extinção ao longo das 6 horas em que se pode estimar sua duração máxima, a partir desse intervalo, ela passa a ser gradativamente substituída pela memória de longa duração.

Caso recente vivido pelo piloto de formula 1 Felipe Massa, que não se lembra do acidente (uma mola atingiu seu capacete durante uma corrida). Outra, da observação de dois pesquisadores, Muller e Pilzecker, de que muitas memórias interferem com outras, adquiridas pouco tempo antes. (IZQUIERDO, 2002)

3.3 O APRENDIZADO, COMO ACONTECE NO CÉREBRO.

Fonseca (2009, p. 46) cita a teoria de Luria em que o cérebro opera como um organizador cognitivo complexo e super articulado em qualquer tipo de aprendizagem, tendo por fundamento o papel multicomponencial do processamento de informação, consistindo o seu trabalho em múltiplas interações neurofuncionais e sistemas abrangendo várias áreas do cérebro. Luria divide o cérebro humano em três unidades básicas:

- 1ª unidade, de alerta e atenção;
- 2ª unidade, de recepção, integração, codificação e processamento sensorial;
- 3ª unidade, de execução motora, planificação e avaliação.

Segundo o mesmo autor, cada uma dessas unidades está envolvida em todos os tipos de comportamento e de aprendizagem, sem exceção; todavia, a relatividade da contribuição de cada uma delas varia conforme o comportamento considerado, isto é, verbal ou não verbal, simbólico ou não simbólico, linguístico ou práxico, etc., a 2ª e a 3ª unidades são igualmente subdivididas em mais áreas distintas.

4 DA LEITURA ATÉ A COMPREENSÃO: O CAMINHO A SER PERCORRIDO

Trabalhar a compreensão é bastante complexo, Marchuschi (2008, pag. 228) afirma que não há uma teoria hegemônica ou que seja a mais correta e definitiva. Ler é um ato de produção e apropriação de sentido que nunca é definitivo e completo. O autor afirma que ler não é um ato de simples extração de conteúdos ou identificação de sentidos. Mas não se pode dizer que ler seja apenas uma experiência individual sobre o texto, oral ou escrito. Compreender o outro é uma aventura, e nesse terreno não há garantias absolutas ou completas.

Nesse capítulo trabalharemos leitura, compreensão, multimodalidade, textos e hipertextos. Com o objetivo de fundamentar a leitura utilizadas na nossa pesquisa.

4.1 LEITURA

Antes de adentrar na questão da leitura e também dos modelos de leitura é necessário pensar sobre o que significa o ato de ler. É difícil encontrar uma resposta única, dadas as várias abordagens que permeiam a atividade de leitura a depender da escola lingüística. Segundo Godoy (2009, pag. 30), As concepções clássicas de leitura são delineadas pelos seguintes elementos:

- a) A interação entre o leitor e o texto (processos cognitivos);
- b) Os papéis do autor ao escrever um texto (seus valores, conceitos e experiências);
- c) As características físicas do texto (mecanismos lingüísticos, ortografia, pontuação, sintaxe e estilo, aliados a sua forma e função);
- d) Contexto (considerado em abordagens mais recentes quando se pensa em leitura como a interação entre leitor, texto e contexto).

Nossa investigação analisa os itens a) e c), isto é, como funciona a interação entre o leitor e o texto nos seus processos cognitivos e como funciona essa leitura em características físicas diferentes do texto, assim vamos abordar esse aspectos teóricos da leitura e compreensão (para o item a) e texto e hipertexto (para o item b), nesse tópico trabalharemos a leitura.

Definindo Leitura:

“A leitura tradicionalmente é entendida como um processo de decodificação do código lingüístico com a finalidade de decifrar sinais gráficos e organizar as estruturas sintáticas nas sentenças. Esse conceito é tributário da concepção de leitura 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, pag. 74).

Essa concepção de leitura como decifração de signos linguísticos percebe a leitura como um processo linear e cumulativo.

Segundo Kleiman (2004, pag. 26), a leitura é uma atividade complexa devido aos múltiplos processos cognitivos utilizados pelo leitor ao construir o sentido de um texto.

Ainda o mesmo autor (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.

Assim a autora afirma que leitura é um processo interativo, devido interligar níveis de conhecimento como o linguístico, o textual e o de mundo. O leitor para compreender um texto, utiliza-se deste 3 conhecimentos.

Compreender bem um texto não é uma atividade natural nem uma herança genética; nem uma ação individual isolada do meio e da sociedade em que se vive. Não basta apenas ler para chegar a compreensão de um texto. No próximo tópico descreveremos a compreensão de textos.

4.2 COMPREENSÃO

Segundo Marcuschi (2008, pag. 232) para compreender o leitor é exigido que tenha habilidade, interação e trabalho. Na realidade, sempre que ouvimos alguém ou lemos um texto, entendemos algo, mas nem sempre essa compreensão é bem-sucedida. Compreender não é uma ação apenas linguística ou cognitiva. É muito mais uma forma de inserção no mundo e um modo de agir sobre o mundo na relação com o outro dentro de uma cultura e uma sociedade.

Ainda o mesmo autor afirma que de maneira geral, hoje podemos distribuir os modelos teóricos que tratam da compreensão em dois grandes paradigmas que podem ser por sua vez desmembrados em muitos outros subconjuntos específicos. Esses dois conjuntos poderiam ser agrupados em duas hipóteses:

- (a) Compreender é decodificar (metáfora do conduto)
- (b) Compreender é inferir (metáfora da planta baixa)

Com isso teríamos, de um lado as teorias de compreensão como decodificação, baseadas na noção de língua como código e, de outro lado, aquelas baseadas na noção de língua como atividade, tomando a compreensão como inferência ou pelo menos como processo de construção baseada numa atividade mais ampla e de base sócio-interativa.

Nosso objetivo nesse tópico é apenas conceituar a compreensão então não vamos ser mais detalhados nas duas teorias aqui citadas. No tópico 4.6 voltaremos a falar de compreensão no processamento da leitura.

4.3 TEXTO

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 (Marcuschi, 2008, pag. 72).

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, texto é uma reconstrução do mundo, uma entidade comunicativa que gera uma unidade de sentido. Ainda segundo Beaugrande em Marcuschi (2008, pag. 72) o texto é um evento comunicativo em que convergem ações linguísticas, sociais e cognitivas.

Ainda para Marcuschi (2008, pag. 77), 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.

Logo, só pode existir um texto com a condição de existir alguém para fazer sua interpretação, caso contrario não existirá sentido no texto. Porém 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, expectativas, conhecimentos linguísticos e não-linguísticos e, com isso, interpreta a informação veiculada no texto.

4.4 HIPERTEXTO

Cavalcante afirma que o termo hipertexto tem seu surgimento em meados dos anos 60 criado por Theodoro Nelson para “expressar a idéia de escrita/leitura não linear em sistema de informática.” (CAVALCANTE, 2005, pag. 164).

É definido por Marcuschi (1999, pag. 21) como “nem gênero textual nem um simples suporte de gêneros diversos, mas como um tipo de escritura. É uma forma de organização cognitiva e referencial cujos princípios constituem um conjunto de possibilidades estruturais que caracterizam ações e decisões cognitivas baseadas em séries referenciações não contínuas e não progressivas”.

Como comentado no capítulo 1, os avanços tecnológicos fazem do hipertexto um novo espaço para escrita, caracterizado pela escrita eletrônica utilizando a tecnologia como base. O hipertexto tem inúmeras características que pode facilitar a compreensão e a memória, tais como, segundo a classificação proposta por Marcuschi (2007, pag. 150):

- a) Não-linearidade: que cada texto aponta pro melhor caminho que o leitor deseja seguir, facilitando o entendimento;
- b) Volatilidade: o hipertexto não tem a mesma estabilidade dos textos, tais como livros e apostilas, e todas as escolhas são tão passageiras quanto às conexões estabelecidas por seus leitores;
- c) Topografia: não é um texto hierárquico ou em tópicos, é um lugar onde não se tem limites para escrever e desenvolver;
- d) Fragmentariedade: consiste na constante ligação de porções, em geral, breves, com sempre possíveis retornos ou fugas;
- e) Acessibilidade ilimitada: o hipertexto acessa todo tipo de fonte, sejam elas dicionários, enciclopédias, museus, obras científicas, literárias, arquitetônicas, etc. E, a princípio, não experimenta limites quanto às ligações que queria fazer;
- f) Multisemiose: este traço caracteriza-se pela possibilidade de interconectar simultaneamente a linguagem verbal e não-verbal;
- g) Interatividade: refere-se à interconexão interativa que por um lado, é propiciada pela multisemiose e pela acessibilidade ilimitada e, por outro lado, pela contínua relação de um leitor-navegador com múltiplos autores em quase sobreposição em tempo real, chegando a simular uma interação verbal face a face.

De um modo geral essas características do hipertexto fazem dele um fenômeno essencialmente virtual e descentralizado. Para a ciência cognitiva estudar os possíveis avanços para a compreensão e a memória que esse fenômeno possa trazer, passa a ser fundamental para avanço das pesquisas cognitivas.

4.5 MULTIMODALIDADE

Em geral, no estado de uso, gêneros textuais são compostos além dos caracteres alfanuméricos, por figuras, rodapés, gráficos, fotos e outros. Isso representa a presença da multimodalidade nesses eventos comunicativos.

Toda riqueza visual existente no gênero, ou seja, caixa de texto, fotos diagramação, cores, figuras, tipo de papel são chamamos de multimodalidade.

Descardesi (2002, pag. 20) citou 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. “(DESCARDESI, 2002, p. 20)

Com as novas práticas sociais condicionadas pela tecnocracia, o evento comunicativo (texto) se utiliza cada vez mais de tecnologias digitais. Pereira Jr (2009, pag. 02) afirmou que “a noção fundamental dentro do conceito de multimodalidade é aquilo que Kress (1998) cunhou de “guinada para o visual”, e a maneira como ela interage com as tecnologias eletrônicas. Parte-se do pressuposto de que a expressão material do texto é sempre importante. Textos, nesse sentido, são os objetos materiais que resultam de uma variedade de práticas representacionais que fazem uso de uma diversidade de sistemas dotados de significação.”(PEREIRA JR, 2009, pag. 02).

A disponibilidade é, portanto, um conceito-chave para esta teoria: as formas de inscrição estão intimamente ligadas à tecnologia disponível. Enquanto a internet atual, por exemplo, permite um amplo trabalho com imagens em movimento, cores, disposição, som e interatividade, um jornal do século XIX constituía-se basicamente a partir de textos alfabéticos e lay-out.

Na verdade, o condicionamento tecnológico está presente na linguagem desde que o texto passou a constituir-se enquanto tal. A possibilidade de construção de sumários e índices, na Idade Média, para citar um novo exemplo, tornando possíveis referências entre um e outro ponto do texto – que passou a constituir o suporte livro tal como hoje o conhecemos – foi fruto de uma tecnologia nova.

Da mesma maneira, a mudança do modelo de textos monomodais para o de textos multimodais, conforme se pôde observar nas últimas décadas, foi propiciado pelas novas tecnologias de digitalização, que possibilitam a escolha entre um ou outro modo de linguagem para determinada representação, de acordo com o efeito semiótico buscado. (PEREIRA JR, 2009, pag. 03).

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

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

4.6 PROCESSAMENTO DA LEITURA

Kato (2007) afirma que na área de compreensão e leitura, onde temos processos inacessíveis à observação direta, tivemos também, até recentemente, duas concepções radicalmente opostas, oposição essa que se manifesta na denominação com que elas são conhecidas hoje: hipótese ascendente (bottom-up), ou de dependente do texto e a hipótese descendente (top-down), ou dependente do leitor.

O leitor idealizado pelo modelo ascendente é aquele que analisa cuidadosamente o input visual e que sintetiza o significado das partes menores para obter o significado do todo.

O leitor idealizado pelo modelo descendente é aquele que se apóia principalmente em seus conhecimentos prévios e sua capacidade inferencial para fazer previsões sobre o que o texto dirá, utilizando os dados visuais apenas para reduzir incertezas.

Tanto no primeiro como no segundo modelo pretendem descrever os comportamentos do leitor ideal e são calcados em observações empíricas de sujeitos leitores proficientes como também ineficientes.

Existem autores, tais como Taglieber (1998) e Kato (2007), que salientam que o leitor idealizado no modelo ascendente é mais preciso na compreensão da leitura por não utilizar de conhecimentos fora do texto, como também existem autores que ressaltam que o leitor idealizado no modelo descendente é mais preciso devido ter uma base de conhecimentos prévios utilizada na compreensão do texto.

4.7 METACOGNIÇÃO

A metacognição é um conceito muito discutido. Cunhado por Flavell (DAVIS et al., 2005) os quais referem à “cognição sobre a cognição”, entendendo-se por “cognição” mais o processo de conhecimento do que os conhecimentos resultantes desse processo. Pode-se, assim, dizer que metacognição é a atividade mental por meio da qual outros processos mentais se torna alvo de reflexão, de acordo com Flavell citado em Davis (2005):

“Metacognição refere-se ao conhecimento que se tem sobre os próprios processos cognitivos e produtos ou qualquer coisa relacionada a eles, isso é o aprendizado das propriedades relevantes da informação ou dos dados.”

Ainda em Davis (2005):

“Ou ainda (e na mesma página), “metacognição” se refere, entre outras coisas, ao monitoramento ativo e à consequente regulação e orquestração desses processos em relação aos objetos cognitivos ou dados sobre os quais eles incidem usualmente a serviço de alguma meta ou objetivo concreto.”

Numa primeira classificação, Flavell e Wellman (1977) citado em Figueira (2003) consideram o meta conhecimento ou conhecimento meta cognitivo composto por dois componentes: a sensibilidade (sensitivity) e o conhecimento das variáveis da pessoa, da tarefa, das estratégias e a interação entre elas. A sensibilidade diz respeito ao conhecimento da necessidade de se utilizar ou não estratégias em tarefas ou atividades específicas. Referem-se a esse propósito as atividades induzidas, isto é, aquelas em que são dadas instruções que conduzirão à adoção de uma determinada estratégia e as atividades espontâneas, ou seja, todas as tarefas em que embora não sejam dadas indicações num determinado sentido, o sujeito deve saber o que fazer com elas, em função dos objetivos das mesmas (FIGUEIRA, 2003).

Segundo Brown (1978) citado em Ribeiro (2003), o reconhecer a dificuldade na compreensão de uma tarefa ou tornar-se consciente de que não se compreendeu algo é uma habilidade que parece distinguir os bons dos maus leitores. Os primeiros sabem avaliar as suas dificuldades e/ou ausências de conhecimento, o que lhes permite nomeadamente superá-las, recorrendo muitas vezes a inferências feitas a partir daquilo que sabem. Esta autora chama assim a atenção para a importância do conhecimento, não só sobre aquilo que se sabe, mas também sobre aquilo que não se sabe, evitando assim, o que designa de ignorância secundária não saber que não se sabe.

5 METODOLOGIA

De acordo com Cervo (2007), método é a ordem que se deve impor aos diferentes processos necessários para atingir certo fim ou um resultado desejado. Assim, será apresentado nesse capítulo a descrição da metodologia utilizada para coleta dos dados (os participantes da pesquisa e as etapas de composição do corpus), a descrição dos métodos de análise dos dados.

Metodologia Científica é um conjunto de técnicas e processos utilizados pela ciência para formular e resolver problemas de aquisição objetiva do conhecimento de maneira sistemática.

O aporte teórico apresentado nos capítulos anteriores (os estudos da cognição, memória e leitura) será utilizado para tratamento dos dados desta pesquisa.

5.1 OBJETIVO DA PESQUISA

O objetivo geral desta pesquisa é descrever como o tempo e memória, utilizados no processo de leitura, influenciam diretamente a compreensão de texto de igual grau de multimodalidade em suportes diferentes, como também o que essas variáveis (tempo e memória) influenciam diretamente a compreensão em texto de baixo grau de multimodalidade.

Como objetivos específicos temos:

- Elaborar dois experimentos com finalidades de investigar os as variáveis apresentadas;
- Analisar o grau de memorização em suportes diferentes e modalidades textuais diferentes;
- Investigar o tempo de leitura em suportes diferentes e modalidades textuais diferentes;
- Entender caminhos utilizados para a formação do conhecimento obtido pelos alunos.

5.2 DESCRIÇÃO DO EXPERIMENTO

A investigação será realizada através de dois experimentos, e cada um dos experimentos divididos em dois grupos.

Chamaremos de experimento I o experimento que abrange dois grupos de alunos onde um grupo é submetido a aulas com material impresso (grupo experimental A) e o outro grupo é submetido a aulas através de um computador com material digital (grupo experimental B).

Chamaremos de experimento II o experimento que abrange dois grupos de alunos onde um grupo é submetido a aulas com material impresso, porém mono modal (grupo experimental A) e o

outro grupo são submetidos a aulas através de um computador com material multimodal (grupo experimental B).

5.3.1 Experimento I

O experimento I que o grupo experimental A é submetido a duas aulas multimodal em material impresso e o grupo experimental B é submetido duas aulas através do computador, as aulas, tanto impressa quanto digital têm o mesmo conteúdo, o ambiente utilizado para o grupo experimental A foi uma sala de aula enquanto para o grupo experimental B foi utilizado um laboratório de informática.

Após as duas aulas submetidas para cada grupo será aplicado um teste de memorização onde constarão questões de múltipla escolha, visando medir o grau de memorização dos textos lidos nos dias anteriores (as duas primeiras aulas).

5.3.1.1 Sujeitos da Experiência I

Grupo experimental A

Grupo formado por 15 (quinze) alunos do segundo ano do curso técnico em informática integrado ao ensino médio do Instituto Federal do Tocantins, campus de Palmas-TO.

Grupo experimental B

Grupo formado por 14 (quatorze) alunos do segundo ano do curso técnico em informática integrado ao ensino médio do Instituto Federal do Tocantins, campus de Palmas-TO.

5.3.1.2 Procedimentos da Experiência I

Na elaboração das aulas foi utilizado como referência, os capítulos 2 e 3 do livro Desenvolvendo Aplicações com UML 2.0, de Ana Cristina de Melo, onde o capítulo 2, Visão Geral da UML foi utilizado para primeira aula e o capítulo 3, Diagrama de Classes foi utilizado na segunda aula.

Os grupos foram separados, onde o grupo experimental A foi levado para uma sala de aula para fazer a leitura de seus textos e o grupo experimental B foi levado para o laboratório de informática para fazer a leitura dos hipertextos.

Assim foram submetidos os grupos a dois dias de aulas para que no terceiro dia fossem submetidos a um teste de memorização. Teste que contém 10 (dez) questões de múltipla escolha que explora o conhecimento das aulas dos dias anteriores com apenas uma alternativa correta.

5.3.1.3 Ferramentas Experimentais Utilizadas

Para cada grupo foi submetido uma aula diferente, neste tópico segue a descrição de como a aula chegou a cada grupo.

5.3.1.3.1 Suporte Impresso

Cada aluno recebeu uma apostila de 10 (dez) páginas no primeiro dia e de 14 (quatorze) páginas no segundo dia. Essa apostila foi formatada num editor de texto, contendo textos, figuras, desenhos e notas de rodapé. Impressos em impressora a laser (preto e branco) em papel do formato A4 (Figura 3.0).

Cada aluno ao proceder a leitura vai registrando o momento que inicia cada tópico de leitura de acordo com o relógio da sala de aula, fazendo a anotação na própria apostila. Ao término da leitura os alunos entregam o material com todos os registros de horários das leituras.

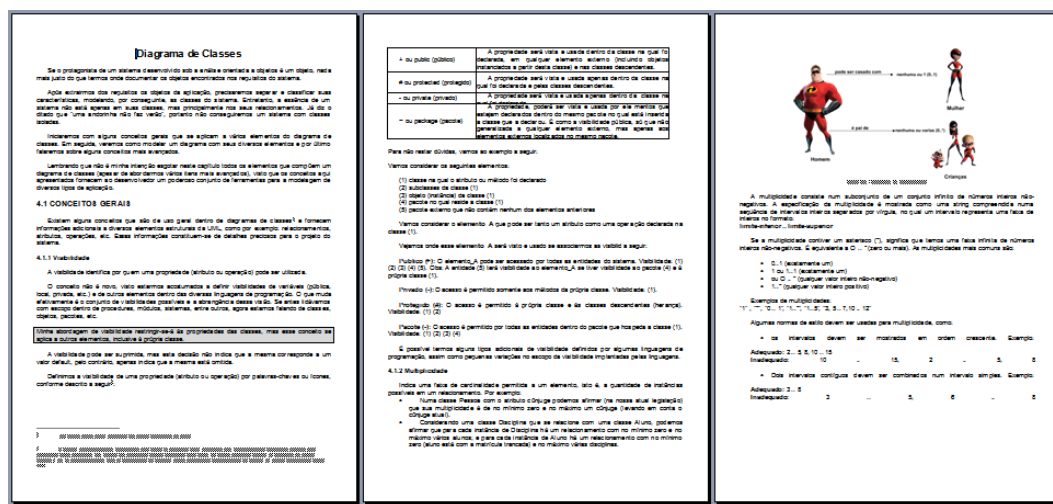


Figura 03 – Texto impresso utilizado.

5.3.1.3.2 Suporte Digital

Cada aluno utilizou um microcomputador, cada aluno acessa um sistema web, feito com tecnologia Java¹, onde após registrar o login e senha (ver figura 1), tem acesso a aula.

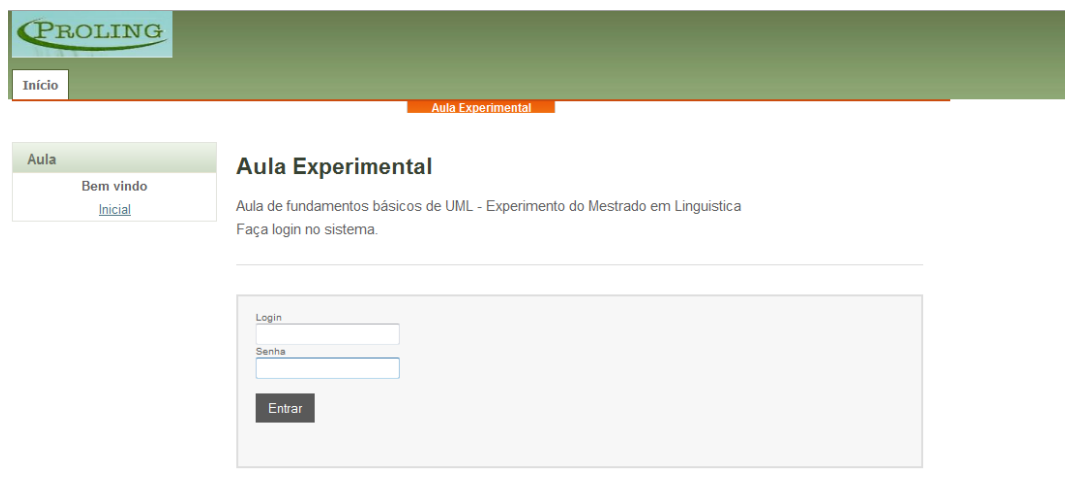


Figura 04 – Tela de acesso ao sistema web.

Após o login, o aluno tem acesso ao conteúdo (ver figura 2) e poderá ir clicando pelos tópicos até terminar o conteúdo.



Figura 05 – Tela do início da aula do primeiro dia.

O sistema web, utiliza de um banco de dados em Postgresql², onde as informações dos alunos e seus registros de tempos de leitura são armazenados.

5.3.1.3.3 Teste de memorização

¹ Anexo I segue o detalhamento da tecnologia utilizada, onde consta a descrição da tecnologia Java.

² Anexo I segue o detalhamento da tecnologia utilizada, onde consta a descrição da Banco de Dados Postgresql

O teste de memorização acontece no terceiro dia com o suporte digital. É utilizado o mesmo sistema web para a resolução do teste pelos alunos, onde com o login de acesso o sistema tem como registrar a resposta marcada por cada aluno e também o tempo levado na resolução dos testes.

O teste é formado por 10 (dez) questões de múltipla escolha com apenas uma alternativa correta. Todas as questões são sobre o conteúdo lido nas aulas dos dias anteriores do experimento.

5.3.2 Experimento II

O experimento II, onde o experimental A é submetido a uma aula mono modal em material impresso, as aulas serão no formato de um artigo apenas com texto e o grupo experimental B é submetido a duas aulas multimodais em material digital.

Assim como no experimento I, após as duas aulas submetidas para cada grupo será aplicado um teste de memorização onde constarão questões de múltipla escolha visando medir o grau de memorização dos textos lidos nos dias anteriores (as duas primeiras aulas) e também serão submetidos ao questionário de compreensão, memorização e formação do conhecimento.

5.3.2.1 Corpus da Experiência

Grupo Experimental A

Grupo formado por 20 (vinte) alunos do segundo ano do curso técnico em informática integrado ao ensino médio do Instituto Federal do Tocantins, campus de Palmas-TO.

Grupo Experimental B

Grupo formado por 20 (vinte) alunos do segundo ano do curso técnico em informática integrado ao ensino médio do Instituto Federal do Tocantins, campus de Palmas-TO.

5.3.2.2 Procedimentos da Experiência

Na elaboração das aulas digitais foi utilizado a mesma aula do experimento I, já para a elaboração do conteúdo mono modal, foi extraído texto do conteúdo da aula do experimento I,

porém em formato de artigo utilizando somente texto, ou seja, descartando as figuras e gráficos envolvidos.

Os grupos foram separados, assim o grupo experimental A foi levado para uma sala de aula para fazer a leitura de seus textos e o grupo experimental B foi levado para o laboratório de informática para fazer a leitura dos hipertextos.

Assim foram submetidos os grupos a dois dias de aulas para que no terceiro dia fossem submetidos a um teste de memorização. Teste que contém 10 (dez) questões de múltipla escolha que explora o conhecimento das aulas dos dias anteriores com apenas uma alternativa correta.

5.3.2.3 Ferramentas Experimentais Utilizadas

5.3.2.3.1 Texto com baixo grau de multimodalidade

Cada aluno recebeu um artigo de 7 (sete) páginas no primeiro dia e de 9 (nove) páginas no segundo dia. Esse artigo foi formatado num editor de texto, descartando figuras, desenhos. Impressos em impressora a laser (preto e branco) em papel do formato A4.

Cada aluno ao proceder a leitura vai registrando o momento que inicia cada tópico de leitura de acordo com o relógio da sala de aula, fazendo a anotação no próprio artigo. Ao término da leitura os alunos entregam o material com todos os registros de horários das leituras.

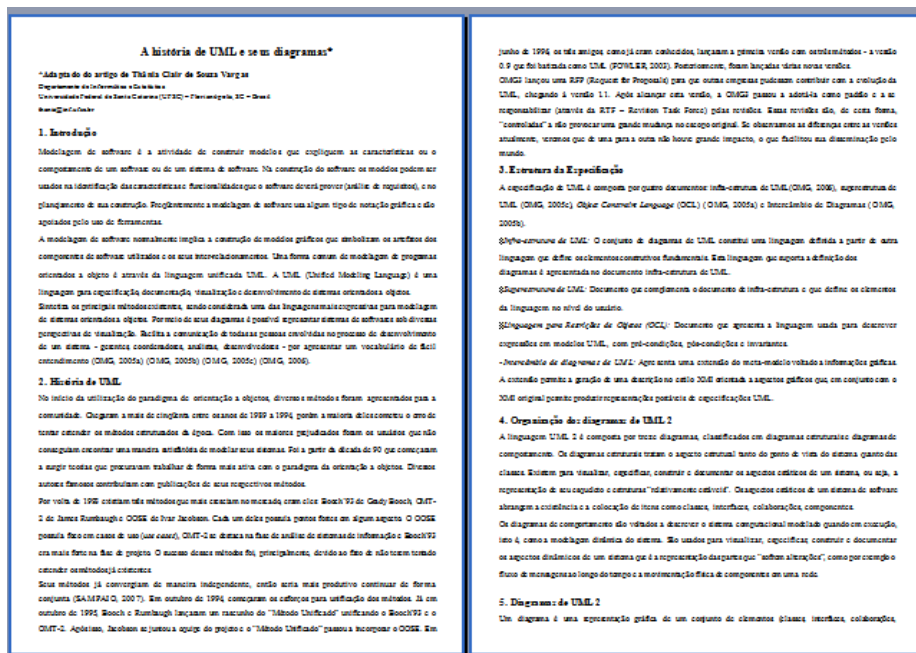


Figura 06 – Texto utilizado no experimento II

5.3.2.3.2 Hipertexto com alto grau de Multimodalidade

Idêntico ao multimodal do experimento I.

5.4 CORPUS DE ANÁLISE – CRITÉRIO DE EXCLUSÃO

Nosso corpus de análise será composto pelos integrantes do grupo de controle e do grupo objeto de estudo, onde obedecerão aos seguintes critérios de exclusão para os procedimentos de análise:

- Para análise do tempo de leitura:
 - o Será excluído do corpus o aluno que não comparecer a um dos dias do experimento;
 - o Será excluído do corpus o aluno que tiver um tempo de leitura incompatível ou muito discrepante do tempo médio de seu grupo.
- Para análise de memorização:
 - o Será excluído do corpus o aluno que não comparecer a um dos dias do experimento;
 - o Será excluído do corpus o aluno que tiver um tempo de resolução do questionário de memorização incompatível ou muito discrepante do tempo médio de resolução do grupo.
 - o Será excluído do corpus o aluno que na entrevista informar que não houve memorização e as questões certas foram mera sorte ao marcar umas das alternativas.

5.5 PROCEDIMENTO DE ANÁLISE (VERIFICAÇÃO DAS VARIÁVEIS)

É preciso salientar que o processo de leitura não está sendo visto como extração do conteúdo do texto, mas como uma construção do sentido do texto, logo então, utilizará procedimentos de análise que farão a verificação da formação do conhecimento e memorização, para essa análise da formação do conhecimento utilizaremos procedimentos de análise que tratarão os dados da variável tempo e ainda teremos procedimentos de análise que manipularam os dados da variável memorização. Sendo assim, temos dois procedimentos de análise e por último um procedimento, chamado aqui de procedimento global, onde terá o cruzamento das variáveis utilizadas nos procedimentos anteriores.

5.5.1 Procedimento de análise da variável tempo

A variável tempo, nesta etapa do procedimento de análise da pesquisa será utilizada como uma variável quantitativa, ou seja, uma variável mensurável onde utilizaremos alguns recursos dos métodos quantitativos de pesquisa para demonstrar o tempo de leitura do corpus de análise.

Será feita a distribuição de frequência dos tempos de leitura dos sujeitos da experiência, em cada dia de aplicação do experimento (primeiro e segundo dia).

Após a distribuição de frequência serão calculadas as medidas:

- A média aritmética, que é a soma de todo o tempo de leitura divididos pela quantidade de leitores;
- A moda, que é o valor mais freqüente no conjunto de dados registrado (tempos) e;
- A mediana que é a medida central do conjunto de valores da variável tempo assumiu no corpus da pesquisa.

O tempo gasto nas leituras nos dois primeiros dias do experimento, após a distribuição de frequência, será representado em gráfico do tipo histograma, que é um gráfico que relaciona a frequência do evento (tempo de leitura) na amostra com o tipo de evento no qual serão agrupados os intervalos onde o evento aconteceu.

5.5.2 Procedimento de análise da variável de memorização

Nesta pesquisa, determinaremos que quanto maior o número de acertos no teste de memorização maior será o grau de memorização do aluno submetido ao procedimento. Sendo assim, teremos também uma variável quantitativa, onde faremos a distribuição de frequência dos graus obtidos e, também, serão calculados as médias aritméticas moda e mediana das quantidades de acertos dos alunos. A utilização do gráfico do tipo histograma, aqui será utilizado para representar a distribuição dos alunos.

5.5.4 Procedimento de análise global

O procedimento de análise global, será a análise cruzando todas as variáveis, observaremos as categorias cognitivas formadas em função do tempo de leitura, em função suporte para a leitura e em função da modalidade do texto.

Serão observadas as teorias cognitivas para demonstrar o conhecimento adquirido pelo leitor em cada tipo de variável analisada.

6 ANÁLISE DOS DADOS

Esse capítulo tem como objetivo descrever, analisar e tratar os dados coletados. A análise dos dados será dividida em dois momentos, um analisará os dados do experimento I e outro analisará os dados do experimento II.

6.1 EXPERIMENTO I

O Experimento I que busca investigar as hipóteses 1 e 3 apresentadas no capítulo 1.

6.1.1 Investigando a variável tempo

Investigando a hipótese I, onde a variável investigada é o tempo e a variável manipulada é o suporte (digital ou impresso). Na tabela abaixo segue os dados de tempos de leitura nos suportes diferentes do primeiro dia de leitura do experimento I.

Tabela 02 – Tempos de leitura do primeiro dia de leitura do experimento I.

Suporte	Total de Alunos	Média	Moda	Mediana
Impresso	14	30 minutos	27 minutos	30 minutos
Digital	15	28 minutos	30 minutos	28 minutos

A seguir a representação gráfica da tabela 2.

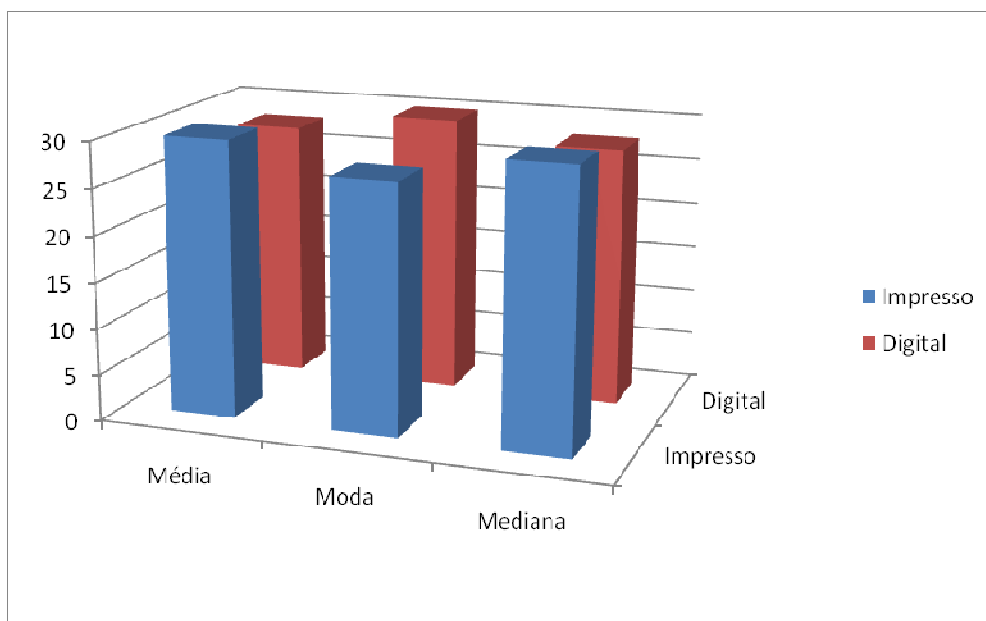


Figura 07 – Gráfico da distribuição da Média, Moda e Mediana no primeiro dia do experimento I

O fato de a média ser igual à mediana caracteriza que a distribuição é simétrica, ou seja a média é exatamente o dado central que divide os maiores tempos dos menores tempos. E também, para o suporte impresso a moda estar menor que a média, caracteriza que mais alunos fizeram a leitura mais rápido que a média da turma, uma vez no digital, mas alunos fizeram a leitura mais demorados que a média da turma, já que a moda foi maior que a média. Devido à proximidade de resultados entre a média dos dois suportes, podemos caracterizar que para análise da variável tempo, o suporte diferente, nesse primeiro dia de leitura, pouco influenciou, já analisando a moda, que sua definição é a frequência de resultados que mais se repete, vemos que a moda para o grupo que fez a leitura do texto impresso, leu mais rapidamente, contrariando a hipótese I, que afirma que a leitura em textos com suporte digital é mais rápido que a leitura em textos com suporte impresso.

Para entendermos o que levou esse resultado vamos analisar a navegabilidade dos alunos, ou seja, o percurso utilizado na leitura dos textos. No grupo experimental A, a leitura é linear, característica do texto, então concluímos que todos os alunos fizeram o mesmo caminho na leitura, já o grupo experimental B, fez uma leitura não-linear, que Marcuschi (2007, pag. 150) afirma que uma das características principais do hipertexto é a “não-linearidade: que aponta para a flexibilidade desenvolvida na forma de ligações permitidas/sugeridas entre nós que constituem redes que permitem a elaboração de vias navegáveis, a não-linearidade é tida como a característica principal do hipertexto” (MARCUSCHI, 2007, pag. 150), assim a justificativa pela a moda do grupo experimental B ser mais alta que do grupo experimental A é o fato de os leitores do experimento B não terem um navegabilidade já definida apenas sugerida.

Para representação da navegabilidade sugerida, segue a legenda abaixo, para auxiliar o seu entendimento:

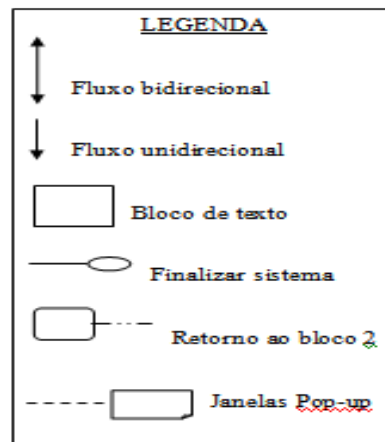


Figura 08 – Legendas para representação da navegabilidade no hipertexto.

Também é necessário identificar cada tópico (subtítulo) do texto utilizado no hipertexto com um número, para isso segue a lista abaixo:

Tabela 03 – 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

Agora abaixo segue a navegabilidade sugerida, representada em fluxograma.

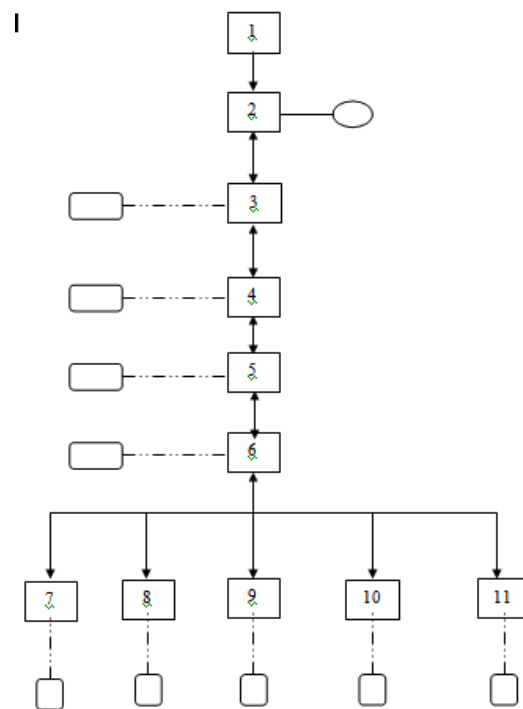


Figura 09 – Navegabilidade sugerida do hipertexto utilizado no Experimento I.

Segue abaixo navegabilidade do aluno 1, que teve seu tempo de leitura exatamente igual a moda, 27 minutos e a navegabilidade sugerida pelo hipertexto do experimento 1.

Tabela 04 – Navegabilidade do aluno 1

Navegabilidade nos tópicos (ver tabela 3)																	
Aluno 1	1	2	3	4	5	6	7	6	8	6	9	6	10	6	5	6	11
Navegabilidade Sugerida	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11		

Para melhor visualização, vamos utilizar o gráfico de linhas comparando a navegabilidade do aluno 1 e a navegabilidade sugerida abaixo.

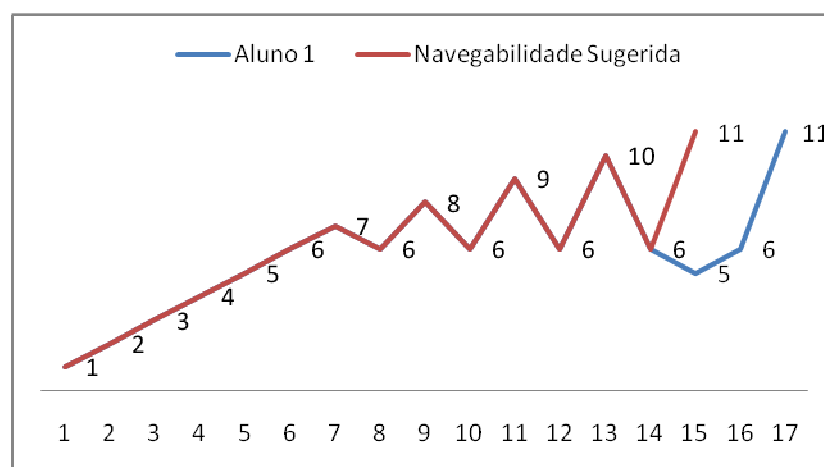


Figura 10 – Gráfico de navegabilidade do Aluno 1 x Navegabilidade sugerida.

Na representação gráfica, a comparação fica bem definida, e notamos que o Aluno 1 fez do uso da característica de não-linearidade, e seguiu uma navegabilidade diferente da navegabilidade sugerida mesmo que apenas nos últimos pontos. Notem que para a navegabilidade sugerida é necessário apenas 15 acessos (cliques) já o aluno 1 utilizou de 17 acessos (cliques).

Segue abaixo a representação da navegabilidade do aluno 2 em tabela e gráfico de linha.

Tabela 05 – Navegabilidade aluno 2

Navegabilidade nos tópicos (ver tabela 3)																				
Aluno 2	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	1	4	5	6	9
Navegabilidade Sugerida	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11					

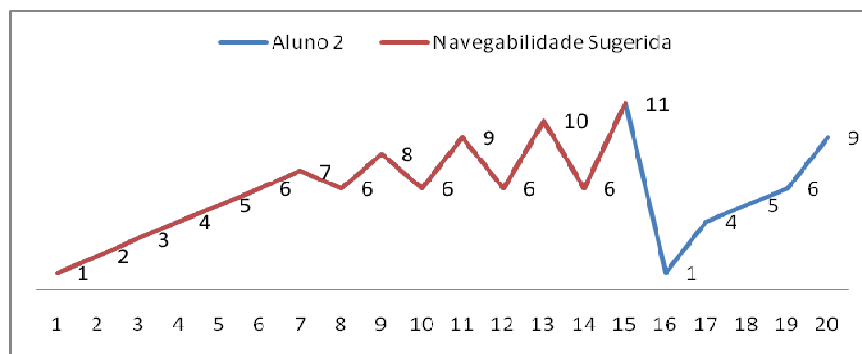


Figura 11 – Gráfico de navegabilidade do Aluno 2 x Navegabilidade sugerida.

Notamos na figura 12 que o aluno 2 utilizou que 5 acessos a mais para concluir sua leitura. Notemos que o aluno 3 já fez 6 acessos (cliques) a mais que o necessário na leitura sugerida.

Tabela 06 – Navegabilidade do aluno 3

Navegabilidade nos tópicos (ver tabela 3)																					
Aluno 3	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11	5	6	9	6	5	4
Navegabilidade Sugerida	1	2	3	4	5	6	7	6	8	6	9	6	10	6	11						

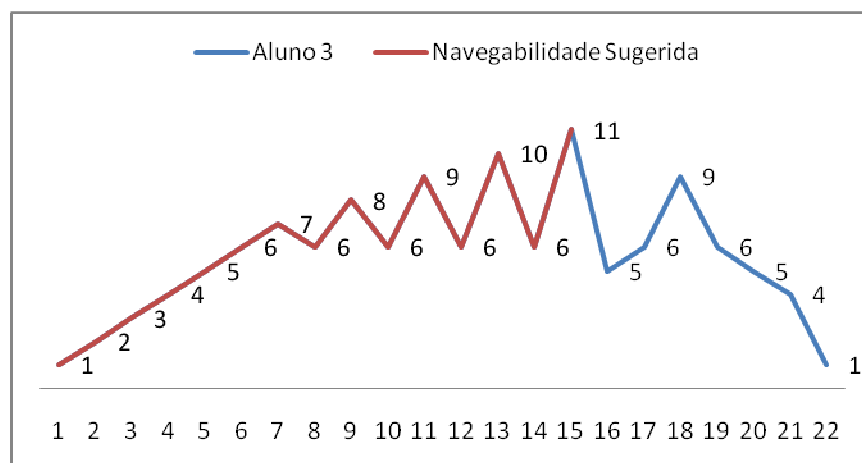


Figura 12 – Gráfico de navegabilidade do Aluno 2 x Navegabilidade sugerida.

De um modo geral, mesmo passando para o leitor (alunos) o controle cognitivo e informacional do hipertexto, o leitor segue a navegabilidade sugerida na maioria das vezes, apenas fazendo retornos para concluir novos conceitos que esta lendo.

Podemos concluir até então que a leitura do hipertexto pode ser mais lenta, como os dados representados pela moda confirmaram, mais o leitor faz os elos e as ligações necessárias para a formação de conhecimento que ele deseja. A releitura de alguns tópicos, como foi detectada nos 3 alunos, demonstra a consolidação das novas categorias formadas, no caso o conhecimento da UML que o texto ensina.

Para o segundo dia de leitura tivemos os seguintes resultados:

Tabela 07 – Tempos de leitura do segundo dia de leitura do experimento I.

Suporte	Total de Alunos	Média	Moda	Mediana
Impresso	10	36 minutos	31 minutos	33 minutos
Digital	12	45 minutos	43 minutos	43 minutos

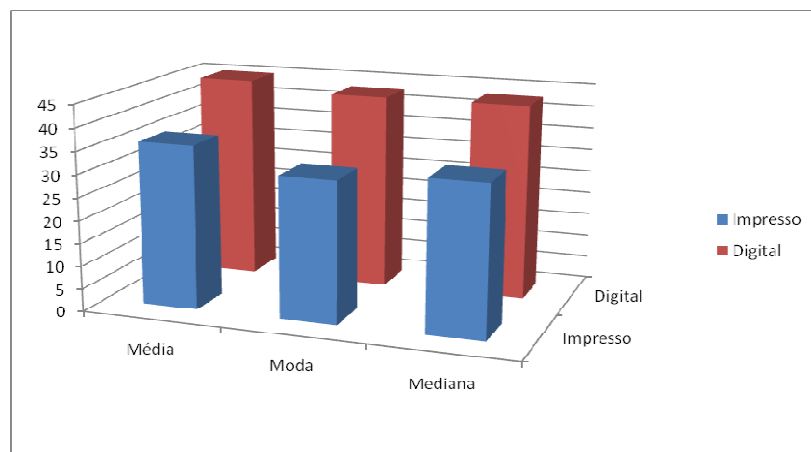


Figura 13 – Gráfico da distribuição da Média, Moda e Mediana no segundo dia do experimento I.

Analisando o tempo de leitura gasto no segundo dia de leitura, notamos que a distribuição dos dados coletados não é assimétrica nem para o impresso nem para o digital, já que as médias são diferentes das medianas, mas vale observar que tanto a média quanto a mediana os tempos foram bem maiores no suporte digital. Já para a moda, que é a representação dos tempos que mais se repetem, notamos que no digital o tempo gasto para leitura foi bem maior do que o tempo gasto para leitura no impresso. Esta comprovação nega a hipótese 1, onde afirmava que a leitura em suporte digital era de menor tempo do que a leitura em suporte impresso.

Para uma avaliação cognitiva de como foi feita a leitura, selecionamos 3 alunos para grupo experimental B para acompanhar a navegação dos alunos no tempos, tais alunos selecionados por terem seus tempos de leitura igual ou próximo o da moda.

Para melhor entendimento da navegabilidade vamos utilizar da mesma padronização de representação de navegabilidade utilizada para o primeiro dia.

Segue as lista de tópicos do conteúdo do segundo dia:

Tabela 08 – Lista de blocos do hipertexto 2

Lista de bloco de texto do hipertexto 2	
1	<i>Login</i> e senha (início)
2	Boas vindas – Registro do início da navegabilidade
3	Diagrama de classes
4	Conceitos gerais
5	Visibilidade
6	Multiplicidade
7	Criando diagramas de classes
8	Finalização – Agradecimentos
9	Atributos e operações
10	Relacionamentos
11	Associação
12	Generalização
13	Dependência
14	Agregação
15	Composição

Tabela 09 – Lista de janelas popup do hipertexto 2

Lista de janelas <i>popup</i> do hipertexto 2	
1	Ferramentas da UML
2	Normas de estilo
3	Delphi

Segue abaixo a navegabilidade sugerida pelo hipertexto

.

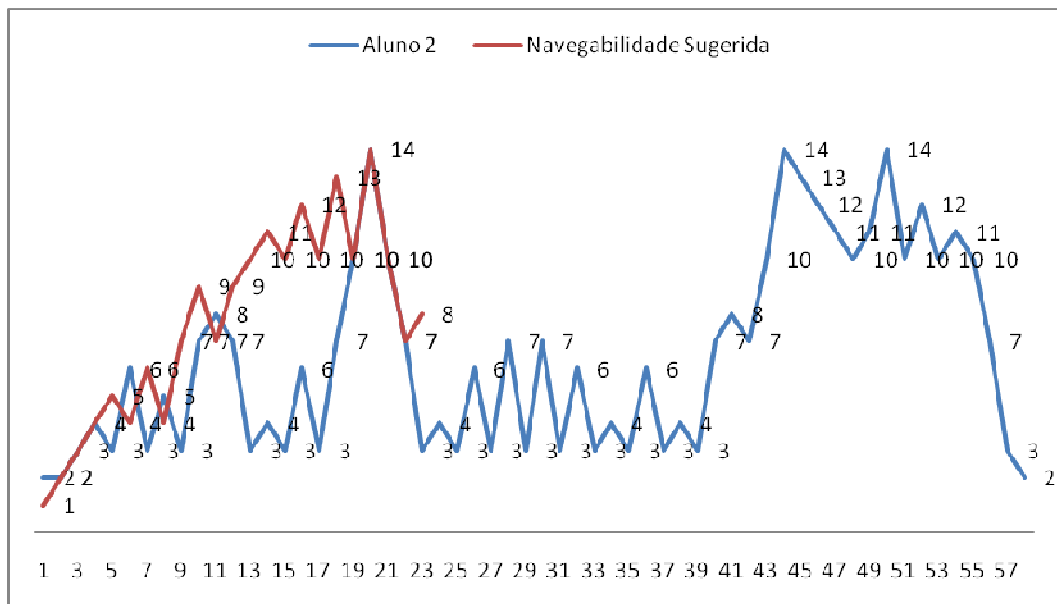


Figura 16 – Gráfico da navegabilidade do Aluno 2 x Navegabilidade sugerida.

No gráfico aqui representado o aluno 2 faz mas de 30 acessos (cliques) do que o sugerido para concluir sua leitura.

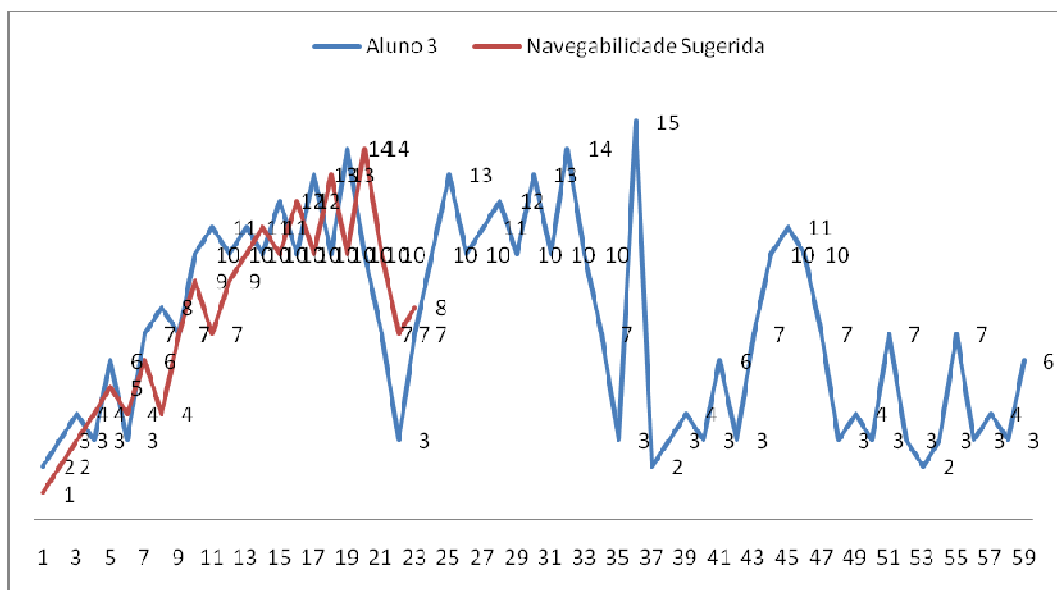


Figura 17 – Gráfico da navegabilidade do Aluno 3 x Navegabilidade sugerida

Assim como o aluno 2, o aluno 3 faz mais que 30 acessos a mais que o sugerido para concluir sua leitura.

6.1.2 Investigando a variável grau de memorização

O questionário de memorização no qual os alunos foram submetidos no terceiro dia foi todo executado eletronicamente, ou seja, foi um questionário digital, composto por perguntas objetivas com 5 alternativas, sendo apenas uma correta. Com a participação 16 alunos, 8 do grupo do experimento A e 8 do grupo do experimento B. A figura abaixo mostra uma questão no questionário.

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

Figura 18 – Questão do questionário de memorização.

Analisando as respostas dadas pelo grupo de alunos, tivemos a tabela abaixo como resultado da tabulação.

Tabela 10 – Tabulação das respostas do questionário de memorização.

Questões	Grupo Experimental A		Grupo Experimental B	
	Quantidade de alunos que acertaram	Quantidade de alunos que erraram	Quantidade de alunos que acertaram	Quantidade de alunos que erraram
1	6	2	2	6
2	1	7	1	7
3	5	3	7	1
4	1	7	1	7
5	0	8	0	8
6	7	1	6	2
7	8	0	7	1
8	8	0	6	2
9	7	1	5	3
10	2	6	1	7

Totais	45	35	36	44
Percentual	56,25%	43,75%	45,00%	55,00%

Conforme a tabela 10 podemos constatar que o resultado do grupo de experimental A, o suporte impresso, só não foi melhor na questão número 3, que apresentou menor quantidade de acertos, e nas questões números 2,4 e 5 que são as questões são de nível maior de aprendizado. É possível verificar também que os grupos tiveram graus de acertos baixos e iguais.

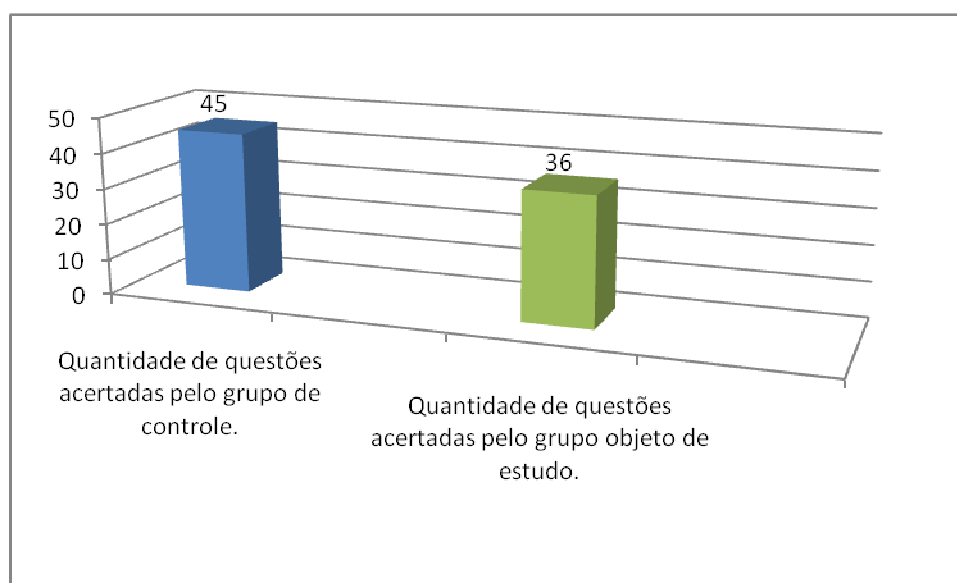


Figura 19 – Gráfico da quantidade de questões acertadas por cada grupo do experimento I.

A representação gráfica demonstra em termos de acertos o grupo de controle na soma de todos os integrantes acertaram 45 questões das 80 (10 questões para cada um dos 8 alunos) possíveis, dando um percentual de 56,25%, enquanto o grupo objeto de estudo acertou apenas 36 questões das 80 possíveis, dando um percentual de 45%.

Observando agora os resultados por questão, representaremos a questão, e os gráficos dos resultados de cada grupo experimental.

Tabela 11 – Resultados do questionário de memorização para questão 1.

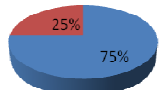
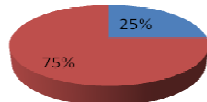
Qual é a versão atual utilizada na UML? Selecione uma opção: a) 1.1 b) 2.0 c) 3.0 d) 2009 e) 1.2	Grupo Experimental A Questão 1 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram  <table><thead><tr><th>Resultado</th><th>Porcentagem</th></tr></thead><tbody><tr><td>Quantidade de alunos que acertaram</td><td>75%</td></tr><tr><td>Quantidade de alunos que erraram</td><td>25%</td></tr></tbody></table>	Resultado	Porcentagem	Quantidade de alunos que acertaram	75%	Quantidade de alunos que erraram	25%	Grupo Experimental B Questão 1 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram  <table><thead><tr><th>Resultado</th><th>Porcentagem</th></tr></thead><tbody><tr><td>Quantidade de alunos que acertaram</td><td>25%</td></tr><tr><td>Quantidade de alunos que erraram</td><td>75%</td></tr></tbody></table>	Resultado	Porcentagem	Quantidade de alunos que acertaram	25%	Quantidade de alunos que erraram	75%
Resultado	Porcentagem													
Quantidade de alunos que acertaram	75%													
Quantidade de alunos que erraram	25%													
Resultado	Porcentagem													
Quantidade de alunos que acertaram	25%													
Quantidade de alunos que erraram	75%													

Tabela 12 – Resultados do questionário de memorização para questão 2.

Representar 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 Sequência
- c) de Toda UML
- d) Diagrama de Caso de Uso (Use Case)
- e) Diagrama de Componentes

Grupo Experimental A

Questão 2

- Quantidade de alunos que acertaram
- Quantidade de alunos que erraram

Resultado	Porcentagem
Quantidade de alunos que acertaram	12%
Quantidade de alunos que erraram	88%

Grupo Experimental B

Questão 2

- Quantidade de alunos que acertaram
- Quantidade de alunos que erraram

Resultado	Porcentagem
Quantidade de alunos que acertaram	12%
Quantidade de alunos que erraram	88%

Tabela 13 – Resultados do questionário de memorização para questão 3.

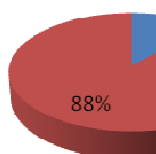
Relacionamentos: realizam a ligação, entre si, dos elementos do modelo. São eles: Selecione uma opção: a) dependência, associação, generalização e realização. b) dependência, associação, generalização e maturação. c) dependência, associação, maturação e realização. d) dependência, associação, maturação e independência. e) dependência, associação e desassociação	Grupo Experimental A Questão 3 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram 38% 62%	Grupo Experimental B Questão 3 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram 13% 87%
--	--	--

Tabela 14 – Resultados do questionário de memorização para questão 4.

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) Publica c) Protegida d) Explicita e) Pacote ☐

Grupo
Experimento
Questão

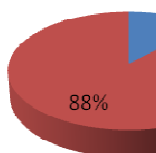
- Quantidade de
acertaram
- Quantidade de
erraram



☐

Grupo
Experimento
Questão

- Quantidade de
acertaram
- Quantidade de
erraram



☐ ☐

☐

Tabela 15 – Resultados do questionário de memorização para questão 5.

A propriedade será vista e usada apenas dentro da classe na qual foi declarada. É um tipo de visibilidade:

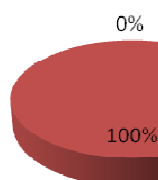
Selecione uma opção:

- a) Privada
- b) Publica
- c) Protegida
- d) Explicita
- e)

Pacote ☐

**Grup
Experime
Questâ**

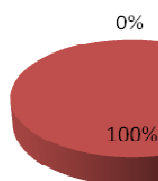
■ Quantidade de acertaram
■ Quantidade de erraram



☐

**Grup
Experime
Questâ**

■ Quantidade de acertaram
■ Quantidade de erraram



☐
☐
☐

Tabela 16 – Resultados do questionário de memorização para questão 6.

Sobre multiplicidade. Sabendo que um homem pode se casar apenas com uma mulher ou ser solteiro, porem 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 0..*	
Grupo Experimental A Questão 6 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram 13% 87%	

Grupo
Experimento
Questão

- Quantidade de
acertaram
- Quantidade de
erraram

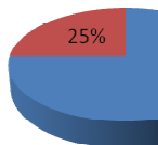


Tabela 17 – Resultados do questionário de memorização para questão 7.

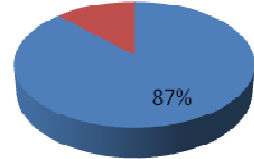
Observe a figura:												
<table><tr><th>Aluno</th></tr><tr><td>Matricula: string</td></tr><tr><td>Nome: string</td></tr><tr><td>DataNascimento: date</td></tr><tr><td>DataMatricula: date</td></tr><tr><td>ValorMensalidade: Real</td></tr><tr><td>MatricularAluno()</td></tr><tr><td>ReajustarMensalidade()</td></tr><tr><td>Trancar()</td></tr><tr><td>EmitirBoleto()</td></tr></table>	Aluno	Matricula: string	Nome: string	DataNascimento: date	DataMatricula: date	ValorMensalidade: Real	MatricularAluno()	ReajustarMensalidade()	Trancar()	EmitirBoleto()	Grupo Experimental A Questão 7 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram 0%  100%	Grupo Experimental B Questão 7 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram 13%  87%
Aluno												
Matricula: string												
Nome: string												
DataNascimento: date												
DataMatricula: date												
ValorMensalidade: Real												
MatricularAluno()												
ReajustarMensalidade()												
Trancar()												
EmitirBoleto()												
Qual o nome da Classe representada?												
Selecione uma opção:												
a) Data matricula												
b) Matricula												
c) nome												
d) Data nascimento												
e) Aluno												

Tabela 18 – Resultados do questionário de memorização para questão 8.

Observe a figura

Aluno
Matricula: string
Nome: string
DataNascimento: date
DataMatricula: date
ValorMensalidade: Real
MatricularAluno()
ReajustarMensalidade()
Trancar()
EmitirBoleto()

Qual das propriedades abaixo não pertencem a classe aluno?

Selecione uma opção:

a) Data matricula

b) Matricula

c) nome

d) Data nascimento

e) Aluno

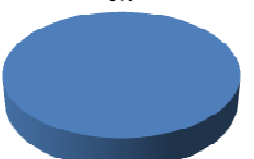
Grupo Experimental A

Questão 8

■ Quantidade de alunos que acertaram

■ Quantidade de alunos que erraram

0%



100%

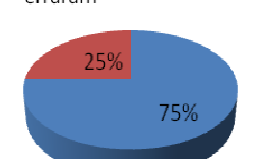
Grupo Experimental B

Questão 8

■ Quantidade de alunos que acertaram

■ Quantidade de alunos que erraram

25%



75%

Tabela 19 – Resultados do questionário de memorização para questão 9.

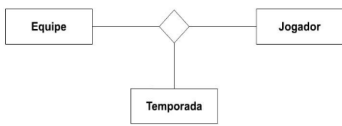
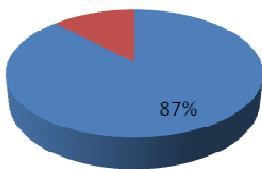
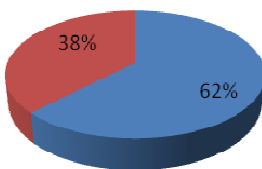
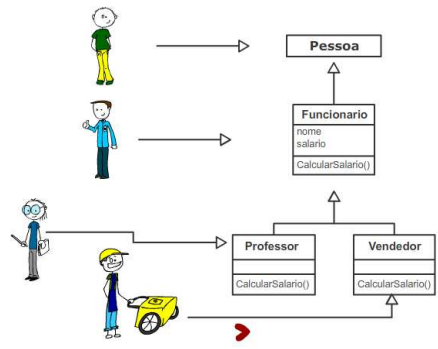
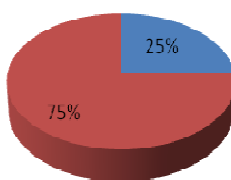
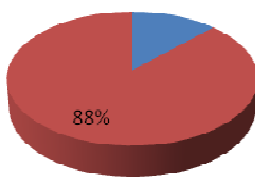
Observe a figura:  A figura é uma representação de que tipo de associação? Selecione uma opção: - Binária - Direta - pré-requisito - Ternária - auto associação	Grupo Experimental A Questão 9 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram  87% 13%	Grupo Experimental B Questão 9 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram  62% 38%
---	---	---

Tabela 20 – Resultados do questionário de memorização para questão 10.

Observe a figura  É qual tipo de relacionamento? Selecione uma opção: - Associação - Dependência - Agregação - Generalização - Composição	Grupo Experimental A Questão 10 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram  25% 75%	Grupo Experimental B Questão 10 ■ Quantidade de alunos que acertaram ■ Quantidade de alunos que erraram  12% 88%
--	--	--

Observando o resultado por questão, constatamos que realmente o grupo experimental A, manteve-se o em 9 das 10 questões com resultados iguais ou melhores que o grupo experimental B. Apenas a questão 3, o grupo do hipertexto obteve melhor resultado.

6.2 EXPERIMENTO II

O Experimento II que busca investigar as hipóteses 2 e 4, apresentadas no capítulo 1.

6.2.1 Investigando a variável tempo

Para o experimento II tivemos os seguintes resultados para média, mediana e moda dos tempos de leitura do grupo experimental A e do grupo experimental B. Abaixo segue a tabela que representa os resultados obtidos nos tempos de leitura.

Tabela 21 – Tabela de tempos de leitura do experimento II

	Grupo Experimental A	Grupo Experimental B
Média	22 minutos	30 minutos
Mediana	22 minutos	29 minutos
Moda	22 minutos	29 minutos

Para melhor ilustração dos resultados, segue abaixo os gráficos dos resultados obtidos:

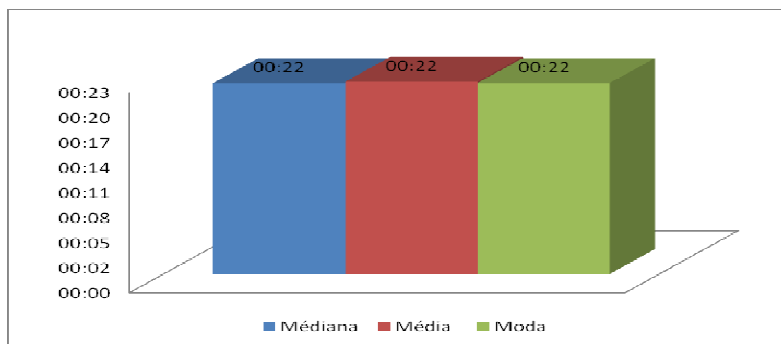


Figura 20 – Resultados dos tempos de leitura do experimento II – Grupo experimental A.

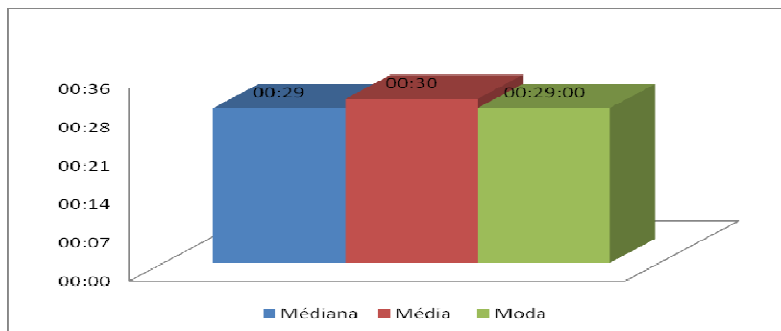


Figura 21 – Resultado dos tempos de leitura do experimento II – Grupo experimental B.

Temos nesses resultados um tempo gasto para leitura muito maior no grupo experimental B, vimos no experimento I na análise do segundo dia que o resultados também foram maiores, só que a diferença entre os resultados de Moda foi de 7 minutos, ou seja, aproximadamente 34% de tempo a mais na leitura. Isso se deve porque o texto utilizado pelo grupo experimental A é de baixo grau de multimodalidade, não contendo imagens ou notas de rodapé. Fazendo com que o leitor executasse uma leitura totalmente linear e mais rápida. Assim vamos fazer uma análise da navegabilidade realizada por 3 alunos que tiveram resultado igual a moda do grupo experimental B.

Abaixo segue lista do conteúdo do hipertexto utilizado.

Tabela 22 – Lista de blocos do hipertexto do experimento II

Lista de bloco de texto no hipertexto 3	
1	<i>Login</i> 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 23 – 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

A navegabilidade sugerida pelo hipertexto do experimento II é representada na figura abaixo:

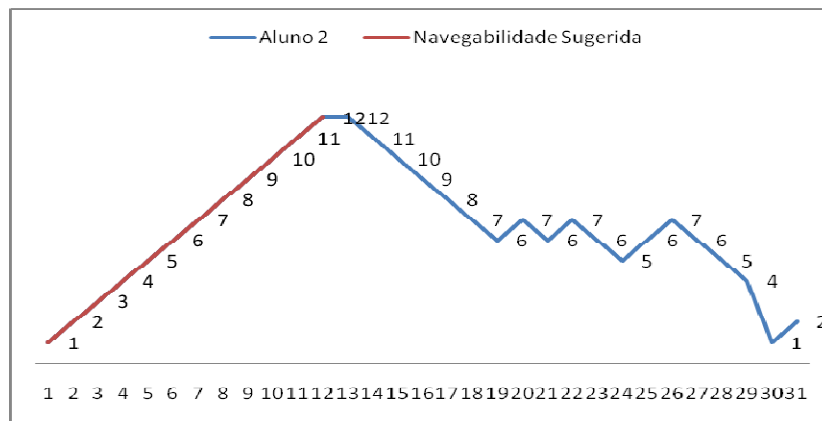


Figura 24 – Gráfico de navegabilidade Aluno 2 no experimento II.

É claro, observando a figura 22, que o tempo gasto na leitura é maior, já que os leitores fazem retorno a leituras anteriores, para melhor obter o conteúdo do texto na memória.

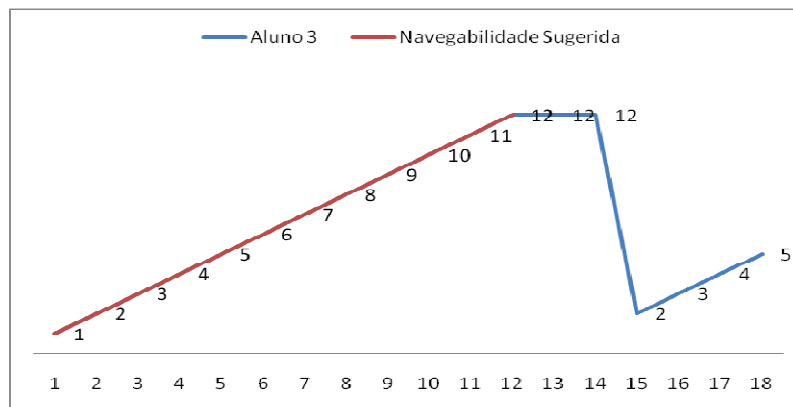


Figura 25 – Gráfico de navegabilidade aluno 3 no experimento II.

Observamos que a quantidade de acessos (cliques) ao conteúdo é muito maior que o necessário para percorrer o hipertexto todo, ou seja, os alunos fizeram releitura para buscar a compreensão, assim justificando os tempos maiores nas leituras em comparação ao grupo experimental A.

6.2.2 Investigando a variável memória

Para o questionário de memorização conseguimos os seguintes resultados para o experimento II.

Tabela 24 – Resultado do questionário de memorização

Alunos	Questões	1	2	3	4	5	6	7	8	9	10	Total	%
Grupo A	Acertos	11	4	11	3	4	11	17	13	13	12	99	49,5
	Erros	9	16	9	17	16	9	3	7	7	8	101	50,5
Grupo B	Acertos	16	8	15	4	10	11	18	14	12	10	118	59
	Erros	4	12	5	16	10	9	2	6	8	10	82	41

Notamos que a diferença nos acertos em relação ao experimento anterior é bem grande, e o grupo experimental B obteve 9,5% maior acertos que o grupo experimental A.

Esse resultado indica que o uso do hipertexto com grau de multimodalidade textual maior que o texto, facilita a memorização dos leitores. De acordo com Marcuschi (2008, pag. 230):

“Compreender bem um texto não é uma atividade natural nem uma herança genética; nem uma ação individual isolada do meio e da sociedade em que se vive. Compreender exige habilidade, interação e trabalho. Na realidade, sempre que ouvimos alguém ou lemos um texto, entendemos algo, mas nem sempre essa compreensão é bem sucedida.”(MARCUSCHI, 2008, pag. 230).

Essa afirmação justifica o melhor resultado na memorização dos alunos, já que o hipertexto o hipertexto tem como uma das suas características centrais a interatividade, que refere-se a interconexão interativa que é propiciada pela multissemiose e pela acessibilidade ilimitada.

CONSIDERAÇÕES FINAIS

Esta pesquisa trata-se de uma pesquisa 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 grau de multimodalidade e a forma de apresentação (suporte impresso e digital).

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 grau de multimodalidade presente no texto.

Assim pudemos constatar os seguintes resultados a partir da pesquisa:

a) No experimento I: com os dois grupos experimentais com texto com o mesmo grau de multimodalidade com alteração somente no suporte (digital e impresso).

Para a variável tempo: os tempos de leitura foram maiores no suporte digital (Grupo experimental B), que se justifica devido a maior interação e caminhos seguidos pelos leitores no hipertexto, relendo varias vezes em busca de melhor compreensão.

Para a variável grau de memorização: os resultados foram muito próximos, já que os textos tinham graus de multimodalidade iguais, notamos então que para a memorização o suporte não altera os resultados obtidos.

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

Para a variável tempo: os tempos de leitura foram maiores no suporte digital (Grupo experimental B), que se justifica devido a maior interação e caminhos seguidos pelos leitores no hipertexto, relendo varias vezes em busca de melhor compreensão.

Para a variável grau de memorização: os resultados foram diferentes, e melhores para o grupo experimental B, isso é justificado isto confirma que o grau de multimodalidade quando maior no texto (ou hipertexto) favorece a memorização e a formação de conhecimento.

Entendemos que a presença de multimodalidade em maior ou menor grau nos textos nos dois suportes é que 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 memorização.

Entendemos também que a utilização do suporte digital (hipertexto) os tempos de leitura são de maior tamanho devido as características do hipertexto, de não-linearidade e de interatividade.

Voltando as hipóteses levantadas no capítulo 1, apenas temos a confirmação da quarta hipóteses e a negação das outras três.

Confirma-se que a memorização tem maior grau em textos com suporte digitais e com maior grau de multimodalidade.

REFERÊNCIAS

BARDINI, Thierry. "Hypertext." The Blackwell Guide to the Philosophy of Computing and Information. Floridi, Luciano (Ed). Blackwell Publishing, 2003. Blackwell Reference Online. 07 April 2009 <http://www.blackwellreference.com/subscriber/tocnode?Id=g9780631229193_chunk_g978063122919321>

CARVALHO, Alex et al. Aprendendo Metodologia Científica. São Paulo: O Nome da Rosa, 2000, pp. 11-69.

CERVO, Amado Luiz, BERVIAN, Pedro Alcino, SILVA, Roberto, Metodologia científica, São Paulo: Pearson Prentice Hall, 2007. 6 ed.

COUCEIRO-FIGUEIRA, Ana Paula, Metacognição e seus contornos, Revista Ibero americana de Educación, 2003, Disponível em <http://www.rieoei.org/deloslectores/446Couceiro.pdf>.

DAVIS, NUNES, NUNES, Claudia, Marina, César, Meta cognição e sucesso escolar: articulando teoria e prática. Cadernos de Pesquisa, v. 35, n. 125, maio/agosto de 2005. Disponível em <http://www.scielo.br/pdf/cp/v35n125/a1135125.pdf>.

FELTES, H. P. M.; MACEDO, A. C. P.; FARIAS E. M. P. (Orgs.). Cognição e Linguística: explorando territórios, mapeamentos e percursos. 1. Ed., 2008.

FLAVELL, J. H. Cognitive development. New Jersey: Prentice Hall; Englewood Cliffs, 1985.

FONSECA, Vitor da. Cognição, neuropsicológica e aprendizagem: abordagem neuropsicológica e psicopedagógica, Petrópolis, Vozes, 2009.

GERVAIS, Bertrand. "Is There a Text on This Screen? Reading in an Era of Hypertextuality." A Companion to Digital Literary Studies. Siemens, Ray and Susan Schreibman (Eds). Blackwell Publishing, 2007. Blackwell Reference Online. 07 April 2009 <http://www.blackwellreference.com/subscriber/tocnode?Id=g9781405148641_chunk_g978140514864111>

IZQUIERDO, Ivan, Memória, Porto Alegre: Artmed, 2002.

KATO, Mary Aizawa, O aprendizado da leitura, 6ª Edição, São Paulo: Martins Fontes, 2007.

KINTSCH, Walter and Katherine A. Rawson. "Comprehension." The Science of Reading: A Handbook. Snowling, Margaret J. and Charles Hulme (Eds). Blackwell Publishing, 2005. Blackwell Reference Online. 07 April 2009 <http://www.blackwellreference.com/subscriber/tocnode?id=g9781405114882_chunk_g978140511488217>

LOMBROSIO, Paul. Aprendizado e Memória. Revista Brasileira de Psiquiatria, Volume 26 Nº. 3 - Setembro 2004. Disponível em http://www.psiquiatriageral.com.br/cerebro/aprendizado_memoria.htm. Acessado em 01/02/2020.

LYONS, John, Língua (gem) e Lingüística uma introdução, Rio de Janeiro: LTC, 1987.

MARCUSCHI, Luiz Antônio. Cognição, linguagem e práticas interacionais, Rio de Janeiro: Lucerna, 2007.

MATURANA, Humberto R., Cognição, ciência e vida cotidiana, organização e tradução Cristina Magro, Victor Paredes, Belo Horizonte: Ed. UFMG, 2001.

MEDEIROS, João Bosco, Redação científica: a prática de fichamentos, resumos, resenhas, São Paulo: Atlas, 2009. 11. Ed.

MEDRADO, Betânia Passos, Espelho, espelho meu: um estudo sociocognitivo sobre a conceptualização do fazer pedagógico em narrativas de professoras. Recife, 2006. Tese (Doutorado em Letras e Linguística) – Centro de Artes e Comunicação, Universidade Federal do Pernambuco.

MUSSALIN, BENTES, Fernanda, Anna Christina, organizadoras, Introdução à linguística: fundamentos epistemológicos, volume 3, 3ª ed., São Paulo: Cortez: 2007.

OLIVEIRA, Silvio Luiz de. Tratado de metodologia científica: projetos de pesquisa, TGI, TCC, Monografias, Dissertações e Teses. São Paulo: Pioneira, 1997.

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

RIBEIRO, Célia, Metacognição: Um Apoio ao Processo de Aprendizagem. *Psicologia: Reflexão e Crítica*, 2003, 16(1), pp. 109-116. Disponível em <http://www.scielo.br/pdf/prc/v16n1/16802.pdf>.

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

ROEDIGER, Henry L., Iii and Lyn M. Goff. "Memory." *A Companion to Cognitive Science*. Bechtel, William and George Graham (Eds). Blackwell Publishing, 1999. Blackwell Reference Online. 21 February 2009 <http://www.blackwellreference.com/subscriber/tocnode?Id=g9780631218517_chunk_g978063121851720>

TAGLIEBER, L. K. A leitura na língua estrangeira. In: BOHN, H. I.;

VANDRESEN, P. (org) Tópicos de Linguística Aplicada. Florianópolis: Editora da UFSC, 1998.

VARELA, THOMPSON, ROSCH, Francisco, Evan, Eleanor, *A mente incorporada: ciências cognitivas e experiência humana*, trad. Maria Rita Secco Hofmeister, Porto Alegre: Artmed, 2003.

GODOY, Roberta Guimarães de, *Hipertexto, Leitura e ensino*: Universidade Federal de Pernambuco, Recife, 2009.

PEREIRA JR, Antonio Davis, *A Constituição Multimodal De Textos Digitais: A Superposição E A Integração De Modos Diversos De Linguagem Na Construção Hipertextual*, Fapesp, São Paulo, 2002. Disponível em <http://ead1.unicamp.br/e-lang/publicacoes/down/09/09.pdf>.

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

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

- 1) 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;
- 2) 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.
- 3) 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?

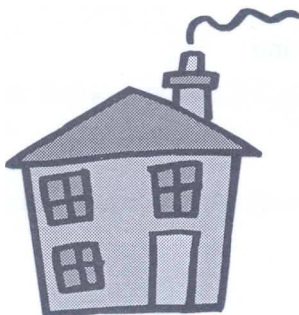


Figura 2.1 - Representação de uma casa

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 Collaboration¹³); 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 software²⁴ 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 colaboradores³⁵, 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.

³ 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.

⁴ 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.

⁵ 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 desenvolvimento⁶. 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 artefatos⁷ 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

⁶ Cabe ao processo decidir quais artefatos serão produzidos, a equipe e atividades necessárias para criar e gerenciar esses artefatos.

⁷ 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 atores⁸ 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.

❖ 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 freqüentemente 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 classes⁹ 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 seguir¹⁰:

+ 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.

⁹

Em alguns casos estão presentes em outros diagramas da UML.

¹⁰

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.

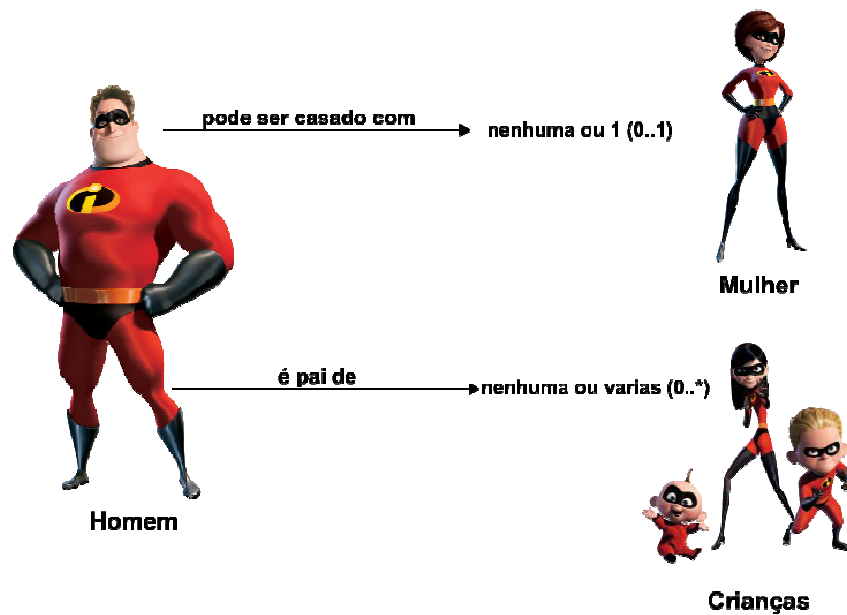


Figura 4.1 - Ilustração de Multiplicidade

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 seqüência 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:

1. 0..1 (exatamente um)
2. 1 ou 1..1 (exatamente um)
3. ou $0 \dots *$ (qualquer valor inteiro não-negativo)
4. 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

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.

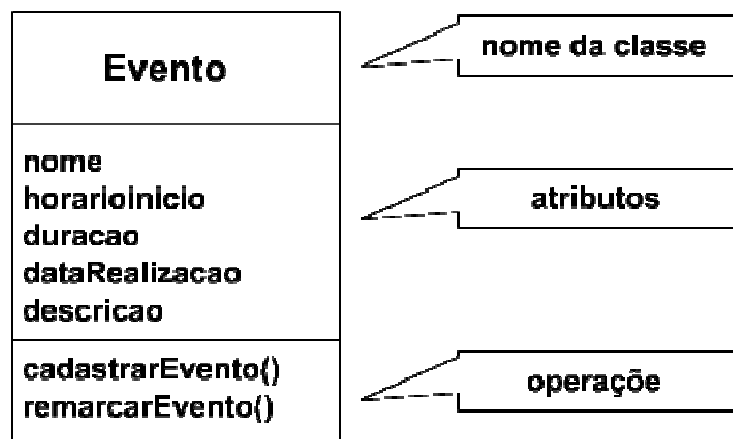


Figura 4.3 - Representação de uma classe

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

- 1- O nome da classe seja centralizado e negrito
- 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: 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 visibilidade¹¹ 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

11

Conforme visto no item 4.1.1 na página 94.

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: TEditora¹²

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]

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-default¹³ 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: real**piso**Salarial: 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: notal 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.

¹² Representação de tipos em Delphi.

¹³ 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.

BoletimEscolar nota1:real nota2: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.

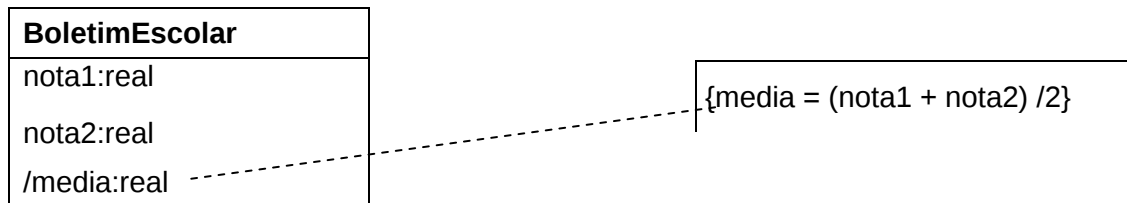


Figura 4.8 - Exemplo da especificação de um atributo derivado

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}

A visibilidade¹⁴ 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.

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.

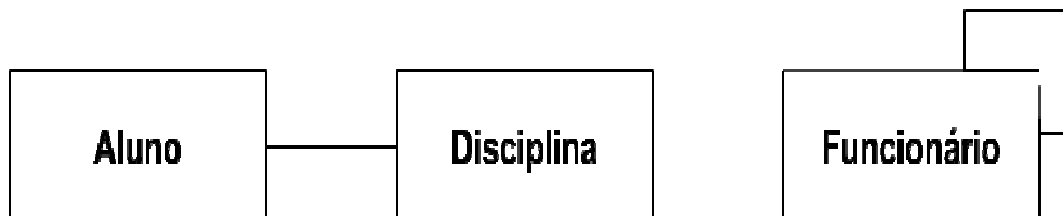


Figura 4.10 - Exemplo da representação de associação

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.

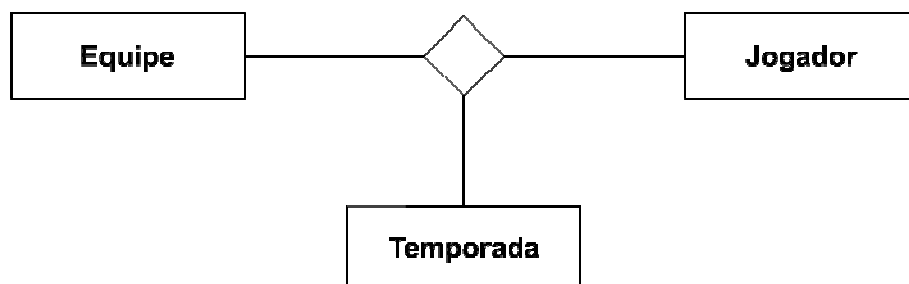


Figura 4.11 - Exemplo da representação de uma associação ternária

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".

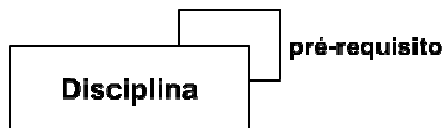


Figura 4.13 - Exemplo de uso do nome de associação

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.

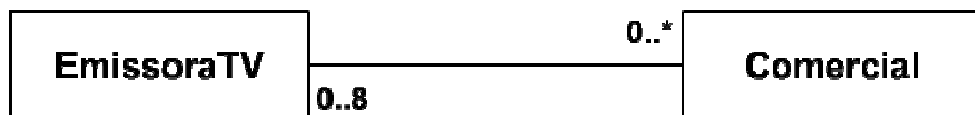


Figura 4.14 - Exemplo de multiplicidade numa associação

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.

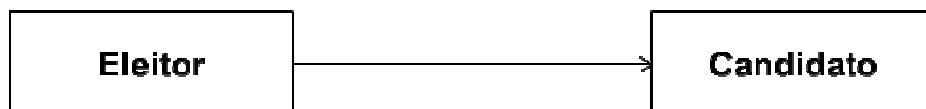
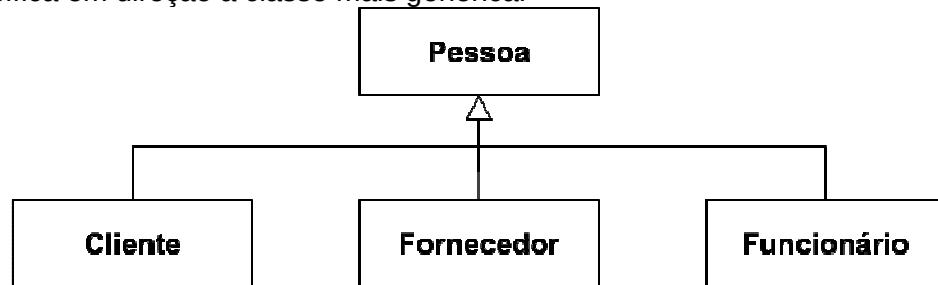


Figura 4.16 - Exemplo de navegabilidade

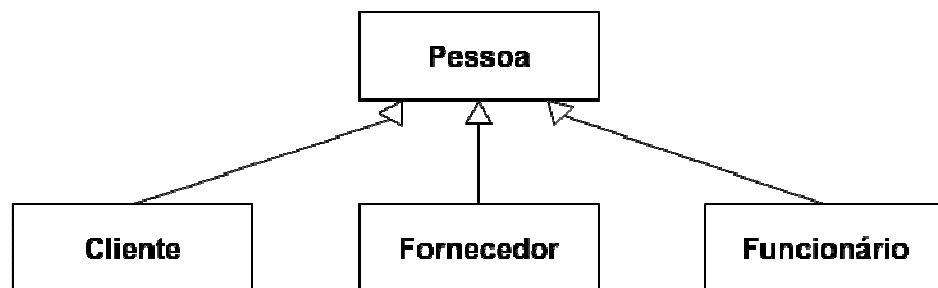
❖ GENERALIZAÇÃO

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

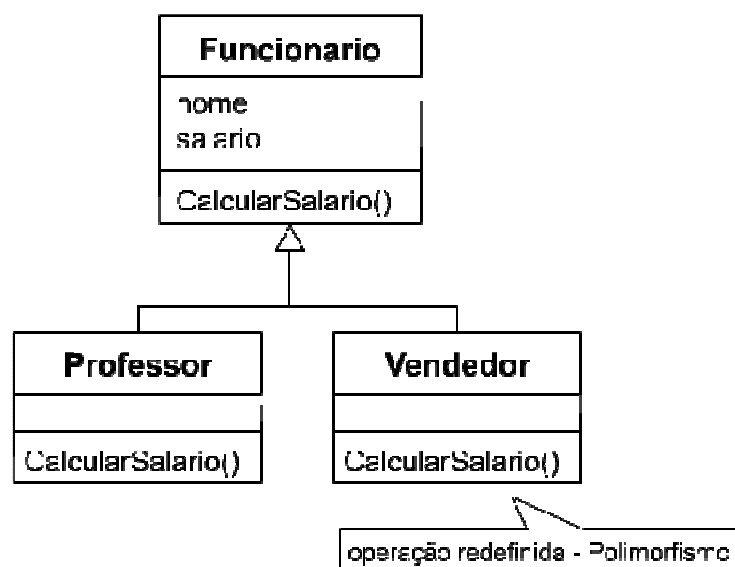


Figura 4.21 - Exemplo da representação de Polimorfismo

❖ 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.

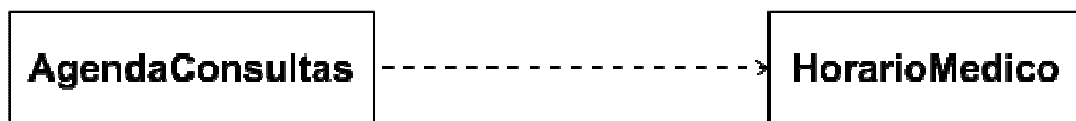


Figura 4.22 - Exemplo de dependência

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

depende 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.

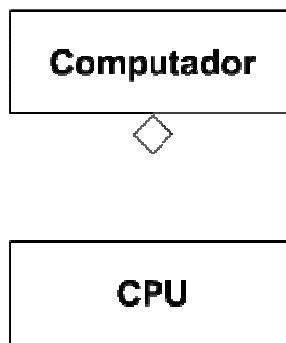


Figura 4.24 - Exemplo de uma agregação

❖ 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.

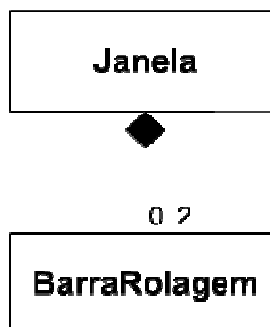


Figura 4.25 - Exemplo de uma composição

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 'Aula Experimental' web 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' (highlighted in orange). Below the header, on the left, is a sidebar with 'Aula' and 'Bem vindo' (with a link to 'Inicial'). 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 text is a login form with fields for 'Login' (containing 'kikopalmas') and 'Senha' (masked with dots), and an 'Entrar' button. At the bottom of the page is a green footer bar with the text: 'Minter em Linguística UFPB/FTO - Sistema para aulas em hipertexto experimentais - 1.0'.

The screenshot shows the 'Aula Experimental' web interface for Experiment II. It has the same header and sidebar as Experiment I. 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 text is a menu with three buttons: 'Visão Geral de UML', 'Diagramas de Gazeo', and 'Questionário de Memorização'. At the bottom of the menu area, it says: 'Apostila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4'. At the bottom of the page is a green footer bar with the text: 'Minter em Linguística UFPB/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). Dúvida? 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:

Aluno
matricula: string
nome: string
dataNascimento: date
dataMatricula: date
valorMensalidade: real
MatricularAluno()
ReajustarMensalidade (percentual: real)
TrancarMatricula()
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.

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 procuraram trabalhar mais ativamente a ideia 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, parava 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 Proposal), emitido pelo 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 limpeza 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.

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 (no-tação 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.

Booth, 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. Sem o caso! 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 (no-tação 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.

Booth, 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. Sem o caso! 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
 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](#)

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
Diagramas Dinâmicos	Diagrama de Componentes
	Diagrama de Pacotes
	Diagrama de Implantação
	Diagrama de Estrutura Composta
	Diagrama de Casos 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

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 aplicando, 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-tu-da-se, entre outros, as concordâncias nominal e verbal.

Concordância Nominal:
As palavras menos, aienta, 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
FIM

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)

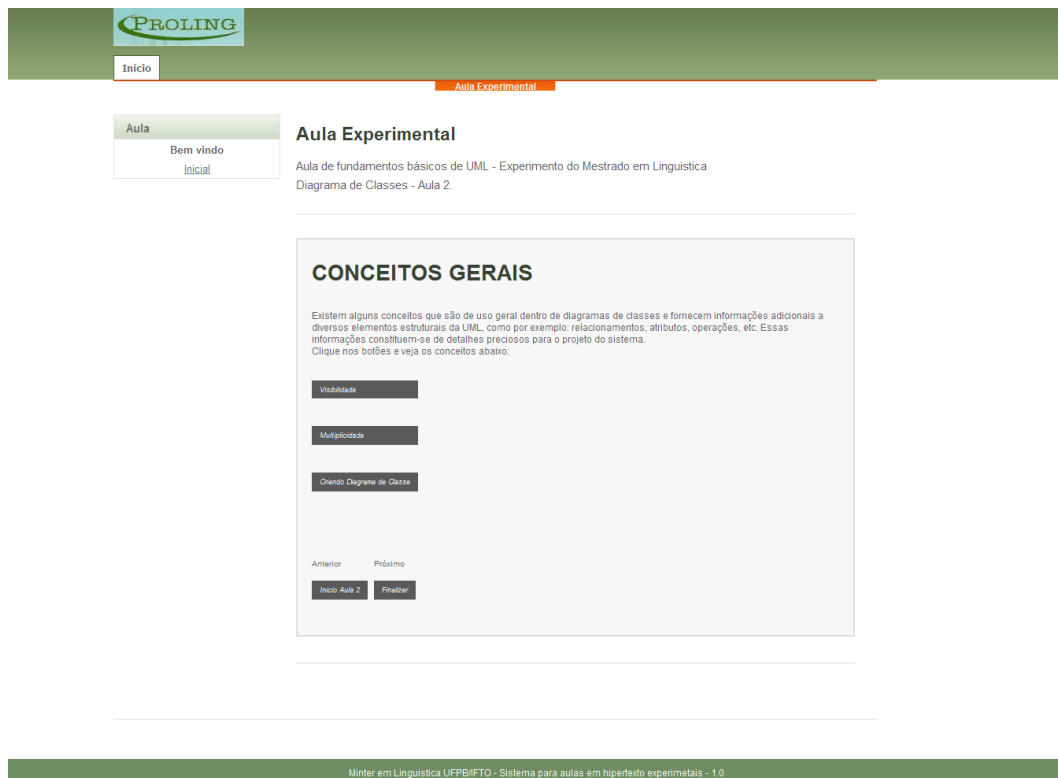
Os modelos criados durante o processo de desenvolvimento passam por uma evolução natural. Essa evolução frequentemente leva a modelos incompletos e inconsistentes, ou seja, é comum que os desenvolvedores criem seus modelos de forma gradativa.

Anterior

Modelando com a UML



- Telas do Sistema WEB do segundo texto do experimento II.



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 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 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.

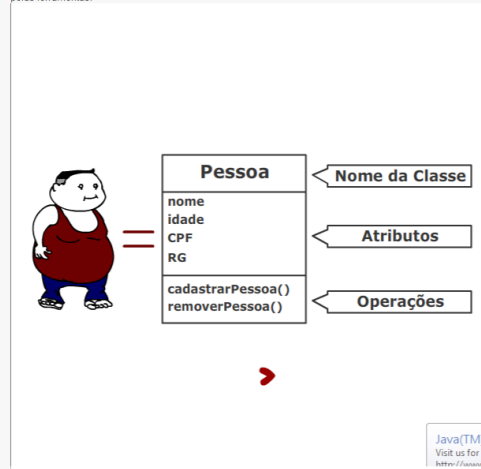
Diagramas de classes podem ser organizados dentro de pacotes, assim como um pacote pode ser representado por um ou mais diagramas de classes.

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 a-linhados 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 tornam-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 correspondem a tipos básicos (inteiro, real, string, booleano) mas, também podemos associar tipos definidos pela linguagem de implementação, como por exemplo DateTime 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 determinado 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:

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

Funcionario: nome: string; salário: real; piso: real; salário: 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 tem necessidade de aparecer em forma de atributo. Todavia sua existência 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 BoletimEscalar possui, entre outros, os atributos: nota1 e nota2. Sabemos que a média é calculada pela divisão por dois do resultado de 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 ITI 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.

BoletimEscalar: nota1: real; nota2: real; média: 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.

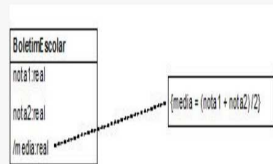


Figura 4.8 - Exemplo da especificação de um atributo derivado

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)

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

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

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

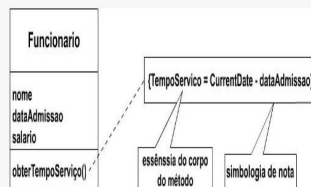


Figura 4.9 - Exemplo da especificação do corpo de um método

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](#)

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 coligação 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.

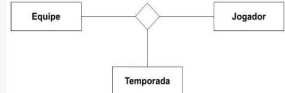


Figura 4.11 - Exemplo da representação de uma associação ternária

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 admissos que melhoraram, em alguns casos, a compreensão do modelo. Todos esses admissos 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 classe. 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".

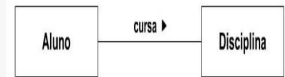


Figura 4.12 - Exemplo de nome de associação e direção

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 esclarecer o relacionamento: "Uma disciplina é pré-requisito de outra disciplina".

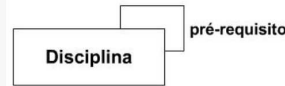
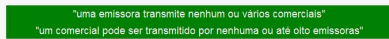


Figura 4.13 - Exemplo de uso do nome de associação

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 oculta que se relacionará na associação. Apesar de ser opcional, só deve ser omitida nas primeiras visitas ao diagrama. Uma vez que a modelagem avança, 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 obter:



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

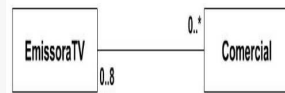


Figura 4.14 - Exemplo de multiplicidade numa associação

Se a multiplicidade da associação for superior a um, então os elementos relacionados podem ser ordenados ou não. Se não indicarmos nenhum admissos na associação, estamos 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 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 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 única" 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

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.

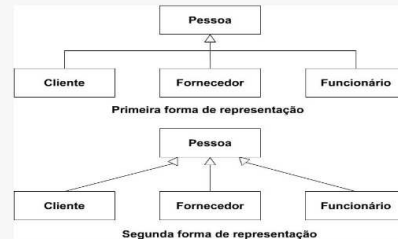


Figura 4.18 - Exemplo de representação de generalização

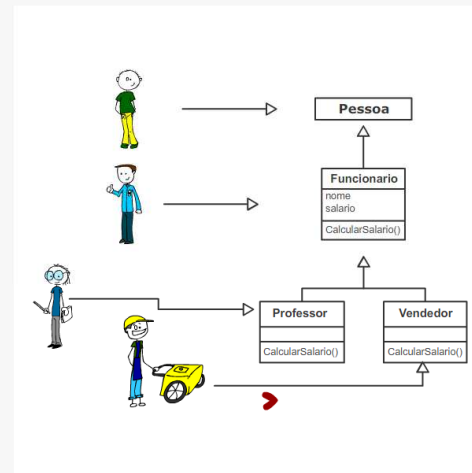


Figura 4.21 - Exemplo da representação de Polimorfismo na função CalcularSalario().

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. 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.

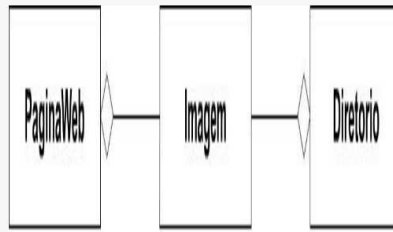


Figura 4.23 - Representação da agregação envolvendo a classe Imagem

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

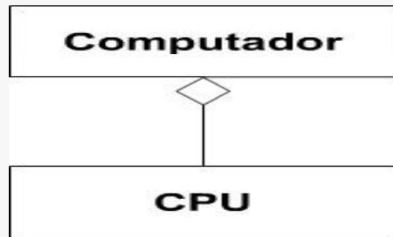


Figura 4.24 - Exemplo de uma agregação

Anterior

Relacionamentos

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 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.

Segundo 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.

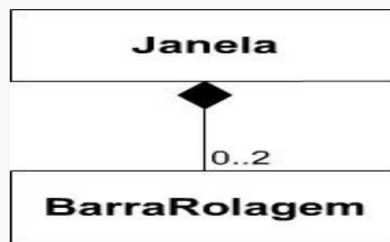


Figura 4.25 - Exemplo de uma composição

Anterior

[Relacionamentos](#)

Obrigado por participar do nosso experimento, agora responda os questionários.

- Telas do Sistema WEB do primeiro dia de experimento

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

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
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

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
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

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
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

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
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

[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
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 *.*

[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
Escolha no Menu Abaixo qual a aula que irá fazer:

Observe a figura

Aluno	
matricula: string	
nome: string	
dataNascimento: date	
data Matricula: date	
valorMensalidade: real	
MatriculaAluno()	
ReajustarMensalidade (percentual: real)	
TrancaMatricula()	
EmitirBoletoPagto (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

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	
data Matricula: date	
valorMensalidade: real	
MatriculaAluno()	
ReajustarMensalidade (percentual: real)	
TrancarMatricula()	
EmitirBoletoPagto (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

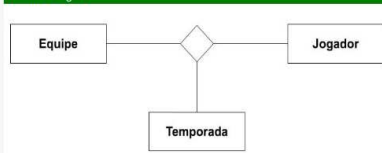
Confirma Resposta

Apostila baseada no livro Desenvolvendo aplicações UML, de Ana Cristina de Melo, capítulos 2, 3 e 4

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

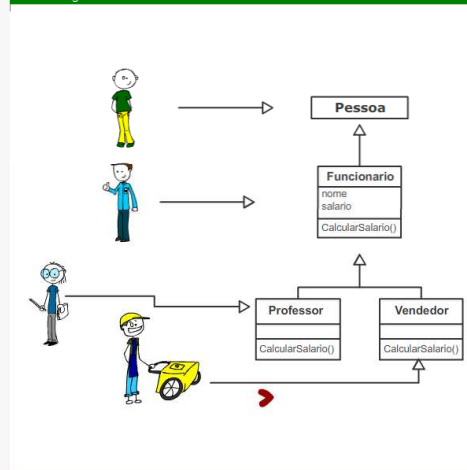
Confirma Resposta

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

[Confirma Resposta](#)

Apófila baseada no livro Desenvolvendo aplicações UML de Ana Cristina de Melo, capítulos 2, 3 e 4



[Início](#)

Aula Experimental

Aula

Bem vindo

[Inicial](#)

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

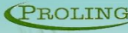
Sua data de nascimento

Fale sobre sua opinião sobre o experimento

Confirma Resposta

Minter em Linguística UFPB/IFTO - Sistema para aulas em hipertexto experimentais - 1.0

- 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 da 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 conseguiram encontrar uma maneira satisfatória de modelar seus sistemas. Foi a partir da década de 90 que começaram a surgir tentativas de procurar trabalhar de forma mais ativa com o paradigma da orientação a objetos. Diversos autores famosos contribuíram com publicações de seus respectivos métodos.

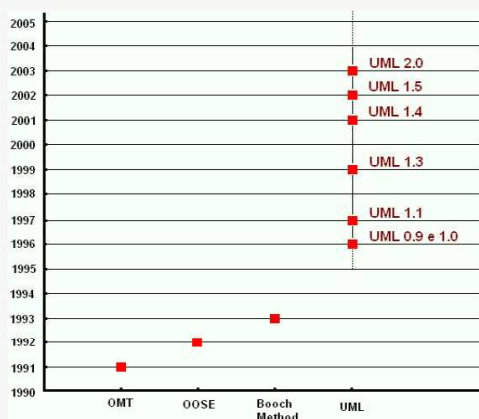


Figura 1. Linha do Tempo de UML

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 (SAMPÃO, 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 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.

[Leia Mais sobre Histórico da UML](#)

[Anterior](#)

[Próximo](#)

[Introdução](#)

[Estrutura da Especificação](#)

Aula

Bem vindo

[Inicial](#)

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 2.4 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).

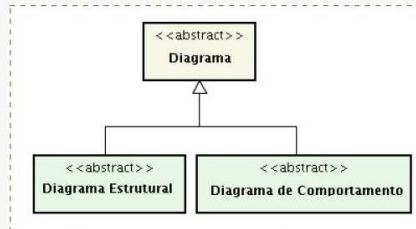


Figura 2. Organização Geral dos Diagramas de UML 2

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.

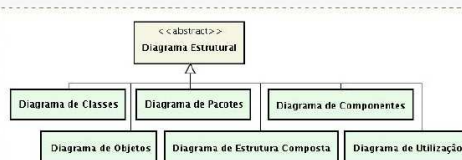


Figura 3. Diagramas Estruturais

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.

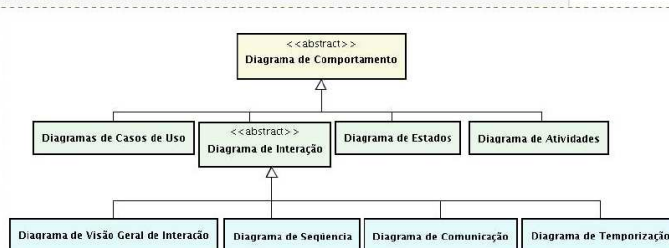


Figura 4. Diagramas Comportamentais

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).

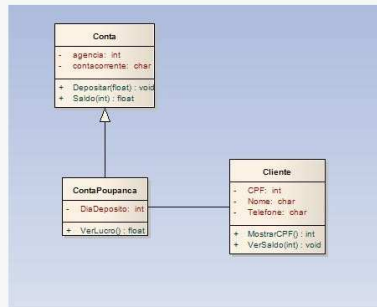


Figura 5. Exemplo de Diagrama de Classes

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).

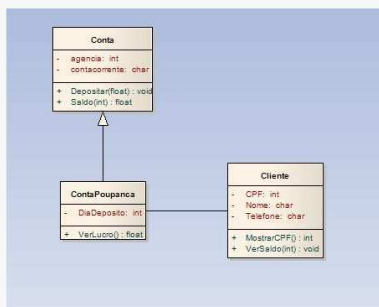


Figura 5. Exemplo de Diagrama de Classes

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).

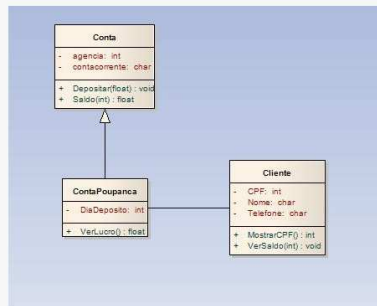


Figura 5. Exemplo de Diagrama de Classes

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.

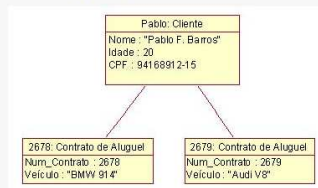


Figura 6. Exemplo de Diagrama de Objetos

[Anterior](#)

[Próximo](#)

[Diagramas UML 2](#)

[Diagrama de Pacotes](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
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.

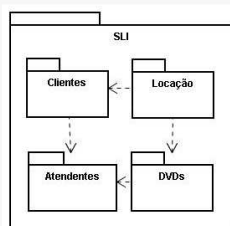


Figure 1. Exemplo de Diagrama de Pacotes

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).

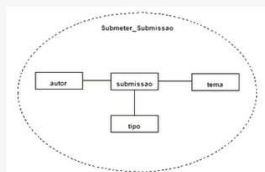


Figure 8 - Diagrama de Estrutura Composta

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.

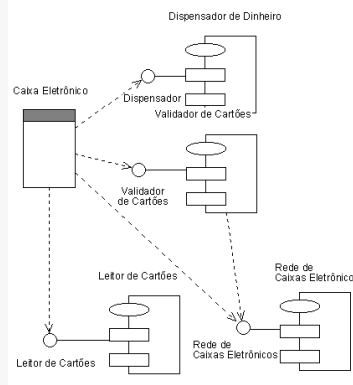


Figura 9 - Diagrama de Componentes

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.

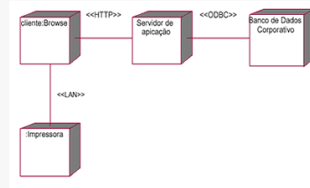


Figura 10 - Diagrama de Implantação

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.

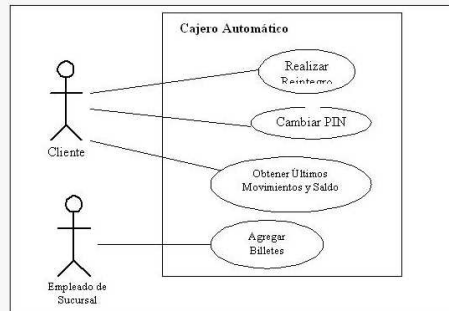


Figura 11 - Diagrama de Caso de Uso

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.

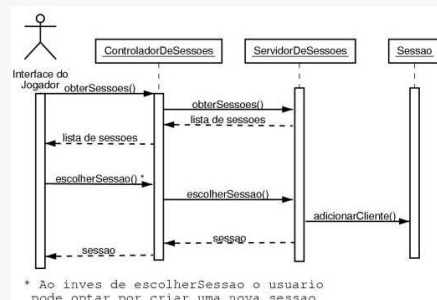


Figura 12 - Diagrama de Sequência

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.

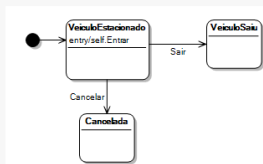


Figura 13 - Diagrama de Máquina de Estados

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.

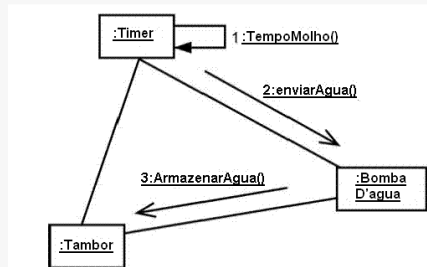


Figura 14 - Diagrama de Comunicação

Anterior: [Diagrama de Casos de Uso](#)

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).

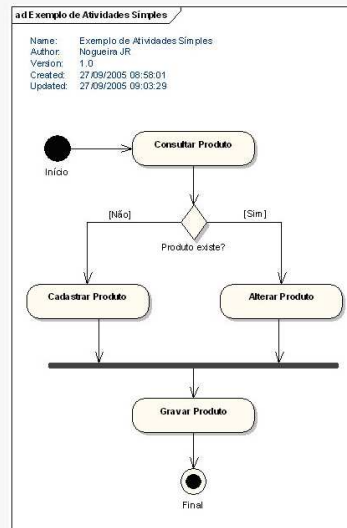


Figura 15 - Diagrama de Atividades

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.

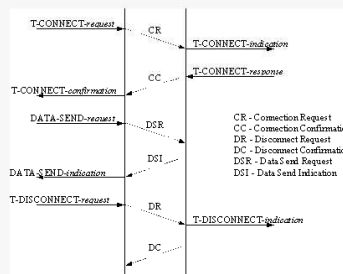


Figura 16 - Diagrama de Temporização

Anterior: Diagrama de Máquina de Estados

Próximo: Conclusão

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.

Antes: [Diagrama de Componentes](#)

Proximo: [Finalizar](#)

Aula

Bem vindo

[Inicial](#)

Aula Experimental

Aula de fundamentos básicos de UML - Experimento do Mestrado em Linguística
Final da leitura

Agora responda o questionário