

Tecnologias Web Aplicadas ao Ensino de Engenharia de Estruturas

AGRADECIMENTOS

Agradeço à CAPES pela bolsa concedida para o desenvolvimento deste projeto.

Agradeço aos professores e funcionários do DEES, em especial ao Franco e ao Felício pelas inúmeras orientações recebidas.

Agradeço ao Augusto pela resolução de diversas dúvidas.

Agradeço às minhas irmãs, amigos e familiares pela força.

Agradeço aos meus pais pelo apoio dispensado.

Enfim, agradeço à Deus pelas diversas oportunidades concedidas.

SUMÁRIO

1	Introdução	1
1.1.	Objetivos.....	1
1.2.	Motivação	3
1.3.	Metodologia de Desenvolvimento.....	5
1.4.	Conteúdo por Capítulos	8
2	Referencial Teórico.....	11
2.1.	A internet e a Web.....	11
2.1.1.	Definições	11
2.1.2.	Histórico	15
2.2.	A Educação a Distância.....	17
2.2.1.	Definições	17
2.2.2.	Histórico	19
2.2.3.	Legislação da EAD no Brasil.....	25
2.2.4.	Contexto Social Econômico e suas aplicações para EAD	28
2.3.	Sistemas Educacionais Informatizados.....	29
2.3.1.	Sites Educacionais	31
2.3.2.	Sistemas de Autoria para Cursos a Distância.....	36
2.3.3.	Salas de Aula Virtuais	38
2.3.4.	<i>Frameworks</i> para Aprendizagem Cooperativa	40
2.4.	Sistemas Educacionais Informatizados para Engenharia de Estruturas.....	41
3	Recursos Tecnológicos Utilizados.....	44
3.1.	Programação Orientada a Objetos	44
3.1.1.	Histórico	44
3.1.2.	Definições	45
3.1.3.	Vantagens da POO.....	51
3.2.	Linguagens para <i>Internet</i>	52
3.2.1.	HTML.....	53
3.2.2.	XML	55
3.2.3.	CSS	57
3.2.4.	JAVA.....	58
3.2.5.	<i>JavaScript</i>	59
3.2.6.	<i>Action Script</i>	61
3.2.7.	ASP Clássico	63
3.2.8.	UML	65
3.3.	O Framework .NET.....	66

3.3.1.	Abstração de <i>Hardware</i> e do Sistema Operacional	67
3.3.2.	<i>Class Libraries</i>	69
3.3.3.	Arquitetura em N Camadas.....	70
3.3.4.	ASP.NET	72
3.3.5.	C# (C Sharp).....	75
3.4.	Banco de Dados.....	77
3.4.1.	Definições	77
3.4.2.	Modelos de Dados	77
3.4.3.	Conceitos dos Bancos de Dados Relacionais.....	79
3.4.4.	Projeto de Bancos de Dados	81
3.4.5.	Sistemas Gerenciadores de Banco de Dados	83
3.5.	<i>Software</i> Utilizados	88
3.6.	Observações	89
4	Concepção Conceitual de uma Plataforma para EAD	92
4.1.	O que é uma Plataforma	92
4.2.	Proposta da Base Tecnológica	93
4.2.1.	Sistema Operacional.....	94
4.2.2.	Servidor http	95
4.2.3.	Linguagens.....	95
4.2.4.	Banco de Dados	95
4.3.	Estudo e Proposição Conceitual de uma Plataforma para EAD	96
4.3.1.	Objetivos.....	96
4.3.2.	O NucleoEAD x Plataforma para EAD	96
4.3.3.	Diretrizes da Plataforma Tecnológica para EAD.....	99
4.3.4.	Módulos de Desenvolvimento da Plataforma Tecnológica para EAD.....	102
4.3.5.	Diagramas UML Desenvolvidos.....	104
5	O Curso de Introdução ao Método dos Elementos Finitos – Conteúdo Digital para <i>web</i>	108
5.1.	Tecnologias <i>Web</i> Aplicadas ao Desenvolvimento de Conteúdos Digitais.....	108
5.2.	Objetivos.....	110
5.3.	Estruturação de <i>Sites</i>	111
5.3.1.	Interface	111
5.3.2.	Navegação	113
5.4.	O Conteúdo Digital.....	120
5.4.1.	Definição do Escopo do Programa do Curso	120
5.4.2.	O Conteúdo.....	124
5.5.	Papel da Computação Gráfica.....	127
5.6.	Bases Tecnológicas e Configurações do Sistema	127
5.6.1.	Bases Tecnológicas.....	127
5.6.2.	Configurações do Sistema	128

6	Ferramentas Desenvolvidas para Apoio a EAD.....	129
6.1.	Bloco de Notas.....	130
6.2.	Recados.....	131
6.3.	Agenda.....	132
6.4.	O Banco de Dados do Curso	133
6.5.	Programação em ASP.NET	135
7	O Aplicativo <i>WEB</i> para o Método dos Elementos Finitos	137
7.1.	Introdução.....	137
7.2.	Interface	138
7.3.	O Aplicativo do Método dos Elementos Finitos e a Programação Orientada a Objetos.....	145
8	Conclusão.....	153
8.1.	Introdução.....	153
8.2.	Contribuições do Presente Projeto.....	156
8.3.	Proposições para Desenvolvimentos Futuros	156
	Referências Bibliográficas.....	158
	Anexo A.....	164

LISTA DE SIGLAS E ABREVIATURAS

AJAX	JavaScript e XML assíncronos
ANSI	American National Institute
ARPA	Advanced Research Projects Agency
ASP	Active Server Pages
BD	Banco de Dados
BLOB	Binary large object
CADTEC	Centro de Apoio, Desenvolvimento Tecnológico e Computação Gráfica
CAI	Computer Aided Instruction
CGI	Common Gateway Interface
CLR	Comon Language Runtime
CLS	Common Language Specifications
CSS	Cascading Style Sheets
D.O.U	Diário Oficial da União
DEES	Departamento de Engenharia de Estruturas da UFMG
DLL	Dynamic Link Libraries
DTD	Document Type Definition
EAD	Educação a Distância
ECMA	European Computer Manufactures Association
EXE	Extensão de arquivos executáveis
FAQ	Frequently asked questions
FTP	File Transfer Protocol
GD	Grupo de Discussão
HTML	Hipertext markup language
HTTP	Hypertext Transport Protocol
IBASE	Instituto Brasileiro de Análises Sociais e Econômicas
IBM	International Business Machine
IDC	Internet Database Conector
IIS	Internet Information Service
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IRC	Internet Relay Chat
ISAPE	Internet Server Application Programing Interface
ISO	International Standard Organization
JIT	Just-in-time
LAGEAR	Laboratório Gráfico para Ensino de Arquitetura
LAN	Local Área Network
MEB	Movimento de Educação de Base
MEC	Ministério da Educação e Cultura
MEF	Método dos Elementos Finitos
MIME	Multipurpose Internet Mail Extensions

MSDE	Microsoft SQL Server 2000 Desktop Engine
MSIL	Microsoft Intermediate Language
MVS	Multiple Virtual Storage
NEAD	Núcleo de Educação Aberta e a Distância
NNTP	Network News Transfer Protocol
NSF	National Science Foundation
ODBC	Open Data Base Connectivity
OLEDB	Object Linking and Embedding, Database
OMG	Object Management Group
OSI	Open Systems Interconnection
PEC	Programas Educacionais por Computador
PHP	Hypertext Preprocessor
PL / SQL	Procedural Language/Structured Query Language
POO	Programação Orientada a Objetos
POP3	Post Office Protocol
ProInfo	Programa Nacional de Informática na Educação
PRONTEL	Programa Nacional de Tele-Educação
PTV	Princípio dos Trabalhos Virtuais
PWS	Personal Web Server
RNP	Rede Nacional de Pesquisa
Seed	Secretaria de Educação a Distância
SESU	Secretaria de Educação Superior
SGBD	Sistema de Gerenciamento de Banco de Dados
SGBDR	Sistemas de Gerenciamento de Bancos de Dados Relacionais
SGML	Standard Generalized Markup Language
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
T - SQL	Transaction - Structured Query Language
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
UAB	Universidade Aberta do Brasil
UDDI	Universal Description, Discovery and Integration
UDF	User Defined Functions
UFPR	Universidade Federal do Paraná
UML	Unified Modeling Language
UNED	Universidad Nacional de Educación a Distancia
UniRede	Universidade Virtual Pública do Brasil
URL	Universal Resource Locator
URSS	União das Repúblicas Socialistas Soviéticas
UVB	Universidade Virtual Brasileira
WAN	Wide Area Networks

WWW
XHTML
XML

World Wide Web
eXtensible Hipertext Markup Language
eXtensible Markup Language

LISTA DE FIGURAS

FIGURA 1 – Internet e alguns de seus serviços	14
FIGURA 2 - Usuários da Web	17
FIGURA 3 - Infra-estrutura da internet	17
FIGURA 4 - Definição de classes.....	47
FIGURA 5 - Exemplo do mecanismo de herança	49
FIGURA 6 - Definição de encapsulamento.....	50
FIGURA 7 – Exemplo do código de uma página HTML.....	54
FIGURA 8 – Exibição do código da figura 4 em <i>browser</i>	54
FIGURA 9 – Exemplo de documento XML	56
FIGURA 10 – Exemplo de um arquivo CSS	58
FIGURA 11 – Exemplo da Linguagem <i>JavaScript</i> dentro de um arquivo HTML ..	60
FIGURA 12 – Arquivo “.asp” com códigos HTML e códigos ASP, em destaque ..	65
FIGURA 13 – Organização do framework.NET	67
FIGURA 14 – Mecanismo de multilinguagem permitido pelo <i>Framework</i> .NET...	69
FIGURA 15 – Arquitetura em n Camadas.....	72
FIGURA 16 – Níveis de Abstração de um Sistema de Banco de Dados	78
FIGURA 17 – Modelos utilizados em um projeto de BD	82
FIGURA 18 – Diagrama UML da Classe Usuários e suas especificações	106
FIGURA 19 – Diagramas UML das Classes de Cursos e suas relações com os usuários	107

FIGURA 20 – Hierarquia Global de Navegação para acesso ao curso de Introdução ao MEF	114
FIGURA 21 – Estruturação das páginas do curso de Introdução ao MEF.....	114
FIGURA 22 – Rodapé do curso de Introdução ao MEF	115
FIGURA 23 – Página Inicial do curso	115
FIGURA 24 – Índice do Conteúdo Digital do Curso.....	116
FIGURA 25 – Página da Ferramenta Agenda	117
FIGURA 26 – Página da Área de Transferência	118
FIGURA 27 – Página da Ferramenta de Busca	119
FIGURA 28 – Página FAQ	120
FIGURA 29 – Modelo das Páginas do Conteúdo Digital	120
FIGURA 30 – Ferramenta Bloco de Notas.....	130
FIGURA 31 – Listagem das anotações existentes.....	131
FIGURA 32 – Ferramenta Recados	132
FIGURA 33 – Listagem dos recados existentes.....	132
FIGURA 34 – Banco de dados do curso	135
FIGURA 35 – Diagrama do aplicativo do Método dos Elementos Finitos.....	139
FIGURA 36 – Página inicial do aplicativo do Método dos Elementos Finitos.....	140
FIGURA 37 – Página Nós	141
FIGURA 38 – Página Seções.....	141
FIGURA 39 – Página Materiais	142
FIGURA 40 – Página Elementos	143

FIGURA 41 – Página Resultados	144
FIGURA 42 – Página ferramentas.....	144
FIGURA 43 – Diagrama das classes básicas do aplicativo do Método dos Elementos Finitos	146
FIGURA 44 – Diagrama estendido da classe Elemento e suas derivadas	149
FIGURA 45 – Diagrama estendido da classe Secao e suas derivadas.....	150
FIGURA 46 – Diagrama estendido da classe No e das classes Material e suas derivadas.....	151
FIGURA 47 – Diagrama estendidos das classes Modelo, UtilModelo e Matriz....	152

LISTA DE TABELAS

TABELA 3.1 – <i>Namespaces</i> do ASP.NET.....	74
TABELA 4.1 – Comparação entre o NucleoE@D e a Plataforma para E@D proposta	98
TABELA 4.2 – Símbolos e Conceitos da UML	104
TABELA 5.1 – Ícones Utilizados no Curso de Introdução ao MEF	112

RESUMO

A proposta deste trabalho é contribuir para o desenvolvimento e exploração de tecnologias e metodologias para o ensino e aprendizagem de Engenharia de Estruturas através da Educação a Distância (EAD) via *web*.

Foi realizada inicialmente, uma revisão bibliográfica da EAD, compreendendo definições, legislação, histórico e terminando com um levantamento sobre o estado da arte da modalidade predominante de EAD nos dias de hoje, que é a educação a distância via *web*. Uma pesquisa sobre os sistemas educacionais informatizados existentes e as possibilidades tecnológicas disponíveis utilizadas para o desenvolvimento da EAD via *web* também foi realizada, objetivando embasar teoricamente a proposta de uma plataforma para EAD e o desenvolvimento de um curso de Introdução ao Método dos Elementos Finitos.

Alguns recursos tecnológicos utilizados para desenvolvimento de ambientes *web* foram apresentados por terem sido aplicados na execução dessa dissertação. A justificativa da escolha de utilização desses recursos foi exposta, apresentando uma comparação sucinta das outras tecnologias existentes. A grande possibilidade de desenvolvimento de aplicações *web* a partir dos recursos pesquisados motivou a proposição de uma plataforma para EAD via *web* e o desenvolvimento de um aplicativo para o Método dos Elementos Finitos para ser executado em ambiente *web*.

O conteúdo digital do curso de Introdução ao Método dos Elementos Finitos foi desenvolvido em formato para internet, aplicando conceitos de hipertexto, hiperímia e explorando recursos multimídias como animações. O *site* do curso foi implementado objetivando aproximar as relações interpessoais entre alunos e professores, e oferecendo facilidades adicionais que estimulem esta forma de ensino e o aprendizado.

Durante o desenvolvimento do trabalho foi confirmada a necessidade de equipes multidisciplinares em desenvolvimentos de projetos desse tipo, para que a EAD via *web* continue sendo uma forte alternativa para democratização do conhecimento no país.

Palavras Chave: EAD, Elementos Finitos, Tecnologias Web, *Framework*.

ABSTRACT

This work aims to develop and explore technologies and methodologies for E-learning teaching and learning experiments in the field of Structural Engineering through the *web*.

An initial bibliographical revision was done encompassing definitions, legislation, history and ending with a survey about the state of the art on E-learning via *web*. A thorough survey about available computational educational systems and about available technologies for the development of E-learning via *web* has also been done with the objective of providing the basis for a proposal for an E-learning platform and to develop the contents of an introductory course on Introduction to the Finite Element Method.

Some technological resources commonly used for the development of *web* environments have been presented, because they have been applied on this project. The choice of such resources has been justified by comparing them with other available technologies. The resources listed on the survey showed the great potential of applying such technologies on the development of *web* applications, which encouraged the proposal of a platform for E-learning via *web* and the development of a computational system for teaching the Finite Element Method to be run in the *web* environment.

The digital content of the course “Introduction to the Finite Element Method” was also developed in a format for the Internet, applying concepts of hypertexts, hypermedia and exploring multimedia resources, such as animation, images, photos, etc... The site environment for the course has been implemented aiming to approximate the interpersonal relation among students and instructors and offering additional facilities to stimulate and facilitate this form of teaching and learning.

During the development this work, has been confirmed the need for multidisciplinary teams in the development of such projects so that the E-Learning through the web remains a strong alternative to democratization of knowledge in this country.

Key-Words: E-learning, Finit Elements, Web Technologies, *Framework*.

1

INTRODUÇÃO

1.1. OBJETIVOS

O desenvolvimento, a democratização da internet e o surgimento de novas Tecnologias da Informação (TI) trouxeram grande avanço para a Educação a Distância e instigaram a possibilidade de desenvolvimento de ferramentas e ambientes ainda mais intuitivos e completos para o auxílio dessa modalidade de ensino.

Pensando nisso, esse projeto propõe o desenvolvimento da concepção inicial de uma plataforma tecnológica ou *framework*, que fornecerá uma base para aplicação e gerenciamento de cursos e de todas as atividades complementares a Educação à Distância (EAD). A concepção inicial desta plataforma tecnológica disponibiliza um sistema (*software*) para execução na *web* e acesso a banco de dados, composta de uma infra-estrutura para facilitar a oferta e gerenciamento de atividades relacionadas à educação à distância. A plataforma disponibilizará ferramentas auxiliares para a criação de escolas virtuais e de seus componentes. A concepção desta plataforma foi proposta a partir da análise da literatura existente sobre o assunto, da análise de outros ambientes educacionais para a *web* e seguindo a evolução natural da concepção de **Sistemas Educacionais via web para Plataformas Tecnológicas para E@D.**

O desenvolvimento de ambientes web mais dinâmicos e interativos para o ensino e aprendizagem de Engenharia de Estruturas é também um dos objetivos deste trabalho. Para possibilitar, bem como ilustrar, a aplicação de algumas tecnologias da informação em ambientes específicos, foi desenvolvido um curso introdutório do Método dos Elementos Finitos e um aplicativo *web* para o cálculo de esforços e deformações em elementos de barra, utilizando os conceitos apresentados no curso.

Em todo o processo de desenvolvimento do curso foram utilizados recursos multimídias (gráficos/visuais) ou sistemas de hipermídia simplificados na busca de experimentos educacionais que pudessem, de alguma forma, melhorar o aprendizado. A multimídia e a computação gráfica foram utilizadas como recurso didático e de redundância com um apelo visual consistente, sempre com o cuidado de evitar que o usuário esteja exposto a sobrecargas sensoriais, seguindo as orientações de TORRES e MAZZONI (2004).

O curso de Introdução ao Método dos Elementos Finitos apresenta uma estrutura de *sites* próprios, abrangendo: o conteúdo digital, um grupo de discussão, uma agenda, bloco de notas, uma ferramenta de recados, uma área de transferência e uma busca por palavras chave. Toda essa estrutura tem a finalidade de apoiar o aluno durante o processo de aprendizagem, tentando suprir, ainda que parcialmente, a ausência física de um professor.

O Método dos Elementos Finitos é um dos métodos mais utilizados para análise estrutural, possibilitando, a partir de seus resultados, a descrição do comportamento mecânico do problema estudado. É um método numérico aproximado para cálculo de sistemas contínuos, onde o sistema é discretizado, ou seja, é subdividido em subdomínios conectados entre si denominados Elementos Finitos. O curso desenvolvido pode ser utilizado como uma ferramenta auxiliar aos currículos regulares da graduação, a título de complementação às aulas presenciais e como auxílio no processo de consolidação do conhecimento.

As ferramentas agenda, bloco de notas e recados foram desenvolvidas utilizando as facilidades do “framework” ASP.NET (ferramenta para o desenvolvimento de aplicativos *web*). Além disto, conceitos e fundamentos de programação orientada a objetos foram aplicados na construção de ambientes educacionais tecnológicos. Essa

modalidade ou paradigma de programação pode ajudar na criação de ambientes de aprendizado mais dinâmicos e intuitivos de tal forma que a relação ensino/aprendizado pode ser continuamente melhorada com novos desenvolvimentos. O aplicativo (sistema computacional para web) desenvolvido calcula esforços e deformações nas extremidades de elementos de barra como treliças, vigas, pórticos planos e grelhas. Este programa foi desenvolvido utilizando o *framework* ASP.NET com a linguagem de programação C#. Todo o programa está estruturado seguindo os paradigmas da programação orientada a objetos, que não só permite posteriores ampliações, mas facilita a manutenção do sistema

Dentro dessa proposta foram estabelecidos os seguintes objetivos específicos deste trabalho:

- Desenvolvimento da concepção inicial da Plataforma Tecnológica para EAD;
- Desenvolvimento de um curso de Introdução ao Método dos Elementos Finitos;
- Desenvolvimento de ferramentas para auxílio à EAD (agenda, bloco de notas e recados).
- Desenvolvimento de um aplicativo, executado via *web*, para cálculo de esforços e deformações em elementos de barra, através do Método dos Elementos Finitos, aplicando os fundamentos e conceitos expostos no curso desenvolvido.

Merece destaque o fato de que o *design* gráfico do curso de Introdução ao Método dos Elementos Finitos teve uma importante colaboração do laboratório LAGEAR da Escola de Arquitetura da UFMG para o seu desenvolvimento.

1.2. MOTIVAÇÃO

Segundo NEVES (2003), a educação a distância cresceu em todo mundo com a popularização da internet e foi natural o surgimento de novas Tecnologias da

Informação e Comunicação (TICs). Esta aliança entre as novas tecnologias da informação e os novos recursos da comunicação está sendo explorada na busca de metodologias educacionais para melhorar o aprendizado e permitir uma importante propagação do conhecimento em todas as áreas.

As novas facilidades serviram para despertar a ousadia e a criatividade e tornaram possível o emprego de ferramentas gráficas, vídeos, locuções e animações ainda mais intuitivas e estimulantes para melhorar o aprendizado e torná-lo mais prazeroso. Ferramentas poderosas de programação para internet também estão cada vez mais disponíveis, aumentando a facilidade de se projetar páginas e aplicativos em ambiente *web* mais dinâmicos e interativos, atendendo de forma eficaz as expectativas e necessidades dos usuários.

NEVES (2003) observou que cada vez mais cidadãos e instituições vêm na Educação a Distância através da internet, um meio de democratizar o acesso ao conhecimento e de expandir oportunidades de trabalho e de aprendizagem ao longo da vida, devido principalmente à expansão das redes de comunicação e à conseqüente globalização das informações. O volume de informações acessíveis através da internet é imenso, e a facilidade que estas informações podem ser acessadas é um dos fatores de credibilidade da EAD através da internet.

A qualificação e instrução continuadas dos profissionais, além de constante renovação de conhecimentos, é hoje uma exigência do mercado de trabalho. A EAD através da internet apresenta-se como uma alternativa viável para muitos profissionais já que não obriga a contigüidade espaço-tempo durante o processo de aprendizagem além de possibilitar a criação de cursos dinâmicos e flexíveis.

A EAD pode ser vista, dentro de uma abordagem sócio-econômica, como uma alternativa para levar a educação em áreas de difícil acesso ou mais distantes dos centros de pesquisa.

A importância da EAD no mundo atual é incontestável e, conseqüentemente, surge a necessidade de ferramentas para a aplicação da EAD de forma eficiente, explorando todas suas possibilidades e satisfazendo as exigências de formação acadêmicas.

Todos esses fatores serviram de motivação para esse projeto somados à necessidade e à possibilidade do desenvolvimento de uma plataforma tecnológica intuitiva e eficaz para a EAD.

A área de engenharia de estruturas apresenta um potencial importante para o desenvolvimento de conteúdos digitais; assim, decidiu-se desenvolver um curso introdutório de Elementos Finitos e aplicar neste curso vários recursos e tecnologias disponíveis e facilitadores da aprendizagem. Este curso pode ser utilizado como um reforço aos currículos regulares da graduação, como ferramenta complementar as aulas presenciais e auxílio no processo de aprendizagem e como material didático complementar aos profissionais que desejam ampliar ou reciclar seus conhecimentos.

Visando a utilização de forma prática dos conceitos apresentados no curso do Método dos Elementos Finitos (MEF), foi desenvolvido um programa para cálculo de estruturas através do MEF. Este é um aplicativo que, embora calcule apenas elementos de barra com ações externas aplicadas nos nós, apresenta um grande potencial didático para a apresentação dos conceitos iniciais deste método. Além disso, o programa foi desenvolvido utilizando conceitos de programação orientada a objetos, possibilitando a sua ampliação posterior, para um sistema mais completo, com grande facilidade.

1.3. METODOLOGIA DE DESENVOLVIMENTO

Inicialmente, para desenvolvimento do projeto, foi realizada uma breve revisão bibliográfica das tecnologias que seriam utilizadas, bem como uma investigação sobre projetos semelhantes existentes.

As tecnologias estudadas foram:

- a internet e a web, com suas definições, como funcionam e a evolução ao longo do tempo;
- a conceituação de *framework* ou plataforma tecnológica;
- o paradigma da programação orientada a objetos;

- as linguagens de programação para internet como o *Hipertext markup language* (HTML), *Cascading Style Sheets* (CSS), *JavaScript*, ASP.NET, C#.

Durante estas pesquisas, procurou-se conhecer as tendências atuais de *design* e de programação para internet; banco de dados, suas definições, os modelos de banco de dados existentes, os *softwares* de banco de dados existentes no mercado e suas principais aplicações.

Foram realizadas ainda, pesquisas sobre os ambientes ou plataformas educacionais disponíveis, assim como cursos *on-line* que abordam o Método dos Elementos Finitos, artigos e textos relativos aos assuntos.

Em seguida ao processo de revisão bibliográfica, foi iniciado o desenvolvimento da plataforma tecnológica para EAD.

Inicialmente foram pensadas as características esperadas da plataforma, do ponto de vista do aluno e também do ponto de vista dos professores, administradores, monitores ou qualquer outro usuário da plataforma. A partir dessas características foi gerado um documento que estabelece as diretrizes para o desenvolvimento da plataforma.

A plataforma foi concebida por etapas, com o objetivo de permitir que a mesma possa ser desenvolvida por uma equipe de programadores. Para isso ela foi dividida em módulos com características e finalidades específicas. Estes módulos foram definidos a partir das funcionalidades esperadas para a plataforma. Padrões de programação esperados para a plataforma foram também estabelecidos.

Posteriormente iniciou-se o projeto (ou “desenho”) da plataforma. Este desenho foi realizado através de diagramas *Unified Modeling Language* (UML). Foram desenhadas as classes relativas ao gerenciamento dos usuários e dos cursos.

A concepção do curso de Introdução ao Método dos Elementos Finitos foi iniciada através da elaboração de seu conteúdo digital. Para tal, definiu-se primeiramente os tópicos que seriam abordados e o grau de profundidade com que seriam tratados. Esta etapa foi fundamentada com base nos trabalhos didáticos de LOGAN (2002) e ALVES

(2000), e nos conhecimentos adquiridos pela aluna durante as aulas de graduação e pós-graduação.

Após a definição do escopo do curso, deu-se início ao desenvolvimento do seu conteúdo digital, com a seleção e a confecção das figuras e animações utilizadas.

Paralelamente ao desenvolvimento do conteúdo digital, foi desenvolvido e criado o *design* e o *layout* do *site* em parceria com o laboratório LAGEAR da Escola de Arquitetura da UFMG.

As ferramentas de apoio e comunicação presentes neste curso seguiram o padrão adotado em outros cursos existentes do Centro Avançado de Desenvolvimento Tecnológico e Computação Gráfica (CADTEC) pertencente ao Departamento de Engenharia de Estruturas da UFMG (DEES). Além disso, outras três ferramentas foram desenvolvidas e acrescentadas ao ambiente desse curso: agenda, bloco de notas e de recados. A escolha destas três ferramentas deveu-se às facilidades adicionais que elas disponibilizam para os alunos. Durante a revisão bibliográfica, a aluna observou que alguns cursos já apresentavam essas funcionalidades.

A montagem do *site* e enquadramento do mesmo no *layout* proposto foram realizados em conjunto com o desenvolvimento das ferramentas bloco de notas, agenda e recados. Os requisitos fundamentais para o desenvolvimento dessas ferramentas foram os estudos aprofundados sobre as linguagens de programação para Internet, ASP.NET e C#, bem como do *framework.NET*.

Os estudos das linguagens ASP.NET e C# viabilizaram também o desenvolvimento do aplicativo para implementação na “*web*” do MEF para cálculo de elementos de barra, utilizando o *framework.NET* e essas linguagens. A base deste programa e sua estruturação de classes já haviam sido desenvolvidos pela aluna durante a realização de disciplinas do mestrado, onde a linguagem de programação utilizada foi o Java e o aplicativo não podia ser executado em ambiente web.

A etapa final do desenvolvimento foi a publicação na “*web*” e testes do ambiente digital do curso e do aplicativo (sistema computacional), com suas ferramentas de navegação e

de interação com os alunos. Esse ambiente está publicado no NúcleoEAD, Escola Virtual do CADTEC, cujo endereço é www.cadtec.dees.ufmg.br/NucleoEAD.

1.4. CONTEÚDO POR CAPÍTULOS

Além deste capítulo introdutório, esta dissertação será apresentado em outros sete capítulos: capítulo 2 – Revisão Bibliográfica, capítulo 3 – Recursos Tecnológicos Utilizados, capítulo 4 – Concepção da Plataforma para EAD, capítulo 5 – O Curso de Introdução ao Método dos Elementos Finitos, capítulo 6 – Ferramentas de Apoio a EAD Desenvolvidas, capítulo 7 – O Programa do Método dos Elementos Finitos e capítulo 8 - Conclusão.

Neste primeiro capítulo foram apresentados os objetivos da dissertação em questão, a motivação e a metodologia de desenvolvimento.

No capítulo 2 foi realizada uma revisão bibliográfica de todo o conteúdo abrangido durante o desenvolvimento da dissertação como:

- a internet e a *web*, o histórico e definições;
- a educação a distância, expondo sobre os conceitos e definições sob o ponto de vista de vários autores e suas principais características;
- o histórico da EAD até os dias atuais; quando a EAD através da “*web*” ganhou um papel predominante;
- a legislação vigente para esta modalidade de ensino e o contexto social econômico atual e como este contexto afeta a educação a distância;
- alguns sistemas educacionais informatizados pesquisados, com enfoque nas tecnologias utilizadas para o desenvolvimento e em suas características funcionais.

Os recursos tecnológicos utilizados para o desenvolvimento deste projeto foram expostos no capítulo 3. Primeiro, foram revisadas as linguagens utilizadas para internet, como:

- *Hipertext markup language* (HTML),
- *eXtensible Markup Language* (XML),
- *Cascading Style Sheets* (CSS),
- JavaScript, que foi a linguagem de script utilizada para dar mais interatividade às páginas,
- *Action Script* que é a linguagem utilizada pelo Flash (programa utilizado para desenvolver as animações),
- *Active Server Pages* (ASP) clássico,
- *Unified Modeling Language* (UML), utilizada para modelagem de sistemas computacionais.

O *Framework.NET*, tecnologia da *Microsoft* para aplicações e serviços *web*, também está apresentado no capítulo 3, incluindo as linguagens ASP.NET e o C#, utilizadas no desenvolvimento do curso e do aplicativo de Introdução ao Método dos Elementos Finitos.

Ainda no capítulo 3 foi feita uma introdução de conceitos necessários para a escolha de um banco de dados para integrar a proposta da plataforma para EAD e para a montagem do banco de dados do curso desenvolvido como parte desse projeto de mestrado. Os *software* utilizados para desenvolvimento da concepção da plataforma e do curso estão descritos no capítulo 3.

O capítulo 4 apresenta a concepção da plataforma tecnológica para EAD. Esta concepção foi elaborada a partir da proposição da base tecnológica (sistema operacional, servidor, linguagens e banco de dados) e da definição de objetivos e

diretrizes para a mesma, estabelecidos a partir da experiência do CADTEC, da observação de sistemas existentes e da análise de bibliografia pertinente ao assunto.

Um outro produto deste trabalho de mestrado apresentado no capítulo 5 é o curso de Introdução ao Método dos Elementos Finitos. Primeiramente, é realizada uma fundamentação teórica para desenvolvimento de conteúdos digitais. Posteriormente, os objetivos do curso são apresentados, seguidos da estruturação das páginas e integração ao NucleoEAD, Escola Virtual do CADTEC. A definição do escopo do conteúdo digital e a explicação de cada unidade, que compõem o curso, também são apresentadas.

O capítulo 6 apresenta as três ferramentas adicionais desenvolvidas e implantadas no curso de Introdução ao Método dos Elementos Finitos, que servem de apoio a EAD. Estas ferramentas são o bloco de notas, recados e agenda.

O aplicativo do Método dos Elementos Finitos para cálculo de elementos de barra está apresentado no capítulo 7, com a fundamentação da formulação utilizada no programa. A estruturação das classes e suas funções também estão apresentadas neste capítulo. Além disso, há uma breve explanação das tecnologias utilizadas no desenvolvimento do programa.

O capítulo 8 apresenta as conclusões sobre o trabalho desenvolvido e levanta possibilidades de trabalhos futuros a partir desta dissertação de mestrado.

2

REFERENCIAL TEÓRICO

2.1. A INTERNET E A WEB

2.1.1. Definições

A *Internet* é um grupo de redes de computadores organizadas logicamente, segundo uma hierarquia, cuja arquitetura compõe-se de três camadas: o cliente, o servidor de aplicação e o servidor de dados, permitindo a interação entre pessoas, dispersas geograficamente e temporalmente, para troca de informações. A *internet* oferece um conjunto de serviços como acesso e compartilhamento de arquivos de dados, incluindo sons, imagens e textos, troca de mensagens eletrônicas (*emails*), compartilhamento de informações *on-line*, a partir de servidores (*web* ou *World Wide Web* – WWW), dentre outros.

A *internet* possibilita a troca de informações de todos os gêneros e amplia a capacidade de comunicação e cooperação entre os povos e culturas. A velocidade e a quantidade de informações disponíveis e acessíveis aumentaram enormemente com a *internet*, possibilitando colaborações em pesquisas e desenvolvimentos científicos entre pessoas localizadas geograficamente distantes entre si. O correio eletrônico, por exemplo, está

re-estruturando o modo como as pessoas comunicam. Segundo definição de SANTOS¹, “A Internet causa a impressão de ser a maior biblioteca do mundo, sendo, de fato, um banco de dados *on-line* com tal escopo e alcance que permite o acesso a maior quantidade de informação à qual o ser humano jamais teve acesso.”.

A internet vem ajudando no processo de descentralização da informação. Qualquer pessoa pode (facilmente e a um custo muito baixo) não só ter acesso a informações localizadas nos mais distantes pontos do globo como também criar, gerenciar e distribuir informações em larga escala, no âmbito mundial, algo que somente uma grande organização poderia fazer usando os meios de comunicação convencionais. Isso com certeza afeta substancialmente toda a estrutura de disseminação de informações existente no mundo, a qual é controlada primariamente por grandes empresas.

A internet funciona a partir de protocolos ou *frameworks* de aplicação, ou seja, um conjunto de aplicativos abertos para que os usuários criem seus aplicativos para a internet. O protocolo para transmissão de aplicações e comunicação pela internet é o *Transmission Control Protocol/Internet Protocol* (TCP/IP).

O TCP/IP suporta o endereçamento de pacotes a partir de um nó de rede, definindo o melhor caminho que os pacotes irão tomar, orientando o tráfego de informações, definindo o endereçamento e envio de dados. Estes protocolos definem como aparelhos eletrônicos devem ser conectados à internet e como os dados devem ser transmitidos entre eles, ou seja, gerenciam o empacotamento de dados a serem enviados para múltiplos destinos. Representam, na verdade, um conjunto de protocolos responsáveis pelo gerenciamento da comunicação: TCP – responsável pela comunicação entre aplicações, IP – responsável pela comunicação entre computadores. Alguns outros protocolos que fazem parte do TCP/IP são:

FTP - *File Transfer Protocol* para "baixar" arquivos

NNTP - *Network News Transfer Protocol* para acessar grupos de discussão

¹ Santos, N. Internet e Web, Notas de Aula, março, 2000, disponível em <http://www.persocom.com.br>, 07/11/06.

Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3) e Internet Message Access Protocol (IMAP) para acessar correio eletrônico

IRC - *Internet Relay Chat*, para conduzir conversações multiusuários apoiadas em rede

URL - *Universal Resource Locator*, padrão para localizar recursos da rede

MIME - *Multipurpose Internet Mail Extensions*, padrão para especificar a forma pela qual documentos transmitidos estão codificados.

A arquitetura do TCP/IP se dá através de camadas. Cada camada encapsula funções e presta serviços à camada superior. Essa arquitetura foi baseada no modelo *Open Systems Interconnection (OSI)* da *International Standard Organization (ISO)*, (TOLENTINO, 2002).

Os serviços da Internet são acessados através dos protocolos descritos e da interação cliente-servidor. A figura 1 ilustra alguns serviços da internet e esquematiza o conceito de rede de computadores. Cliente pode ser entendido como a máquina que faz as requisições de serviços e os servidores são as máquinas onde estes serviços estão armazenados para possibilitar a disponibilização aos usuários. Uma transação na internet acontece sem que o servidor saiba o que está acontecendo no cliente até que este lhe envie dados. O cliente envia dados ou faz requisições através de uma interface gráfica, a requisição é processada nos servidores de aplicação (onde as aplicações estão armazenadas) que por vezes, consulta o servidor de dados (onde os dados necessários para execução das aplicações estão armazenados) e envia a apresentação dos dados trabalhados ao cliente através da internet. Os clientes são identificados na rede através de seu endereço IP.

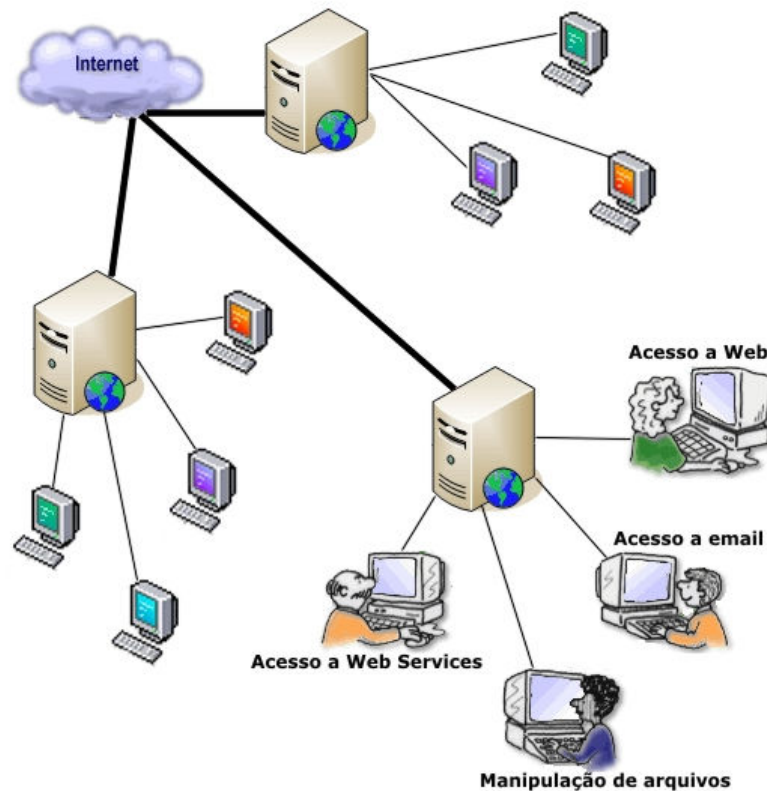


FIGURA 1 – Internet e alguns de seus serviços

A *web* ou a *world wide web* é um serviço da Internet, uma rede virtual a partir da qual foi possível o acesso de multi-usuários a programas (páginas) gerados por plataformas diferentes, tornando os serviços disponíveis na Internet totalmente transparentes para o usuário e possibilitando, ainda, a utilização da multimídia no processo de transferência da informação. A *web* é um banco de dados ou servidor de aplicações que contém informações que podem ser manipuladas através de *software* de navegação (*browsers*). Um servidor WWW pode incluir texto, som, imagem e até mesmo imagem de vídeo.

O padrão de comunicação utilizado pela *web* é o *Hypertext Transport Protocol* (HTTP), que é o padrão capaz de requisitar e enviar páginas *web*, ou seja, cuidar da comunicação entre um servidor e um *browser* da *web* (o cliente).

As conhecidas páginas da *web* estão armazenadas em servidores. Os servidores transmitem estas páginas aos clientes quando são solicitadas através de um endereço

URL, que identificam cada página *web*, utilizando o protocolo (ou serviço) HTTP, como por exemplo `http://www.nome.com.br`. Ao solicitar esta página, o cliente está solicitando uma conexão com o servidor WWW dentro do domínio `nome.com.br`. HTTP também é um protocolo contido no TCP/IP.

Os servidores HTTP são programas que dependem de uma solicitação para executar uma função. Quando o cliente solicita páginas estáticas (contêm apenas dados estáticos, que não precisam ser trabalhados) o servidor apenas recupera a página solicitada, localizando-a em seu domínio através do endereço solicitado e a envia através do protocolo HTTP. Quando o cliente solicita páginas dinâmicas (algum conteúdo precisa ser trabalhado antes de ser enviado ao cliente), programas são executados nos servidores e apenas os resultados obtidos são enviados ao cliente através do HTTP.

2.1.2. Histórico

As redes de computadores tiveram início nos anos 60 e 70, mas eram redes locais (*Local Area Network* - LAN'S). Em conjunto com o desenvolvimento das redes locais, começaram a ser desenvolvidas as *Wide Area Networks* (WANs) para comunicação entre computadores distantes. Estas duas tecnologias eram, entretanto, incompatíveis. No intuito de compatibilizá-las, surgiu o projeto *Advanced Research Projects Agency* (ARPA) em 1957, financiado pelo Departamento de Defesa Americano, que passou a chamar o projeto de *Internetwork*. (LIMA, 2002)

A tecnologia de divisão do conteúdo em partes ou pacotes antes de ser transmitido, surgiu em 1962. Através dessa tecnologia, as informações poderiam trafegar em redes interligadas, mesmo utilizando máquinas e sistemas operacionais diferentes.

O primeiro programa de e-mail foi criado em 1972 por Roy Tomlinson, quando nasceu também o símbolo @. O IRC, serviço de bate papo pela internet foi criado em 1988. A *web* ou *world wide web* iniciou sua operação em 1989.

Nos anos 80, muitas universidades se conectaram a essa rede, desviando o uso da rede para uma motivação mais cultural e acadêmica. Algumas universidades que primeiro se conectaram a essa rede foram: *Harvard University*, *Stanford University*, *University of*

Illinois e Carnegie Mellon University. Ainda nos anos 80 a *National Science Foundation* (NSF) dos EUA desenvolveu uma rede de fibra ótica de alta velocidade conectando centros de computação. Essa rede teve um papel fundamental no desenvolvimento da Internet por reduzir substancialmente o custo da transmissão de dados para os computadores existentes. Vários outros trechos de transmissão de dados em rede de grande capacidade e velocidade surgiram: NSINET da NASA, ESNET do *U.S. Department of Energy*, NORDUNET na Europa. Com isso, apareceram os primeiros provedores comerciais, tornando público o acesso a Internet (TOLENTINO, 2002).

No Brasil, em 1975, a EMBRATEL começou a instalar e explorar a rede nacional de transmissão de dados. Em 1979, políticas globais para os serviços de teleinformática começaram a ser estabelecidos pela Secretaria Especial de Informação juntamente com o Ministério das Comunicações. Em 1989, o Instituto Brasileiro de Análises Sociais e Econômicas (IBASE), iniciou a operação do primeiro serviço internacional de correio e conferências eletrônicas operado por uma entidade privada no país, o Alternex. A Rede Nacional de Pesquisa (RNP) foi criada no mesmo ano, com objetivo de desenvolver a infra-estrutura de rede no âmbito nacional. Em 1990, a FAPESP tornou-se o órgão responsável pelo registro da internet no Brasil, administrando o domínio br e sendo responsável pela distribuição dos números IP. O modelo de rede utilizado aqui foi o modelo do *U.S. National Science Foundation*, estruturando a rede nos níveis nacional, regional e institucional. Apenas em 1994 é que os primeiros servidores *web* do Brasil começaram a funcionar, na Universidade Federal do Rio de Janeiro e na Universidade Federal de Santa Catarina. Em 1995, a internet deixou de ser restrita ao meio acadêmico, atendendo todos os setores da sociedade.

Deixando sua estrutura essencialmente acadêmica, a internet, nos últimos quinze anos, apresentou um crescimento enorme, como pode ser observado na FIG. 2. Em 1991 eram 4,4 milhões de usuários, em 2006 esse número passou para 1,086 bilhão. O número de *sites* estimados na *web* hoje é de 80 milhões e o número de contas de e-mail ativas é 1,4 bilhão, segundo Heitor Shimizu (2006). A capacidade instalada da internet cresce cada vez mais, como pode ser observado na FIG. 3, figurando um futuro cada vez mais

promissor para a internet e para as relações que se processam através da mesma, como por exemplo, a EAD.

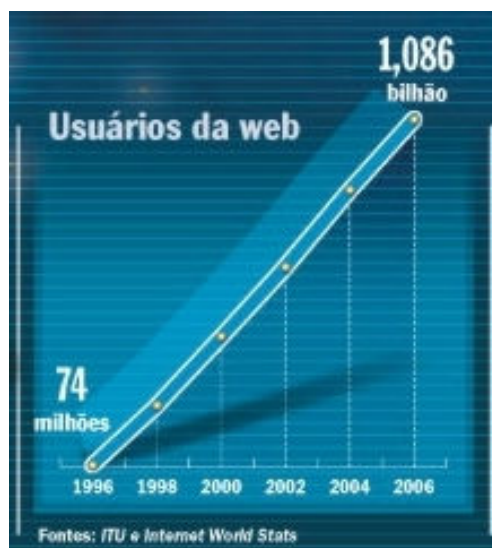


FIGURA 2 - Usuários da Web

Fonte: SHIMIZU (2006)



FIGURA 3 - Infra-estrutura da internet

Fonte: SHIMIZU (2006)

2.2. A EDUCAÇÃO A DISTÂNCIA

2.2.1. Definições

O artigo 1 do Decreto 5.622 de 20 de dezembro de 2005 caracteriza a Educação a Distância como modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos. NUNES (1993/1994) faz uma coletânea das definições de Educação a Distância, segundo vários autores.

Educação a distância é uma forma sistematicamente organizada de auto-estudo onde o aluno se instrui a partir do material de estudo que lhe é apresentado, onde o acompanhamento e a supervisão do sucesso do estudante são levados a cabo por um grupo de professores. Isto é possível de ser feito a distância através da aplicação de meios de comunicação capazes de vencer longas distâncias. O oposto de "educação a distância" é a "educação direta" ou "educação face a face": um tipo de educação que tem lugar com o contato direto entre professores e estudantes (DOHMEM, 1967 apud NUNES, 1993/1994).

Educação/ensino a distância é um método racional de partilhar conhecimento, habilidades e atitudes, através da aplicação da divisão do trabalho e de princípios organizacionais, tanto quanto pelo uso extensivo de meios de comunicação, especialmente para o propósito de reproduzir materiais técnicos de alta qualidade, os quais tornam possível instruir um grande número de estudantes ao mesmo tempo, enquanto esses materiais durarem. É uma forma industrializada de ensinar e aprender (PETERS, 1973 apud NUNES, 1993/1994).

Ensino a distância pode ser definido como a família de métodos instrucionais onde as ações dos professores são executadas a parte das ações dos alunos, incluindo aquelas situações continuadas que podem ser feitas na presença dos estudantes. Porém, a comunicação entre o professor e o aluno deve ser facilitada por meios impressos, eletrônicos, mecânicos ou outros (MOORE, 1973 apud NUNES, 1993/1994).

O termo "educação a distância" esconde-se sob várias formas de estudo, nos vários níveis que não estão sob a contínua e imediata supervisão de tutores presentes com seus alunos nas salas de leitura ou no mesmo local. A educação a distância se beneficia do planejamento, direção e instrução da organização do ensino (HOLMBERG, 1977 apud NUNES, 1993/1994).

Embora possam ser notadas pequenas transformações ao longo dos anos nas definições de EAD, em todas elas, o principal ponto observado é a separação física entre alunos e professores. Outros aspectos que podem ser observados é a importância da organização, planejamento e sistematização, a necessidade da utilização de diferentes meios de comunicação, sistemas de gestão e operacionalização específicos para transmissão do conhecimento e superação das barreiras físicas, a utilização de materiais didáticos bem organizados, a necessidade da comunicação bidirecional, estabelecendo relações interpessoais participativas entre alunos e professores de qualidade, auxiliando que o aluno construa seu conhecimento e a importância dos alunos envolvidos no processo adotarem uma postura ativa e consciente, através de estudos individualizados e independentes. Em outras palavras, educação a distância pode ser definida como todas as formas de ensino-aprendizagem nas quais os alunos e/ou os professores se comunicam de qualquer maneira além de reuniões presenciais em sala de aula. Neste contexto, EAD pode ser compreendida como uma forma de educação independente de distâncias.

Cada vez mais pessoas vêm na Educação a Distância uma maneira de democratização do saber, formação, capacitação e atualização profissional, já que possibilita grande

flexibilidade espacial e temporal entre os envolvidos no processo educativo. A EAD possibilita ainda, uma maior adaptação às possibilidades e aspirações individuais dos estudantes, sem que isto venha em detrimento da qualidade acadêmica do material instrucional. Para muitos, a Educação a Distância representa o único meio de se conseguir uma educação aberta e continuada.

A principal diferença entre a Educação a Distância e a Educação presencial é que, na educação a distância, o aluno é obrigado a adotar uma postura ativa diante do processo de aprendizagem. O aluno constrói e é responsável pelo seu conhecimento, desenvolve competências, habilidades, atitudes e hábitos relativos ao estudo, à profissão e à sua própria vida, no tempo e local que lhe são adequados. Não há a presença física constante de professores ou orientadores.

2.2.2. Histórico

A Educação a Distância apresenta uma longa história de experimentações. Há muito tempo, pessoas já se comunicam e transmitem informações através de cartas. No final do século XVII algumas experiências de educação por correspondência haviam sido iniciadas, como por exemplo, o curso de taquigrafia por correspondência do professor Cauleb Philips anunciado em 1728 pela Gazeta de Boston, EUA. Porém, foi no século XIX que os cursos por correspondência tiveram um grande desenvolvimento. Alguns cursos dessa época podem ser citados: em 1840, na Inglaterra, Issac Pitman elabora outro curso de taquigrafia por correspondência; em 1873 é realizado um curso sobre medidas de segurança no trabalho de mineração no *International Correspondence Institute*, na Pennsylvania.

Após estas experiências, várias outras iniciativas foram tomadas, como em 1922 foi criado pela União das Repúblicas Socialistas Soviéticas (URSS) um programa de ensino por correspondência objetivando formar trabalhadores em dois anos. A *Universidad Nacional de Educación a Distancia* (UNED) em Madri, Espanha, criada em 1922 é outra organização que merece destaque em EAD por oferecer diversos cursos de graduação, pós-graduação e extensão, utilizando esta forma de ensino (LIMA, 2002).

Com o grande avanço tecnológico e dos meios de comunicação, a EAD ganhou grande impulso e houve a possibilidade de ser realizada através de outros meios de comunicação, como a TV, o rádio, telefone e por último o computador e a internet.

O surgimento dos computadores trouxe amplas perspectivas para a educação, provocando uma verdadeira revolução na concepção de ensino e de aprendizagem. O computador vem se mostrando bastante útil nos processos de aprendizagem. Os programas e *software* educacionais desenvolvidos inicialmente, tentavam imitar as salas de aula convencionais e se comportavam apenas como “máquinas de ensinar”. Atualmente, têm surgido usos próprios e específicos do computador como uma ferramenta de auxílio à educação.

O marco inicial do ensino utilizando a informática foi em 1924, quando o Dr. Sidney Pressey criou uma máquina para corrigir testes de múltipla escolha, e depois em 1950, quando Skinner, utilizando o conceito de instrução programada (divisão do conteúdo a ser ensinado em módulos organizados logicamente), desenvolveu uma máquina para ensinar. A produção de materiais didáticos utilizando o conceito de instrução programada era difícil e não possuía padronização, fazendo com que essa forma de educação não desse certo.

Com o surgimento dos computadores houve a possibilidade de implementação das técnicas de instrução programada com maior flexibilidade, nascendo assim, a instrução auxiliada por computador (*Computer Aided Instruction* – CAI), ou Programas Educacionais por Computador (PEC).

Nos anos 60 houve grande investimento do governo americano para desenvolvimento de CAIs mas, como os computadores eram muito caros, isto só era viável em universidades, onde diversos cursos puderam ser ministrados por computador. O CAI mais conhecido foi o PLATO, elaborado na década de 70, pela *Control Data Corporation*, uma fábrica de computadores, e a Universidade de Illinois. Este sistema utilizava terminais sensíveis a toque e vídeo com alta capacidade gráfica. Em sua última versão, o PLATO apresentava cerca de 8.000 horas de material instrucional produzido por cerca de 3.000 autores.

Apenas após os microcomputadores terem se tornado acessíveis é que a utilização de CAI na educação proliferou, permitindo ampla diversificação: tutoriais, programas de demonstração, exercício-e-prática, avaliação do aprendizado, jogos educacionais e simulação. Outros usos do computador como auxílio à educação também surgiram.

Atualmente, são inúmeros os *software* educacionais existentes no mercado, mas o mais interessante é que o computador passou a ser não só uma máquina de ensinar, mas uma ferramenta educacional, de apoio, complementação, aperfeiçoamento, mudando a abordagem, foco e qualidade do ensino e da maneira de ensinar. O computador não ensina, mas promove o ensino. O computador pode ser visto também, como um estímulo para os estudantes exercitarem capacidades de procurar e selecionar informações, aprender independentemente. Estas mudanças estão de acordo com as mudanças ocorridas na conceituação dos processos de aprendizagem. A escola, professores e computador não devem ser responsáveis por ensinar, mas por promover o aprendizado. O computador é o criador de ambientes de aprendizado e facilitador dos processos pelos quais os alunos adquirem conhecimentos.

Na EAD, as diferentes modalidades de usos do computador vão sempre existir e são importantes porque cada estudante se adapta melhor a cada uma delas, já que atendem a objetivos específicos.

O termo Tecnologia Educacional refere-se ao estudo dos meios como geradores da aprendizagem, ou o estudo do ensino como processo tecnológico. O termo EAD traz um significado mais amplo de todo o processo de ensino/aprendizado que apresenta barreiras espaços-temporais e que utiliza recursos tecnológicos e meios de comunicação para vencê-las. Assim, a história da EAD está intimamente ligada com o desenvolvimento e surgimento da Tecnologia Educacional.

Nos anos 50 e 60 o termo Tecnologia Educacional era definido como o estudo dos meios geradores de aprendizagem. Nos anos 70, passou a ser entendido como o estudo do ensino como processo tecnológico. Atualmente, e de uma maneira mais geral, Tecnologia Educacional refere-se à aplicação da técnica à resolução de problemas educativos, ou seja, é o estudo científico das práticas educativas. Assim, a tecnologia educacional ocupa-se do surgimento e desenvolvimento das novas tecnologias

(impressos, vídeo, TV, rádio, áudio, internet, computador) aplicadas aos processos de educação.

Pode-se observar que o desenvolvimento da tecnologia Educacional coincide com o desenvolvimento da EAD. Pode-se identificar a origem da tecnologia educacional nos anos 50, principalmente nos Estados Unidos com grande desenvolvimento de aparelhos e introdução de outros campos científicos como a psicologia, nos processos de aprendizagem. O emprego de novas tecnologias na educação (como os computadores, por exemplo) teve início nos anos 70, mas apresentava-se ainda de forma limitada, principalmente na América Latina devido aos altos custos dessas novas tecnologias. A medida que essas tecnologias foram se tornando mais acessíveis, a EAD apresentou grande avanço já que o desenvolvimento dessas tecnologias representaram também o surgimento de novos meios para vencer as barreiras físicas da EAD, dando mais flexibilidade à mesma.

A história da EAD no Brasil teve início em 1922 com um programa na Rádio Sociedade do Rio de Janeiro. Em 1941 houve a criação do Instituto Universal Brasileiro, programa de EAD por correspondência com grande repercussão nacional. Na década de 50 surgiu o Movimento de Educação de Base (MEB), utilizando o rádio para alfabetização da população nas regiões Norte e Nordeste. Durante as décadas de 60 e 70 surgem os programas TV Escola (TVE Ceará), TV Educativa (Maranhão), Programa Nacional de Tele-Educação (PRONTEL) e as fundações Padre Anchieta (TV Cultura – SP), Roberto Marinho (TV Globo), que cumprem até hoje o importante papel no desenvolvimento de programas educativos.

No final do século passado, após grande avanço tecnológico e democratização de meios de comunicação como a internet, a EAD aparece com todo vigor, como modalidade de ensino que revisa seus princípios fundamentais e incorpora toda a tecnologia possível, como hipertextos e multimídias para auxiliar no processo educativo e para espelhar a nova representação da sociedade. Uma nova tendência que surge a partir desse período é a criação de universidades virtuais, através de consórcio de cooperação de várias universidades, integradas pela internet e redes de videoconferência.

Algumas iniciativas foram marcantes, representando esta nova etapa da EAD:

- A Universidade Virtual Pública do Brasil (UniRede), é uma rede das universidades públicas brasileiras. Seu objetivo é pesquisar os processos de aprendizagem, difundindo os processos metodológicos de EAD, através dos sistemas de comunicação. A UniRede foi lançada em 1999, oferece cursos a distância em níveis de graduação, pós-graduação, extensão e educação continuada. Fazem parte dessa rede 80 instituições públicas do ensino superior. Além de cursos, a UniRede oferece uma série de serviços acadêmicos, como ferramentas de tradução, bancos de teses, catálogos de pesquisadores, etc.
- O Instituto Universidade Virtual Brasileira (Uvb) é uma associação de 10 instituições de ensino de nível superior. O objetivo da Uvb é modificar os paradigmas educacionais e buscar novas práticas e metodologias para EAD. Recebeu em 8 de maio de 2003 o credenciamento e a autorização do MEC para lançar quatro bacharelados que serão oferecidos pela Internet.
- O Programa TV Escola é uma das ações prioritárias da Secretaria de Educação a Distância (SEED). Lançado experimentalmente no Piauí, em setembro de 1995, foi ao ar para todo o País, em caráter definitivo, em 4 de março de 1996. Os principais objetivos da TV Escola são o aperfeiçoamento e a valorização dos professores da rede pública, o enriquecimento do processo de ensino-aprendizagem e a melhoria da qualidade do ensino.
- O Programa Nacional de Informática na Educação (ProInfo) foi criado em 9 de abril de 1997 e desenvolvido pela SEED, do Ministério da Educação, em parceria com governos estaduais e municipais. O objetivo principal é introduzir o uso das tecnologias de informação e comunicação nas escolas da rede pública
- A UAB Universidade Aberta do Brasil foi criada em 2005, destinada principalmente à capacitação de professores da rede pública de ensino, tem o objetivo de formar um sistema integrado de instituições públicas para levar o ensino superior a municípios onde a oferta não existe ou é insuficiente para atender a população. A meta da UAB é abrir, até 2010, 830 pólos de ensino espalhados pelo Brasil. Em 2007 foram abertos 291 pólos e em 2008 espera-se abrir mais 271 pólos.

- O e-Tec-Brasil é um programa, lançado pela SEED em 2007 que propõe articular as instituições públicas federais, estaduais e municipais que oferecem ensino técnico nível médio a interessados em ofertar seus cursos na modalidade a distância.

As novas tecnologias da informação, incorporadas à educação a distância, promoveram significativas mudanças nos paradigmas educacionais, em função das possibilidades interativas desses instrumentos. As novas tecnologias estimulam o desenvolvimento de metodologias educacionais que propiciam a busca incessante do auto-aprimoramento, no sentido de educação continuada. O que se vê atualmente é a emergência de uma política de ação para a EAD, em todos os níveis, já que a sociedade exige qualificações cada vez mais elevadas, ampliando as necessidades educacionais da população. A divulgação de experiências, criação de espaços para discussão do tema e formação profissional própria é outra necessidade visível, por tratar de um assunto complexo e que envolve novas concepções da aprendizagem.

Segundo dados extraídos do *site* do MEC (<http://www.mec.gov.br>, acesso em: 10/02/08), no censo da Educação superior de 2006, o crescimento do número de cursos de educação superior a distância é o maior destaque. De 2003 a 2006 o número passou de 52 para 349, um aumento de 571%. O número de estudantes de educação a distância também teve um aumento em 315%, de 40 mil em 2003 para 207 mil em 2006. Assegurar a qualidade destes cursos torna-se o maior desafio dos educadores e legisladores em EAD.

Existe hoje, um documento elaborado pela Secretaria de Educação a Distância (SEED) para orientar a elaboração de um projeto de curso de graduação a distância: “Indicadores de qualidade para cursos de graduação a distância”. A primeira versão deste documento foi elaborada em 2003 e revisada em 2007. Estes referenciais de qualidade circunscrevem-se no ordenamento legal vigente em complemento às determinações específicas da Lei de Diretrizes e Bases da Educação, do decreto 522 de 20 de dezembro de 2005, do decreto 5773 de junho de 2006 e das portarias Normativas 1 e 2 de 11 de Janeiro de 2007. Este documento foca principalmente os seguintes tópicos:

- Concepção de educação e currículo no processo de ensino e aprendizagem
- Sistemas de comunicação
- Material didático
- Avaliação
- Equipe multidisciplinar
- Infra-estrutura de apoio
- Gestão Acadêmico – administrativa
- Sustentabilidade financeira

2.2.3. Legislação da EAD no Brasil

A Secretaria de Educação a Distância (SEED) é o órgão do governo destinado ao desenvolvimento e regulamentação da EAD no Brasil. A SEED trabalha com os demais órgãos do Ministério da Educação, em conjunto com as Secretarias de Educação dos estados, municípios e Distrito Federal, com universidades, centros de pesquisas, televisões e rádios educativas e outras instituições que utilizam a metodologia de educação a distância. A SEED foi criada em 27 de maio de 1996. Os três objetivos principais da SEED são: desenvolvimento de projetos estratégicos, institucionalização da educação a distância no país e articulação do campo institucional e da sociedade civil.

A lei n.º 9394 de 20 de dezembro de 1996 (Lei de diretrizes e Bases da Educação nacional) que foi regulamentada pelo Decreto número 5622 publicado no D.O.U. de 20/12/05 e com normatização definida na Portaria Ministerial n.4361 de 2004, estabelece as bases legais da Educação a Distância no Brasil, colocando a EAD como uma modalidade de educação plenamente integrada ao sistema de ensino.

Esta lei estabelece diretrizes para a EAD no ensino médio e fundamental, no ensino superior (graduação) e educação profissional em nível técnico. Em 3 de abril de 2001, a Resolução n.º1 estabeleceu as normas para programas de pós-graduação.

Relativamente à educação básica, os cursos a distância que oferecem certificados e diplomas deverão ser oferecidos por instituições devidamente credenciadas para esse fim. O credenciamento é realizado pelos órgãos dos sistemas estadual ou municipal que avaliarão as propostas de cursos à distância. De acordo com o artigo 30º da Lei de diretrizes e bases, e o decreto que trata dessas regulamentações (nº 5622, que revogou o n.º 2494/98 e o n.º 2561/98), as instituições credenciadas para a oferta de educação a distância, poderão solicitar autorização, junto aos órgãos normativos dos respectivos sistemas de ensino, para oferecer os ensinamentos fundamental e médio a distância exclusivamente para complementação de aprendizagem ou em situações emergenciais.

Para oferecer cursos de graduação e educação profissional em nível tecnólogo, a instituição precisa se credenciar junto ao Ministério da Educação e Cultura (MEC). Como para os cursos presenciais, o processo será analisado na Secretaria de Educação Superior - SESU e pelo Conselho Nacional de Educação. A qualidade do projeto da instituição é o foco da análise.

Os cursos de pós-graduação (pós-graduação stricto sensu, mestrado e doutorado e pós-graduação lato sensu, especialização) a distância devem ser oferecidos somente por instituições credenciadas pela União e devem obedecer exigências de autorização, reconhecimento e renovação de reconhecimento estabelecidas na resolução. Os cursos de pós-graduação lato sensu oferecidos a distância deverão incluir, necessariamente, provas presenciais e defesa presencial de monografia ou trabalho de conclusão de curso. As resoluções e/ou artigos que tratam dessas regulamentações são: capítulo V do decreto n.º 5622/05, artigo 11 da resolução n.º1, artigo 3 da lei n.º 9394/96, artigo 80 da lei n.º 9394/96, artigo 24 do decreto n.º 5622/05.

Algumas outras regulamentações da EAD podem ser encontradas nos seguintes meios normativos:

- Leis, Decretos e Portarias

- Lei nº 9.394, de 20 de dezembro de 1996
Lei de Diretrizes e Bases da Educação Nacional.
- Decreto nº 5.622, de 19 de dezembro de 2005
Regulamenta o art. 80 da Lei nº 9.394, de 20 de dezembro de 1996, que estabelece as diretrizes e bases da educação nacional.
- Decreto nº 2.561, de 27 de abril de 1998
Altera a redação dos artigos 11 e 12 do Decreto nº 2.494.
- Decreto nº 2.494, de 10 de fevereiro de 1998
Regulamenta o Art. 80 da LDB (Lei nº 9.394/96).
- Portaria nº 4.363, de 29 de dezembro de 2004
Dispõe sobre a autorização e reconhecimento de cursos sequenciais da educação superior.
- Portaria nº 301, de 7 de abril de 1998
Normatiza os procedimentos de credenciamento de instituições para a oferta de cursos de graduação e educação profissional tecnológica a distância.
- Resoluções e Pareceres do CNE
 - Resolução CNE/CES nº 1, de 3 de abril de 2001
Estabelece normas para o funcionamento de cursos de pós-graduação.
 - Resolução nº 1, de 26 de Fevereiro de 1997
Fixa condições para validade de diplomas de cursos de graduação e de pós-graduação em níveis de mestrado e doutorado, oferecidos por instituições estrangeiras, no Brasil, nas modalidades semi-presenciais ou a distância.
 - Parecer nº 78/96, aprovado em 7 de outubro de 1996
Assunto: Solicita estudo sobre a adoção de medidas coibindo a revalidação de diplomas de graduação e pós-graduação na modalidade de ensino a distância, oferecidos pelo Colégio Brasileiro de Aperfeiçoamento e Pós-Graduação-COBRA.
- Portarias que regulamentam o Credenciamento de Instituições de Ensino Superior (disponível em www.mec.gov.br)
 - Portaria nº 335, de 6 de fevereiro de 2002
Criar a Comissão Assessora para a Educação Superior a Distância
 - Portaria nº 4.059, de 10 de dezembro de 2004
Substitui a portaria 2.253/01 que normatizava os procedimentos de autorização para oferta de disciplinas na modalidade não-presencial em cursos de graduação reconhecidos.

Atualmente, são inúmeras as instituições credenciadas para oferta de cursos superiores a distância (graduação, cursos sequenciais e pós graduação *latu senso*). As instituições e os cursos oferecidos podem ser consultados no site do MEC.

2.2.4. Contexto Social Econômico e suas aplicações para EAD

Devido ao grande avanço das tecnologias da informação e comunicação, a educação a distância vem apresentando-se cada vez mais como uma forma das pessoas conseguirem uma educação continuada de qualidade, garantindo a democratização do acesso ao conhecimento e expandindo oportunidades de trabalho. A EAD oferece a flexibilidade espaço-tempo, tão importante atualmente, já que o tempo e os deslocamentos, tanto entre cidades como dentro das grandes cidades, estão cada vez mais difíceis. A EAD mostra-se ainda economicamente viável e possível mesmo em países como o Brasil, com grande extensão territorial e com a maioria da população dentro dos índices de pobreza.

Atualmente é imensa a quantidade de conhecimentos e informações produzidas nas diversas áreas de conhecimento, excedendo em muito, o número de informações que uma pessoa pode absorver ao longo da vida. O mercado mostra-se cada vez mais competitivo, deixando claro a necessidade das pessoas se auto-instruírem e estarem sempre em busca de uma educação continuada de qualidade. A EAD associada às novas tecnologias, além de apresentar a flexibilidade de espaço e tempo, é uma forma de potencializar o aprendizado das pessoas, favorecendo o desenvolvimento da autonomia dos alunos, oferecendo oportunidades de aprendizagem individualizada e respeitando o ritmo e estilo de aprender próprios de cada indivíduo. Além da auto-instrução, a EAD com as TIs possibilitam também que pessoas, independente de cultura, etnia ou economia participem, de forma cooperativa na construção do conhecimento; isto é a democratização do conhecimento.

A alternativa da aprendizagem em um ambiente de rede possibilita ainda que o aluno amplie seu horizonte além da realidade escolar devido ao grande número de informações disponibilizadas e o modo como essas informações estão estruturadas (o material didático deve ser fornecido corretamente estruturado), os conteúdos deixam de ser estanques, criando interfaces entre as diversas áreas do conhecimento e levando os alunos a desenvolverem habilidades para utilizar, relacionar, analisar e avaliar as informações recebidas, ou seja, a EAD pode se tornar uma poderosa ferramenta para que o aluno construa uma visão crítica do mundo que o cerca, o que é também uma exigência atual do mercado de trabalho.

A *web*, uma das tecnologias da comunicação surgidas recentemente, impulsionou consideravelmente a EAD através da internet. A *web* apresenta uma estrutura dinâmica e portátil, oferecendo possibilidade de integrar texto, som, vídeo, imagem, etc, resultando na possibilidade de desenvolvimento de materiais riquíssimos, estimulantes e adaptáveis ao estilo de cada aluno. As tecnologias da informação possibilitaram, acima de tudo, a criação de ambientes propícios para aprendizagem bastante motivadores, intuitivos e amigáveis o suficiente para compensarem, de certa forma, a falta de contato físico entre aluno e professor.

A viabilidade econômica da EAD cresceu bastante com o uso da internet, maximizando os recursos financeiros empregados em relação ao tamanho da população que pode ser atingida com a confecção de materiais didáticos bem estruturados e com o desenvolvimento e manutenção das estruturas de apoio necessárias, além dos professores e monitores.

A EAD pode ser vista também como uma questão sócio-econômica, ou seja, como uma alternativa para levar a educação em áreas de difícil acesso ou mais distantes dos centros de pesquisa.

O Brasil é um país de grande extensão territorial e, se não fosse pela falta de infraestrutura (disponibilidade de redes e máquinas) a EAD, através da utilização de computadores e da internet, seria uma excelente alternativa para a propagação do conhecimento e educação, até mesmo de comunidades distantes e com poucos recursos financeiros. Como menciona DOVICCHI (2003): "...a distribuição de documentos em hipertexto pode se tornar viável se for implementada em máquinas de baixo custo, com sistemas operacionais, servidores de *HyperText Transport Protocol* (http) e *software* educacionais gratuitos (...)”.

2.3. SISTEMAS EDUCACIONAIS INFORMATIZADOS

De acordo com DOVICCHI (2003), existem centenas de plataformas de EAD. Foi realizada pelo autor uma pesquisa com base na internet, e 246 destes programas foram

analisados e descritos em um documento de referência, sendo cerca de 20% deles gratuitos e 12% de código aberto.

Este item, Sistemas Educacionais Informatizados, foi escrito com o objetivo de apresentar algumas plataformas para EAD existentes, verificar suas principais características e ferramentas disponíveis. Esta pesquisa baseia-se em SANTOS (2005) e DOVICCH (2003), além de outras pesquisas realizadas na internet.

Segundo SANTOS (2005), os softwares educacionais ou plataformas tecnológicas para EAD existentes apresentam-se distribuídas em seis modalidades: aplicações hipermídia, *sites* educacionais, sistemas de autoria para cursos à distância, salas de aula virtuais, *frameworks* para aprendizagem cooperativa e ambientes distribuídos para aprendizagem cooperativa.

Seguindo a conceituação de SANTOS (2005), os sites educacionais reúnem conjuntos de funcionalidades como: bibliotecas, espaços para comunicação síncrona e assíncrona, áreas de transferências, etc, ou seja, reúnem diferentes formas de apoio ao trabalho docente e ao aprendizado autônomo dos estudantes.

Sistemas de autoria para cursos a distância são aplicações que permitem o desenvolvimento de páginas web de forma automática para o desenvolvimento de cursos. Algumas dessas aplicações permitem o desenvolvimento das páginas sem que o usuário tenha conhecimento do HTML e de outras linguagens para web e este processo pode ser realizado *online*.

As salas de aula virtuais, além de apresentarem ferramentas para o desenvolvimento de cursos *online*, ampliam os espaços para a interatividade entre professores e alunos, favorecendo a comunicação e cooperação. São sites educacionais associados aos sistemas de autoria de cursos. Nas salas de aula virtuais há o acompanhamento do aluno durante a aplicação do curso que é realizada por intermédio de um professor.

Os *Frameworks* para atividades cooperativas são aplicativos que permitem o desenvolvimento de ambientes para cooperação através da utilização de ferramentas disponíveis.

Ambientes Distribuídos para aprendizagem colaborativa são ambientes que apresentam ferramentas síncronas e/ou assíncronas de comunicação e que estimulam as discussões através de simulações ou metáforas de elementos educativos do mundo real. Os Ambientes Distribuídos para aprendizagem colaborativa não serão abordados nesse texto, pois fogem ao objetivo, já que apresentam as mesmas ferramentas de comunicação dos softwares educacionais anteriores.

A seguir serão apresentados alguns sistemas educacionais informatizados existentes. Estes sistemas apresentados estarão enquadrados nas categorias definidas por SANTOS (2005).

2.3.1. Sites Educacionais

Study Web

O *Study Web* é uma coleção de 28000 *sites*, sobre conteúdos curriculares separados por série escolar. Neste site há uma ferramenta de busca poderosa, que permite ao usuário encontrar os assuntos desejados de forma rápida e segura. O endereço para acesso é: <http://www.studyweb.com>

The Internet Public Library

Este site é uma biblioteca virtual, apresenta uma coleção variada de assuntos e temas, classificados por faixa etária, além de fornecer ferramentas de busca e navegação eficientes. O endereço para acesso é: <http://www.ipl.org/ref/>

The World Lecture Hall

Neste site há uma coleção de *links* para cursos a distância criados em todo o mundo, classificados em diversas categorias. O site apresenta ferramenta de tradução e inserção da referência de algum curso. Página para acesso: <http://www.utexas.edu/world/lecture/>

Escolanet

Este site faz parte de uma rede de sites comerciais que reúne informações e serviços educacionais como conteúdos das principais disciplinas do ensino fundamental, grupos de discussão, dicas para desenvolvimento de sites relacionados a EAD, consultoria, convênios, informação e sistemas de interatividade virtual para organizações educativas. Outros sites da rede são o “Ao Mestre Com Carinho” (www.aomestrecentcarinho.com.br) e “Publicado” (www.publicado.com.br). O site apresenta uma boa compilação de notícias sobre educação à distância. A navegação é livre. O endereço de acesso é: <http://www.escolanet.com.br>

Projeto Aprendiz

Este projeto trata-se de um site de caráter informativo. Nele são encontradas notícias na área de educação, além de grupos de discussão. Pode ser acessado pelo endereço: <http://www.uol.com.br/aprendiz/>

UOL Educação

O canal de educação do portal UOL oferece aos usuários notícias de última hora sobre o mundo da educação, materiais de apoio aos professores e ferramentas educativas para estudantes. Na seção Sala do Professor, de livre acesso, há planos de aula para educadores. Os pontos fortes do UOL Educação são os dicionários e tradutores online e os bancos de dados informativos (com arquivos do jornal Folha de S. Paulo e de diversas revistas brasileiras). O acesso é livre somente para a parte do conteúdo oferecido. O endereço de acesso é: <http://www.uol.com.br/educacao>

FLE3

É uma plataforma de aprendizagem baseada na Web ou, mais especificamente, uma aplicação servidora para aprendizagem colaborativa mediada por computador. Este *software* foi desenvolvido pela *University of Art and Design Helsinki*, apresenta código aberto e é gratuito. O objetivo do FLE3 é dar apoio aos aprendizes e grupos de trabalho nas áreas de arte e design. Possui uma ferramenta denominada “*Jamming Tool*”, que é um espaço compartilhado para construção colaborativa de artefatos digitais como

imagem, texto, áudio e vídeo. O endereço para acesso é: <http://fle3.uiah.fi/index.html> ou <http://www.nonio.uminho.pt/kitfle/> (versão modificada)

10 em tudo

O 10 em tudo funciona como um centro de estudos online voltado tanto para professores como estudantes. Os vestibulandos, por exemplo, têm a disposição um banco de dados que apresenta 27 mil questões de múltipla escolha divididas por disciplina e tópico. Aos educadores, o site oferece uma ferramenta chamada “gerador de provas”. O acesso é restrito aos assinantes. O endereço para acesso é: www.10emtudo.com.br

Alfabetização e Letramento

Lançado pela Editora Scipione, o Alfabetização e Letramento destina-se aos professores que atuam no início do ensino fundamental. Um dos principais objetivos do projeto é oferecer aos docentes uma síntese dos conhecimentos disponíveis sobre o processo de aquisição da leitura e da escrita das crianças de seis anos. Um bom diferencial do site são os *podcasts* (arquivos sonoros), atualizados mensalmente. A navegação é livre. O endereço para acesso é: <http://www.scipione.com.br/letramento/default.asp>

EdukBr

O EdukBr busca oferecer múltiplas possibilidades e propostas educacionais de qualidade. O portal agrega vários sites temáticos autônomos, como “Profs Online”, “Oficina de aprendizagem”, “Celeiro de projetos”, “Leitura & Escrita”, “Mochila nas Costas” entre outros. Destina-se a um público formador de opinião: professores e jovens brasileiros que usam a Internet de forma sistemática em seu cotidiano. O acesso é livre. endereço para acesso é: <http://www.edukbr.com.br>

EduOnline

Site simples e de fácil navegação, o EduOnline funciona como um guia de navegação para os usuários interessados em saber mais sobre a legislação brasileira de educação, as associações e entidades educacionais, os eventos da área, os softwares educativos entre outros assuntos. O acesso é livre em <http://www.eduline.com.br>.

EduTecNet

Criado pelo educador Eduardo Chaves em 1998, o EduTecNet reúne professores interessados em discutir o uso de tecnologia na educação. O ponto forte do projeto é a sua lista de discussão, que atualmente agrega cerca de 800 pessoas interessadas em debater o tema. Oferece uma ampla oferta de conteúdos ligados ao mundo da educação e tecnologia: legislação, lista de teses e dissertações, resenhas, sugestões práticas para o uso de tecnologia na educação, informações sobre associações de profissionais da área, links para sites afins etc. A navegação é livre em <http://www.edutecnet.com.br>.

Ensino net

O Ensino Net é um portal educacional que busca atender três frentes de trabalho: alunos, professores e escolas. O site apresenta uma variada oferta de conteúdos, recursos e serviços que auxiliam o desenvolvimento de atividades escolares, tais como tradutor de texto e sistema de pesquisa por imagens. O acesso ao portal é restrito para assinantes. <http://www.ensino.net/>

Escola 24 horas

A idéia do Escola 24 horas é estender a presença da escola na casa do aluno através da Internet. Trata-se de um projeto de educação online voltado aos alunos da pré-escola ao vestibular. Oferece serviços como aulas online, plantão 24 horas para tirar dúvidas de qualquer disciplina, bate-papos temáticos, cursos de especialização à distância para professores entre outros. O acesso é restrito aos usuários cadastrados. <http://www.escola24horas.com.br>

Interaula Clube

O Interaula Clube oferece aos alunos em fase pré-vestibular uma espécie de reforço aos seus estudos. O auxílio funciona através de aulas gravadas em vídeo e áudio e disponibilizadas para download aos usuários mediante assinatura. Professores também podem se associar ao site e utilizar o ambiente de aprendizagem como ferramenta de apoio às suas aulas. <http://www.interaula.com.br/>

KidLink do Brasil

O site KidLink do Brasil faz parte de um fórum internacional de língua portuguesa que tem por principal objetivo ajudar crianças de diversas partes do mundo a trabalharem em rede, se comunicarem e se tornarem amigas. A idéia é encorajar os participantes a dar valor as diferenças étnicas e culturais. Destaques do site são o KidSpace (usuários podem criar páginas e publicar nelas os seus trabalhos e imagens) e o KidProj Br (fórum de professores). O acesso é livre. <http://www.kidlink.org/portuguese/brasil/>

Klick Educação

Trata-se de uma iniciativa privada criada em 2000 pelo Grupo Klick. O portal é voltado ao desenvolvimento e à produção de conteúdos educacionais e de referência para a mídia impressa e eletrônica. O acesso ao material educativo do site é feito mediante cadastro e/ou assinatura paga, mas os professores podem ter um ano de acesso livre aos conteúdos e serviços desde que comprovem a sua atividade. <http://www.klickeducacao.com.br>

Microsoft Educacional

A Microsoft oferece em seu site educacional do Brasil um apoio aos professores e às escolas interessados em usar as tecnologias de informação e comunicação no dia-a-dia escolar. Os usuários encontram no site uma boa oferta de propostas de aula e de projetos educativos disponíveis para download. O acesso é livre em <http://www.microsoft.com/brasil/educacional>

Net Educação

O Net Educação é um projeto do programa Net de Responsabilidade Social, mantido pela empresa de televisão a cabo Net, vinculada às Organizações Globo. O site destina-se ao professor da rede pública e oferece acesso gratuito a planejamentos de cursos, planos de aulas, exercícios, artigos, infoteca e informações relevantes ao dia-a-dia dos educadores. Professores, coordenadores pedagógicos e diretores de escolas são convidados a enviarem artigos, ensaios, projetos, planos de aula, exercícios e informações para o portal. <http://www.neteducacao.tv.br>

Portal Educacional

É a porta de entrada do Grupo Positivo na Internet. A idéia do portal é encorajar e disseminar o uso de tecnologias na educação. Lá encontram-se vastos conteúdos e ferramentas online criados para professores e alunos dos ensinos fundamental e médio. O site também atende às necessidades dos pais de alunos. A maior parte das seções do portal é de navegação restrita às escolas associadas ao grupo. <http://www.educacional.com.br/>

Projeto Telemar Educação

O site do Projeto Telemar Educação busca atuar no sentido de oferecer um ambiente propício à criação de uma comunidade virtual de aprendizagem e de prática. A comunidade é animada por escolas e seus projetos comunitários, agenda de eventos e notícias em geral. O usuário encontra no portal uma série de tutoriais sobre o mundo da informática, dicionários e tradutores, ferramentas de busca e links para outros sites educacionais. A navegação é livre. <http://pte.futuro.usp.br/homepage.do>

Protagonismo juvenil

O Protagonismo juvenil oferece online uma série de textos, dicas e links sobre como criar projetos sociais e educativos em que o jovem ocupa uma posição de centralidade. O site define “protagonismo juvenil” como uma forma superior de educação para a cidadania, não pelo discurso das palavras, mas pelo curso dos acontecimentos. A navegação é livre. <http://www.protagonismojuvenil.org.br>

2.3.2. Sistemas de Autoria para Cursos a Distância

HM-Card

HM-Card é um aplicativo que permite a construção de páginas web para cursos a distância. O *software* permite a inserção de imagens e vídeos. Os cursos desenvolvidos através do HM-Card podem ser acessados via *web* e suas páginas podem estar ligadas de forma linear ou formando complexas unidades interativas. O sistema é composto de

três módulos independentes e ainda oferece suporte para cooperação. O endereço para acesso é: http://coronet.iicm.tugraz.at/HM_Tools/hmcard15/hmcard.htm

LearningSpace

O *LearningSpace* é um ambiente para desenvolvimento e aplicação de cursos, desenvolvida pela Lotus Development Corporation, que permite fazer cursos não presenciais usando o Lotus Notes ou um browser da WEB. Algumas ferramentas que oferece são: agenda, centro de mídia para acesso a materiais externos ao curso, sala de curso que é um ambiente para discussões privadas e públicas, *homepage* dos participantes com as descrições dos mesmos e gerenciamento de avaliações. Pode ser acessado através do site da Lotus: www.ibm.com/software/lotus/support/learningspace/

TopClass

O *TopClass* dispõe de ferramentas para autoria de cursos, de aprendizagem colaborativa, para disponibilização de cursos via *web*, grupos de discussão e um sistema de mensagens. Usuários cadastrados como professores no *TopClass* têm a possibilidade de gerenciar cursos, grupos de discussão, monitorar progresso dos alunos, etc. Cada usuário do *TopClass* tem informações a respeito do *status* do material do curso definido para ele, que pode ser: novo, velho, lido, não lido. O endereço de acesso é: <http://www.wbtsystems.com/solutions/technology>

Virtual-U

O Virtual-U é uma ferramenta para criação de curso via *web*. Este sistema oferece outras ferramentas de apoio como um sistema para disponibilização de conferências *online*, sistema para acompanhamento do desempenho dos alunos e as atividades realizadas por eles e ferramentas de administração do sistema para criação e manutenção de cursos. Pode ser acessado através do endereço:

<http://emediadesign.com/sfu/index.html>

WebCT

Além de ferramentas para criação de cursos via *web*, o WebCT fornece outras ferramentas como: *chat*, trilha do progresso do aluno, organização de projeto em grupo,

auto-avaliação, controle de acesso, ferramentas de navegação, investigações auto-marcadas, correio eletrônico, geração de índice automático, calendário de curso, *homepages* dos alunos e pesquisas do conteúdo do curso. O endereço para acesso é: <https://webct41.gmu.edu/webct/public/home.pl>

ClassWeb

ClassWeb foi desenvolvida pela *University of Califórnia*, Los Angeles, é um *software* gratuito para desenvolvimento de cursos via *web* sem os usuários conhecerem HTML ou as tecnologias necessárias para *web*. <http://classweb.ucla.edu/>

Eledge

Eledge foi desenvolvido pela *University of Utah*, é um ambiente para criação de *websites* para instrução *online*. Roda em Linux, utiliza o servidor Apache e o banco de dados MySQL. <http://eledge.sourceforge.net/>

O-LMS

Foi desenvolvido pela *University of Utah* e apresenta código aberto. Este *software* é composto de bibliotecas montadas para facilitar a criação de cursos a distância. Há ferramentas de *chat* e quadro negro. Foi desenvolvida em Java, utiliza o servidor Apache e o banco de dados Oracle. <https://uonline.utah.edu/olms/>

2.3.3. Salas de Aula Virtuais

AulaNET

O AulaNET foi desenvolvido pelo Laboratório de Engenharia de Software do Depto. de Informática da PUC-Rio. Este software disponibiliza um ambiente para criação, manutenção e assistência de cursos baseados na *web*. As ferramentas de apoio disponibilizadas pelo AulaNET são: correio eletrônico, listas de discussão, grupos de discussão, sessões de *chat* e videoconferências. Este software considera três grupos de participantes: professor, aluno e administrador. O endereço de acesso é: <http://www.eduweb.com.br/downnet/index.asp>

BolinOs

BolinOs é uma plataforma para comunicação e publicação de textos/artigos na *web*. Foi desenvolvida para atuar na área médica e pode ser utilizada para treinamentos a distância. <http://www.bolinos.com>

Ilias

Desenvolvido pela Universidade de Colônia, na Alemanha. É uma plataforma de treinamento com código aberto. Apresenta um ambiente de aprendizagem com anotações pessoais, avaliações, ferramentas de impressão, busca, mensagens, suporte ao gerenciamento de cursos e trabalho colaborativo, sistema de autoria de cursos a distância e suporte a várias línguas. Utiliza o servidor Apache, a linguagem PHP4 e o banco de dados MySQL. <http://www.ilias.de/>

MANIC

O MANIC é uma plataforma usada para desenvolver e distribuir cursos na *web*. Apresenta código aberto. Foi desenvolvida pela University of Massachussets. Utiliza o servidor Apache, a linguagem PHP, o banco de dados MySQL e o Real Áudio Server. <http://ripples.cs.umass.edu/research.php#dotnet>

ProInfo

O ProInfo é um Ambiente Colaborativo de Aprendizagem que, através da internet permite a concepção, administração e desenvolvimento de cursos a distância, complementos para cursos presenciais e diversos suportes para cursos a distância e para o processo ensino-aprendizagem. O ProInfo é desenvolvido pela Secretaria de Educação a Distância (SEED), por meio do Departamento de Infra-Estrutura Tecnológica (DITEC), em parceria com as Secretarias de Educação Estaduais e Municipais. <http://www.eproinfo.mec.gov.br/default.php>

2.3.4. *Frameworks para Aprendizagem Cooperativa*

Habanero

Habanero foi desenvolvido pelo National Center for Supercomputing Applications. Esta aplicação fornece uma biblioteca de ferramentas desenvolvidas em Java para o desenvolvimento de ambientes cooperativos para alunos do primeiro e segundo grau. O endereço para acesso é: <http://www.isrl.uiuc.edu/isaac/Habanero/>

Claroline

Claroline foi desenvolvido pela Universidade de Louvain, Bélgica. É um sistema de desenvolvimento e oferecimento de cursos gratuito e código aberto. Utiliza o servidor Apache, a linguagem PHP4 e o banco de dados MySQL. A Universidade Federal de Uberlândia vem adaptando esta plataforma para o oferecimento de cursos na UFU Virtual e UniRede. <http://www.claroline.net/>

Uma experiência no desenvolvimento de *software* educacional da Universidade Federal de Minas Gerais que merece ser lembrada é o NucleoEAD, desenvolvido pelo CADTEC, laboratório do Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais e que apresenta uma forte relação com este projeto de mestrado.

O NucleoEAD é uma escola virtual composta de uma estrutura organizada de sites, que disponibiliza recursos computacionais para a implementação de múltiplas atividades de ensino-aprendizado e objetiva incentivar a exploração de experimentos de ensino em busca de abordagens pedagógicas necessárias a esta modalidade de ensino. Atualmente o NucleoEAD disponibiliza quatro cursos: Introdução a linguagem HTML, Curso de Estruturas Isostáticas, Curso de Estruturas Metálicas e curso de Introdução ao Método dos Elementos Finitos. Este último curso é parte do projeto de mestrado em questão.

Como ferramentas de apoio a EAD, o NucleoEAD apresenta ferramentas de comunicação síncrona e assíncronas como grupos de discussão e *chat*, ferramentas de

busca, área de transferência, FAQ, além de um sistema gerenciador dos cursos e dos grupos de discussão.

O NucleoEAD representou também a base prática para a proposta de concepção de uma plataforma tecnológica para EAD, outro produto deste projeto.

De uma maneira geral, pode-se observar que todos os *softwares* educacionais pesquisados apresentam as mesmas ferramentas como: sistemas de autoria para cursos a distância, ferramentas de comunicação síncrona e assíncrona (*email*, fórum, *chat*), outras ferramentas de apoio (bloco de anotações, mural de recados, FAQ, bibliotecas, área de transferência) e sistemas gerenciadores de cursos e alunos.

2.4. SISTEMAS EDUCACIONAIS INFORMATIZADOS PARA ENGENHARIA DE ESTRUTURAS

Nesta seção serão descritos alguns sistemas existentes que auxiliam no ensino de engenharia de estruturas. Nem todos os sistemas comentados aqui podem ser utilizados via *web*, mas representam contribuições para E@D na área de engenharia de Estruturas.

ETOOLS – Ferramentas Computacionais para o Ensino de Estruturas

Trata-se de um projeto cooperado iniciado a partir de coordenação da PUC-Rio e com os seguintes participantes iniciais: PUC-Rio, UFAL, UFMG, UFPR, UPF e USP. Pretende ser um repositório de software e links educacionais.

O ETOOLS tem o objetivo de desenvolver programas computacionais gratuitos para serem usados como elementos pedagógicos na aprendizagem de análise e dimensionamento de estruturas ou outras áreas da engenharia. Para o desenvolvimento dos recursos, utiliza o paradigma de orientação a objetos e principalmente, as linguagens DELPHI, C e C++.

<http://www.tecgraf.puc-rio.br/etools/> ou www.etools.upf.br

OE³ / ETools – Objetos Educacionais para Engenharia de Estruturas

Desenvolvido pela UFPR (universidade federal do Paraná), destina-se à criação de software ou ferramentas que poderão ser utilizadas para o ensino de engenharia de estruturas. No site do projeto há um repositório de objetos educacionais, classificados por assuntos e categorias estruturais. Este projeto faz parte do ETOOLS.

<http://www.cesec.ufpr.br/etools/oe3/>

INSANE

O projeto INSANE (*Interactive Structural Analysis Environment*) tem o objetivo de desenvolver recursos para auxílio ao ensino presencial de Análise Estrutural além de encontrar soluções tecnológicas para desenvolvimento de software que auxiliem na pesquisa de modelos discretos de análise estrutural. Este projeto está sendo desenvolvido pela Universidade Federal de Minas Gerais (UFMG), no departamento de Engenharia de Estruturas (DEES). Os produtos deste projeto são desenvolvidos utilizando a programação orientada a objetos, linguagem Java, XML, padrões de projeto de software, dentre outras tecnologias. Dentre os sistemas desenvolvidos, apresenta o INSANE Web que disponibiliza um Serviço Web para resolução de modelos de elementos finitos através da Internet, através do núcleo numérico do INSANE.

INSANE - <http://www.insane.dees.ufmg.br/insane>

INSANE Web - <http://insane.dees.ufmg.br:8080/InsaneWeb/app>

FEMOOP – Finite Element Method Object Oriented Program

O FEMOOP foi desenvolvido pela PUC do Rio de Janeiro e pelo laboratório de mecânica computacional da Escola Politécnica da Universidade de São Paulo. O FEMOOP é um programa de análise estrutural baseado no Método dos Elementos Finitos. Utiliza a programação orientada a objetos e a linguagem C++. Este programa apresenta toda a sua documentação na *web*.

<http://www.lmc.ep.usp.br/people/tbitten/femoop/home.htm>

FTOOL - Two-dimensional Frame Analysis Tool

O FTOOL é um programa gratuito desenvolvido pela Tecgraf /P.U.C-R.J (Grupo de Tecnologia em Computação Gráfica) e com apoio do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico). para análise estrutural bidimensional. Se destina principalmente para o ensino do comportamento de estruturas. Com o FTOOL é possível montar estruturas e, após a especificação de alguns parâmetros, construir gráficos de momento fletor, esforço normal e cortante, linha elástica e gráfico da configuração deformada da estrutura.

<http://www.tecgraf.puc-rio.br/ftool/>

CIMNE - International Center for Numerical Methods in Engineering

O CIMNE é um centro autónomo de pesquisa e desenvolvimento dedicado a promover e fomentar os avanços na produção de aplicações de métodos numéricos e técnicas computacionais para a solução de problemas de engenharia. CIMNE organiza um vasto leque de atividades que visam o ensino e a difusão do conhecimento, tais como cursos, seminários, conferências e publicações. Foi criado em 1987 pela Generalitat de Catalunya e pela Universitat Politècnica de Catalunya (UPC). O site o CIMNE oferece cursos a distância, além de possuir salas de aula virtuais, ou seja, ambientes para colaboração, pesquisa, desenvolvimento e ensino de tópicos relacionados a engenharia.

<http://www.cimne.upc.es/cdl1/itshome/start.aspx>

3

RECURSOS TECNOLÓGICOS UTILIZADOS

A seguir serão mostrados e discutidos aspectos fundamentais de importantes tecnologias existentes, muitas das quais utilizadas na execução dos produtos dessa dissertação. Estas tecnologias englobam linguagem de programação ou de marcação para a *web*, o *Framework.NET*, Banco de Dados e *software* utilizados para variadas finalidades.

3.1. PROGRAMAÇÃO ORIENTADA A OBJETOS

3.1.1. Histórico

A Programação Orientada a Objetos, POO, foi desenvolvida devido às dificuldades e limitações encontradas nas abordagens anteriores de programação, quando sistemas computacionais cada vez maiores e mais complexos começaram a ser exigidos.

O paradigma (princípio organizacional) de programação orientada a objetos surgiu na área acadêmica, há uns quarenta anos, mas apenas recentemente, na década de 90 é que começou a ser realmente utilizado e aceito nas empresas de desenvolvimento de software.

A primeira linguagem a realmente utilizar as idéias de programação orientada a objetos foi a linguagem Simula 67, criada por Ole Johan Dahl e Kristen Nygaard em 1967, na Noruega. Esta linguagem foi criada para fazer simulações. No entanto, a Orientação a

Objetos apenas se concretizou após a criação da linguagem de programação Smalltalk 80, quando surgiu o termo Programação Orientada a Objetos criado por Alan Kay, autor também da linguagem de programação Smalltalk.

A partir dos anos 80, algumas linguagens estruturadas tentaram se adaptar para incluir alguns conceitos de Orientação a Objetos, como por exemplo Pascal, C, Lisp, Cobol, que posteriormente evoluíram com a inclusão de classes. A linguagem C++ foi um marco para aplicações em todos os níveis. É também na década de 80 que surge o Visual Object Oriented Cobol (c/ win95) e o Visual Basic.

A maioria das linguagens de programação existentes atualmente adota parcialmente as idéias de orientação a objetos, dando espaço ao antigo modelo procedural. Isso acontece por exemplo, no C++ onde tem-se a possibilidade de usar POO, mas a linguagem não força o programador a adotar este paradigma de programação, sendo ainda possível programar da forma procedural tradicional. Outras linguagens são construídas do zero focando-se sempre nas idéias por trás da orientação a objetos como é o caso das linguagens Smalltalk, Self e IO, onde TUDO é orientado a objetos.

Alguns exemplos de linguagens de programação orientadas a objetos são: Smalltalk, Python, Ruby, C++, Object Pascal, Java e C#.

3.1.2. Definições

Programação Orientada a Objetos

A **orientação a objetos**, também conhecida como **Programação Orientada a Objetos (POO)** ou ainda em inglês ***Object-Oriented Programming (OOP)*** é uma forma natural e lógica pela qual os seres humanos e, especificamente, os programadores raciocinam, tentando aproximar o mundo real do mundo virtual: a idéia fundamental é tentar simular o mundo real dentro do computador através de uma coleção de objetos que interagem entre si. A POO utiliza conceitos básicos como objetos e atributos, todos e partes, classes e membros.

Em uma linguagem técnica, a POO é um paradigma de análise, projeto e programação de sistemas de *software* baseado na composição e interação entre diversas unidades de *software* chamadas de objetos.

A análise e projeto orientados a objetos têm como meta identificar o melhor conjunto de objetos para descrever um sistema de *software*. O funcionamento deste sistema se dá através do relacionamento e troca de mensagens entre estes objetos.

Na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos presentes no sistema de *software*. Cada classe determina o comportamento (definidos nos métodos) e estados possíveis (atributos) de seus objetos, assim como o relacionamento com outros objetos, combinando assim em uma única entidade (o objeto) tanto os dados quanto as funções que operam esses dados.

As linguagens de programação que aceitam o paradigma de orientação a objetos são denominadas linguagens orientadas a objetos (OO). Formalmente, para ser considerada uma linguagem OO, esta precisa implementar quatro conceitos básicos: **abstração, encapsulamento, herança e polimorfismo**

Alguns conceitos importantes na programação orientada a objetos estão descritos a seguir.

Classes

Uma classe pode ser definida como a declaração ou abstração de objetos. Uma classe define o comportamento dos objetos, através de métodos, e quais estados ele é capaz de manter, através de atributos.

Cada classe possibilita a criação de inúmeros objetos. Cada objeto é dito ser uma *instância* de uma classe. Assim, cada instância de uma classe tem seus próprios valores para cada atributo. Uma classe é apenas um conceito estático pertencente ao programa de computador, utilizada para definir, ou declarar os objetos, elementos dinâmicos manipulados pelos programas e pertencentes à memória do computador (local onde os objetos ocupam espaço durante a execução do programa).

A definição de classe pode ser resumida como uma estrutura modular completa que descreve (encapsula) as propriedades estáticas e dinâmicas dos elementos manipulados pelo programa, ou seja, descreve um conjunto de dados estruturados que são caracterizados por propriedades comuns. Uma classe captura a estrutura e o comportamento comum a todos os objetos que são relacionados, por meio de um conjunto uniforme de atributos e serviços. Na FIG. 4 há a ilustração esquemática e um exemplo do diagrama de uma classe.

Uma classe é definida por:

- um nome da classe;
- nome da sua superclasse (se houver uma classe da qual é derivada);
- nome de suas variáveis;
- os nomes e as definições de todas as operações associadas a esta classe;

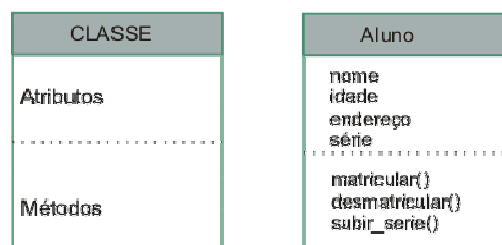


FIGURA 4 - Definição de classes – estrutural e exemplo

Objetos

Objetos são exemplares de uma classe qualquer, são os elementos dinâmicos em um programa. Os objetos são criados em tempo de execução de um programa a partir de uma classe, por uma ação denominada de *instance* (que significa em inglês exemplificar). Os objetos apresentam uma identidade e podem representar entidades concretas (um arquivo no computador, uma bicicleta) ou entidades conceituais (uma estratégia de jogo, uma política de escalonamento em um sistema operacional). Cada objeto tem sua identidade, significando que dois objetos são distintos mesmo que eles apresentem exatamente as mesmas características.

Os objetos armazenam suas características através de seus atributos e interagem entre si através das mensagens enviadas a ele ou enviadas dele a outros objetos. Assim, a estrutura de um objeto é representada em termos de *atributos*. O comportamento de um objeto é representado pelo conjunto de *operações* que podem ser executadas sobre o objeto. Objetos com a mesma estrutura e o mesmo comportamento são agrupados em *classes*.

De maneira simples, um objeto é uma entidade lógica que contém dados (atributos) e código para manipular esses dados (métodos).

Um objeto é composto por:

- uma estrutura de dados, ou uma memória interna em que valores podem ser armazenados e modificados ao longo da vida do objeto.
- um conjunto de ações pré-definidas (métodos), através das quais o objeto responderá a demanda de processamento por parte de outros objetos.

Resumidamente, um objeto é composto por estrutura e processos, sendo que esses processos giram em torno da estrutura.

Atributos

Os atributos são a estrutura de dados de um objeto que representa uma determinada classe, são as características de um objeto, ou em termos técnicos, são as variáveis onde são armazenados os dados relacionados aos objetos de uma classe.

Os atributos possuem sempre um nome e um tipo. Cada objeto apresenta um valor específico para o atributo correspondente.

Métodos

Os métodos são as ações ou funcionalidades que os objetos de uma determinada classe podem realizar. Os métodos manipulam a estrutura de dados de uma classe.

Um método é apenas uma definição em uma classe. A ação só ocorre quando o método é invocado através de um objeto específico e este método afetará apenas um objeto em particular.

Herança

Herança é o mecanismo pelo qual uma classe pode herdar as características de outra classe, isto é, aproveitar seus comportamentos (métodos) e estados possíveis (atributos) e adicionar outros dados (atributos e métodos) a mais. O mecanismo de herança estabelece uma hierarquia entre as classes, como pode ser observado na FIG. 5.

A classe criada a partir de uma outra classe é chamada de classe filha ou sub-classe, a classe principal é chamada de super-classe, ou classe mãe.

A herança permite altos graus de reutilização de código, já que evita a repetição de código comum a mais de uma classe.

Uma classe mãe pode sempre ter várias classes filhas, mas normalmente as linguagens de programação orientadas a objetos exigem que cada classe filha tenha apenas uma classe mãe. A linguagem C++ permite que uma classe herde as características de várias classes (herança múltipla).

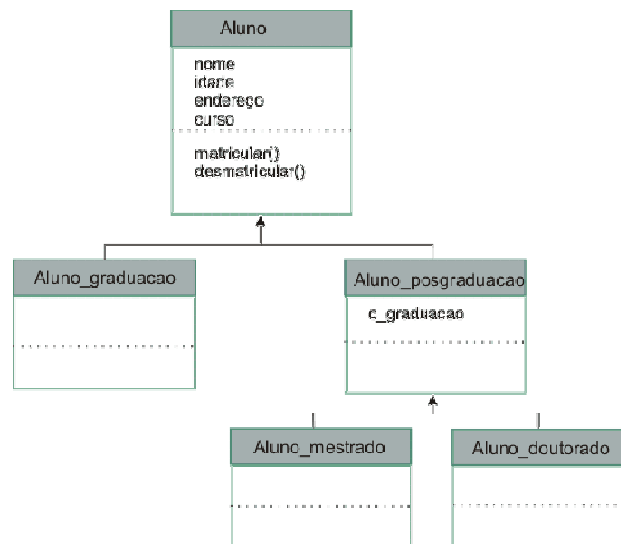


FIGURA 5 - Exemplo do mecanismo de herança

Encapsulamento

O encapsulamento é a capacidade de ocultar dados dentro de um modelo, deixando visível ao usuário apenas as operações ou funções especializadas criadas para a manipulação dos dados. O usuário não conhece os mecanismos internos de desenvolvimento do programa. Um dado está encapsulado quando não há meios de acessá-lo externamente, a não ser por meio de métodos, o mesmo acontece com uma rotina, que sendo encapsulada, suas operações internas são invisíveis às outras rotinas. No encapsulamento há a separação de aspectos internos e externos de um objeto. Este conceito está esquematizado na FIG. 6.

Encapsular significa separar o programa em partes, o mais isoladas possível, tornando o software mais flexível, fácil de modificar e de criar novas implementações, com menos erros e de entendimento mais fácil aos programadores.

Algumas das vantagens do encapsulamento são:

- toda parte encapsulada pode ser modificada sem que os usuários da classe em questão sejam afetados.
- proteção ao acesso direto (referência) aos atributos de um objeto fora da classe onde estes foram declarados. Esta proteção consiste em se usar modificadores de acesso mais restritivos sobre os atributos definidos na classe. Depois devem ser criados métodos para manipular de forma indireta os atributos da classe.
- auxilia a garantir que o estado e o comportamento de um objeto se mantenha coeso, funcionando exatamente como previsto pelo programa, evitando que os usuários mudem o estado ou comportamento dos objetos sem necessidade ou acidentalmente, ou seja, protege os dados e faz o uso do objeto ser mais seguro.

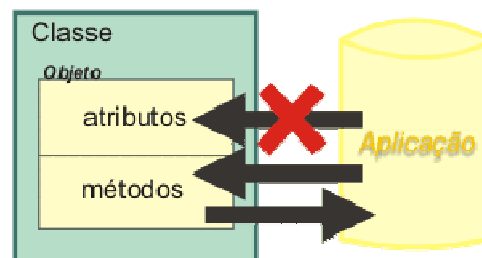


FIGURA 6 - Definição de encapsulamento

Abstração

Abstração é a capacidade de modelar características do mundo real, seguindo uma determinada lógica e implementando os aspectos necessários para a resolução do problema em questão. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no mundo real e são descritas com objetivos específicos, contendo as informações necessárias para cumprir sua função.

Polimorfismo

Polimorfismo é a capacidade de um código ou um método produzir comportamento diferentes dependendo da classe do objeto que produziu a ação, em tempo dinâmico de execução. Ou seja, uma mesma mensagem é enviada a objetos de classes distintas e eles poderão reagir de maneiras diferentes. Um método polimórfico é aquele que pode ser aplicado à várias classes de objetos sem que haja qualquer inconveniente.

Utilizando o polimorfismo novas classes podem ser adicionadas sem necessidade de modificação de código já existente, pois cada classe apenas define os seus métodos e atributos.

3.1.3. Vantagens da POO

Algumas vantagens da Programação Orientada a Objetos podem ser enumeradas:

- Reutilização de código;
- Desenvolvimento de programas mais complexos e de grande porte com menor volume de código;
- Manutenção mais simples dos programas, já que apresentam uma divisão de código mais lógica e melhor encapsulada, devido à utilização dos conceitos de herança e encapsulamento;
- Maior capacidade de adequação a mudanças;

- Gerenciamento do desenvolvimento do software mais simples, já que o programa pode ser dividido em módulos, estruturas que englobam as classes que apresentarem objetivos similares;
- Rapidez no desenvolvimento de programas: várias equipes podem trabalhar simultaneamente no desenvolvimento de um programa, já que este pode ser dividido em módulos com características e objetivos afins.

3.2. LINGUAGENS PARA *INTERNET*

Para o desenvolvimento de páginas *web* são utilizadas linguagens específicas, as quais nem todas são linguagens de programação. Algumas são linguagens de marcação como, por exemplo, a linguagem HTML, provavelmente a mais conhecida para *web*. Outras, como a UML, são linguagens para estruturação lógica de dados. A UML foi utilizada para conceber os diagramas de classes da plataforma tecnológica proposta. Abaixo estão relacionadas algumas linguagens utilizadas para o desenvolvimento deste projeto. As linguagens de programação para internet se diferenciam das linguagens de marcação pela possibilidade de exibição de conteúdo dinâmico, processados no servidor e enviados ao usuário, geralmente em HTML.

Para o HTML, XML, XHTML e CSS, a World Wide Web Consortium desenvolveu padrões que referenciam o desenvolvimento de páginas contendo essas linguagens.

O *world wide web consortium* (W3C) é um consórcio de empresas que promovem a evolução da internet através do desenvolvimento de protocolos comuns, fóruns abertos para troca de informações, de comércio, comunicação e conhecimento coletivo. O W3C foi fundado por Tim Berners-Lee em 1994, hoje conta com aproximadamente quinhentos membros.

O W3C possui diversos escritórios e comitês no mundo que estudam as tecnologias existentes para apresentação de conteúdo na internet e desenvolvem padrões e recomendações para a utilização dessas tecnologias. Estas discussões favorecem a

evolução da internet e asseguram a sua interoperabilidade, através das especificações, manuais, fóruns e ferramentas criadas.

Os *sites* desenvolvidos segundo os padrões do W3C podem ser acessados e visualizados por qualquer pessoa ou tecnologia, independente do *hardware* ou *software* utilizados.

3.2.1. HTML

HTML ou *HyperText Markup Language* (Linguagem de Marcação para Hipertextos), como o próprio nome diz, é uma linguagem de marcação, que tem a finalidade de formatação de dados. É a linguagem utilizada para a produção de páginas estáticas para publicações na *web*.

Um arquivo HTML é um arquivo de texto comum, contendo marcações ou *tags* pré-definidas, que dizem ao *browser* como mostrar a página. Cada marcação representa um elemento a ser exibido pelo navegador. A linguagem HTML foi desenvolvida para a apresentação de dados em *browsers*. Os *browsers* (*Internet Explorer da Microsoft*®, *Netscape da Netscape Communications*®, etc) interpretam as marcações ou *tags* existentes nos arquivos HTML e exibem as páginas. As figuras 7 e 8 são exemplos de uma página HTML e de sua interpretação por um *browser*, respectivamente.

As páginas HTML são armazenadas estaticamente em um servidor HTTP e disponibilizadas ao usuário quando requisitadas através de um endereço URL e exibidas em navegadores ou *browser*.

Uma página HTML apresenta uma estrutura pré-definida que pode ser observada na FIG. 7: a *tag* de abertura do documento <HTML>; o cabeçalho do documento, definido pela *tag* <HEAD>, onde estão disponíveis informações a respeito do documento, como por exemplo o título da página; e o corpo do documento, definido pela *tag* <BODY>, onde está organizado e devidamente disposto o conteúdo para apresentação na página.

A normalização para desenvolvimento e utilização de páginas HTML encontra-se disponível no *site* da *World Wide Web Consortium*.

```

<html><head>
<title>Untitled Document</title>
</head><body>
<table width="46%" border="0">
  <tr><td><p></p>
  <table width="398" border="0" cellspacing="0" cellpadding="0">
    <tr><td width="183" height="43"><dl><dt>&nbsp;</dt></dl></td>
    <td colspan="2"><br><b><font size="2" face="Arial">Pesquisar alunos pelo:</font> </b> </td>
  </tr><tr><td width="183" height="39"><p align="right">
    <input type="radio" name="tipo" value="nome" checked></p></td>
    <td width="183"><font size="4" face="Arial">Nome</font></td>
    <td width="183"><input type="text" name="nome" size="20"></td>
  </tr><tr><td width="183" height="36"><p align="right"><input type="radio" name="tipo" value="email"
  </p></td><td width="183"><font size="4" face="Arial">E-mail</font></td>
    <td width="183"><input type="text" name="email" size="20"></td></tr>
  </table><p>&nbsp;</p></td>
</tr>
</table>
</body>
</html>

```

FIGURA 7 – Exemplo do código de uma página HTML

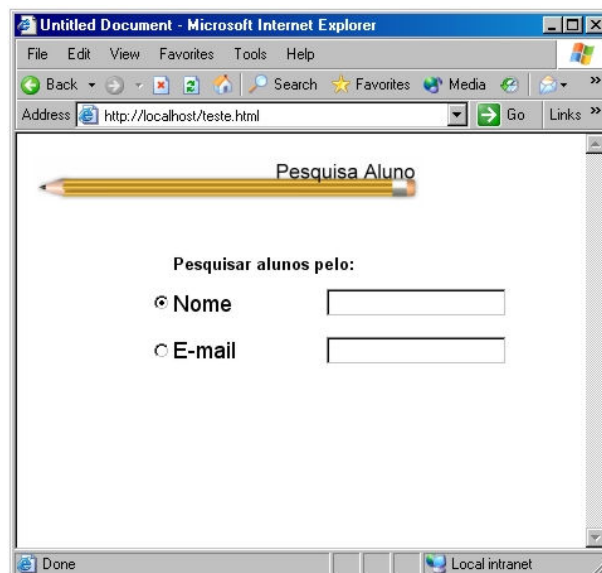


FIGURA 8 – Exibição do código da figura 4 em um *browser*

A HTML é uma linguagem bastante simples, o que contribuiu em muito para a popularização da internet. Páginas HTML podem ser produzidas em editores de texto comuns, bastando apenas salvar o arquivo com a extensão .html ou .htm e utilizar a estruturação pré-definida das páginas HTML. Editores próprios para desenvolvimento de páginas *web* têm sido desenvolvidos e aperfeiçoados, facilitando o trabalho dos desenvolvedores. Um exemplo de um editor de páginas HTML é o *DreamWeaver*, *software* da *Macromedia*®.

A linguagem HTML apresenta ainda alguns problemas que têm sido aos poucos superados através de padronizações estabelecidas pela *World Wide Web Consortium*. Entretanto, ainda há o problema de que a combinação das diferentes marcações existentes nem sempre produzem o mesmo resultado visual em cada navegador diferente. Outro problema intrínseco do HTML é a sua falta de estruturação de dados. A linguagem HTML não foi desenvolvida para o armazenamento ou organização lógica de dados, mas apenas para a apresentação dos dados em *browser*.

Para solucionar este problema estão sendo desenvolvidas outras linguagens como o *eXtensible Markup Language* (XML) e o *Cascading Style Sheets* (CSS), por exemplo. Desde 1999 vem sendo desenvolvida a *eXtensible HyperText Markup Language* (XHTML), uma nova linguagem de marcação que combina a HTML com a XML, dando mais poder à HTML.

3.2.2. XML

XML ou *eXtensible Markup Language* é outra linguagem de marcação, muito utilizada para transmissão de dados via *web*. As marcações desta linguagem são feitas através de “*tags*” não pré-definidas. A XML é mais poderosa que a HTML, sobretudo porque a XML é uma metalinguagem, ou seja, uma linguagem utilizada para definir outras linguagens de marcação. A linguagem XML não faz nada, ela foi criada para estruturar, armazenar e enviar informações.

Um arquivo XML auto-descritivo contém arquivos *Document Type Definition* (DTD) ou *XML Schema* para descrever os dados contidos no arquivo XML. Os elementos de um documento XML são criados de acordo com a necessidade do programador. A linguagem XML é uma linguagem clara, simples de ser criada, suporta um grande número de aplicações, independe de sistemas operacionais ou de outros recursos e por isso permite que um mesmo documento seja apresentado sob várias formas e dispositivos (PCs, Palms, Celulares, convertidos em som, etc).

As principais vantagens da XML são: apresenta declarações mais precisas dos conteúdos; fornece resultados mais significativos de busca; permite a integração de diferentes fontes de dados; permite o desenvolvimento de aplicações *web* mais flexíveis;

por apresentar completa separação da interface visual e dos dados, permite múltiplas formas de visualização e a criação de aplicações mais poderosas, simples e flexíveis; a atualização do conteúdo pode ser feita de forma granular, uma pequena alteração em parte do documento não resultará na busca pelo documento inteiro novamente, novos dados podem ser adicionadas sem a necessidade de reconstrução da página, diminuindo assim, a sobrecarga do servidor e o tráfego na rede; facilidade da distribuição dos dados na rede e na *web*, pois, podem ser fornecidos de forma mais comprimida e escalável.

A linguagem XML foi publicada em fevereiro de 1998 e surgiu a partir de uma simplificação da *Standard Generalized Markup Language* (SGML), linguagem que deu origem a todas as metalinguagens.

A FIG. 9 apresenta um exemplo de um documento XML utilizado para organizar livros em uma livraria. As marcações utilizadas definem as informações a respeito dos livros da livraria como título, autor, preço, gênero. Como pode ser observado, cada categoria da informação tem o seu nome próprio e indica também o seu significado.

As linguagens XML e HTML foram criadas com objetivos diferentes. A XML foi criada para descrever dados, realçando o que cada dado é, enquanto que a HTML foi criada para mostrar e formatar dados, enfatizando como os dados deverão ser apresentados.

```
<bookstore>
  <book genre= "novel" style="hardcover">
    <title>The Handmaids's Tale</title>
    <price>19.95</price>
    <author>
      <first-name>Margaret</first-name>
      <last-name>Atwood</last-name>
    </author>
  </book>
  <book genre= "novel" style="paper-back">
    <title>The Possonwood Bible</title>
    <price>11.99</price>
    <author>
      <first-name>Barbara</first-name>
      <last-name>Kingsolver</last-name>
    </author>
  </book>
</bookstore>
```

FIGURA 9 – Exemplo de documento XML

3.2.3. CSS

CSS ou *Cascading Style Sheets* (Folhas de estilo em cascata) são estilos que definem a aparência dos elementos HTML. Foi criada para dar suporte a HTML, separando, de modo adequado, o estilo do conteúdo da página.

Os estilos podem ser armazenados em folhas de estilos dentro do próprio arquivo HTML ou em folhas de estilos externas: arquivos CSS. O nome *Cascading* foi utilizado porque um estilo definido é aplicado para todos os elementos dentro de um documento.

O CSS foi criado em acréscimo à linguagem HTML 4.0, mas pode ser utilizada na linguagem XML.

Algumas vantagens do CSS podem ser enumeradas:

- Facilita a manutenção de páginas, devido à completa separação entre interface e código;
- Diminui o tráfego de informações;
- Maior flexibilidade nas aplicações.

O CSS define como as marcações dos elementos HTML devem ser apresentadas, como pode ser observado na FIG 10. Por exemplo, a marcação <p> que define um parágrafo na página HTML, aparecerá com o tamanho da fonte 8pt, nome da fonte Geórgia, sem margem superior e justificado.

```

body {
    font: 8pt/16pt georgia;
    color: #555753;
    background: #fff url(blossoms.jpg) no-repeat bottom right;
    margin: 0px;
}
p {
    font: 8pt/16pt georgia;
    margin-top: 0px;
    text-align: justify;
}
h3 {
    font: italic normal 12pt georgia;
    letter-spacing: 1px;
    margin-bottom: 0px;
    color: #7D775C;
}
a:link {
    font-weight: bold;
    text-decoration: none;
    color: #B7A5DF;
}
a:visited {
    font-weight: bold;
    text-decoration: none;
    color: #D4CDDC;
}
a:hover, a:active {
    text-decoration: underline;
    color: #9685BA;
}
/* specific divs */
#container {
    background: url(zen-bg.jpg) no-repeat top left;
    padding: 0px 175px 0px 110px;
    margin: 0px;
}

```

FIGURA 10 – Exemplo de um arquivo CSS

3.2.4. JAVA

Java é uma linguagem de programação desenvolvida pela *Sun Microsystems*®. Esta linguagem é bastante utilizada para desenvolvimento de aplicativos para *web*, ou seja, através desta linguagem podem ser criados programas para serem executados dentro de um *browser* da *web*. Os aplicativos desenvolvidos em Java são pré-compilados em um ambiente de desenvolvimento, hospedados no servidor web e referenciados pela página HTML que chamará o aplicativo. Após a requisição desses aplicativos pelo cliente

através de um *browser*, o aplicativo é compilado no cliente e para isso, é necessário que ele tenha a máquina virtual Java (compilador final da linguagem).

As aplicações desenvolvidas em JAVA independem do sistema operacional em que foram desenvolvidas, podendo ser executadas em diferentes sistemas operacionais, por isso a sua grande utilização na internet.

Applets são aplicativos Java abrigados no servidor e executados no cliente. Java *applets* oferecem alto grau de interação com usuários, mas deve-se tomar cuidado com o uso excessivo porque necessita ser transmitido ao cliente. Os *servlets* são programas criados utilizando a linguagem Java que executam acesso a dados e geram conteúdo HTML dinamicamente no servidor.

A linguagem Java possibilita o desenvolvimento de aplicativos, como aplicativos *desktop* e são executados no cliente, diferentemente das linguagens ASP e PHP que executam e processam o código dos aplicativos no servidor e envia para o cliente apenas os resultados.

3.2.5. JavaScript

JavaScript é uma linguagem de *scripting*, ou seja, uma linguagem de programação leve e menos complexa que outras como o Java ou C++, além de ser interpretada (não é compilada). Uma linguagem interpretada não é convertida em um arquivo executável, o código é executado utilizando o interpretador que lê o código fonte e o interpreta diretamente, durante a sua execução.

O *JavaScript* dá mais interatividade às páginas e são muito utilizados, por exemplo para a validação de formulários no cliente, antes de serem enviados ao servidor. O código Java Script pode aparecer como janelas, com as quais o usuário irá interagir.

A linguagem *JavaScript* aparece como linhas de código no meio da página HTML. Os blocos dos códigos de programação são delimitados por chaves, as linhas de código são terminadas com ponto e vírgula. Esta linguagem apresenta vários métodos, objetos e funções já definidas, mas há ainda a possibilidade de definição de novos métodos, objetos e funções.

Esta linguagem foi baseada no *ECMAScript*, especificação ECMA-262 (padrão internacional para linguagem de *script* desenvolvido pela *European Computers Manufactures Association* - ECMA).

A FIG. 11 mostra um exemplo de um bloco *JavaScript* dentro de um arquivo HTML. No exemplo, a função *alert(...)* exibirá uma janela com a mensagem escolhida assim que a página HTML for solicitada.

```

<html>
<head>
<title>Teste</title>
<style type="text/css">
<!--
.style1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
    font-size: 12px;
}
</style></head>
<body>
<table width="100%" border="0">
  <tr><td colspan="3"><span class="style1">Alerta</span></td></tr>
  <tr>
    <td width="7%"><span class="style1">Alunos:</span></td>
    <td width="25%"><input type="text" name="textfield"></td>
    <td width="34%" rowspan="2">&nbsp;</td>
  </tr>
  <tr><td colspan="2"><form name="form3" method="post" action="">
    <div align="right"><input type="submit" name="Submit" value="Vai &gt;&gt;"></div>
  </form></td>
  </tr>
</table>
<script language="JavaScript">
<!--
alert("Aluno, você não tem acesso ao Módulo 3 da apostila.");
//-->
</script>
<script language="JavaScript">
<!--
alert("Aluna, o tempo para cursar essa etapa acabou!.");
//--></script>
</body>
</html>

```

Bloco de Script

FIGURA 11 – Exemplo da Linguagem *JavaScript* dentro de um arquivo HTML

3.2.6. *Action Script*

O *Action Script* é a linguagem de *script* utilizada pelo Flash, programa da *Macromedia*® para confecção de animações. O *Action Script* é a linguagem utilizada para desenvolver aplicações e adicionar mais interatividade nas animações produzidas no Flash.

Action Script é semelhante à linguagem de programação *JavaScript*, apresentando algumas diferenças, dentre as principais: não suporta alguns objetos específicos dos *browser* como *document*, *window*; também não suporta alguns construtores do *JavaScript* e a função *eval()* comporta-se de maneira diferente nas duas linguagens.

Esta linguagem utiliza o conceito de linguagem orientada a objetos adotando a sintaxe do ponto para obter e definir propriedades e métodos de um objeto. Também é utilizado o conceito de classes, existem várias classes pré-definidas, mas há a possibilidade de serem definidas novas classes.

O *action script* utiliza algumas terminologias: Flash(1993 – 2003)

- **Ações** são sentenças que instruem a animação (arquivo SWF) a fazer algo durante a execução. Alguns exemplos de ação são: parar - *stop()*, continuar – *play()*, ir para o quadro X e parar- *gotoAndStop()*, etc.
- **Classes** são tipos de dados que podem ser criados para definir tipos de objetos.
- **Constantes** são elementos que não mudam.
- **Construtores** são funções utilizadas para definir propriedades e métodos de uma classe. Os construtores são funções que aparecem dentro da definição de uma classe com o mesmo nome da classe.
- **Tipos de dados** descrevem o tipo de informação que uma variável ou elemento do *Action Script* pode armazenar. Os tipos de dados possíveis são: *string* (seqüência de caracteres), *number* (número), *boolean* (booleano – *true or false*), *object* (objeto), *movieClip* (clipe de filme), *function* (função), *null* (nulo) e não definido.

- **Eventos** são ações que ocorrem enquanto a animação está sendo executada.
- **Controladores de Eventos** são ações especiais que controlam os eventos, por exemplo: *on()* – *on(mouseover)*; indica a ação que deverá ocorrer quando o mouse estiver sobre o objeto.
- **Expressões** são combinações de símbolos do *action script* que representam um valor. Exemplo: *x+2*.
- **Funções** são blocos de código reusáveis que podem passar parâmetros ou retornar valores.
- **Instâncias** são objetos que pertencem a uma classe. As instâncias de uma classe contêm as propriedades e métodos da classe.
- **Keywords** ou palavras chave são palavras reservadas que apresentam um significado especial. Por exemplo, *var* que é utilizada para declarar variáveis locais.
- **Métodos** são funções associadas a uma classe.
- **Objetos** são coleções de propriedades e métodos. Cada objeto tem seu nome próprio e é uma instância de uma classe. O *Action Script* apresenta alguns objetos padrão, como por exemplo, o objeto *Date* que fornece informações a respeito de datas e horários.
- **Operadores** são termos que calculam um novo valor de um ou mais valores. Por exemplo: *+* (soma), *-* (subtrai), *** (multiplica), etc. Há também os operadores lógicos, que executam regras a partir da obediência ou não de lógicas estabelecida, alguns desses operadores são: *if-else*, *while*, etc.
- **Parâmetros** ou **Argumentos** são lugares próprios que permitem passar valores às funções, aparecem entre parênteses após o nome da função.
- **Pacotes** são diretórios que contêm um ou mais arquivos de classes, localizados em um caminho específico.

- **Propriedades** ou **Atributos** definem as características de um objeto. Por exemplo, um *movieclip* tem as propriedades: visível ou escondido (*visible or hidden*).
- **Variáveis** são identificadores que guardam valores de qualquer tipo.

Estas terminologias utilizadas pelo *Action Script* são semelhantes às terminologias utilizadas pelas linguagens orientadas a objetos.

O *Action Script* segue um fluxo lógico. O Flash executa os comandos a partir do primeiro e continua em ordem até alcançar o comando final ou um comando que informe ao *Action Script* que vá para outro local.

3.2.7. ASP Clássico

Active Server Pages (ASP) é um programa (ambiente de *script* de servidores), desenvolvido pela *Microsoft*® e que funciona dentro do IIS (*Internet Information Service*) ou PWS (*Personal Web Server*) que são aplicativos do *Windows* responsáveis pelo gerenciamento de serviços da *internet*. O PWS é um servidor desenvolvido para aplicações que precisam de um IIS em escala reduzida, podendo hospedar apenas um *site*.

Um arquivo ASP contém texto HTML e linguagens de *script* (*VBScript* ou *JavaScript*), como pode ser observado na FIG 12, porém, os *scripts* nestes arquivos são executados no servidor. Quando uma página ASP é requisitada, o servidor, com o IIS executa os *scripts* do arquivo e retorna para o *browser* uma página HTML como resposta. Um arquivo ASP apresenta a extensão “.asp” e pode ser criado utilizando um editor de texto comum.

Quando o cliente solicita uma página ASP ao servidor, este verifica a extensão para certificar que o arquivo deve ser anteriormente executado, executa-o e envia a resposta ao cliente.

O ASP é utilizado para o desenvolvimento de páginas dinâmicas para *web*, inclusive com possibilidade de acesso a banco de dados.

Com o aumento da necessidade de aplicações mais complexas e interativas, surgiram algumas desvantagens na utilização do ASP:

- O código ASP é interpretado linha a linha cada vez que a página é requisitada pelo cliente, tornando o processo de navegação pouco ágil, além de suportar poucos tipos de variáveis.
- A colaboração, através da programação em ASP é difícil, já que mistura código *script* com HTML, dificultando a separação das funções de definição de *layout* e programação durante o desenvolvimento de um projeto.
- O reaproveitamento do código em ASP também é outro aspecto limitador, já que elementos HTML e ASP encontram-se misturados em um mesmo documento.
- As ferramentas para desenvolvimento de aplicativos ASP e depuração são limitadas, principalmente porque uma página ASP combina elementos de várias tecnologias como VBScript, JavaScript, CSS, XML, etc.
- O ASP apresenta também dificuldades para configuração da aplicação que é realizada através de um arquivo próprio (*metabase*), para manipular respostas do servidor e dar suporte a diferentes navegadores.

Mesmo com todas estas desvantagens, o ASP, como desenvolvimento de páginas dinâmicas para *web* apresentava-se superior as outras tecnologias utilizadas, como *Common Gateway Interface* (CGI), extensões e filtros *Internet Server Application Programming Interface* (ISAPI) e *Internet Database Conector* (IDC).

CGI são aplicações instaladas e executadas no servidor web. Estas aplicações são a interface entre os dados e as requisições dos usuários, mas apresenta difícil manutenção e depuração e gastam grande volume de recursos e tempo de processamento, pois, a cada requisição, um novo processamento da aplicação é iniciado.

ISAPI são aplicativos desenvolvidos como *Dynamic Link Libraries* (DLL), usados para processar formulários, acessar banco de dados, executar lógicas, etc. Os ISAPI são mais rápidos que os CGIs já que permanecem na memória do computador até que seja

desligado porém, isso ocasiona grande dificuldade em sua utilização, já que uma aplicação deficiente pode provocar sérios danos ao computador, pois, estes aplicativos ficam carregados permanentemente.

O IDC é um recurso especial do IIS que permite o acesso a banco de dados, porém não permite adicionar lógica ao servidor.

As razões acima apresentadas mostram a superioridade do ASP em relação às outras ferramentas para desenvolvimento de aplicações interativas e dinâmicas.

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Agenda - Curso de Análise Estrutural </title>
</head>

<body link="#000000" vlink="#000000" alink="#000000" topmargin="0" leftmargin="0" background=
<table border="0" cellpadding="0" cellspacing="0" width="710" height="192">
  <tr> <td height="20"><font face="Verdana" color="#008080"><b>Agenda</b></font></td></tr>
  <tr><td height="15"><font face="Verdana" size="2">Oi, <%=session("Nome")%>!</font></td>
</tr>
  <tr>
    <td height="20"><font face="Verdana" size="2">Voc&ecirc; est&aacute; no m&ocirc;du&ccedil;o
    <%=Aluno("Modulo")%>.
    </font></td>
  </tr>
  <tr>
    <td height="20"> <font face="Verdana" size="1">| <font color="#FF9933">Agenda</font>
    <%
if Aluno("UltimoArquivo")<>"" then
  response.write "<font size=""2"" face=""Verdana, Arial, Helvetica, sans-serif""><a href="
Else
  response.write"<font face=""Verdana"" size=""1"" color=""#008080"">N&atilde;o h&aacute; &ccedil;o
end if
conexao.close
set conexao=nothing
%>
</td>
</tr>
</table>
</body>
</html>

```

FIGURA 12 – Arquivo “.asp” com códigos HTML e códigos ASP, em destaque

3.2.8. UML

Unified Modeling Language (UML) é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas, inclusive 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 *software* 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. A UML é controlada pelo Grupo de Gerenciamento de Objeto (*Object Management Group* - OMG) e é um padrão da indústria para descrever um *software* graficamente.

A UML é composta por muitos elementos de modelo, que representam as diferentes partes de um sistema de *software*. Os elementos UML são usados para criar diagramas que representam uma determinada parte ou um ponto de vista do sistema.

A UML surgiu em 1996 quando Rumbaugh unificou alguns outros métodos de modelagem de sistemas existentes na época, podendo-se dizer que a UML nasceu das melhores práticas de engenharia que provaram ter sucesso na modelagem de sistemas grandes e complexos.

3.3. O FRAMEWORK .NET

O *Framework .NET* é uma infra-estrutura da Microsoft® para aplicações e serviços *web*. O .NET é um ambiente para desenvolvimento de aplicações, responsável por gerenciar tudo o que roda dentro dele, contendo bibliotecas comuns como ADO.NET, ASP.NET e *Windows Forms*. O *Framework .NET* pode ser entendido como um ambiente de execução e uma biblioteca de classes que espelham as funcionalidades do sistema operacional.

Este *framework* fornece alta produtividade para o desenvolvimento de aplicações, é baseado em padrões, ambiente multi-linguagem, que permite a integração de aplicações existentes, bem como agilidade para resolver os desafios da distribuição e das aplicações da internet em escala. O *framework.NET* consiste de três partes principais: *Common Language Runtime* (CLR), um conjunto hierárquico das *class libraries* unificadas e o componente ASP.NET, que é uma versão do ASP. A figura 13 mostra esta organização do *Framework.NET*, as tecnologias para construção de aplicações:

ASP.NET (aplicações web) e *Windows Forms* (aplicações *desktop*), as *class libraries* e o CLR.

O *Framework .NET* foi desenvolvido com os seguintes padrões da internet: http (o protocolo de comunicação entre aplicações da internet), XML (formato de troca de dados entre aplicações da internet), SOAP (formato padrão para requisição de *web services*), UDDI (padrão para busca de *web services*).

O *Framework.NET* apresenta várias funcionalidades que facilitam o desenvolvimento de aplicativos, podendo-se citar: Abstração de *Hardware* e do Sistema Operacional, bibliotecas (*.NET Framework Class Libraries*) e arquitetura em N camadas.

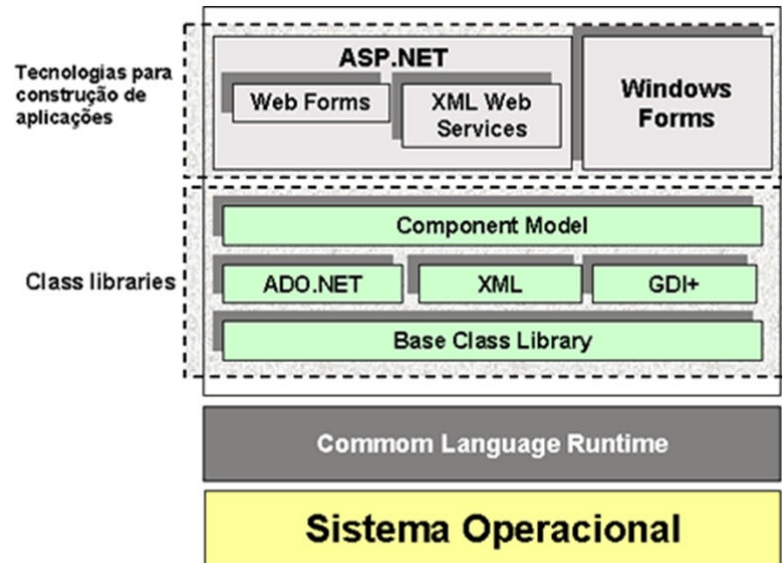


FIGURA 13 – Organização do framework.NET

Fonte: <http://microsoft.com>

3.3.1. Abstração de *Hardware* e do Sistema Operacional

Abstração do Sistema Operacional é a capacidade do .NET de criar uma camada intermediária entre o aplicativo e o sistema operacional. Isto é possível a partir da criação de uma linguagem intermediária, a MSIL (*Microsoft Intermediate Language*). Todo aplicativo .NET é pré compilado nesta linguagem intermediária e, é compilado através do CLR (*Common Language Runtime*) em tempo de execução, na máquina do cliente, quando requisitado.

O CLR é o motor no núcleo da execução de um código gerenciável. O CLR executa os seguintes serviços em um código gerenciável (um código gerenciável é o código que é executado no ambiente do CLR ao invés de ser executado no sistema operacional da máquina): integração de linguagens, segurança de acesso ao código, gerenciamento do tempo de vida dos objetos, gerenciamento de memória etc. Em outras palavras, o CLR gerencia a execução do código das aplicações e componentes, além de fornecer um conjunto de serviços que simplifica o desenvolvimento e distribuição das aplicações.

Este fato permitiu duas evoluções na programação: cada desenvolvedor pode escolher a linguagem em que irá programar: C++, C# (simplificação do C++), VB.NET (evolução do VB), J# (versão da *Microsoft*® do Java), COBOL, etc, sem perda de recursos ou performance. O CLR é capaz de executar, atualmente, mais de vinte diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem. A outra evolução é a independência do sistema operacional, permitindo que usuários do *Windows* 98 a 2003, Linux e Mac possam utilizar o mesmo aplicativo.

A figura 14 apresenta de forma esquemática a abstração do sistema operacional do *framework*.NET e o mecanismo de multilinguagem permitido a partir desta abstração. As linguagens originais ou o código fonte é inicialmente compilado em uma linguagem intermediária pelo CLR e posteriormente, através do compilador JIT (anacrônimo para *Just-in-time*) é transformado em código nativo que será processado pela máquina somente quando o código é requisitado em tempo de execução. Da maneira tradicional, o código fonte é compilado diretamente para código nativo processado pela máquina.

A abstração de *hardware* acontece no *framework*.NET porque o sistema operacional e suas funcionalidades não são acessados diretamente pelo programador, existem inúmeras bibliotecas e classes nativas do .NET que facilitam esse trabalho para o programador.

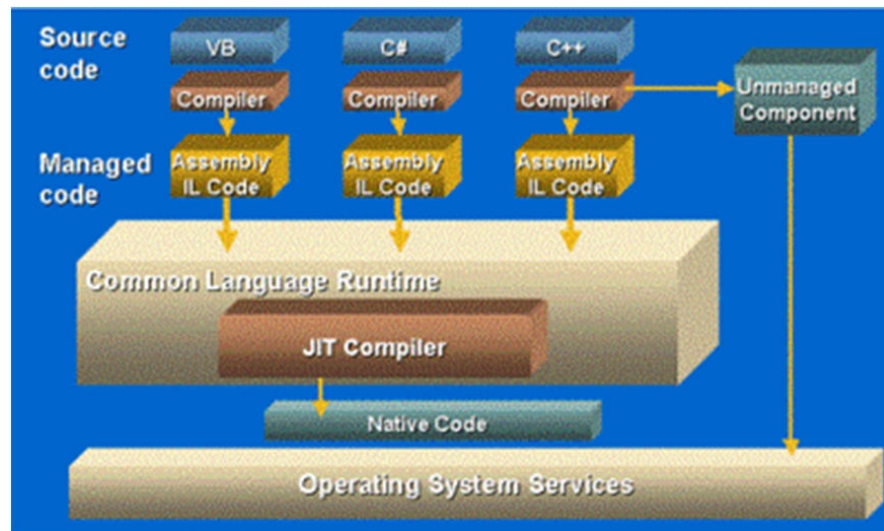


FIGURA 14 – Mecanismo de multilinguagem permitido pelo *Framework* .NET

Fonte: <http://microsoft.com>

3.3.2. *Class Libraries*

O *Framework*.NET é composto por bibliotecas ou *assemblies* (DLL ou EXE) que dão suporte a qualquer linguagem que rode dentro dele, facilitando o desenvolvimento de aplicações e componentes.

O *Framework Class Libraries* está organizado em *namespaces* (esquema de nomeamento lógico para agrupar tipos de funcionalidades relacionadas), tornando fácil a localização de classes e funções específicas. Estas bibliotecas apresentam código aberto, dando possibilidade para utilização de objetos diretamente ou especialização dos mesmos, além de prover acesso às funcionalidades do sistema e foram desenvolvidas para ser a base das aplicações, componentes e controles do *framework* .NET.

As *Class Libraries* apresentam apenas código dentro dos padrões do CLS – *Common Language Specifications*, que é um subconjunto com características da linguagem suportada pelo *common language runtime*, incluindo características comuns a várias linguagens de Programação Orientada a Objetos (POO).

A figura 13 mostra as bibliotecas constituintes do *framework*.NET.

A *Base Class Library* (BCL) envolve os seguintes *namespaces*: *System*, *System.Collections* e *System.Diagnostics*. Suas funcionalidades abrangem manuseio de tipos primitivos (*Int32*, *String*, *Boolean*), estruturas de dados (*array*), classes para conectividade, implementação de protocolos, leitura e escrita de arquivo, processamento de textos, dentre outras.

A biblioteca ADO.NET provê acesso a dados relacionais e à linguagem XML, é composta pelo *namespace* e *subnamespaces* do *System.Data*.

A biblioteca XML fornece aos desenvolvedores formas eficientes de acessar dados XML (estruturado e não estruturado). Além disso, inclui os *namespaces* e *subnamespaces* do *System.XML*.

A camada GDI+ é a parte orientada a objetos para manipulação gráfica. É composta pelos *namespaces* e *sub-namespaces* do *System.Drawing*, oferecendo facilidades para renderização de objetos primitivos, vetorização e tipografia e ainda permite a criação de arquivos de imagem de vários formatos (JPEG, GIF, BMP e TIFF).

A biblioteca *Component Model* facilita a transformação de uma classe em um componente que pode ser projetado, personalizado em tempo real em uma ferramenta. Consiste dos *namespaces* e *sub-namespaces* *System.ComponentModel* e *System.ComponentModel.Design*.

3.3.3. Arquitetura em N Camadas

O *Framework.NET* permite a separação do código do aplicativo em camadas. Usualmente são utilizadas apenas duas camadas. Em aplicativos cliente-servidor, a requisição é feita no cliente e a compilação do código é feita no servidor e a resposta é devolvida ao cliente.

Outra possibilidade são as aplicações em 3 ou mais camadas, como pode ser observado esquematicamente na FIG. 15; estas camadas são usualmente as seguintes:

- Interface: cuida somente da visualização e apresentação da aplicação.

- Camada de Dados: camada de programação, que manipula os dados armazenados em um Banco de Dados. Pode ser dividida em duas camadas: armazenamento e manipulação de dados. Esta camada costuma ser armazenada em servidores de dados.
- Camada de Aplicações: desenvolve os processos de lógica da aplicação, acesso e atualização de dados. Esta camada também pode ser dividida em duas camadas: desenvolvimento da lógica de dados e desenvolvimento da lógica das aplicações. Esta camada costuma ser armazenada em servidores de aplicação.

Entre as camadas de desenvolvimento pode haver separação física ou não. As camadas podem estar localizadas em servidores distintos ou em um mesmo servidor.

A programação em várias camadas apresenta inúmeras vantagens, dentre as quais, pode-se citar:

- Facilita o trabalho em equipe, pois a aplicação pode ser modularizada e escalonada. Aplicando-se o conceito de modularização, uma aplicação pode ser desenvolvida por partes e o fato da aplicação ser escalonada é que durante ou após o desenvolvimento, novas funcionalidades poderão ser acrescentadas ao seu núcleo.
- Rapidez no desenvolvimento de aplicações, já que podem ser desenvolvidas facilmente por equipes, cada profissional atuando no assunto em que é especialista.
- Completa separação do código e interface resultando em maior facilidade para reutilização, estender e modificar o código sem ter que necessariamente recompilar o código no cliente.
- Mudanças nos métodos de uma das camadas não terão influência nas outras camadas.
- A aplicação cliente não sabe de onde vêm os dados armazenados.

Desde o lançamento do *framework.NET*, há atualizações periódicas deste sistema e atualmente, foi lançada a versão 3.5 que apresenta novas características e ampliou as versões 2.0 e 3.0 com novas ferramentas, novas tecnologias para construção de aplicações, melhoramentos das ferramentas existentes e ampliação da documentação. Algumas bibliotecas consideravelmente ampliadas foram o ADO.NET e o ASP.NET.

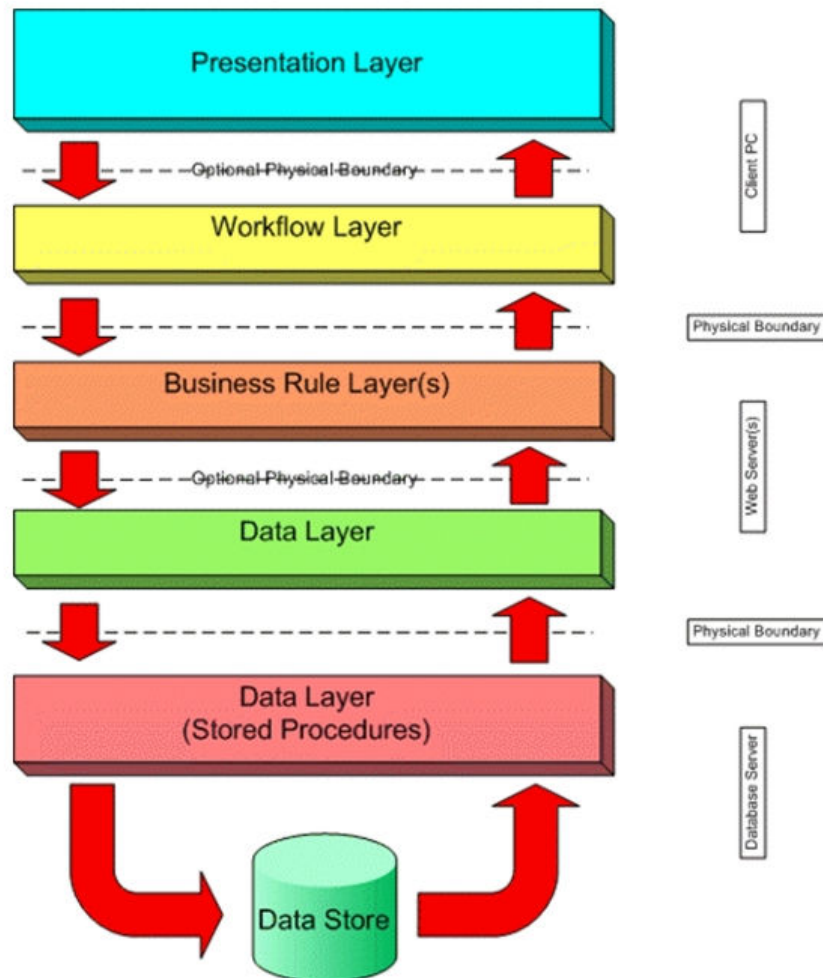


FIGURA 15 – Arquitetura em n Camadas

Fonte: <http://microsoft.com>

3.3.4. ASP.NET

ASP.NET é uma nova versão do ASP, que faz parte do *Framework.NET* da *Microsoft* utilizada para fazer páginas da *web* dinâmicas e interativas. O ASP.NET, através do *Framework.NET* suporta o modelo de programação orientado a objetos e o seu código

pode ser compilado. Suporta ainda o gerenciamento de estado, a utilização de modelo baseado em componentes, além de permitir o uso de várias linguagens e a completa separação do código e da linguagem HTML. Os aplicativos construídos com ASP.NET são hospedados no *Microsoft* IIS e utilizam os protocolos de internet como HTTP e SOAP.

O ASP.NET é uma plataforma completa e extensível para o desenvolvimento de páginas *web*, que permite acesso amplo aos recursos do sistema operacional, a banco de dados e *Web Services*, já que apresenta todo o suporte do *framework*.NET.

A execução das páginas *web* desenvolvidas em ASP.NET é bem mais rápida pois, após a página ter sido requisitada, o *framework* verifica se esta página já foi compilada, e a compila apenas uma vez.

O reaproveitamento de código e a facilidade de programação é muito grande utilizando-se o ASP.NET, pois o desenvolvimento de aplicativos pode ser feito apenas acrescentando controles já implementados pelo ASP.NET

Controles são partes de aplicações customizáveis. Os HTML *Controls* são controles típicos de uma página HTML e representam os elementos HTML, mas podem ser programáveis. Exemplos desses controles são os *textbox*, *checkbox*, *labels*, etc. Os *Web Controls* são controles nativos do ASP.NET, mais sofisticados que os HTML *Controls* e passam pelo servidor para execução, exemplos: *datagrid*, calendários, *datalist*, etc... *User Controls* são controles definidos pelo usuário e podem ser formados a partir da composição de dois ou mais controles já definidos, acrescidos de alguma funcionalidade.

As configurações de uma aplicação ASP.NET são feitas através de um arquivo XML.

Embora se possa desenvolver aplicações ASP.Net utilizando somente um editor de texto simples como o *notepad* e o compilador .net, o ambiente de desenvolvimento mais comum das aplicações ASP.NET é o *Visual Studio*.NET já que possui características e recursos que facilitam o trabalho do programador, como os componentes visuais para criação de formulários de páginas *Web*. Entretanto, há outros ambientes para desenvolvimento como o *WebMatrix*.

Utilizando o ASP.NET, a aplicação pode ser desenvolvida em uma das linguagens orientadas a objetos suportadas pelo *Framework*.NET, porém dentro do código da página deverá ser informada linguagem utilizada.

Existem dois modelos de programação no ASP.NET: *Code Behind* e *Code in Page*. No primeiro, o código encontra-se em página diferente dos HTML. Por exemplo, no caso da linguagem utilizada ser o C#, para cada arquivo *aspx* contendo o código HTML, haverá um arquivo *aspx.cs* onde estará o código C#. O reuso de código nesse modelo é muito grande e há completa separação da linguagem de programação e do HTML. No segundo modelo, a linguagem de programação e o HTML encontram-se separados, porém na mesma página.

Seguindo a mesma estrutura do *framework*.NET, o ASP.NET apresenta-se organizado em *namespaces*. Os *namespaces* do ASP.NET podem ser observados na TAB. 3.1.

TABELA 3.1 – *Namespaces* do ASP.NET

<i>System.Web</i>	Contém classes utilizadas para implementar o protocolo http, requisições http, além de interfaces que permite estender a arquitetura do ASP.NET na lógica de processamento de uma requisição.
<i>System.Web.UI</i>	Coletivamente formam a página do <i>framework</i> . Implementam as características referenciadas como <i>web forms</i> .
<i>System.Web.UI.HtmlControls</i>	
<i>System.Web.UI.WebControls</i>	
<i>System.Web.Services</i>	Funcionalidades usadas na implementação, descrição, descoberta de serviços <i>web</i> desenvolvidos na forma de arquivos <i>.asmx</i>
<i>System.Web.Caching</i>	Fornecer a implementação de um sistema de cachê em memória que mantém os dados em cachê levando em consideração dependências personalizadas e regras de expiração. Pode ser utilizado para melhorar a performance e tempo de resposta de uma aplicação web. O sistema de cachê também será utilizado no cachê de <i>User Controls</i> .
<i>System.Web.SessionState</i>	Funcionalidades utilizadas para gerenciamento de estado – objeto <i>Session</i> . Todo usuário conectado à aplicação <i>web</i> possui um estado (objeto <i>session</i>). O tempo de vida desse objeto será até a sessão expirar ou quando for finalizada via código. Os dados da sessão

	podem ser mantidos em memória ou num banco de dados SQL Server.
<i>System.Web.Security</i>	Funcionalidades para implementação de vários mecanismos de autenticação usados em aplicações <i>web</i> , tais como autenticação via formulário, autenticação <i>windows</i> e <i>Microsoft Passport</i> (quando estamos conectados ao <i>messenger</i> ou conta do <i>email</i> do <i>hotmail</i> estamos utilizando este mecanismo)
<i>System.Web.Mail</i>	Funcionalidades para envio de <i>email</i> utilizando o protocolo SMTP (<i>Simple Mail Transfer Protocol</i>)

Fonte: Guimarães (2004)

O ASP.NET, através do .NET *framework* 2.0 ganhou ainda mais facilidades e novos instrumentos, que facilitam as atividades desenvolvidas pelos programadores.

No ASP.NET 2.0 é possível a utilização de *templates*, pois houve a implementação de cerca de cinco novos controles, construídos sobre os novos serviços e funcionalidades do .NET *framework* 2.0. Outra mudança fundamental foi a possibilidade de desenvolver aplicações *web* sem ter o IIS instalado. Isso se deve ao fato do *framework* apresentar seu próprio servidor *web* embutido. Esse servidor visa, dentre outros objetivos, aumentar a segurança de desenvolvimento, uma vez que o *site* não poderá ser acessado remotamente.

O ASP.NET 2.0 continua sendo utilizado com o .NET *framework* 3.5.

Encontra-se em desenvolvimento o novo ASP.NET AJAX (JavaScript e XML assíncronos). O novo acréscimo do Microsoft ASP.NET AJAX ao .NET Framework inclui um conjunto de bibliotecas JavaScript, o qual é a base para a criação de comportamentos sofisticados e funcionalidades de controle no navegador. Estas funcionalidades estão disponíveis desde outubro de 2005, mas vem sofrendo constantes modificações.

3.3.5. C# (C Sharp)

C# é uma linguagem de programação orientada a objetos desenvolvida pela *Microsoft* em 2001, como parte da plataforma .NET e foi mais tarde, aprovada como padrão pela

ECMA e ISO em 2003 e 2001, respectivamente. Foi revisada pela última vez em 2006. A *Microsoft*® baseou-se nas linguagens C++ e Java para seu desenvolvimento.

A linguagem C# é a linguagem que está mais vinculada ao *framework*.NET. O seu código é 100% gerenciado e suas estruturas de dados primitivas são objetos que correspondem a tipos no .NET. A Microsoft ainda admite um pequeno ganho de desempenho quando é utilizado o C#.

A linguagem C# apresenta as seguintes diferenças em relação ao C++:

- Objetos não são liberados explicitamente, mas somente através do processo *garbage collection*, processo de coleta de lixo próprio do .NET, prevenindo referências inválidas;
- Não existem destrutores;
- Só é permitida herança simples entre classes, mas uma classe pode implementar várias heranças abstratas. O objetivo disso é simplificar a implementação do ambiente de execução;
- C# é mais seguro que C++, não permitindo várias conversões implícitas possíveis no C++;
- A sintaxe para declaração de vetores é mais clara;
- Existem propriedades que permitem que métodos sejam chamados com a mesma sintaxe de acesso a membros de dados.

Em relação a Java, algumas diferenças são importantes:

- Com o C# é possível implementar propriedades e sobrecargas de operadores;
- C# utiliza comentários baseados em XML para gerar documentação automática;
- C# suporta indexadores.

As especificações e padrões para os programas desenvolvidos utilizando C# são estabelecidos pela ECMA-334.

Em resumo, o C# é a linguagem mais vinculada ao *framework*.NET, contempla o mesmo potencial do C++, e ainda funciona de forma mais segura e simplificada. Além disto a linguagem disponibiliza as facilidades inerentes da linguagem Java.

3.4. BANCO DE DADOS

3.4.1. Definições

Um Banco de dados pode ser definido como um conjunto integrado de dados (informações) relacionados a um determinado assunto ou finalidade representando algum aspecto específico do mundo real, tendo alguma relação lógica entre seus componentes, armazenados sob alguma forma física, que tem por objetivo atender a uma comunidade de usuários.

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é um *software* que incorpora as funções de definição, alteração, recuperação, armazenamento, segurança física, controle dos dados e integridade em um banco de dados, além de prover aos usuários uma visão abstrata dos dados, havendo separação entre a parte lógica e física do armazenamento. Um sistema de Banco de Dados (BD) é composto por um banco de dados e um SGBD.

3.4.2. Modelos de Dados

Um modelo de dados é uma descrição dos tipos de informações que estão armazenadas num banco de dados, ou seja, a forma de representação da informação no banco de dados. Os modelos principais são: Modelo Conceitual, Modelo Físico e Modelo Lógico, como podem ser observados na FIG. 16. Há quatro formas de representação da lógica das informações no modelo lógico: Hierárquico, Rede, Relacional, Orientado a Objetos.

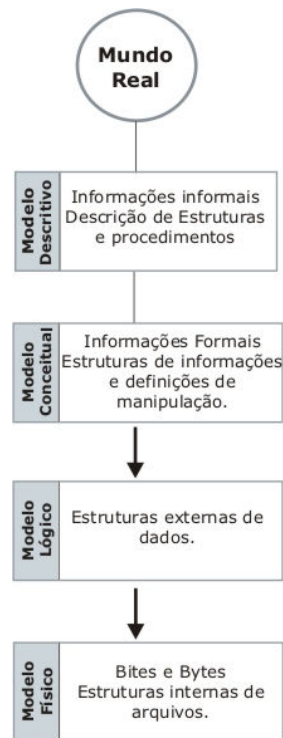


FIGURA 16 – Níveis de Abstração de um Sistema de Banco de Dados

O Modelo Conceitual é o modelo que mostra as entidades e seus relacionamentos em um banco de dados, ou seja, a estrutura de um banco de dados, independente de um SGBD.

O Modelo Físico representa detalhes de armazenamento interno de informações que não tem influência sobre a programação de aplicações no SGBD, mas podem influenciar a o desempenho das aplicações.

O Modelo Lógico é um modelo de dados que representa a estrutura de dados de um BD conforme vista pelo usuário do SGBD. Assim, um SGBD pode ser de um dos quatro tipos citados anteriormente: Hierárquico, Rede, Relacional, Orientado a Objetos.

No SGBD Hierárquico, que representa os dados na forma de árvore, cada ocorrência de um objeto possui, associada a ela, um conjunto de ocorrências de outro tipo de objeto. Linguagens diferentes podem ser utilizadas para manipulação de dados e ligações entre eles.

O sistema em Rede foi proposto pela CODASYL em 1971. Os dados são apresentados por dois elementos básicos: o “*record type*”, que define um objeto e o “*set type*”, que

define o relacionamento entre objetos. As ligações entre tipos de dados diferentes são físicas e são implementadas através de cadeias de apontadores. Também possui linguagens diferentes para manipulação de dados e apontadores.

O sistema de banco de dados mais usado é o Relacional, proposto por E. F. Codd da IBM em 1970. Este sistema é baseado num modelo matemático sólido, que envolve teoria de conjunto, álgebra relacional e cálculo relacional. Os dados estão armazenados em tabelas as quais apresentam relações entre si. Possui somente uma linguagem para manipulação de dados e relacionamentos. Exemplos desse sistema são: DB2 da IBM, *Access* da *Microsoft*, Oracle, *SQL Server* da *Microsoft*, SYBASE, INGRES, INFORMIX, dentre outros.

O SGBD com incorporação da orientação a objetos de modo simplificado permite que cada elemento de dado seja tratado como um objeto único, com sua própria identificação e características. A informação é localizada pela identidade do objeto e não pelos valores associados a ele.

Para a representação de um modelo são necessários os seguintes itens:

- Entidade: objeto do mundo real, um fato, ser, organismo social ou coisa.
- Atributo: informações que se deseja guardar do objeto.
- Relacionamento: associação existente entre elementos de entidades.
- Cardinalidade: número de ocorrências possíveis em cada entidade envolvida em um relacionamento.

3.4.3. Conceitos dos Bancos de Dados Relacionais

Os bancos de dados relacionais, ou SGBD relacionais são compostos de:

- **Tabelas:** objeto criado para armazenar os dados fisicamente. São as estruturas de armazenamento de dados. Cada linha de uma tabela é o registro ou entidade. As colunas representam os campos ou atributos das entidades.

- **Visões:** representação lógica de um banco de dados, não contém dados. São os modos de visualização das tabelas (as tabelas podem ser visualizadas eliminando linhas, colunas ou unindo duas tabelas);
- **Índice:** ferramenta utilizada pelo gerenciador de banco de dados para facilitar a busca de linhas em uma tabela;

Um outro conceito importante no modelo relacional de dados é a “**Chave Primária**”. Ao se definir um campo como sendo uma Chave Primária, está sendo informado ao SGBD que não podem existir dois registros com o mesmo valor no campo que é a Chave Primária, ou seja, os valores no campo Chave Primária precisam ser únicos. O campo Chave Primária identifica de maneira única cada registro de uma tabela. A Chave Primária pode ser formada pela combinação de mais de um campo. Podem existir casos em que um único campo não é capaz de atuar como chave primária, pelo fato deste apresentar valores repetidos. Uma tabela pode ter somente uma chave primária.

Os relacionamentos entre as tabelas de um SGBD podem ser de três tipos: **um para um, um para vários e vários para vários**.

Um para um: Esta relação existe quando os campos que se relacionam são ambos do tipo Chave Primária, em suas respectivas tabelas. Tais campos não apresentam valores repetidos.

Um para vários: Uma das tabelas (o lado um do relacionamento) possui um campo que é a Chave Primária e a outra tabela (o lado vários) se relaciona através de um campo cujos valores relacionados podem se repetir várias vezes.

Vários para vários: Este tipo de relacionamento aconteceria em uma situação onde em ambos os lados do relacionamento os valores poderiam se repetir. Na prática não é possível implementar um relacionamento deste tipo, devido a uma série de problemas que seriam introduzidos no modelo do banco de dados.

Outro conceito importante em um Sistema de Gerenciamento de Bancos de Dados Relacionais (SGBDR) é o da **Integridade Referencial**. A Integridade Referencial é utilizada para garantir a Integridade dos dados entre as tabelas relacionadas. O sistema

verifica a inclusão ou exclusão de dados dependentes de outros e, conforme a escolha dos usuários essas ações podem ser propagadas ou não para as outras tabelas relacionadas.

3.4.4. Projeto de Bancos de Dados

Um projeto de Banco de dados apresenta basicamente duas fases: modelagem conceitual e modelagem lógica, conforme ilustra a FIG 17.

Algumas fases são recomendadas em um projeto de banco de dados:

- **Determinar qual o objetivo do banco de dados:** Isto ajuda na determinação de quais os dados devem ser armazenados.
- **Determinar as tabelas necessárias:** Após definir os objetivos do Banco de Dados, as informações devem ser definidas e separadas em assuntos diferentes.
- **Determinar os Campos de cada Tabela:** Definir quais informações devem ser mantidas em cada tabela.
- **Determinar a Chave Primária de cada tabela:** Sendo que pode haver tabelas onde não exista uma chave primária: Determinar, em cada tabela, quais campos serão utilizados como Chave Primária. Esta é uma etapa importantíssima para a definição dos relacionamentos que vem a seguir.
- **Determinar os Relacionamentos:** Decidir como os dados de uma tabela se relacionam com os dados de outras tabelas.
- **Refinar a Estrutura do Banco de Dados:** verificar se a estrutura contém erros, isto é, verificar se os resultados obtidos são os desejados. Isto, normalmente, pode ser obtido através do processo de Normalização. Caso necessário deve-se alterar a estrutura do banco de dados.

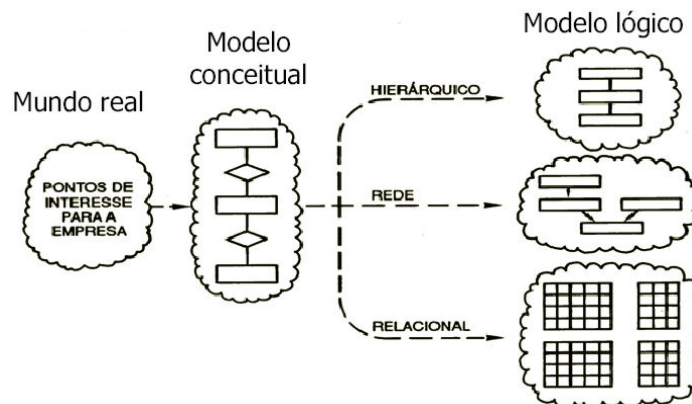


FIGURA 17 – Modelos utilizados em um projeto de BD

É usual que se adote em um projeto de banco de dados a normalização de tabela, cujo objetivo é evitar os problemas provocados por falhas no Projeto do Banco de Dados, bem como eliminar a "mistura de assuntos" e as correspondentes repetições desnecessárias de dados. O Processo de Normalização aplica uma série de regras sobre as tabelas de um Banco de Dados, para verificar se estas estão corretamente projetadas. Embora existam cinco formas normais (ou regras de Normalização), na prática usa-se um conjunto de três Formas Normais.

Primeira Forma Normal: Uma Tabela está na Primeira Forma Normal quando seus atributos não contêm grupos de Repetição, ou seja, campos, que repetirão várias vezes entre as entidades da tabela, podendo provocar erros, inclusive de digitação.

Segunda Forma Normal: Ocorre quando a Chave Primária é composta por mais de um campo. Neste caso, deve-se observar se todos os campos que não fazem parte da chave dependem de todos os campos que compõem a chave. Se algum campo depender somente de parte da chave composta, então este campo deve pertencer à outra tabela.

Terceira Forma Normal: Na definição dos campos de uma entidade podem ocorrer casos em que um campo não seja dependente diretamente da chave primária ou de parte dela, mas sim dependente de outro campo da tabela, campo este que não seja a Chave Primária.

3.4.5. Sistemas Gerenciadores de Banco de Dados

A Linguagem SQL

SQL “*Structured Query Language* - Linguagem Estrutural de Consultas”. SQL é a linguagem padrão mais comum usada para acessar banco de dados e é definida pelo padrão *American National Institute* (ANSI)/ISO SQL.

SQL é uma linguagem simples, em que você facilmente pode gravar, alterar e recuperar informações num *web site* com segurança e rapidez. Ela foi desenvolvida pelo Departamento de Pesquisas da IBM como forma de interface para o Sistema de Banco de Dados Relacionais SYSTEM R no início dos anos 70. Em 1996, a ANSI publicou um padrão SQL. A SQL estabeleceu-se como linguagem padrão de Banco de Dados Relacional.

Atualmente, os vários bancos de dados existentes fazem algumas modificações nesta linguagem para se servirem dela. Por exemplo, o banco de dados Oracle utiliza a PL/SQL, o SQL Server a T-SQL.

Access

O *Access* é um sistema gerenciador de banco de dados relacional da *Microsoft* indicado para aplicações pequenas e médias. O *Access* é um arquivo de dados e não um servidor de dados como o *SQL Server* e os demais bancos de dados apresentados a seguir. A interface do *Access* é bastante didática e intuitiva, por isso, é indicado para iniciantes em BD. A primeira versão deste *software* foi lançada pela *Microsoft* em 1992. Foi a partir da versão 8.0 (97) que o *Access* ganhou maior integração com a *internet*, principalmente trabalhando com a linguagem ASP.

O limite de armazenamento do *Access* é apenas de 2 GB (dois gigabytes) e pode armazenar apenas 10 tipos diferentes de dados. Capacidade muito pequena se comparado a outros bancos de dados.

My SQL

O MySQL é um sistema de banco de dados relacional, multi-usuário e multitarefa. Utiliza a linguagem SQL. O MySQL é um *software* gratuito e aberto “*open source*”, ou seja, qualquer pessoa ou empresa pode usar e modificar o programa.

O MySQL foi originalmente desenvolvido pela empresa sueca TCX, que necessitava de um servidor de banco de dados que operasse com grandes escalas de dados rapidamente sem exigir caríssimas plataformas de *hardware*.

MySQL é muito utilizado com a linguagem *Hypertext Preprocessor* (PHP) de programação para a *internet*, concorrendo com o *Access* para ASP.

O MySQL possui inúmeras opções de variáveis distribuídas nos tipos *strings*, numéricos e data e hora. Essa característica, associada à rapidez do MySQL, faz com que ele seja um dos bancos de dados mais recomendados para sites da *web*.

As características do MySQL são enumeradas da seguinte maneira:

- **Suporta diferentes plataformas:** Win32, Linux, FreeBSD, Unix,;
- **Suporte às API's das Seguintes linguagens:** PHP, Perl, C,C++,Java, Python, etc...;
- **Suporte a múltiplos processadores;**
- Sistema sofisticado: Tem recursos de senhas criptografadas, é flexível e Seguro;
- **Suporte à ODBC:** Pode facilmente conectar o Access a um banco de dados do MySQL;
- **Suporta até 16 índices por tabela;**
- **Código fonte escrito em C e C++:** Código testado com uma variedade de diferentes compiladores;
- **O Cliente se conecta ao MySQL através de conexões TCP/IP.**

DB2

O DB2 é um banco de dados relacional da IBM. Funciona em sistemas MVS (armazenamento virtual múltiplo). O Banco de dados DB2, também chamado de “Base de Dados Universal” funciona nos seguintes sistemas operacionais: Windows NT, AIX, Solaris e Linux. Pode ser utilizado para aplicações grandes.

Interbase

O *Interbase* foi desenvolvido pela Borland. Atualmente o *Interbase* está na versão 6, disponível para os sistemas operacionais Windows, Linux, Unix, Solaris, NetWare, entre outros. A partir desta versão, o *InterBase* se tornou um *software aberto e gratuito* “open-source” e “free”.

O *Interbase* oferece suporte a campos BLOB (Binary Large Object – que são tipicamente objetos de imagens, sons ou multimídias), armazena desde textos até objetos gráficos, binários ou outro tipo qualquer pré-definido pelo usuário. Os campos inteiros armazenados são de até 64 bits. Suporta diversos protocolos: local, TCP/IP, NetBeui, IPX/SPX (Novell). Apresenta funções definidas pelo usuário, podendo ser usada qualquer linguagem que gere DLLs, além de ser multi-transacional.

Segundo alguns programadores, o *Interbase* não perde em quase nada para grandes bancos de dados como Oracle e SQL Server, a não ser por causa de sua pouca popularidade, pois ele possui todos os recursos de um grande BD: *domains*, *stored procedures*, *views*, *roles*, *triggers*, *generators*, *exceptions*, suporte a *User Defined Functions* (UDF), além de uma integridade referencial impecável, controle de usuários e permissões, ferramentas de *backup/restore*, *sweep* e reparação de BD. Além de apresentar outras vantagens: é leve, é grátis e possui versões para Linux e Solaris (O SQL Server não possui). Porém, o *Interbase* apresenta perda de *performance* se muitos usuários fizerem requisições ao mesmo tempo, mas é um banco multi-usuário para grandes processamentos de dados. Outra desvantagem do *Interbase* é que a Borland não desenvolveu *drivers* ODBC ou OLEDB para acessar o *Interbase*, tendo em vista que o acesso por Kylix/Delphi/C++Builder/JBuilder é praticamente nativo. Assim, com as linguagens ASP/VB não seria possível acessar este banco de dados. A solução para este problema é de procurar *drivers* ODBC desenvolvidos por outras iniciativas.

Oracle

O Oracle é também um banco de dados relacional, bastante completo e eficiente (considerado o sistema de banco de dados mais completo em termos de funções e recursos de segurança), é um produto vendido em nível mundial, mas devido o seu elevado custo, é utilizado apenas por grandes empresas. No desenvolvimento de páginas *web* acontece o mesmo: como é um sistema muito caro, não está tão difundido no meio como outras bases de dados, por exemplo, *Access*, *MySQL*, *SQL Server*, etc.

Para desenvolver em Oracle utiliza-se PL/SQL uma linguagem de 5ª geração, bastante potente para tratar e gerenciar a base de dados.

SQL Server 2000

O *SQL Server 2000* é um sistema de gerenciamento de bancos de dados cliente/servidor de alto desempenho com alta integração com o *Windows* e com o *Framework.NET* e dá suporte a linguagem XML. Opera sobre o protocolo TCP/IP, indicado para grandes aplicações e concorre diretamente com o Oracle e o DB2.

Suas características são: Integração com os serviços de *multithreading* (múltiplas tarefas), agendamento, Monitor de Desempenho, e *log* de eventos do *Windows*.

A tecnologia de replicação nativa permite disseminar informações para vários locais, reduzindo a dependência de um servidor único, e deixando a informação necessária mais próxima de quem realmente precisa dela. Arquitetura paralela, que executa as funções de banco de dados simultaneamente para diversos usuários e tira proveito de sistemas com múltiplos processadores. Gerenciamento centralizado de todos os servidores através de uma arquitetura de gerenciamento distribuída, com uma interface visual de gerenciamento.

Para desenvolver em *SQL Server* utiliza-se T-SQL (*Transact-SQL*), uma variação da linguagem *SQL-92 standard*, direcionada e otimizada para este servidor de banco de dados.

MSDE 2000

O *Microsoft SQL Server 2000 Desktop Engine* (MSDE) é uma versão grátis do *SQL Server*. Da mesma forma que o *SQL Server*, o MSDE é um servidor de dados relacional, que pode usado em aplicações de menor porte. Está limitado a 2 (dois) GB de capacidade de armazenamento e suporta *web sites* com até 25 usuários simultâneos. É totalmente compatível com o *SQL Server* e por isso, pode ser uma alternativa para aplicações menores que não necessitam de altos investimentos.

PostgreSQL

O *PostgreSQL* é um sistema de gerenciamento de banco de dados objeto-relacional (SGBDOR) baseado no POSTGRES, Versão 4.2, desenvolvido no Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O projeto POSTGRES, liderado pelo Professor Michael Stonebraker, foi patrocinado pelas seguintes instituições: *Defense Advanced Research Projects Agency* (DARPA); *Army Research Office* (ARO); *National Science Foundation* (NSF); e ESL, Inc.

O *PostgreSQL* descende do código original de Berkeley, possuindo o código fonte aberto. Fornece suporte às linguagens SQL92/SQL99, além de outras funcionalidades modernas. Dispõe de um amplo conjunto de ligações com linguagens procedurais (incluindo C, C++, Java, Perl, Tcl e Python).

O POSTGRES foi pioneiro em muitos conceitos objeto-relacionais que agora estão se tornando disponíveis em alguns bancos de dados comerciais. Os Sistemas de Gerenciamento de Bancos de Dados Relacionais (SGBDR) tradicionais suportam um modelo de dados composto por uma coleção de relações com nome, contendo atributos de um tipo específico. Nos sistemas comerciais em uso, os tipos possíveis incluem número de ponto flutuante, inteiro, cadeia de caracteres, monetário e data. É amplamente reconhecido que este modelo não é adequado para aplicações futuras de processamento de dados. O modelo relacional substituiu com sucesso os modelos anteriores principalmente devido à sua grande simplicidade. Entretanto, esta simplicidade tornou a implementação de certas aplicações muito difícil. O *PostgreSQL* oferece recursos adicionais pela incorporação de outros conceitos (herança, tipos de dado, funções), tornando possível os usuários estenderem o sistema facilmente.

Outras funcionalidades (restrições, gatilhos, regras, integridade da transação) fornecem poder e flexibilidade adicionais.

Estas funcionalidades colocam o *PostgreSQL* dentro da categoria de banco de dados referida como **objeto-relacional**. Diferente daqueles referidos como **orientados a objetos** que, em geral, não são muito adequados para apoiar as linguagens tradicionais de banco de dados relacionais. Portanto, embora o *PostgreSQL* possua algumas funcionalidades de orientação a objetos, está firmemente ligado ao mundo dos bancos de dados relacionais. Na verdade, alguns bancos de dados comerciais incorporaram recentemente funcionalidades nas quais o *PostgreSQL* foi pioneiro.

3.5. SOFTWARE UTILIZADOS

Os *software* utilizados para o desenvolvimento do trabalho foram:

- *Microsoft Visio* para o desenho dos diagramas UML da plataforma tecnológica proposta.
- O *DreamWeaver*, *software* da empresa *Macromedia* utilizado para confecção de páginas *web*. Este *software* apresenta diversas facilidades para inserção dos elementos HTML e disposição dos mesmos e, vêm desenvolvendo as versões de modo a disponibilizar aos programadores diversas facilidades. O *DreamWeaver MX 2004* por exemplo já oferece integração direta com os arquivos CSS e *Scripts*, que são inclusive, editados dentro do mesmo ambiente.
- O *Access* foi o SGBD utilizado para curso de Elementos Finitos. O *Access* é um *software* bastante didático e de fácil manuseio, permitindo o desenvolvimento do banco de dados do curso sem maiores dificuldades.
- O *Microsoft Visual Studio 2005* foi utilizado para desenvolvimento das ferramentas bloco de notas, bloco de recados e agenda e do programa do Método dos Elementos Finitos. Este *software* facilita ainda mais a programação em

ASP.NET, permitindo a depuração do código e a montagem das páginas de maneira bastante intuitiva.

- O Flash é um *software* da *Macromedia* e foi utilizado para desenvolvimento de animações no curso de Elementos Finitos. Este programa é muito utilizado para produção de vídeos, imagens e animações para *web*. O *Flash* utiliza o formato vetorial para produção de imagens, resultando em aplicativos leves, próprios para *web*. O *Flash* proporciona acréscimo de interatividade nas páginas *web* e animações, sendo possível que o usuário tenha controle, através de botões ou entrada de dados das animações. Os aplicativos *flash*, além de imagens podem conter sons. As imagens vetoriais, como é o caso das imagens formadas pelo *flash*, podem ser redimensionadas, exibidas em diversos dispositivos com várias resoluções sem perder a qualidade. Para que um arquivo *flash*, extensão *swf*, possa ser executado, basta que o *flash player* esteja instalado no computador local. O *flash player* é um dispositivo utilizado para exibir filmes criados pelo *flash* e pode ser transferido gratuitamente do site da *Macromedia* (www.macromedia.com). O *flash* é uma excelente ferramenta para produção de animações. Além de seguir as tendências atuais da *web* e permitir mais interatividade nas páginas *web*, nessa dissertação foi utilizado como uma ferramenta pedagógica na exploração do sentido visual para estimular o aprendizado.

3.6. OBSERVAÇÕES

Os objetivos específicos desta dissertação de mestrado são o desenvolvimento da concepção de uma plataforma tecnológica para EAD, o desenvolvimento de um curso de Elementos Finitos (MEF), de um programa do MEF para ser executado em ambiente *web* e de ferramentas de auxílio para E@D.

A partir do estudo das tecnologias expostas acima foi possível uma definição e confirmação a respeito de qual tecnologia seria mais apropriada para os resultados pretendidos.

Para o desenvolvimento da concepção da plataforma tecnológica, foi utilizada a linguagem UML para montagem dos seus diagramas de classes. A proposta é que a plataforma seja desenvolvida utilizando o ASP.NET, tendo em vista os inúmeros benefícios e facilidades disponibilizadas pelo ASP.NET e pelo *framework*.NET, como será explicado no capítulo seguinte.

Observando todos os bancos de dados analisados, nota-se que o banco de dados que apresenta maior integração com *Windows* e com o *framework*.NET é o *SQL Server* 2000 ou MSDE 2000 (conforme volume de dados), da própria *Microsoft*, assim o banco de dados proposto para a plataforma tecnológica é o *SQL Server* 2000 ou MSDE.

Para o desenvolvimento do Curso de Elementos Finitos foram utilizadas as linguagens padrão para web: HTML, CSS e XML, além do ASP.NET e *JavaScript*.

O estudo do ASP clássico foi necessário porque o curso de Elementos Finitos estará integrado a escola virtual do CADTEC, o NucleoEAD, todo desenvolvido em ASP, assim, uma integração entre estas duas linguagens foi necessária.

O Banco de Dados utilizado para o curso de Elementos Finitos foi o *Access*, que é o padrão de banco de dados utilizado hoje pelo NucleoEAD.

Estas ferramentas foram escolhidas pensando na possibilidade de desenvolvimento de aplicativos bastante interativos e complexos, capazes de atender as expectativas e necessidades atuais, além da facilidade de desenvolvimento e suporte tecnológico, itens fundamentais para se alcançar o desenvolvimento bem sucedido de uma aplicação e que geralmente não são encontrados em ferramentas gratuitas (*free*).

O HTML, associado ao CSS resultou em mais leveza das páginas do curso de Elementos Finitos, facilitando a construção de *layouts* e seguindo a tendência de projetos *web* (*web design*) atual. A utilização do CSS será ainda muito importante, pois, qualquer alteração necessária no *layout* do curso será facilmente feita apenas com a alteração dos arquivos CSS associados às páginas HTML.

O *JavaScript* permitiu a construção de páginas e *layouts* mais dinâmicos e intuitivos, dando novas possibilidades para exibição de textos, títulos e para a navegação dos alunos.

Com a linguagem ASP.NET foram desenvolvidas três ferramentas adicionais de apoio a EAD para o NucleoEAD, a saber; o bloco de notas, o bloco de recados e a agenda. Estas ferramentas foram desenvolvidas utilizando todas as facilidades do ASP.NET, através de seus componentes e classes.

O sistema computacional para “*web*” para aplicação do MEF a estruturas de barras foi desenvolvido utilizando a base teórica do curso implementado. As ferramentas que possibilitaram o seu desenvolvimento foram a linguagem UML, o *framework* .NET com o ASP.NET, utilizando a linguagem C# e o HTML. A UML foi utilizada para criação e organização das classes e funções do programa.

4

CONCEPÇÃO CONCEITUAL DE UMA PLATAFORMA PARA EAD

4.1. O QUE É UMA PLATAFORMA

Uma Plataforma tecnológica ou *framework* é uma aplicação reusável, que pode ser especializada para produzir aplicações personalizadas. É um conjunto de classes desenvolvidas para reutilização, provendo um guia para uma solução de arquitetura em um domínio específico de *software*. Uma plataforma é reusável porque pode ser utilizada na construção de diferentes sistemas, é semi-completa porque nunca implementa todas as possibilidades e necessidades de um *software* e é customizável porque permite que cada usuário faça adaptações seguindo suas necessidades.

O desenvolvimento de *frameworks* é uma tendência atual. A produção de *software* sob medida é muito cara e, o tempo requerido para o desenvolvimento é grande, tornando o *software* obsoleto antes mesmo de ser utilizado. Através dos *frameworks*, o desenvolvimento de *software* é realizado de forma mais rápida, já que diversas funcionalidades são reutilizadas e adaptadas às necessidades requeridas. Além disso, as ferramentas desenvolvidas não se tornam obsoletas, mas evoluem. A tecnologia disponível atualmente permite o desenvolvimento e utilização de *frameworks*.

Uma plataforma para EAD seria um aplicativo composto por um conjunto de classes para reutilização na criação de ambientes completos para EAD, com as necessidades e facilidades que estes ambientes devem disponibilizar como grupos de discussão, *chats*, sistemas de autoria para cursos, sistemas de gerenciamento de cursos, sistemas de avaliação, etc.

Segundo DOVICCH et al (2003), as plataformas de EAD devem levar em conta dois fatores fundamentais: primeiro, a arquitetura do ambiente do ponto de vista do controle, e segundo, a facilidade de navegação de forma intuitiva, deixando mais tempo para transmissão de conteúdo e menos tempo para suporte tecnológico. O ambiente virtual desenvolvido deve atender às expectativas do aluno, ser amigável e intuitivo tanto no que diz respeito ao conteúdo quanto à navegação, sem perder de vista a qualidade esperada. O ambiente deve ser ainda, flexível, de modo que atenda a cada tipo de estudante com suas necessidades específicas. Neste sentido, as ferramentas de comunicação síncronas e assíncronas exercem um papel muito importante na relação ensino aprendizagem. O fornecimento de um *feedback* aos alunos é um outro aspecto essencial ao aprendizado.

4.2. PROPOSTA DA BASE TECNOLÓGICA

Para DOVICCH (2003), a base tecnológica dos projetos dos ambientes virtuais deve estar atenta a quatro fatores básicos: portabilidade, flexibilidade, segurança e custo. A portabilidade implica na liberdade de escolha do sistema operacional, ou seja, o *software* desenvolvido deverá funcionar independentemente do sistema operacional do cliente. A flexibilidade está relacionada aos aspectos de modularidade e adaptabilidade do código. A segurança está relacionada aos materiais desenvolvidos e aspectos de avaliações e controle acadêmico. Em seu artigo, DOVICCH (2003) comenta sobre o custo efetivo do projeto, o que não está de acordo com a presente proposição, onde foi mais considerado o custo relativo, ou custo-benefício das tecnologias propostas.

Para o desenvolvimento da plataforma pensou-se na utilização de ferramentas que dessem a maior disponibilidade possível aos programadores, em relação aos recursos

oferecidos para desenvolver as ferramentas necessárias e idealizadas de um ambiente para EAD.

A tecnologia escolhida também teria que fornecer suporte tecnológico aos programadores, como grupos de discussão e suporte técnico *online*. As ferramentas deveriam ainda, garantir que o projeto da plataforma fosse desenvolvido de forma rápida e flexível, isto é, que pudesse ser adaptado facilmente seguindo as necessidades de cada usuário.

O ASP.NET é um recurso para programação de internet que oferece grandes facilidades e possibilita alcançar de forma rápida e segura todos os objetivos pretendidos com a plataforma. A partir da escolha dessa linguagem foram escolhidos os outros recursos tecnológicos apropriados e que mais se adaptam a esta linguagem.

A escolha da utilização de tecnologias pagas ao invés de tecnologias gratuitas foi feita por acreditar que a agilidade, rapidez e facilidades fornecidas pelas ferramentas escolhidas, além do suporte tecnológico existente, superam a vantagem do custo. Ou seja, entende-se que o custo-benefício alcançado durante o desenvolvimento do projeto proposto será maior se forem utilizadas as ferramentas pagas ao invés de tecnologias gratuitas e compensa as facilidades de investimento conseguidas ao se propor um custo efetivo inicial mais baixo, já que custos indiretos com mais horas de programadores e suporte tecnológico são inevitáveis.

Os recursos tecnológicos propostos para o desenvolvimento da plataforma foram descritos no capítulo 2 (Recursos Tecnológicos Utilizados) e serão apenas lembrados aqui.

4.2.1. Sistema Operacional

O sistema operacional proposto para desenvolvimento da plataforma será o *Windows*, que é o sistema mais integrado ao *framework*.NET.

4.2.2. Servidor http

O servidor proposto será o *Windows Server* 2000, 2003 ou 2008 com o IIS, aplicativo do Windows responsável pelo gerenciamento de serviços da *internet* e oferece suporte ao ASP.NET.

4.2.3. Linguagens

As linguagens para internet propostas para desenvolvimento da plataforma são o HTML, XML e CSS que são o padrão para desenvolvimentos *web*.

Para programação, a linguagem proposta é o C#, através do ASP.NET e do *framework*.NET. Foi escolhido o ASP.NET devido as imensas facilidades oferecidas pelo *framework*.NET, além de possibilitar o desenvolvimento de aplicações utilizando o conceito de linguagens orientadas a objetos, que permite o desenvolvimento de aplicações muito mais seguras, claras e consistentes, podendo ser reutilizadas, escalonadas e modularizadas. O C# está sendo proposto por ser uma linguagem poderosa e simples, como comentado no capítulo 2.

Outra linguagem utilizada para adicionar interatividade às páginas é o *JavaScript*.

4.2.4. Banco de Dados

O SQL *Server* 2000 é um sistema de gerenciamento de bancos de dados cliente/servidor de alto desempenho com alta integração com o *Windows* e com o *Framework*.NET. Por esse motivo foi o sistema gerenciador de banco de dados escolhido para compor a plataforma tecnológica. Conforme o volume de dados requeridos para armazenamento e acesso, o MSDE, versão gratuita do SQL *Server* pode ser utilizado, como alternativa mais barata.

4.3. ESTUDO E PROPOSIÇÃO CONCEITUAL DE UMA PLATAFORMA PARA EAD

4.3.1. Objetivos

A plataforma Tecnológica para EAD a ser desenvolvida será um *software* para execução na *web* e acesso a banco de dados, ou seja, uma “infra-estrutura” para facilitar o oferecimento e gerenciamento de atividades relacionadas à educação à distância. A plataforma auxiliará na criação de Escolas Virtuais, com vários componentes.

A Plataforma tecnológica oferecerá algumas funcionalidades, como por exemplo: os usuários não precisarão ter conhecimento da parte técnica (programação para *internet*, banco de dados) para realização de atividades. Assim, professores, administradores poderão concentrar seus esforços no desenvolvimento de suas atividades com maior rapidez e facilidade; diminuição ou eliminação de trabalhos manuais para implementações e montagens de cursos ou outras atividades como fóruns, *chats*, etc. A Plataforma tecnológica pretende ser uma ferramenta completa para educação a distância, presencial e semi-presencial.

A plataforma deverá oferecer ferramentas para montagem ou desenvolvimento de ambientes completos para EAD, agrupando as funcionalidades observadas nos *software* educacionais existentes, como visto na seção 2.7: sistema de autoria de curso, sala de aula virtual com ferramentas de apoio e de comunicação síncrona e assíncrona, sistema de gerenciamento de cursos, sistemas de avaliação.

4.3.2. O NucleoEAD x Plataforma para EAD

A concepção dessa plataforma foi desenvolvida a partir da análise da literatura existente sobre o assunto e a partir da análise e experiências com o NucleoEAD, escola virtual do CADTEC. A seguir mostra-se sucintamente um paralelo entre o NucleoEAD existente e a proposta conceitual da Plataforma para EAD. A Tabela 4.1 resume as diferenças entre os dois sistemas apontados.

O NúcleoEAD é a Escola Virtual do CADTEC, é um portal para *internet*, composto de uma estrutura organizada de sites, que disponibiliza recursos computacionais para a implementação de múltiplas atividades de ensino-aprendizado e objetiva incentivar a exploração de experimentos de ensino em busca de abordagens pedagógicas necessárias a esta modalidade de ensino. Com esta escola, conteúdos digitais foram desenvolvidos, servindo para testar as funcionalidades da estrutura do NucleoEAD existente e ainda permitem utilizar esse ambiente para a realização de experimentos pedagógicos com a aplicação de recursos multimídias e da computação gráfica como meio de facilitar o aprendizado. À medida que novas tecnologias são disponibilizadas, a concepção básica inicial do NucleoEAD vai sendo ampliada e a idéia maior é a incorporação destas tecnologias na criação de uma Plataforma Tecnológica para EAD.

A exemplo do NucleoEAD, que é uma escola virtual com fins específicos, o desenvolvimento da plataforma tecnológica, permitirá a construção de maneira mais rápida, segura e completa, de outras escolas virtuais semelhantes atendendo assim as necessidades de cada grupo de usuário.

No NucleoEAD, os cursos, conteúdos digitais e grupos de discussão são criados, inseridos e gerenciados manualmente. Com a plataforma, os Cursos, Conteúdos e Grupos de Discussão (GDs) poderão ser inseridos programaticamente, gerenciados através de uma interface própria, sem contato direto ou conhecimento sobre banco de dados ou linguagens de programação utilizados. Cursos serão montados sem a necessidade de conhecimentos de linguagens para *internet*.

A parte de Administração do NucleoEAD é restrita, algumas ações são realizadas manualmente. Na plataforma, a parte de administração será ampliada, resultando em um maior controle dos componentes da escola e dos alunos.

A Avaliação do NucleoEAD é apenas somativa. A proposta é que na plataforma tenha um módulo dedicado somente à avaliação, onde haverá meios para implementação de avaliações somativas e formativas, contando inclusive com tratamentos estatísticos de dados dos alunos.

A Linguagem de Programação do NucleoEAD é o ASP puro (clássico). Conforme proposta, na plataforma será utilizada o ASP.NET e o C# para programação, possibilitando a programação orientada a objetos, resultando na criação de um aplicativo modularizado, mais complexo e melhor.

O Banco de Dados do NucleoEAD é o *Access*, um arquivo de dados. A proposta é que na plataforma seja utilizado o Banco de Dados *SQL Server 2000*. Este banco de dados utiliza a tecnologia cliente-servidor, aumentando a rapidez de acessos e consultas e possibilitando armazenamento de maiores volumes de dados.

TABELA 4.1 – Comparação entre o NucleoE@D e a Plataforma para E@D proposta

	NucleoE@D	Plataforma
Criação de Escolas Virtuais	Escola Virtual do CADTEC	Permite a criação de várias escolas virtuais
Nível de Automatização	Ambientes são criados, inseridos e gerenciados manualmente	Ambientes são criados, inseridos e gerenciados programaticamente
Disponibilidade de ferramentas de gerenciamento e administração	Ferramentas para administração reduzidas, algumas ações são realizadas manualmente	Ferramenta completa para administração de todas ações e ambientes
Avaliação	Avaliação apenas Somativa	Módulo para avaliações
Linguagem de Programação	ASP	ASP.NET e C#
Banco de Dados	Access	SQL Server ou MSDE

4.3.3. Diretrizes da Plataforma Tecnológica para EAD

O passo inicial para desenvolvimento da plataforma tecnológica para EAD foi estabelecer diretrizes para a mesma, baseadas nos objetivos pretendidos e na análise de material sobre assunto encontrado na literatura. As diretrizes foram estabelecidas considerando-se dois aspectos:

- o que se espera da plataforma em relação às suas técnicas de desenvolvimento
- e o que se espera da plataforma em relação ao ambiente para EAD

Em relação às diretrizes técnicas e à integração da plataforma com o ambiente de EAD são esperadas as seguintes características:

- **Adaptabilidade:** A adaptabilidade é o conjunto de atributos que evidenciam a capacidade do *software* de se adaptar às necessidades e preferências do usuário e ao ambiente educacional selecionado. Esse requisito inclui atributos como:
 - **Customização:** disponibiliza recursos para facilitar a adaptação da interface para o uso de diferentes usuários (inteligência artificial na EaD);
 - **Adequação ao ambiente:** disponibiliza recursos para facilitar a adequação do *software* ao modelo, métodos e aos objetivos educacionais adotados;
 - **Adequação do código:** permite que o código possa ser adaptado conforme as diversas necessidades e/ou cursos e que dispense suporte para ser utilizados de novas maneiras. Poderá ser estabelecido um padrão mínimo de interface para que um professor sem conhecimento em HTML consiga oferecer um bom curso, mas os arquivos de interface serão deixados abertos para que professores com conhecimento em HTML possam personalizar o seu curso.
- **Documentação:** A documentação deve ser feita de forma a atender as necessidades dos usuários e dos mantenedores do *software* no momento de instalação, ou seja, deve atender a instalação e uso do *software*, de forma completa, consistente, legível e organizada.

- **Help online ou ajuda para usuário:** ajuda sob-demanda dentro do sistema. Conforme o que o aluno/professor estiver fazendo, vão sendo disponibilizadas dicas e ajudas rápidas. Isso também deve ser fornecido em páginas únicas de ajuda, com busca, índice das dúvidas, dentre outros.
- **Ajuda para administrador:** ajuda para o uso do sistema, instalação e manutenção da plataforma no servidor da escola, configuração de segurança, banco de dados, dentre outros.
- **Expansibilidade:** este atributo deve ser observado a fim de permitir que o código da plataforma possa ser expandido facilmente, tanto na fase de desenvolvimento quanto nos novos incrementos que cada escola utilizar. Para isso, o código deve ser bem estruturado, contendo uma hierarquia e com a utilização de comentários, para facilitar o entendimento do mesmo. Esses aspectos permitirão que alunos de iniciação científica, mestrado e doutorado possam adicionar recursos e incrementar a plataforma futuramente.
- **Modularidade/Escalabilidade:** O desenvolvimento da plataforma deverá ser realizado em módulos com o objetivo de dividir competências e especialidades, permitir o desenvolvimento simultâneo de módulos, além de facilitar uma posterior expansão do código.
- **Reutilização (Sob-demanda):** a plataforma poderá ser utilizada sob demanda para vários projetos. Serão criadas grandes quantidades de opções para interfaces, botões, figuras e o professor, conforme seu curso escolherá dentre as opções.
- **Coerência:** os módulos deverão ser desenvolvidos utilizando padrões de variáveis, formatação. Outra preocupação que deve ser observada é que os módulos deverão ser desenvolvidos observando a futura interligação/composição dos mesmos.
- **Suporte a cursos totalmente e parcialmente a distância:** a plataforma deverá abranger a opção de dar suporte a cursos totalmente e parcialmente a distância de forma que os professores de curso presenciais apenas complementem suas aulas com o fornecimento de arquivos e conversas a distância.

- **Desenvolvimento de ferramentas genéricas:** professores de diversas áreas do conhecimento poderão analisar e verificar viabilidade de uso destas ferramentas em seus cursos.

Em relação às diretrizes dos ambientes a serem desenvolvidos são esperadas as seguintes características:

- **Facilidade de uso e Interação:** o projeto deverá prever a facilidade de uso, tanto na criação dos cursos quanto na administração e acesso. Por facilidade, entende-se que o professor/aluno/monitor não tenha que conhecer de HTML e com isso ele consiga criar cursos com um mínimo de qualidade. Para criar cursos melhores, deverá requerer um conhecimento de ferramentas de criação de imagens e animações. A interface também deverá ser amigável, contando com um ambiente efetivo, sem panes e congestionamentos, que permita boa interação entre os participantes, contendo instrumentos motivadores e engajadores, para manter presa, tanto quanto possível, a atenção do aluno e que seja capaz de estimular alunos e professores a usar a plataforma, contribuindo para a auto-aprendizagem.
- **Disponibilidade de Ferramentas:** presença do maior número possível de ferramentas para EAD: programas de comunicação assíncrona (fórum, email, agenda, avisos, troca de arquivos) e síncronas (*Chat*, seminários, conferências), existência de ambientes para realização de seminários, conferências *online*, controle de atividades dos alunos e rastreamento das mesmas através de dados estatísticos, ou outro.
- **Informações:** ambiente que forneça informações relevantes tais como a descrição completa dos participantes, professores, cursos e demais atividades oferecidas.
- **Comunicação entre plataformas:** todas as instâncias da plataforma deverão comunicar entre si, possibilitando transferência de alunos entre as diversas escolas e maior interatividade. Permitindo que alunos busquem informações (textos, artigos) fora de sua escola de origem, ou seja, disponibilidade para pesquisa de materiais complementares e extras.

- **Avaliação:** implementação de módulo/componente/ferramenta eficaz de avaliação do progresso do aluno, possibilitando, inclusive, um *feedback* tanto para o professor quanto para o aluno. Avaliação formativa e somativa.

4.3.4. Módulos de Desenvolvimento da Plataforma Tecnológica para EAD

Visando o aspecto de modularidade da plataforma, o seu desenvolvimento foi dividido em módulos de acordo com a finalidade de cada parte da plataforma. Além de permitir definir etapas de trabalho, a modularização fornece mais agilidade no desenvolvimento da plataforma, e ainda viabiliza o desenvolvimento em equipe, o que é essencial em projetos dessa magnitude.

A plataforma poderá ser acessada em dois modos distintos, o modo administração, que será permitido apenas para professores, monitores e administradores, cada qual com níveis de permissão distintos, e o modo visitante, para alunos e demais usuários. No modo administração, poderão ser feitas as atividades administrativas como gerenciamento de cursos, gerenciamento de usuários, gerenciamento e composição das ferramentas de comunicação e apoio a EAD, criação de novos cursos, aprovação de matrículas, etc. O modo aluno será o modo padrão para apresentação da plataforma, onde estarão disponíveis cursos, as ferramentas para comunicação e demais ferramentas de apoio a EAD.

Divisão (Definição) dos Módulos de Desenvolvimento

1. O primeiro módulo é o módulo de Gerenciamento de Usuários e Permissões. Este módulo cuidará do cadastro de usuários na plataforma, matrícula de alunos, acesso de alunos, professores e monitores aos cursos e acesso dos usuários às demais ferramentas da plataforma.
2. O segundo módulo proposto para a plataforma é o módulo de Conteúdo Digital dos Cursos. Este módulo estará dividido em três partes:




- a. A primeira parte trata das interfaces e do conteúdo dos cursos, disponibilizando sistemas para montagem de interfaces HTML, inserção de dados e conteúdos.
 - b. Outra parte diz respeito aos processos avaliativos dos cursos, onde deverão ser desenvolvidas ferramentas que fornecerão a possibilidade de aplicação de avaliações somativas, formativas e de monitoramento do aluno em toda a plataforma.
 - c. A última parte disponibilizará ferramentas de apoio a EAD específicas para cada curso que poderão ser: área de transferência, FAQ, bibliotecas, etc.
3. Um terceiro módulo pensado foi um módulo de Ferramentas de apoio a EAD. Estas ferramentas poderão ser utilizadas pelos alunos independentemente dos cursos matriculados. As ferramentas foram divididas em três classes:
 - a. ferramentas gerais; alguns exemplos de ferramentas gerais são agendas, quadro de avisos, mural de recados, bloco de notas.
 - b. ferramentas de comunicação síncrona; exemplos de ferramentas para comunicação síncrona são *chat* e conferências.
 - c. ferramentas de comunicação assíncrona; exemplos de ferramentas para comunicação assíncrona são fóruns, troca de mensagens, *email*.
4. Um outro módulo necessário seria um módulo que desenvolvesse ferramentas para comunicação entre plataformas. Este módulo será responsável pela migração de alunos de uma escola virtual para outra, troca de conteúdos e outros materiais.
5. Um último módulo essencial a qualquer *software* é o módulo de Suporte e Ajuda aos usuários, incluindo ferramentas de ajuda em tempo real.

4.3.5. Diagramas UML Desenvolvidos

Partindo de uma concepção inicial, foram desenvolvidos diagramas UML relativos ao módulo de Gerenciamento de Usuários. Estes diagramas fazem parte da concepção inicial do módulo e são a base para o início da programação orientada a objetos. Nestes diagramas são representadas as classes e os relacionamentos entre as classes.

Para entendimento dos diagramas apresentados, é necessário que se conheça alguns símbolos e conceitos da UML, apresentados na TAB. 4.1.

TABELA 4.2 – Símbolos e Conceitos da UML

	Classe	Uma classe descreve um conjunto de objetos com atributos, estrutura, comportamento e relações similares.
	Composição	Uma composição é uma forma de agregação que indica que uma parte pode pertencer a somente um todo e que o tempo de vida do todo determina o tempo de vida da parte.
	Generalização	Generalização é uma relação entre um elemento específico e um elemento genérico, de forma que o elemento específico é totalmente consistente com o elemento genérico e inclui informações adicionais como atributos e associações.

As figuras 18 e 19 (FIG. 18 e FIG. 19) mostram os diagramas UML do módulo de Gerenciamento de Usuários.

A classe mais geral do módulo de Gerenciamento de Usuários é a **classe Usuarios::Usuario**, que apresenta os seguintes atributos: nome, email, endereço, telefone, celular, login, senha e um ID, que é um campo que identificará cada usuário.

Esta Classe **Usuarios::Usuarios** apresenta especificações dos grupos de usuários que serão permitidos na plataforma: **funcionários** e **alunos**, estas duas classes (**Usuarios::Aluno** e **Usuarios::Funcionário**) também são especificações da classe **Usuario::Permissoes**.

A classe **Usuario::Permissoes** será também derivada da classe usuário, mas não representa um tipo de usuário e servirá apenas para implementar controles dos acessos dos usuários.

A classe **Usuarios::Funcionario** apresentará especificações dos tipos de funcionários existentes na plataforma que poderão ser secretária, monitor e professor, cada qual compondo uma classe.

A classe **Usuarios::Aluno** apresenta especificações dos tipos de alunos existentes: alunos de graduação e alunos de pós-graduação. Os alunos de pós-graduação, por sua vez, se especificam entre alunos de mestrado e alunos de doutorado.

A classe **Usuarios::Aluno** apresenta um ou mais objetos da classe **Cursos::Nota**.

Os diagramas da FIG. 19 apresentam classes dos cursos e seus relacionamentos com os usuários.

Uma das classes observadas no diagrama é a classe principal **Cursos::Curso**. Esta classe é composta por um ou mais objetos da classe **Cursos::Disciplina**, e por um objeto da classe **Usuarios::Professor**.

A classe **Cursos::Disciplina** por sua vez, apresenta um ou mais objetos da classe **Cursos::Turma**, um ou mais objetos da classe **Usuarios::Monitor** e um objeto da classe **Usuarios::Professor**.

A classe **Cursos::Turma** apresenta um ou mais objetos da classe **Usuarios::Aluno**, um ou mais objetos da classe **Usuarios::Professor** e um objeto da classe **Cursos::SituacaoTurma**.

A classe **Cursos::SituacaoTurma** é responsável por representar o status de funcionamento da turma: se as pré-matrículas estão abertas, se as matrículas estão encerradas, se está em curso ou se a turma está encerrada.

Todos esses relacionamentos tentam representar os relacionamentos reais existentes entre cursos, disciplinas, turmas e usuários. A construção do diagrama UML é o primeiro passo para representação de sistemas reais em linguagens técnicas de

programação, ou seja, a UML fornece um grande auxílio na modelagem e concepção de sistemas complexos.

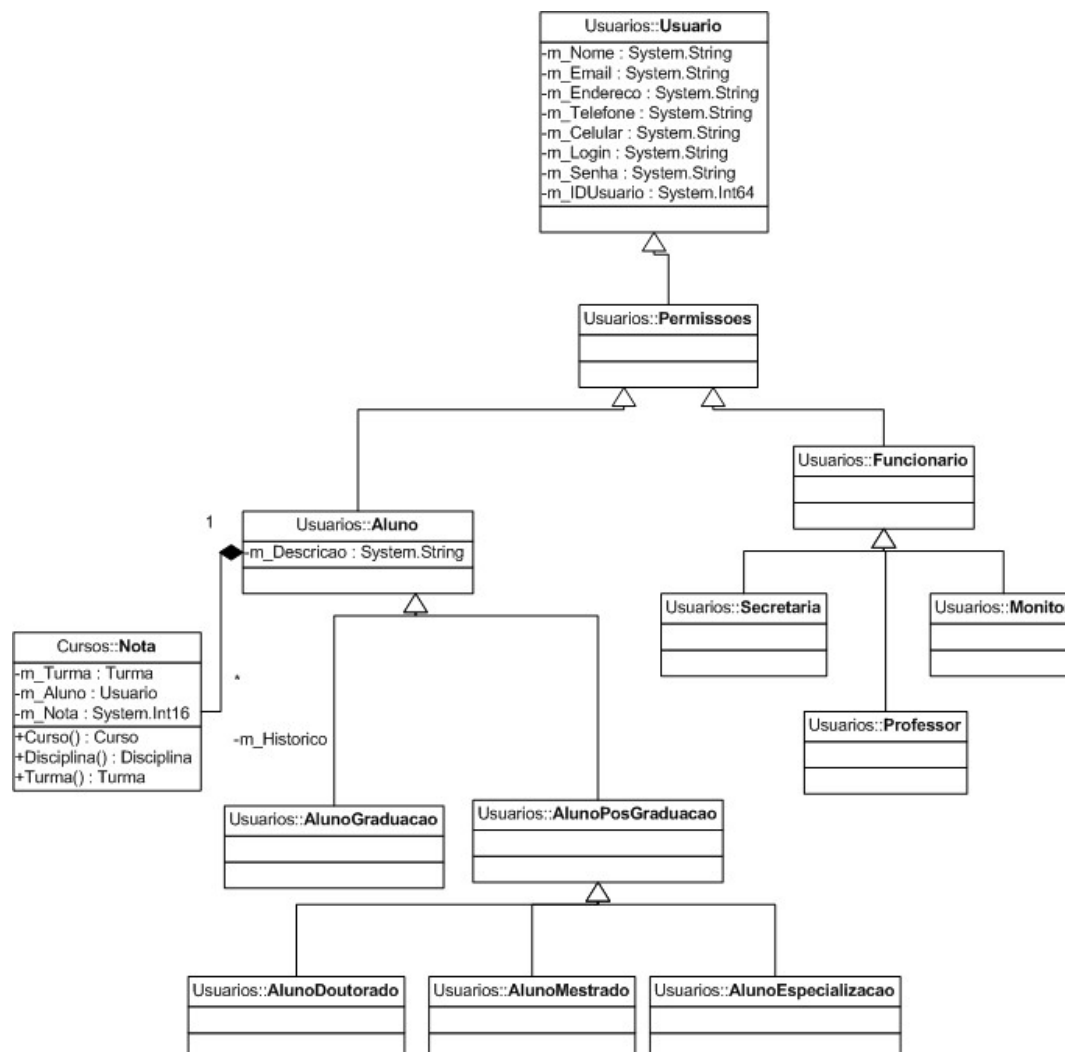


FIGURA 18 – Diagrama UML da Classe Usuários e suas especificações

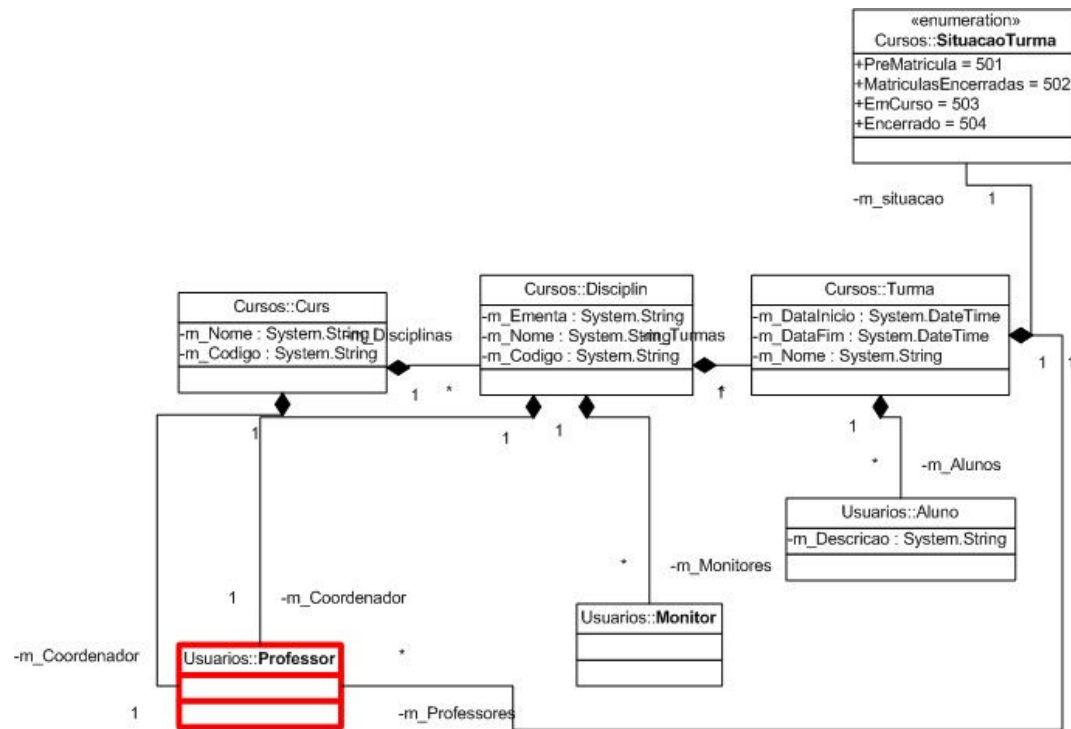


FIGURA 19 – Diagramas UML das Classes de Cursos e suas relações com os usuários

5

O CURSO DE INTRODUÇÃO AO MÉTODO DOS ELEMENTOS FINITOS – CONTEÚDO DIGITAL PARA *WEB*

5.1. TECNOLOGIAS *WEB* APLICADAS AO DESENVOLVIMENTO DE CONTEÚDOS DIGITAIS

Conteúdo Digital é a informação apresentada na forma digitalizada, através de sistemas informáticos digitais, organizada para transmitir conhecimentos, em níveis de profundidade específicos, sobre determinado tema, como mostra TORRES e MAZZONI (2004).

Segundo SANTOS², as possibilidades de utilização dos recursos multimídia trazem grandes contribuições para a produção de conteúdos didáticos digitais, permitindo respeito ao estilo de aprendizagem dos usuários e às suas preferências quanto à forma de recepção da informação. A multimídia deve ser usada como recurso didático e de redundância, cuidando-se sempre, porém, para evitar que o usuário esteja exposto a sobrecargas sensoriais.

² Notas de aula de SANTOS, N. Textos: Internet e Web e Impactos da Internet na Educação, 2000.

Recursos multimídia criam ambientes estimulantes e despertam o caráter ativo do estudante que precisa decidir o que, quando e como estudar. Através dos recursos multimídias disponíveis, cada usuário escolhe a maneira de aprendizado mais adequada ao seu perfil, desenvolvendo métodos mais criativos e flexíveis para a resolução de problemas. O aproveitamento do tempo é outra vantagem da utilização desses recursos, já que os estudantes podem selecionar o que estudar de acordo com suas necessidades (CHAVES, 1991).

A utilização de recursos multimídias sugere ainda a utilização de sentidos, sentidos e meios de comunicação, que não eram utilizados nos processos de educação até pouco tempo atrás. Mais uma vez, a possibilidade de utilização de outros sentidos pelo aluno, especialmente os sentidos visual e auditivo, representa um estímulo a mais para a facilitação do aprendizado e é uma forma de adaptar o conteúdo às preferências de cada aluno.

A estrutura dinâmica do ambiente *web* associada às tecnologias recentemente desenvolvidas traz novas possibilidades aos processos de ensino/aprendizagem via *web*, disponibilizando textos impressos e outros materiais didáticos com mais dinamismo. O Hipertexto surge como uma poderosa ferramenta nesse processo através da integração de diferentes tipos de linguagens como som, vídeo, imagem e texto.

“Hipertextos são sistemas computadorizados, que permitem a criação/utilização de documentos organizados por conjuntos de textos cujos conteúdos se desenvolvem a partir de determinado tópico permitindo a formação de uma rede de informação relacionadas a temática inicial e estruturada de forma não seqüencial.” (STRUCHINER et al, 1997)

Assim, os alunos podem selecionar o que vão estudar e seguir os *links* adequados aos seus objetivos de estudo.

De uma maneira mais simplificada, hipertexto pode ser definido como um texto organizado em forma de rede de itens ou módulos de informação, chamados de nós, interligados entre si, através de *links*, permitindo ao usuário navegar, seguindo sua própria seqüência de estudo.

Hipertextos e hipermídias aparecem no cenário da EAD como poderosas ferramenta de auxílio à educação.

Os sistemas hipertextos, que incorporam outros tipos de dados, tais como som, vídeo, áudio, código fonte e outros além de textos e gráficos, são os denominados sistemas hipermídia. Em outras palavras, sistemas hipermídias são sistemas de base de dados que fornecem um método não seqüencial de acesso a informações, que podem estar nos formatos texto, foto, vídeo, animações, locuções, dentre outros. Os sistemas hipermídias podem possuir complexidades variadas, disponibilizando diversas maneiras de apresentação dos nós e ligações, que indicarão ao usuário os caminhos de navegação fixos ou complexos.

“A construção de ambientes virtuais não é um processo trivial de organização de material didático no novo formato (...). É necessário ter a visão do aluno, organizar o processo de suporte e tutoria e manter aberto o canal de comunicação entre os instrutores e o estudante. Além disso, a construção de um ambiente virtual deve levar em consideração inúmeros aspectos técnicos.”
(DOVICCHI, 2003, p. 5,6)

Outras características essenciais de conteúdo digitais, como descrito por TORRES e MAZZONI (2004), são a usabilidade e a acessibilidade. Acessibilidade é uma característica que assegura que o conteúdo possa ser acessado por todos os seus usuários. Usabilidade determina o grau de facilidade de utilização da aplicação para um usuário que ainda não esteja familiarizado com a mesma.

O curso de elementos finitos foi desenvolvido de maneira a considerar todas as características observadas acima e de maneira a explorar a utilização de recursos multimídias como facilitador do aprendizado através do estímulo dos sentidos. Um dos principais recursos multimídia explorado nesse curso foi a utilização de animações.

5.2. OBJETIVOS

O Curso de Introdução ao Método dos Elementos Finitos destina-se a profissionais na área de engenharia de estruturas que desejam revisar ou iniciar seus conhecimentos

sobre o Método dos Elementos Finitos. Como pré-requisitos desse curso, é necessário que se tenha conhecimentos básicos de Resistência dos Materiais e Análise Estrutural.

O curso fará uma apresentação e desenvolvimento da formulação clássica para alguns elementos de barra: mola e treliça.

Outro objetivo secundário que norteou o desenvolvimento do curso foi a exploração de tecnologias para EAD aplicadas ao ensino de engenharia de estruturas. As tecnologias utilizadas foram os hipertextos, sistemas hipermídia simplificados além de animações que garantem uma apresentação do conteúdo de forma mais dinâmica e intuitiva.

Por último, pretende-se, com a produção deste curso introdutório do método de Elementos Finitos, incentivar a ampliação do número de conteúdos digitais na área de engenharia de estruturas para auxiliar os alunos nas disciplinas presenciais, servindo de estímulo e representando uma fonte adicional de busca do conhecimento.

5.3. ESTRUTURAÇÃO DE *SITES*

5.3.1. Interface

A Interface do curso de Introdução ao Método dos Elementos Finitos foi desenvolvida em conjunto com o laboratório LAGEAR da Escola de Arquitetura da UFMG.

A interface proposta tem o objetivo de ser bastante amigável, intuitiva, clara e motivadora, contribuindo para o processo de aprendizagem do aluno, propiciando uma navegação fácil, sem que este perca tempo com dúvidas sobre a utilização do sistema ou com elementos que possam confundi-lo ou distraí-lo.






O *site* do curso apresenta uma série de ferramentas que estimulam a troca de experiências e comunicações entre alunos e professores. O acesso a estas ferramentas é realizado através de um menu localizado na parte superior da página e de ícones no rodapé como mostra a FIG. 23.









A apresentação do conteúdo digital do curso é realizada através de textos acrescidos de *links*, animações, figuras, seguindo os conceitos de hipertexto e hipermedia. O texto e as multimídias são apresentados de forma integrada, para que o aluno possa visualizar todos os recursos simultaneamente, ganhando agilidade, diminuindo a inserção de fatores que possam causar distração e facilitando que o estudante recorra ao texto em caso de dúvidas. Entretanto, o aluno também tem a opção de visualização do texto e multimídias de forma separada, através da abertura de janelas menores para apresentação das imagens e animações. Essas janelas se abrem sobre a janela principal, quando solicitadas através de clique sobre os ícones a elas relacionados. Em resumo, a apresentação do conteúdo digital do curso foi desenvolvida buscando obedecer aos conceitos anteriormente apresentados, respeitando as preferências e estilos de aprendizado de cada estudante. Durante a exposição do conteúdo digital, houve ainda, a preocupação de deixar os títulos do item e unidade sempre visíveis ao estudante, para que este não se perca no sistema hipermedia proposto.

Durante a exibição do conteúdo digital e nas demais páginas do curso de Introdução ao Método dos Elementos Finitos, foram utilizados ícones e botões com desenhos bastante intuitivos e que disponibilizam uma legenda quando o *mouse* permanece por cima do mesmo.

Alguns ícones utilizados no curso podem ser observados na TAB. 5.1.

TABELA 5.1 – Ícones Utilizados no Curso de Introdução ao MEF

	Ícone Recados – Este ícone abre a ferramenta de recados.
	Ícone Bloco de Notas – Este ícone abre a ferramenta bloco de Notas.
	Este ícone é um link para o aplicativo do Método dos elementos Finitos. Está disponível no rodapé.
	Este ícone aparece na área de conteúdos. Ao clicar sobre este ícone é aberta uma janela da figura ou animação referente.
	Estes ícones aparecem na área de conteúdos e são controles de navegação

	para próxima página e página anterior
	Início – este ícone está presente no menu principal do curso e retorna para a página inicial do mesmo FIG. 20
	Conteúdo – este ícone está presente no menu principal do curso e apresenta um link para o índice do conteúdo digital do curso. FIG 21
	Agenda – este ícone está presente no menu principal do curso e apresenta um link para a ferramenta agenda. FIG 22
	Área de Transferência – este ícone está presente no menu principal do curso e apresenta um link para a página da área de transferência. FIG 23
	Grupo de Discussão – este ícone está presente no menu principal do curso e apresenta um link para o grupo de discussão do curso.
	Busca – este ícone está presente no menu principal do curso e apresenta um link para a ferramenta busca. FIG 24
	FAQ – este ícone está presente no menu principal do curso e apresenta um link para a página FAQ (lista de perguntas mais frequentes). FIG 25
	Sair – este ícone está presente no menu principal do curso. Este botão faz <i>logout</i> do aluno do sistema do NucleoEAD.

5.3.2. Navegação

O curso de Introdução ao Método dos Elementos Finitos será veiculado pela *web*, estando disponível através do site: www.cadtec.dees.ufmg.br/nucleoead. Este “*site*” hospeda a Escola Virtual (NucleoEAD) do CADTEC. Para ter acesso ao curso, os alunos interessados deverão se cadastrar no NucleoEAD e se matricular no curso.

Após matrícula, o aluno terá disponível o acesso ao curso de Introdução ao Método dos Elementos Finitos e às ferramentas de apoio desenvolvidas. A figura 20 ilustra esse processo.

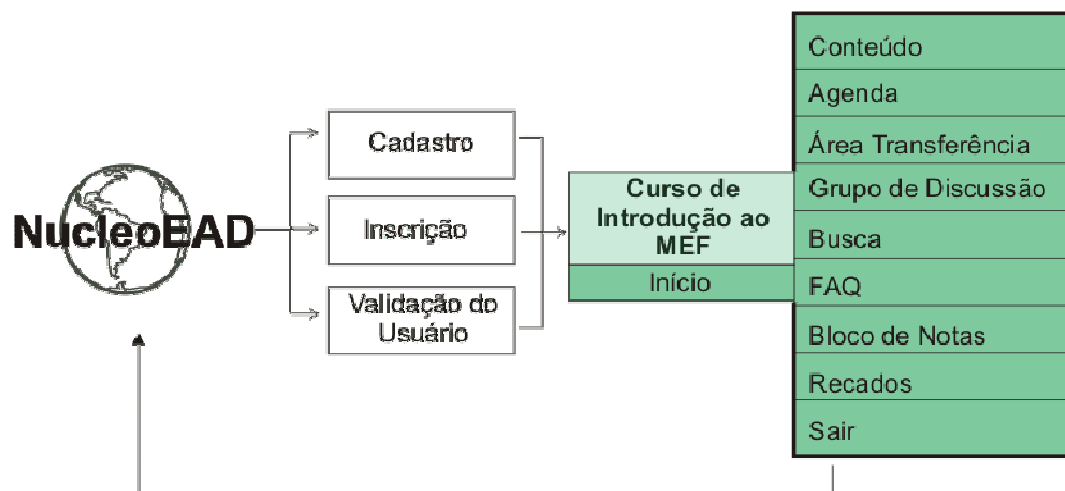


FIGURA 20 – Hierarquia Global de Navegação para acesso ao curso de Introdução ao MEF

O curso foi montado e estruturado a partir de *frames*. *Frame* é um recurso da linguagem HTML que divide a página em quadros e dentro de cada quadro podem ser chamadas diferentes páginas. As páginas do curso foram então, divididas em 3 *frames*, FIG. 21: um *frame* superior para o menu, um *frame* inferior para o cabeçalho e um *frame* principal para exibição do conteúdo principal da página.

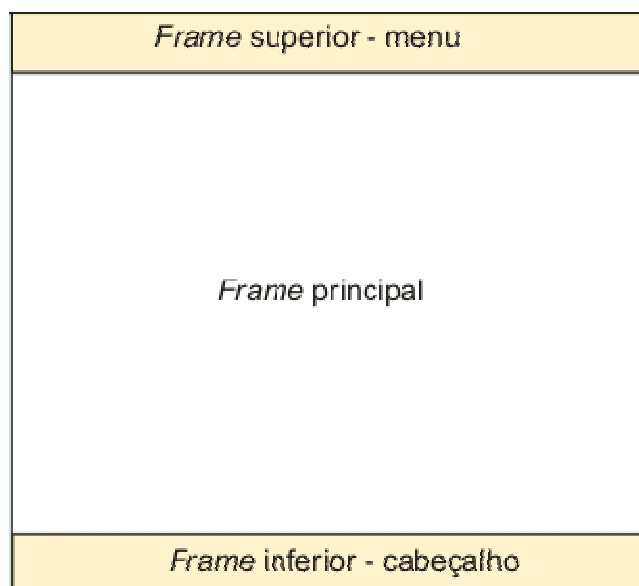


FIGURA 21 – Estruturação das páginas do curso de Introdução ao MEF

O *Frame* superior foi destinado ao *Menu* do curso, onde aparecem os botões para navegação pelas demais páginas: página inicial, conteúdo, agenda, área de transferência, grupo de discussão, busca e FAQ.

O *Frame* inferior foi destinado ao rodapé. No rodapé estão presentes *links* para duas ferramentas de apoio ao curso: bloco de recados e bloco de notas, além do link para o aplicativo do Método dos Elementos Finitos. O rodapé tem a função principal de orientar o usuário sobre sua navegação no curso. Nas páginas do *site* destinadas à exibição do conteúdo, haverá a indicação do módulo e do item pelo qual o aluno está navegando. A figura 22 apresenta o rodapé com a indicação de navegação. Neste exemplo, o aluno está no 2º item do módulo 1. Esta ferramenta foi desenvolvida utilizando *JavaScript*.



FIGURA 22 – Rodapé do curso de Introdução ao MEF

O *Frame* principal é destinado à exibição dos conteúdos principais do curso.

Ao acessar o curso de Introdução ao Método dos Elementos Finitos, a primeira página vista é a página inicial do curso, FIG. 23, onde há uma breve apresentação do curso e do trabalho desenvolvido.

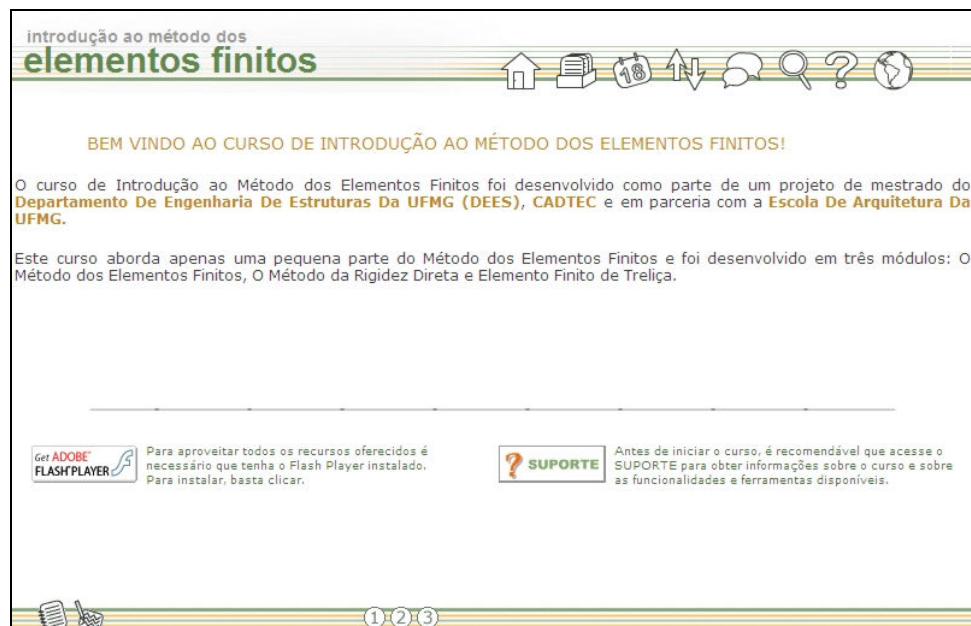


FIGURA 23 – Página Inicial do curso

As demais páginas do curso podem ser acessadas através do *Menu*.

Conteúdo

A primeira página da sessão de conteúdo é uma página contendo o índice de todo o conteúdo abordado pelo curso, FIG. 24. O índice do conteúdo pode ser visualizado de duas formas, de acordo com a preferência do usuário. A página padrão apresenta os módulos desenvolvidos e os itens de cada módulo, que podem ser expandidos quando o mouse é sobreposto, explicando sucintamente o que será abordado em cada item. A segunda forma de visualização é uma página estática com os *links* para cada item dos módulos.

Nesta área do site estará disponível todo o conteúdo do curso: textos, exercícios, animações, ou seja, o material didático propriamente dito do curso.

Nas páginas onde é apresentado o conteúdo do curso, há um espaço reservado para textos e outro destinado a animações, figuras, fotos que são facilmente relacionadas ao texto ao qual se referem e podem ser visualizadas pelo aluno junto ao texto, como pode ser observado na FIG. 29.

A apresentação do conteúdo foi feita seguindo módulos, como será explicado no item seguinte.

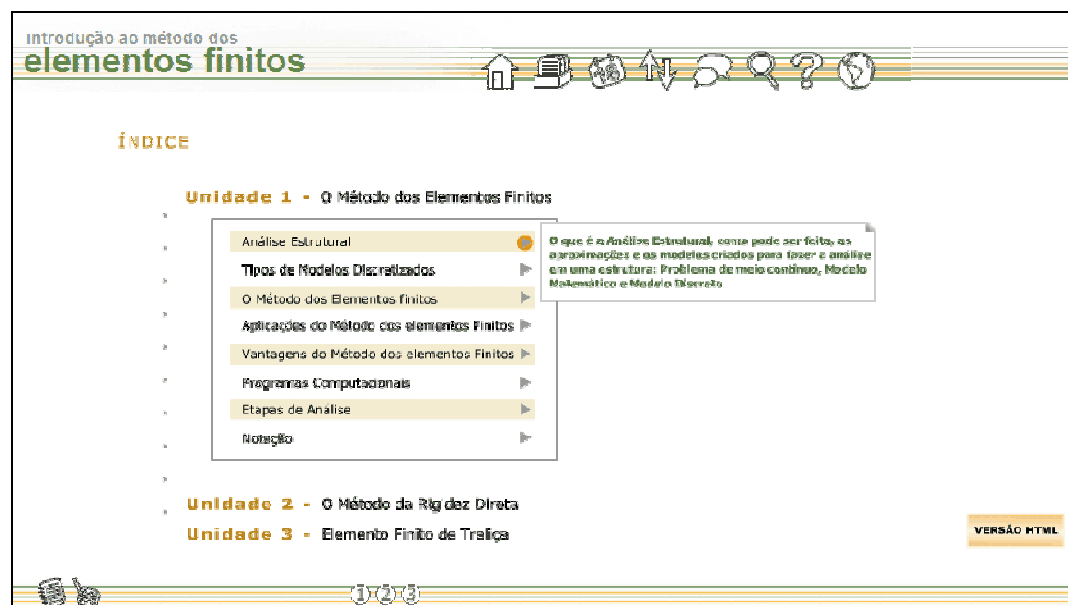


FIGURA 24 – Índice do Conteúdo Digital do Curso

Agenda

Esta é uma nova ferramenta desenvolvida, trata-se de uma agenda *online*, onde os alunos podem programar seus estudos e outras atividades referentes ao curso, através da utilização de um calendário *online* como pode ser observado na FIG. 25.

Introdução ao método dos
elementos finitos

AGENDA - Data: 21/03/08

março de 2008						
dom	seg	ter	qua	qui	sex	sáb
24	25	26	27	28	29	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Adicionar Remover Salvar

Hora:

Formato para entrada da hora: 08:00 - 22:00

FIGURA 25 – Página da Ferramenta Agenda

Área de Transferência

Esta área foi destinada para o professor disponibilizar trabalhos, textos, exercícios complementares ou qualquer material para os alunos e vice-versa, FIG. 26. A versão para impressão do conteúdo digital desse curso estará disponível nessa área.

Grupo de Discussão

Através do botão Grupo de Discussão, o aluno terá acesso direto ao grupo de discussão do curso, onde todos os participantes poderão debater de forma pública assuntos relacionados ao curso. O grupo de discussão a ser utilizado pelo curso estará hospedado no ambiente de grupo de discussão do NucleoEAD.

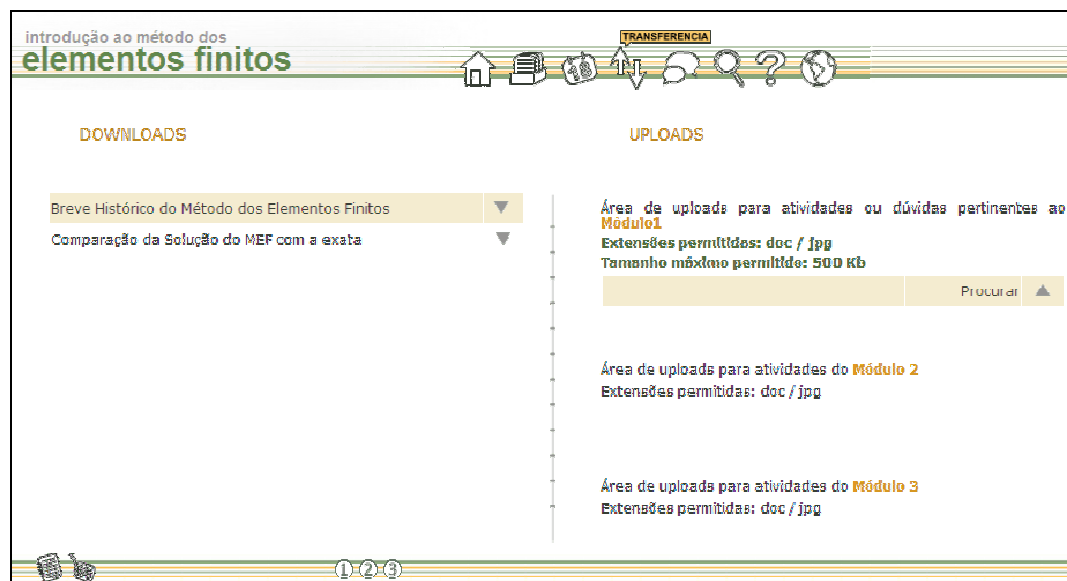


FIGURA 26 – Página da Área de Transferência

Busca

Nesta página estão disponíveis ferramentas para se fazer buscas específicas no material disponibilizado pelo curso. Essa busca pode ser feita através de todas as palavras digitadas pelo aluno no campo específico, por cada palavra ou pela frase digitada, FIG. 27.

Perguntas Frequentes (FAQ)

Nesta página, FIG. 28, estarão as perguntas feitas frequentemente e suas respectivas respostas, relacionadas ao assunto do curso de Introdução ao Método dos Elementos Finitos. Esta página será realmente implementada após o funcionamento e distribuição do curso. Algumas perguntas referentes ao funcionamento da interface também estão disponíveis nessa área, visando facilitar a navegação do aluno e suprir dúvidas que surjam.

NucleoEAD (Sair)

Este botão volta para a página inicial do NucleoEAD, efetuando *logout* do aluno do ambiente do curso na escola virtual, encerrando a sessão do usuário no curso.

Ferramentas Bloco de Notas e Recados

As ferramentas bloco de notas e recados, acessadas através do rodapé, são ferramentas adicionais de apoio à aprendizagem do aluno e serão apresentadas no próximo capítulo. Estas ferramentas não são exibidas no *frame* principal da página, mas sim através de *pop-ups* que abrem sobre a janela principal. Decidiu-se por essa forma de exibição já que estas são ferramentas de apoio ao aluno, possibilitando o seu acesso à medida que conteúdos são estudados.

A ferramenta bloco de notas é para que o aluno faça anotações durante o estudo do conteúdo do curso. Estas anotações podem ser organizadas por data e buscadas pelo aluno a qualquer momento.

A ferramenta página de recados é para que o professor ou mesmo o aluno deixe lembretes ou avisos. O aluno verá esses avisos nas datas agendadas.



FIGURA 27 – Página da Ferramenta de Busca



FIGURA 28 – Página FAQ

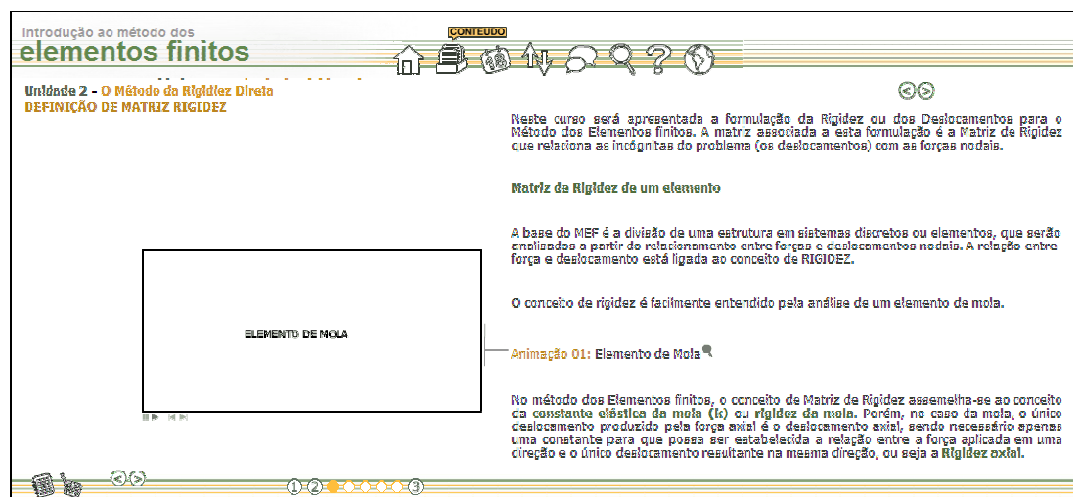


FIGURA 29 – Modelo das Páginas do Conteúdo Digital

5.4. O CONTEÚDO DIGITAL

5.4.1. Definição do Escopo do Programa do Curso

A proposta inicial do programa para um curso de Introdução do Método dos Elementos Finitos era de definir o escopo do conteúdo digital, que abrangesse a formulação

clássica do método, englobando os elementos de barra (treliça e viga), pórticos e grelhas.

A partir dessa idéia, o escopo do programa para o curso foi sendo montado. Para a montagem do programa e posterior desenvolvimento do material didático, foram utilizadas anotações de sala de aula³, dois livros didáticos e uma apostila. Os dois livros didáticos utilizados foram: *A First Course in the Finite Element Method* – Daryl L. Logan LOGAN (2002) e *Elementos Finitos A base da Tecnologia CAE* – Avelino Alves Filho FILHO (2000).

O escopo do programa completo proposto foi o seguinte:

Unidade 1: O Método dos Elementos Finitos

- Análise estrutural;
- Tipos de Modelos Discretizados;
- O método dos elementos finitos.
- Pequeno histórico do método (material complementar);
- Aplicações do MEF.
- Vantagens do MEF.
- Programas computacionais para o MEF.
- Etapas de análise do MEF
- Introdução das condições de contorno;
- Notação

³ Anotações em sala de aula das disciplinas: Introdução ao Método dos Elementos Finitos na graduação e Método dos Elementos Finitos no curso de mestrado

Unidade 2: O Método da Rigidez Direta

- Definição da Matriz de Rigidez.
- Obtenção da matriz de rigidez para um elemento de mola.
- O método da rigidez direta (exemplo com um conjunto de molas).

Unidade 3: Elemento Finito de Treliça

- Apresentação do modelo seguindo as etapas recomendadas para análise;
- Aplicação do método da rigidez direta.
- Estabelecimento das condições de contorno.
- Transformação de vetores em duas dimensões.
- Matriz de rigidez global.
- Solução de treliça plana: tensão, deformação, deslocamento.
- Comparação da solução do MEF com a solução exata.
- Equações de equilíbrio através do PTV.
- Condições de convergência.
- Uso da simetria em uma estrutura.
- Apoios inclinados.
- Exemplos.

Unidade 4: Elemento Finito de Viga

- Apresentação do modelo seguindo as etapas recomendadas para análise;
- Exemplo de análise utilizando e não utilizando o método da rigidez direta.

- Utilização do PTV para obtenção da matriz de rigidez e do vetor de carregamento nodal equivalente.
- Carregamento distribuído.
- Solução.
- Comparação do método de elementos finitos com a solução exata.
- Exemplos.

Unidade 5: Elemento Finito de Pórtico Plano

- Elemento de viga 2D orientado espacialmente e com componentes axiais.
- Obtenção da matriz de rigidez local
- Obtenção da matriz de rigidez global.
- Solução
- Apoios inclinados
- Exemplos
- Pórtico espacial (3D) (material complementar)

Unidade 6: Elemento Finito de Grelha

- Matriz de Rigidez de grelha
- Exemplos

Unidade 7: Outras Considerações

- Análise de subestruturas (subdivisão das estruturas para facilitar a análise);
- Continuidade dos elementos;

- Critérios de convergência e continuidade.

Considerando a complexidade dos desenvolvimentos e a extensão do projeto global desta dissertação, optou-se por limitar abrangência do conteúdo do curso de Introdução do Método dos Elementos Finitos para fins deste projeto. Sendo assim, ficou definido, que o conteúdo digital seria desenvolvido até a Unidade 3 - Elemento Finito de Treliça, do programa completo proposto. O restante do programa será proposto como sugestão para trabalhos futuros.

5.4.2. O Conteúdo

O conteúdo digital foi desenvolvido utilizando os conceitos de hipertexto e utilizando-se recursos multimídias e da computação gráfica para estimular o sentido visual dos estudantes como forma de facilitar e melhorar o aprendizado.

O conteúdo foi organizado em módulos, ou pequenos pacotes, identificando os nós de informação, como apresentado no escopo. A seguir estes módulos e seus itens serão descritos, enfocando o que se pretende transmitir ao aluno.

Unidade 1: O Método dos Elementos Finitos

O primeiro item dessa unidade aborda os fundamentos de Análise Estrutural. Este item apresenta a conceituação da análise estrutural, como pode ser realizada, as aproximações e simplificações usualmente adotadas e os modelos utilizados para análise: modelos matemáticos e modelos discretos. Estes modelos são obtidos após uma análise do problema real, ou problema de meio contínuo.

O segundo item discorre sobre o Método dos Elementos Finitos, em quais situações este método deve ser utilizado para análise, as formulações desenvolvidas para análise e o que será abordado nesse curso.

O item seguinte mostra algumas aplicações práticas do MEF. Posteriormente são apresentadas algumas vantagens desse método e alguns programas computacionais que utilizam o MEF como ferramenta de análise.

O penúltimo item descreve as etapas de análise comumente utilizadas para desenvolvimentos de formulações utilizando este método. Estas etapas são: discretização e escolha dos tipos de elementos, obtenção da função de deslocamento, definição das relações de deformação-deslocamento e tensão-deformação, obtenção das equações de equilíbrio e matriz de rigidez dos elementos; reunião das equações dos elementos para obtenção das equações globais; introdução das condições de contorno; solução das deformações e tensões para os elementos.

O último item da unidade descreve a notação utilizada durante o curso para as variáveis globais e locais.

Unidade 2: O Método da Rigidez Direta

Esta unidade é composta de quatro itens. O primeiro item define e conceitua fisicamente a matriz de rigidez. Esta definição é realizada utilizando como instrumento de comparação elementos de mola e seus coeficientes de rigidez.

No segundo item, são apresentados processos para a obtenção da matriz de rigidez para elementos de mola.

O terceiro item explica o método da rigidez direta, método utilizado para montagem da matriz de rigidez de um conjunto de elementos. Este método é apresentado utilizando como exemplo um conjunto de molas.

O quarto item define elementos reticulares e explica como o MEF pode ser utilizado para análise desses elementos.

Unidade 3: Elemento Finito de Treliça

Esta unidade desenvolve as equações utilizadas pelo MEF para análise de elementos de treliça. A utilização dos conceitos apresentados nas unidades anteriores é fundamental para o aprendizado dessa unidade.

O primeiro item apresenta o modelo discreto de um elemento de treliça seguindo as etapas recomendadas para análise, conforme unidade 1, a saber: observação do

problema de meio contínuo, representação em um modelo matemático, a definição do modelo discreto e obtenção da matriz de rigidez em coordenadas locais.

O item subsequente é Aplicação do Método da Rigidez Direta, conforme explicado na unidade 2.

Os próximos itens ensinam como se faz a transformação de vetores em duas dimensões, para a obtenção da matriz de rigidez global de uma estrutura, quando o sistema é composto por mais de um elemento que não apresentam o mesmo direcionamento de eixos.

O próximo item mostra como devem ser estabelecidas as condições de contorno de um sistema.

Seguindo uma etapa lógica de desenvolvimento, a solução de uma treliça plana é apresentada, determinando-se a tensão, deformação e deslocamento em pontos dos elementos.

O item seguinte apresenta uma comparação de soluções obtidas pelo MEF e pela aplicação de soluções exatas dos modelos matemáticos. Este item desperta no aluno uma grande capacidade de análise crítica a respeito dos resultados obtidos pelo método apresentado.

Outra maneira muito utilizada para determinação das equações de equilíbrio e da matriz de rigidez dos elementos é pela aplicação do Princípio dos Trabalhos Virtuais (PTV), método apresentado neste item e que fornece uma generalização que pode ser aplicada em outros tipos de elementos reticulados tais como de vigas e de pórticos.

Os tópicos finais do capítulo são: descrição de condições de convergência, uso da simetria em uma estrutura e apoios inclinados.

A versão impressa do conteúdo digital do curso está no anexo A deste projeto, juntamente com um CD contendo as animações.

5.5. PAPEL DA COMPUTAÇÃO GRÁFICA

A computação gráfica supre uma necessidade humana de visualização de dados. É a área da computação destinada ao estudo de imagens de uma maneira geral, para representar dados ou informações ou para representar situações do mundo real. A computação gráfica na sua concepção mais abrangente tem um grande apelo na sua aplicação à EAD, por viabilizar a criação de ambientes com interfaces cada vez mais elaboradas, além de utilizar de recursos multimídias como animações e jogos para melhorar o aprendizado.

Animações podem ser consideradas produtos da computação gráfica, já que consistem de formas visuais para repassar informações trabalhadas de maneira intuitiva e agradável, atraindo e mantendo assim, a atenção das pessoas no objeto do aprendizado.

O principal recurso multimídia utilizado pelo curso foi a animação. As animações deixam o conteúdo digital mais dinâmico, servindo de estímulo aos alunos, pois representam um reforço sensorial dos conteúdos apresentados em formato textual.

As animações foram produzidas utilizando o *Flash, software da Macromedia*.

5.6. BASES TECNOLÓGICAS E CONFIGURAÇÕES DO SISTEMA

5.6.1. Bases Tecnológicas

Para desenvolvimento do curso de Introdução de Elementos Finitos, foram utilizadas as linguagens padrão para internet: HTML e CSS. O CSS associado ao HTML ofereceu bastante flexibilidade às páginas do curso, além de deixá-las mais leves.

A linguagem *JavaScript* foi utilizada como linguagem de apoio, dando mais interatividade às páginas do curso e permitindo algumas definições de *layout*, como o título dos itens sempre visíveis, a abertura de *pop-ups*, as indicações de navegação no rodapé, dentre outros.

Para o desenvolvimento das ferramentas de apoio a EAD, como a agenda, bloco de notas e recados, que serão vistos no próximo capítulo, foi utilizado o ASP.NET 2.0, com a linguagem C#. O ASP.NET ofereceu todas as facilidades de programação já apresentadas neste trabalho.

O Banco de dados utilizado pelo curso é o *Access*, seguindo o padrão do NucleoEAD.

Os *software* utilizados no desenvolvimento do curso foram o *Flash*, para produção de animações, o *DreamWeaver*, para composição das páginas HTML, arquivos de *Scripts* e arquivos CSS, o *Corel Draw* para produção de figuras, o *Access* para confecção do banco de dados do curso e o *Visual Studio.NET* para desenvolvimento das páginas “.aspx”.

5.6.2. Configurações do Sistema

Como curso será veiculado pela *web*, o recurso computacional exigido para os usuários do curso é um microcomputador com acesso a *Internet*, com qualquer sistema operacional. É necessário que o usuário tenha ainda o *Flash Player* instalado em seu computador. Esta ferramenta é gratuita e pode ser feito o *download* da mesma através do site: <http://www.macromedia.com/software/shockwaveplayer>. O *Flash player* é uma ferramenta da Macromedia responsável pela execução das animações desenvolvidas em Flash. É recomendada ainda, a utilização de um navegador mais atualizado, já que foram utilizados *scripts* no desenvolvimento do *site*.

6

FERRAMENTAS DESENVOLVIDAS PARA APOIO A EAD

Ferramentas de apoio a EAD são ferramentas que auxiliam de forma direta ou indireta o processo de aprendizagem, tentando suprir em parte, a falta de contato pessoal entre alunos e professores. Essas ferramentas oferecem estímulo e facilidades para os estudantes e professores realizarem suas atividades.

As ferramentas de apoio a EAD desenvolvidas como parte deste projeto e que representam ferramentas novas no contexto da Escola Virtual (NucleoEAD) atual são: bloco de anotações, agenda e recados.

A idéia para desenvolvimento dessas ferramentas surgiu da observação dos recursos existentes nos *sites* educacionais pesquisados e por achar necessário que o estudante, durante o estudo de um texto, faça anotações a respeito do tema para posterior acesso e revisão.

As ferramentas agenda e recados foram desenvolvidas com o objetivo de auxiliar os alunos a se organizarem, deixando para o sistema a função de lembrar eventos, enquanto alunos e professores se concentram em aprender e ensinar.

Cada ferramenta será descrita a seguir.

6.1. BLOCO DE NOTAS

A ferramenta bloco de notas permite que o aluno faça anotações durante o estudo do conteúdo do curso. Estas anotações são armazenadas e organizadas por data e assunto para cada usuário e podem ser acessadas pelo estudante a qualquer momento, bastando que este selecione a anotação desejada na lista de anotações existentes. As anotações de cada usuário são armazenadas em um banco de dados *Access*.

O bloco de notas é acessado através do clique do *mouse* no ícone localizado no rodapé referente a ele. Esta ferramenta está disponível através de um *pop-up* que abre sobre a janela principal do curso. A figura 30 mostra o *pop-up* da ferramenta bloco de notas e a figura 31 mostra como é realizado o acesso às anotações existentes. A listagem das anotações existentes é acessada através do clique no botão “ver todas” da ferramenta bloco de notas.

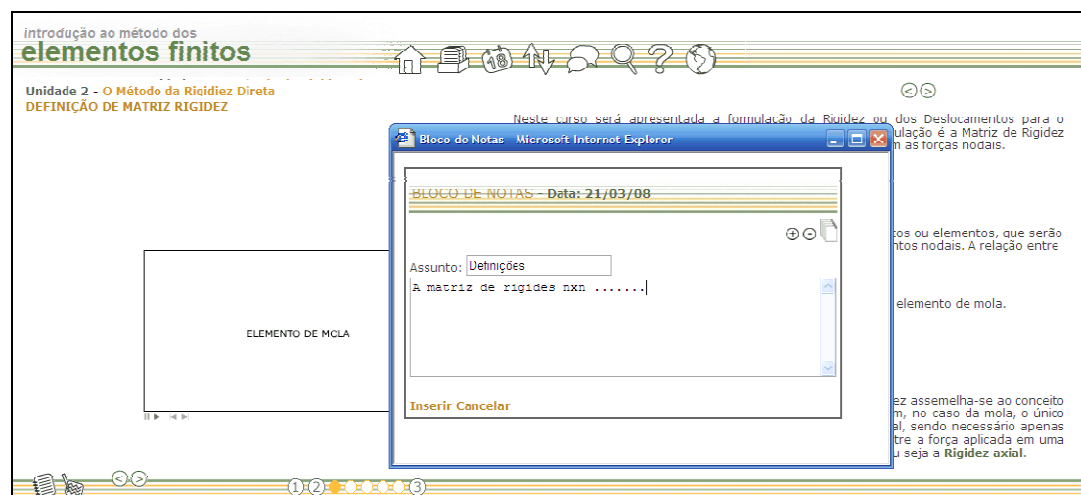


FIGURA 30 – Ferramenta Bloco de Notas

A ferramenta bloco de notas é disponível ao usuário através de um *pop-up* para permitir sua utilização enquanto é realizada leitura ou estudo do conteúdo digital.

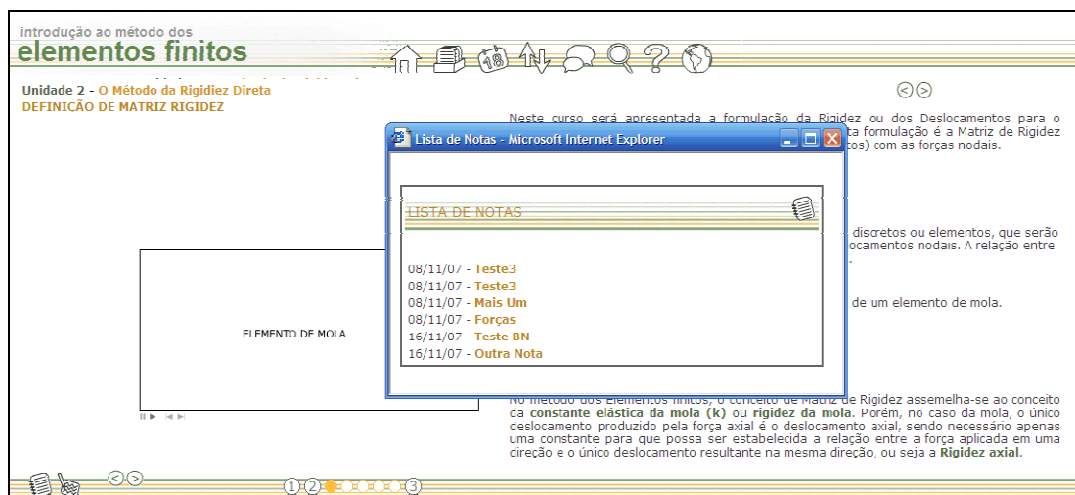


FIGURA 31 – Listagem das anotações existentes

6.2. RECADOS

A ferramenta recados é um sistema para aviso de recados. Posteriormente, esta ferramenta poderá ser modificada permitindo que os professores de um curso deixem recados para seus alunos. Essa implementação é possível de ser realizada, já que a aplicação foi desenvolvida utilizando uma linguagem orientada a objetos e considerando os aspectos de modularidade, flexibilidade.

Através de um calendário, os alunos selecionam o dia e acrescentam recados naquele dia e escolhem se desejam ser lembrados daquele recado e quando. A ferramenta oferece três opções para lembrar os usuários dos seus recados: no dia, um dia antes ou uma semana antes. Os recados e opções de lembrete de cada usuário são armazenados em um banco de dados *Access*.

A lista com os lembretes escolhidos aparecerá na data correta para cada usuário na página principal do curso.

A ferramenta recados é acessada através do clique do *mouse* no ícone referente a ele localizado no rodapé. Esta ferramenta é disponibilizada em um *pop-up* que abre sobre a janela principal do curso. A figura 32 mostra o *pop-up* da ferramenta recados e a figura 33 mostra como é realizado o acesso aos recados existentes. A listagem dos recados

existentes é acessada através do clique no botão “ver todos” da ferramenta recados. Esta listagem também é exibida em um *pop up*.

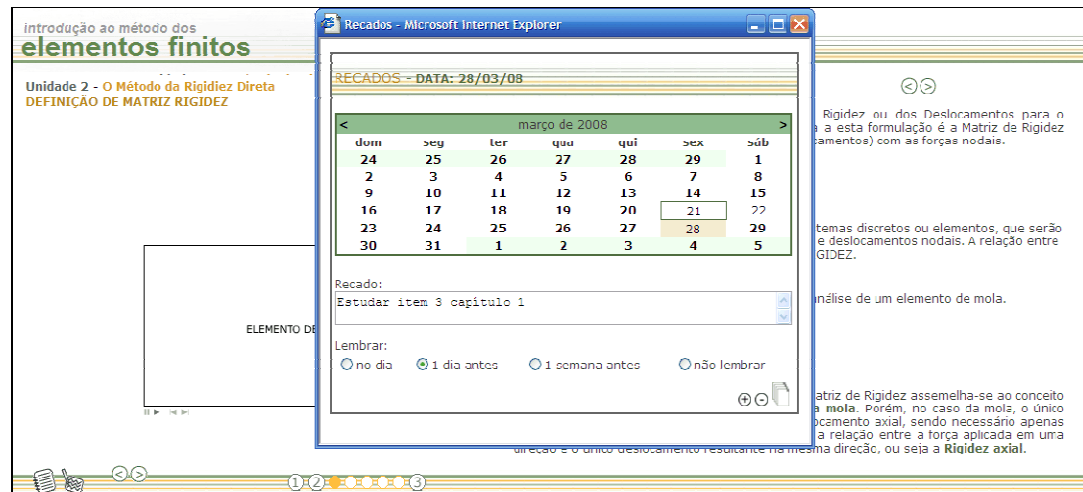


FIGURA 32 – Ferramenta Recados

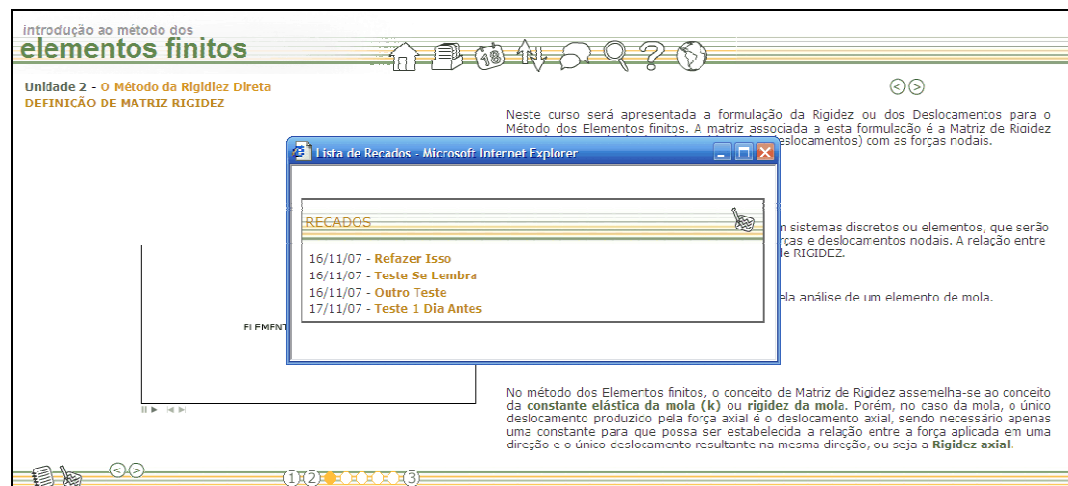


FIGURA 33 – Listagem dos recados existentes

A ferramenta recados é disponibilizada através de *pop-up* para permitir que os usuários realizem atividades simultâneas ao agendamento de recados.

6.3. AGENDA

A ferramenta agenda permite que os alunos se organizem melhor a partir da inserção de compromissos em dia e hora selecionados. Os dias são selecionados através do calendário disponível e a hora é acrescida em um campo apropriado pelo aluno. Os dias

que contêm compromissos agendados são destacados no calendário, permitindo que os usuários os identifiquem.

A página da agenda é acessada através de ícone específico no *menu* superior e está disponível na janela principal do curso. A figura 25 mostra a página agenda.

6.4. O BANCO DE DADOS DO CURSO

O banco de dados utilizado pelo curso de Introdução ao Método dos Elementos Finitos é o *Access*, seguindo o padrão do NucleoEAD. As tabelas deste curso encontram-se no mesmo arquivo dos demais cursos da escola.

O curso de Introdução ao Método dos Elementos Finitos apresentará duas tabelas, FIG. 34. A primeira tabela é uma tabela padrão para os cursos do NucleoEAD, apresentando campos padrões e essenciais para a disponibilização do curso na escola. Estes campos são:

- **CodigoUsuario:** é o campo onde estará definido o código do usuário na escola, este campo é a chave primária desta tabela;
- **DataUltimoAcesso:** campo onde é armazenada a data do último acesso do aluno;
- **Modulo:** campo onde é armazenado o módulo do conteúdo digital onde o aluno se encontra;
- **UltimoArquivo:** nesse campo é armazenado o último arquivo do curso acessado pelo aluno;
- **Mensagem:** esse campo é necessário para que o usuário possa utilizar a ferramenta de mensagens do NucleoEAD.

A tabela disponibilizada para armazenamento e acesso das novas ferramentas desenvolvidas apresenta os seguintes campos:

- **Chave:** este campo é a chave primária da tabela;
- **CodigoUsuario:** este campo se relaciona com o campo **CodigoUsuario** da tabela do curso e indica a qual usuário pertence os registros dos próximos campos.
- **Data:** este campo armazena a data selecionada pelo usuário em uma das ferramentas utilizadas, agenda ou recados. No bloco de notas a data será sempre a atual.
- **Compromisso:** este campo é preenchido quando a ferramenta agenda é utilizada e refere-se ao compromisso estabelecido pelo usuário.
- **Assunto:** este campo é preenchido quando a ferramenta bloco de notas é utilizada e refere-se ao assunto da nota estabelecido pelo usuário.
- **Nota:** este campo é preenchido quando a ferramenta bloco de notas é utilizada e refere-se às notas escritas pelo aluno.
- **Recado:** este campo é preenchido quando a ferramenta recados é utilizada e refere-se ao recado escrito pelo usuário.
- **Lembrete:** neste campo estará a indicação se o usuário deve ser lembrado do recado e quando será lembrado. O número -1 indica que o recado não deve ser lembrado, o número 0 indica que o lembrete deve ser mostrado no dia, o número 1 indica que o lembrete deve ser mostrado um dia antes e o número 7 indica que o lembrete deve ser mostrado uma semana antes.

As três ferramentas Bloco de Notas, agenda e Recados, poderiam apresentar tabelas diferentes, o que evitaria a existência de campos não utilizados nos registros. Porém, por se tratar de uma aplicação pequena, decidiu-se fazer toda a estrutura em uma tabela, reduzindo o número de conexões necessárias com o banco de dados.

Para acesso das páginas das ferramentas ao banco de dados foi utilizado o *WebControl* do ASP.NET denominado *AccessDataSource*. Este controle realiza a conexão com o

banco de dados e apresenta ferramentas desenvolvidas para operações de busca, inserção, alteração e exclusão de elementos.

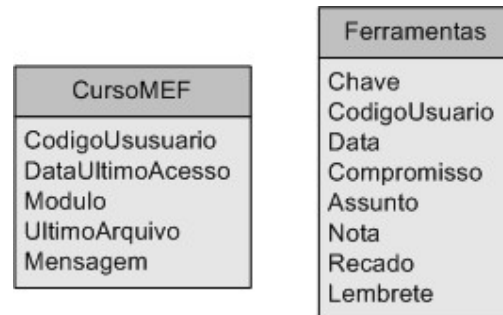


FIGURA 34 – Banco de dados do curso

6.5. PROGRAMAÇÃO EM ASP.NET

Estas Ferramentas foram desenvolvidas utilizando o ASP.NET 2.0 e a linguagem de programação C#. Conforme discutido na seção 2, C# é uma linguagem de programação orientada a objetos. Estando todo o código desenvolvido enquadrado nesse paradigma, foram utilizados os conceitos de variáveis, classes, funções, além da utilização de componentes do ASP.NET, que dão agilidade no desenvolvimento do trabalho.

Para a programação foi utilizado o modelo de programação *Code Behind*, no qual o código encontra-se em arquivo separado dos elementos HTML. Assim, para cada arquivo .aspx há um arquivo .aspx.cs. Foram seis páginas desenvolvidas: BlocoDeNotas.aspx, ListaNotas.aspx, Recados.aspx, ListaRecados.aspx, Agenda.aspx e Lembretes.aspx. Cada uma destas páginas apresenta uma página correspondente com a extensão .aspx.cs, onde estarão as funções necessárias relacionadas à cada uma.

Os principais controles (*WebControls*) utilizados para desenvolvimento dessas ferramentas foram: *Calendar*, *FormViews*, *AccessDataSource*, além de *WebControls* comuns como *label*, *textbox*, *radiobutton*.

Calendar é um *WebControl* responsável pela apresentação de calendários e apresenta já implementadas várias funcionalidades para visualização e manuseio de calendários digitais.

FormView é um controle responsável pelo gerenciamento de ações comuns no recebimento e envio de formulário via web, como atualização de campos, exclusão de itens, inserção e salvamento de dados.

AccessDataSource é um controle responsável pelas operações e conexão com o banco de dados.

As funções desenvolvidas em cada página foram funções para salvar, excluir, recuperar os dados do banco de dados e apresentá-los corretamente nos locais devidos.

O NucleoEAD foi desenvolvido utilizando a linguagem de programação ASP (clássico), as ferramentas apresentadas foram desenvolvidas utilizando o ASP.NET. Para que o curso de Introdução ao Método de Elementos Finitos, desenvolvido utilizando ASP.NET, pudesse ser inserido no NucleoEAD, foi necessário o desenvolvimento de funções, organizadas em uma classe que manipulasse o objeto de sessão (*Object Session*) de cada usuário nas páginas desenvolvidas em ASP e convertesse para a versão ASP.NET. A partir dessa transformação foi possível integrar o curso e ferramentas desenvolvidas no NucleoEAD.

7

O APLICATIVO *WEB* PARA O MÉTODO DOS ELEMENTOS FINITOS

7.1. INTRODUÇÃO

O aplicativo para análise de estruturas pelo Método dos Elementos Finitos desenvolvido como parte desta dissertação de mestrado é um sistema computacional com o objetivo principal de ser uma ferramenta para auxílio didático durante o ensino deste método. O sistema foi desenvolvido para ser aplicado via “web”.

O aplicativo foi desenvolvido utilizando o paradigma de programação orientada a objetos e o *framework* ASP.NET, com a linguagem C#.

Este programa calcula esforços e deslocamentos em elementos unidimensionais (treliça, viga, pórtico e grelha) no espaço bi-dimensional, com ações externas aplicadas em seus nós. Este aplicativo pretende complementar o curso de Introdução ao Método dos Elementos Finitos desenvolvido, representando mais uma ferramenta para estimular o aprendizado.

A formulação utilizada para o desenvolvimento do programa foi a formulação clássica do método, desenvolvida a partir de equações polinomiais que descrevem os

deslocamentos ao longo dos elementos e das equações de equilíbrio dos elementos, das quais é possível estabelecer uma relação entre as forças externas aplicadas e os deslocamentos produzidos, ou vice-versa.

Neste capítulo decidiu-se por restringir as exposições aos aspectos relativos ao desenvolvimento e metodologia específica utilizada no aplicativo, pois, as demais considerações a respeito do papel da computação gráfica, recursos e bases tecnológicas utilizadas, configurações do sistema, já foram tratadas no capítulo 5: “*O Curso de Introdução ao Método dos Elementos Finitos*”, e se repetem aqui.

7.2. INTERFACE

O site do programa é composto de seis páginas, como pode ser observado na FIG.35. Destas seis páginas, quatro são para que o usuário faça a entrada dos dados da estrutura a ser calculada. Na quinta página o usuário pode visualizar os resultados desejados e na última página, denominada ferramentas, poderá ser feita a importação de dados a partir de um arquivo do tipo ASCII em um formato válido ou a exportação de dados para um arquivo do tipo ASCII.

A página inicial do aplicativo foi desenvolvida objetivando uma pequena introdução dos tipos de estruturas que podem ser analisadas. As demais páginas são acessadas através do *menu*, fixo e disponível permanentemente, a FIG. 35 ilustra esse as possibilidades de acesso às páginas através do *menu*.

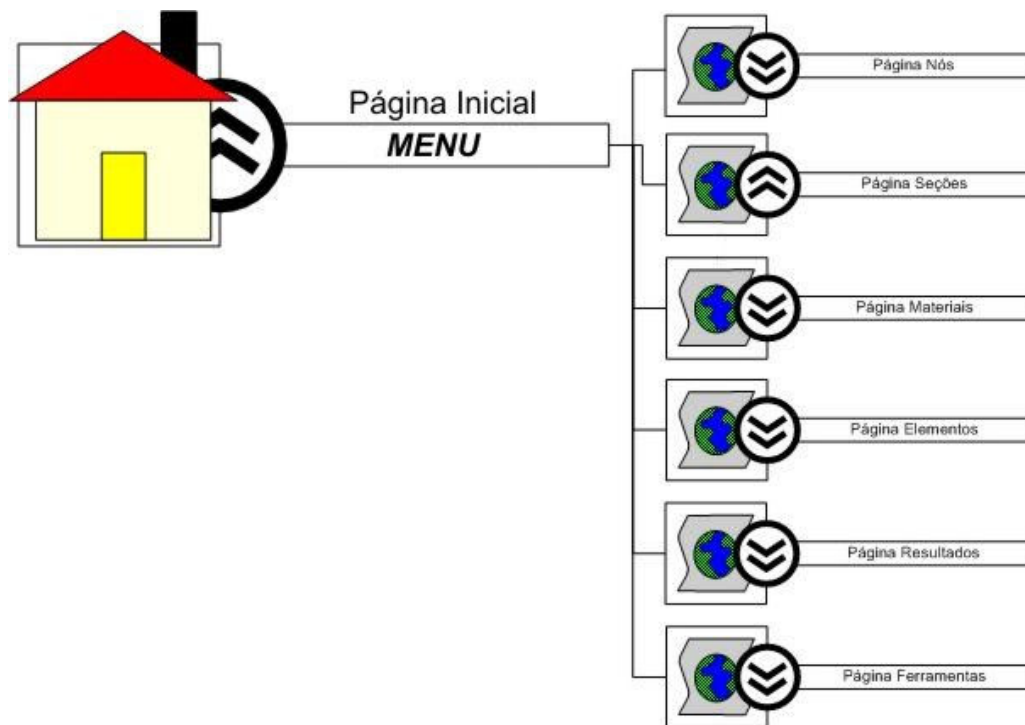


FIGURA 35 – Diagrama do aplicativo do Método dos Elementos Finitos

O layout deste aplicativo foi desenvolvido mantendo a mesma aparência do curso de Elementos Finitos, como pode ser observado na FIG.36, com a proposta do curso e programa serem disponibilizados em conjunto, para facilitar o aprendizado do Método dos Elementos Finitos.

A primeira página existente para a entrada dos dados da estrutura é a página **Nós**, FIG.37. Nesta página o usuário escolhe o número de nós da estrutura discretizada e coloca nos campos apropriados os dados de cada nó: coordenadas x , y ou z ; forças nodais e restrições de cada nó associado aos apoios.

A segunda página para a entrada de dados da estrutura é a página **Seções**, FIG.38. Nesta página, o usuário escolhe o número de seções transversais que compõem os elementos da estrutura. Para cada seção transversal, deverão ser preenchidos os campos correspondentes às propriedades geométricas de cada seção transversal, a saber; área, momento de inércia I e constante de torção J .

A página **Materiais** é destinada para a entrada dos dados dos materiais existentes na estrutura. Nesta página, como pode ser observado na FIG.39, o usuário escolhe o número de materiais do sistema e informa, para cada material, o seu módulo de elasticidade e o coeficiente de *Poisson*. De forma padrão, o programa permite a escolha de dois materiais (aço e concreto), para os quais seus coeficientes são preenchidos automaticamente, devendo o usuário estar atento para as unidades utilizadas.

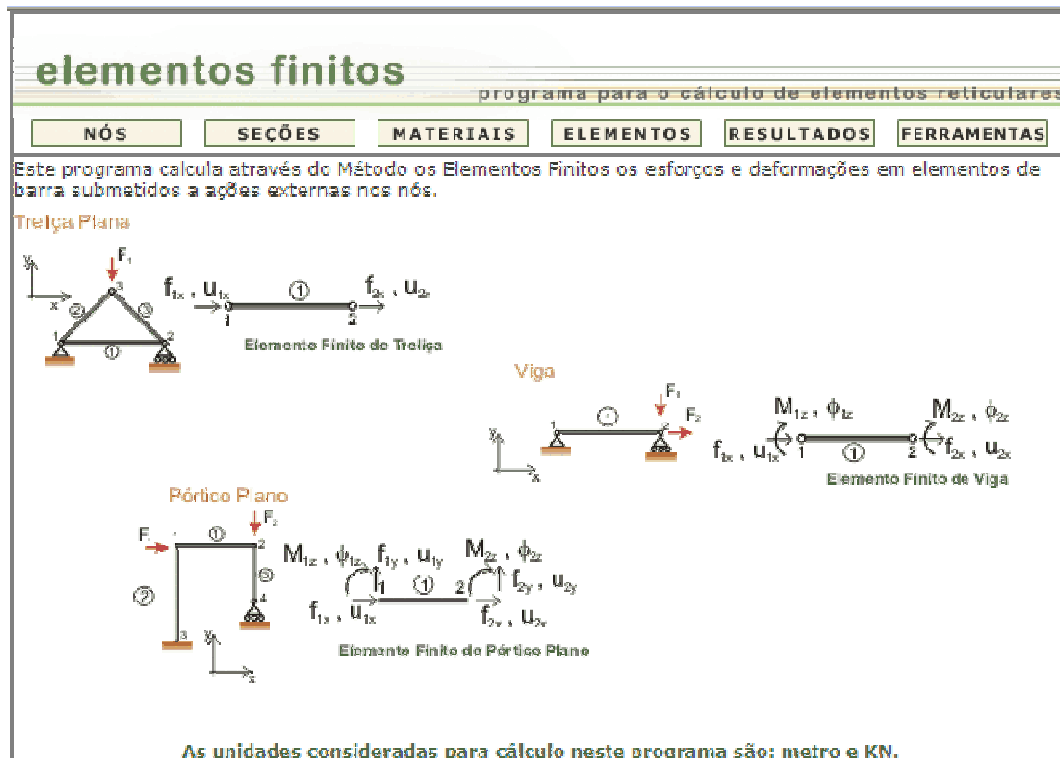


FIGURA 36 – Página inicial do aplicativo do Método dos Elementos Finitos

elementos finitos programa para o cálculo de elementos reticulares

NÓS **SEÇÕES** **MATERIAIS** **ELEMENTOS** **RESULTADOS** **FERRAMENTAS**

Nós

Número de nós:

Coordenadas Nodais

Nó	X	Y	Z
1	0	0	0
2	5	0	0
3	6	6	0

Forças Nodais

Nó	Fx	Fy	Fz
1	0	0	0
2	0	0	0
3	0	50	0

Restrições Nodais

Nó	Rx	Ry	Rz	ox	oy	oz
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FIGURA 37 – Página Nós

elementos finitos programa para o cálculo de elementos reticulares

NÓS **SEÇÕES** **MATERIAIS** **ELEMENTOS** **RESULTADOS** **FERRAMENTAS**

Seções

Número de seções:

Seção **Área** **I** **J**

1	2	2	2
2	4	4	4

FIGURA 38 – Página Seções

elementos finitos
programa para o cálculo de elementos reticulares

NÓS SEÇÕES MATERIAIS ELEMENTOS RESULTADOS FERRAMENTAS

Materials

Número de materiais:

Material		E	Ni
1	Aço	205000	0.3
2	Concreto	250000	0.2
3	Outro	100000	0.1

FIGURA 39 – Página Materiais

A última página para entrada de dados da estrutura é a página Elementos, FIG.40. Nesta página, o usuário entra com a quantidade de elementos que compõem a estrutura, escolhe o tipo de cada elemento (elemento de treliça, viga, pórtico ou grelha), indica a conectividade de cada elemento (nó inicial e nó final), o material e a seção transversal de cada elemento.

Após a entrada de todos os dados, a estrutura pode ser calculada. O botão para cálculo da estrutura está na página resultados, FIG.41. Após o cálculo, ou processamento da estrutura, podem ser visualizados os resultados. A escolha dos resultados a serem visualizados é feita através do sub-menu presente na página resultados. O primeiro botão do sub-menu resultados é: **graus de liberdade**, ou seja, podem ser visualizados os deslocamentos generalizados (translação e rotação) possíveis aos nós da estrutura. Estes graus de liberdade são definidos em função do tipo de elemento escolhido. O segundo botão é **matriz de rigidez**, a partir do qual pode ser visualizada a matriz de rigidez global da estrutura ou a matriz de rigidez de cada elemento. Outros resultados que podem ser visualizados são o sistema de equações da estrutura, representado de forma

geral pela equação 7.1 abaixo, as reações de apoio de cada nó, e os esforços de extremidade de cada elemento. Estes resultados disponíveis para visualização são dados básicos e necessários durante a resolução de um sistema utilizando o MEF.

$\underline{F} = \underline{K}\underline{u}$	(7.1)
--	-------

Onde \underline{F} é o vetor de forças da estrutura, \underline{K} é a matriz de rigidez da estrutura e \underline{u} é o vetor de deslocamentos da estrutura.

elementos finitos

programa para o cálculo de elementos reticulares

NÓS
SEÇÕES
MATERIAIS
ELEMENTOS
RESULTADOS
FERRAMENTAS

Elementos

Número de elementos

Elemento	Tipo	Nó 1	Nó 2	Seção	Material
1	Treça	1	3	1	3
2	Treça	2	3	2	1

FIGURA 40 – Página Elementos



FIGURA 41 – Página Resultados



FIGURA 42 – Página ferramentas

A última página do site do aplicativo da MEF é a página **Ferramentas**, FIG.42. Esta página apresenta as ferramentas para importação e exportação de arquivos do tipo ASCII com extensão .txt contendo os dados do programa. Há a possibilidade de o usuário realizar a entrada de dados através de um arquivo ASCII importando-o e a possibilidade do usuário verificar em um arquivo ASCII os resultados de uma estrutura processada, através da exportação dos dados e resultados do programa.

A importação/exportação através de arquivos ASCII permite a troca de informações com outros programas mais complexos ou com aplicativos que apresentam uma interface gráfica, para a visualização dos dados através de desenhos ou diagramas.

Todo este layout foi desenvolvido buscando ser o mais intuitivo possível, de forma a facilitar o manuseio do programa pelos usuários.

7.3. O APLICATIVO DO MÉTODO DOS ELEMENTOS FINITOS E A PROGRAMAÇÃO ORIENTADA A OBJETOS

O aplicativo do Método dos Elementos Finitos foi desenvolvido utilizando o paradigma de programação orientada a objetos. Toda a sua estrutura foi concebida baseando-se em classes, tentando agrupar as funcionalidades semelhantes.

As vantagens do desenvolvimento de aplicativos considerando a programação orientada a objetos já foi citada anteriormente, cabendo ressaltar a grande facilidade de ampliação posterior do sistema, a facilidade de manutenção e o reaproveitamento de código.

O MEF é um método amplo e seu escopo abrange muito além de elementos de barra. Portanto, este programa foi desenvolvido pensando em sua posterior ampliação, tanto através da inserção de novas funcionalidades quanto na utilização de outros tipos de elementos finitos para o cálculo de estruturas, apenas criando novas classes para representar elementos afins e derivando-as das classes mães pertinentes.

A FIG.43 mostra o diagrama geral do programa, com suas classes e o relacionamento entre elas. Basicamente, este aplicativo é composto de seis grupos de classes.

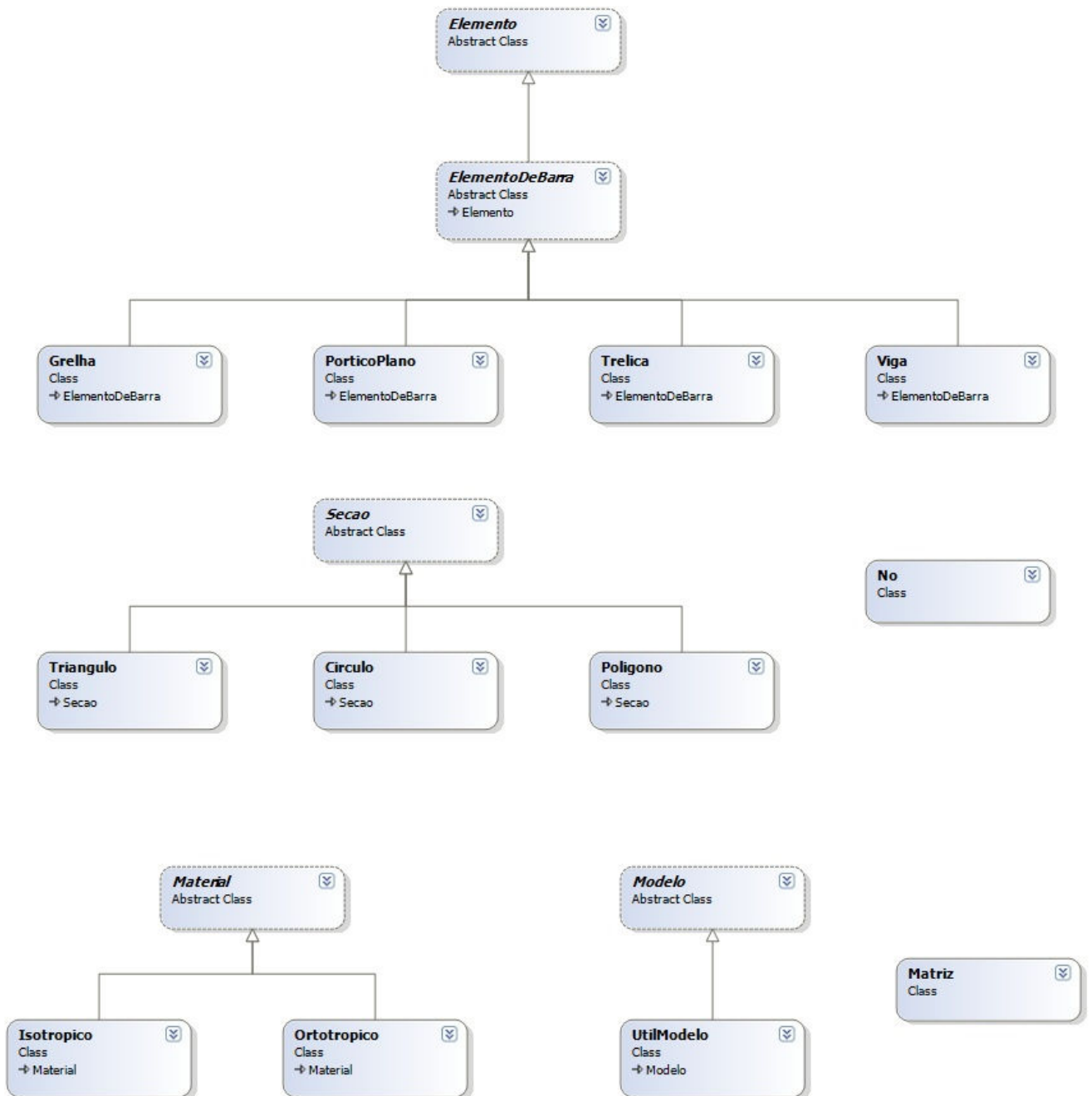


FIGURA 43 – Diagrama das classes básicas do aplicativo do Método dos Elementos Finitos

O primeiro grupo de classes descreve as características e funcionalidades dos elementos do MEF necessários para a resolução de problemas. A classe mãe é a classe **Elemento**, descrevendo características gerais para qualquer tipo de elemento, seja elemento de barra, elemento bidimensional ou outro. Alguns exemplos dessas características estão mostrados na FIG.44, que é o diagrama estendido (contendo os métodos e atributos) da

classe **Elemento** e de suas derivadas. A classe **ElementoDeBarra** apresenta todas as características da classe **Elemento** além de outras específicas apenas para elementos unidimensionais, como seção transversal. Assim, essa classe é derivada da classe **Elemento**. As classes **Trelica**, **Viga**, **Grelha** e **PorticoPlano** são tipos específicos de elementos unidimensionais e por isso, derivam desta classe (**ElementoDeBarra**). Cada uma dessas classes apresentam características distintas. Pode-se observar que as classes **Elemento** e **ElementoDeBarra** são classes abstratas, portanto não podem ser instanciados objetos destas classes. Apenas objetos dos tipos específicos: treliça, viga, grelha e pórtico plano podem ser instanciados.

Outro grupo de classes existente é a classe **Secao**, FIG.45 e suas derivadas. Este grupo foi desenvolvido objetivando facilitar o cálculo das propriedades geométricas de seções transversais. A classe **Secao** é uma classe abstrata e dela derivam três classes que representam tipos específicos de seções transversais, como seções transversais em forma de triângulo, círculo ou um polígono qualquer. Cada uma das classes derivadas de **Secao** apresenta métodos específicos para o cálculo das propriedades geométricas. Por exemplo, a fórmula para cálculo da área da seção transversal de um círculo é diferente da fórmula utilizada para um triângulo e pode ser diferente para um polígono. O mesmo vale para o cálculo das demais propriedades. Por este se tratar de um aplicativo simples, o usuário já deve fornecer as propriedades geométricas da seção transversal, mas uma alternativa seria permitir que o usuário fornecesse o tipo da seção transversal e o programa calcularia automaticamente as características geométricas necessárias.

O grupo de classes responsável pela descrição das características dos materiais que compõem uma estrutura é formado pela classe mãe abstrata **Material** e pelas suas duas classes filhas: **Isotropico** e **Ortotropico**, FIG.46. Os dois atributos dessas classes são o módulo de elasticidade e o coeficiente de *Poisson*. Há a diferenciação entre as classes Isotrópico e Ortotrópico pois, estes atributos são representados de forma diferentes nessas duas classes. Para os materiais isotrópicos, estes coeficientes são representados com apenas um valor, pois este tipo de material comporta-se de forma equivalente para qualquer direção da deformação. Já para os materiais ortotrópicos, esses atributos são representados com três valores, (um para cada direção - x, y e z), pois este tipo de

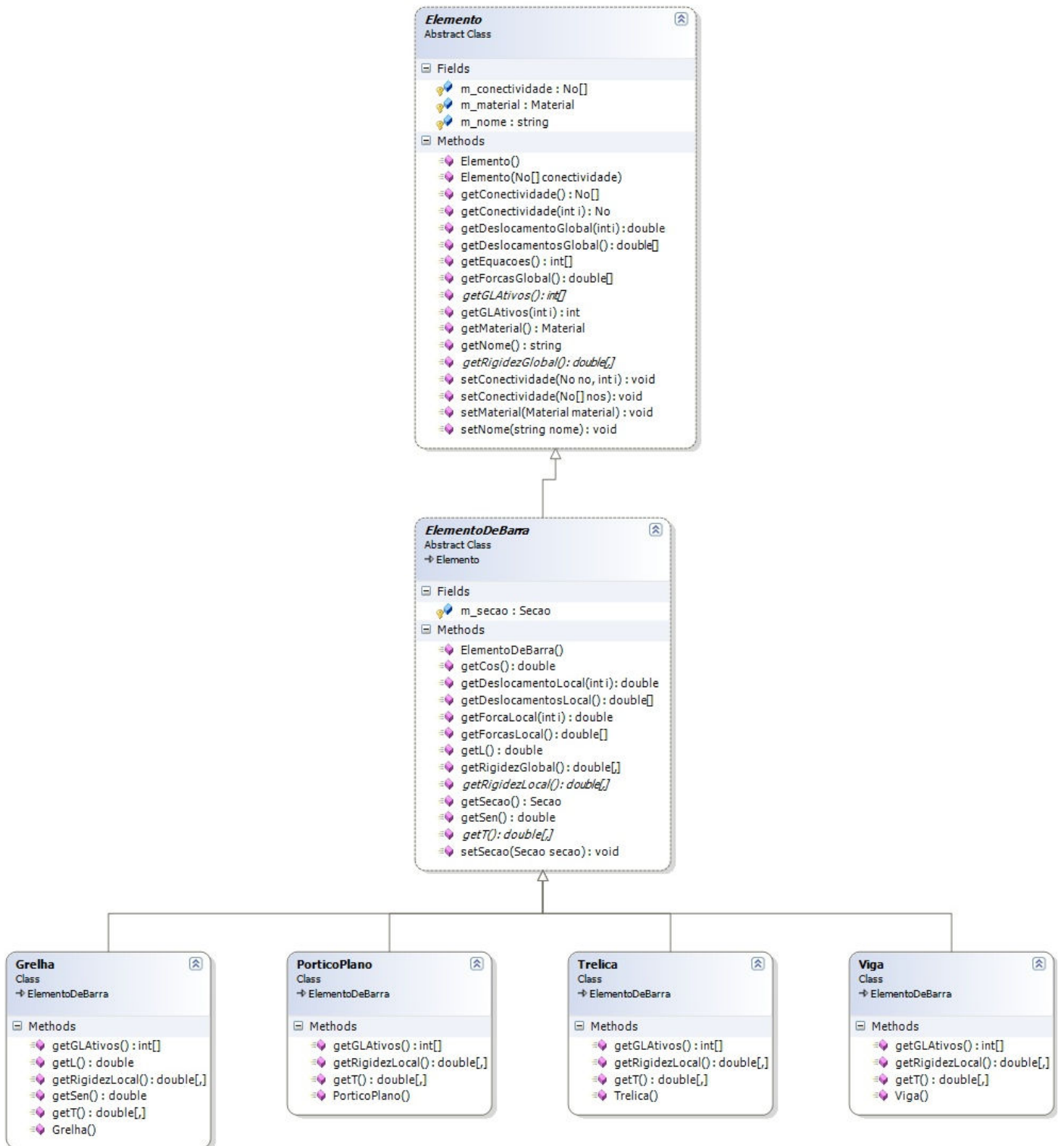
material comporta-se de forma desigual para as deformações dependendo do sentido. Este aplicativo considera apenas materiais isotrópicos em seus cálculos, ou seja, o usuário informa apenas um valor para o módulo de elasticidade e para o coeficiente de *Poisson*, considerando tratar-se um material isotrópico. Porém, quando forem implementados outros tipos de elementos, bidimensionais ou tridimensionais, esta característica se encontra pronta para ser utilizada. Considerando, também, só materiais elásticos.

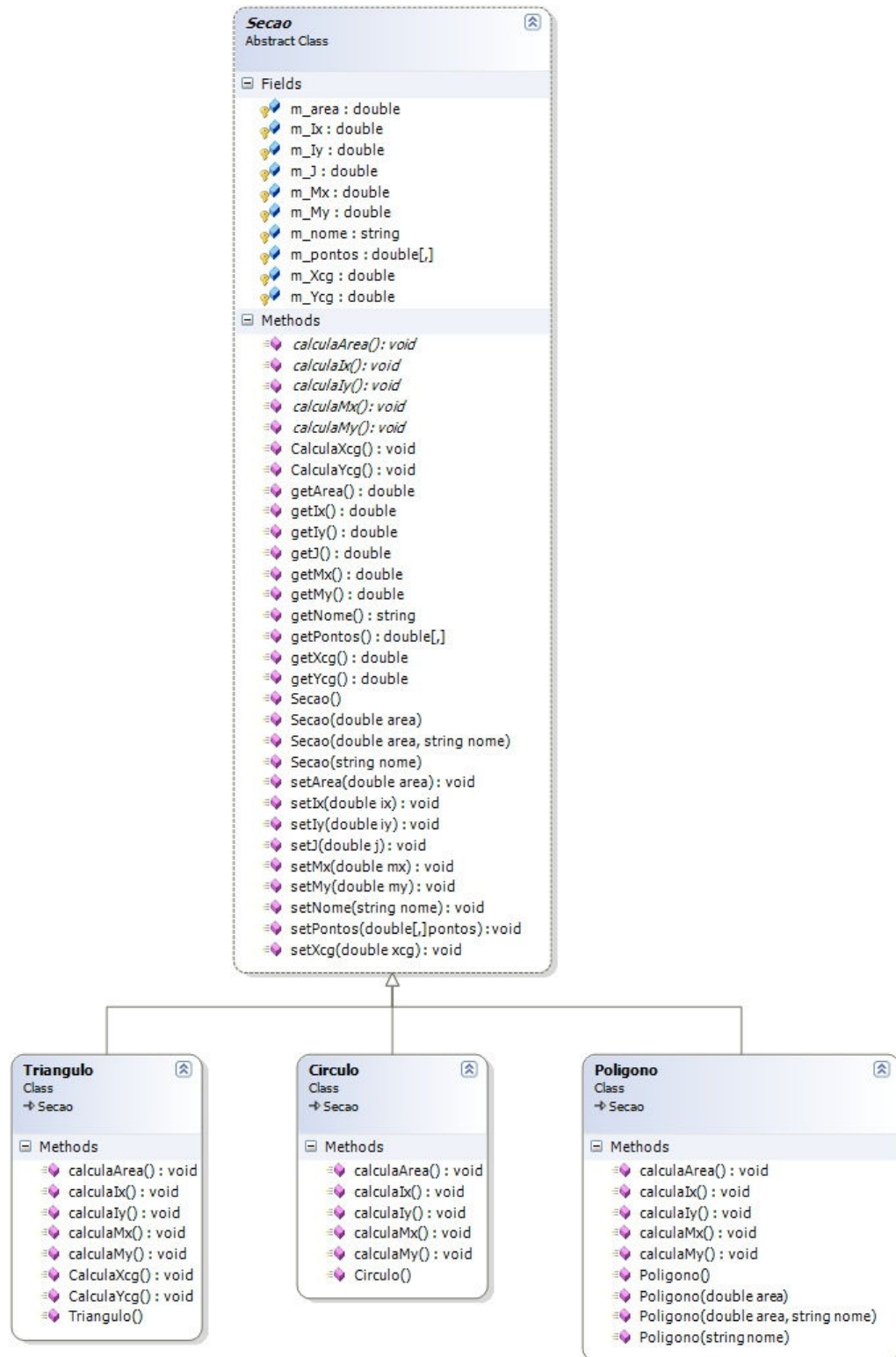
Outra classe extremamente importante para qualquer aplicativo do MEF é a classe **No**. Esta classe representa os nós dos elementos finitos e engloba todos os atributos e métodos necessários para a resolução de problemas, como pode ser observado na FIG.46.

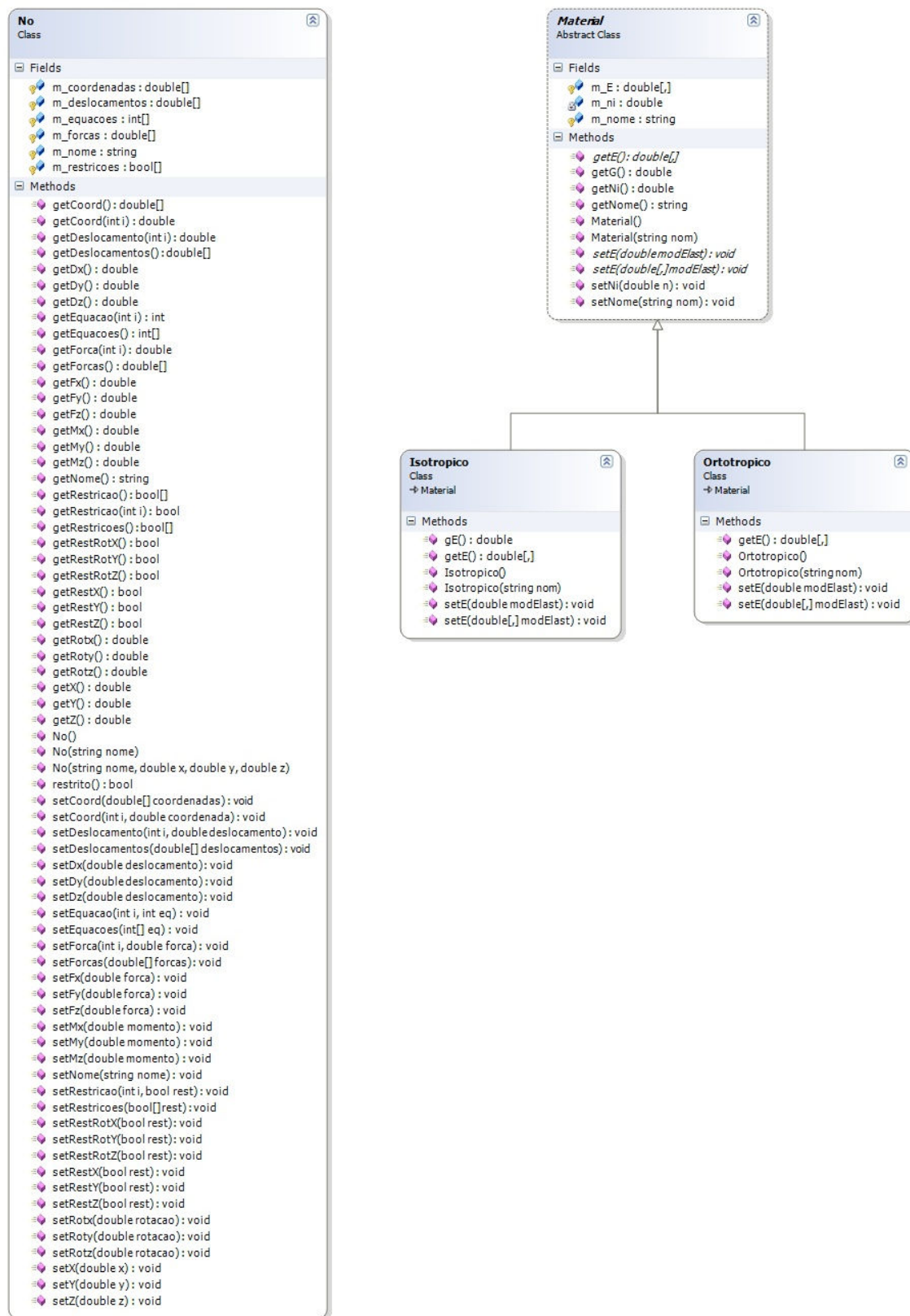
A classe **Matriz** é responsável pelas operações matriciais necessárias para a resolução de problemas do MEF. O diagrama estendido dessa classe está na FIG.47.

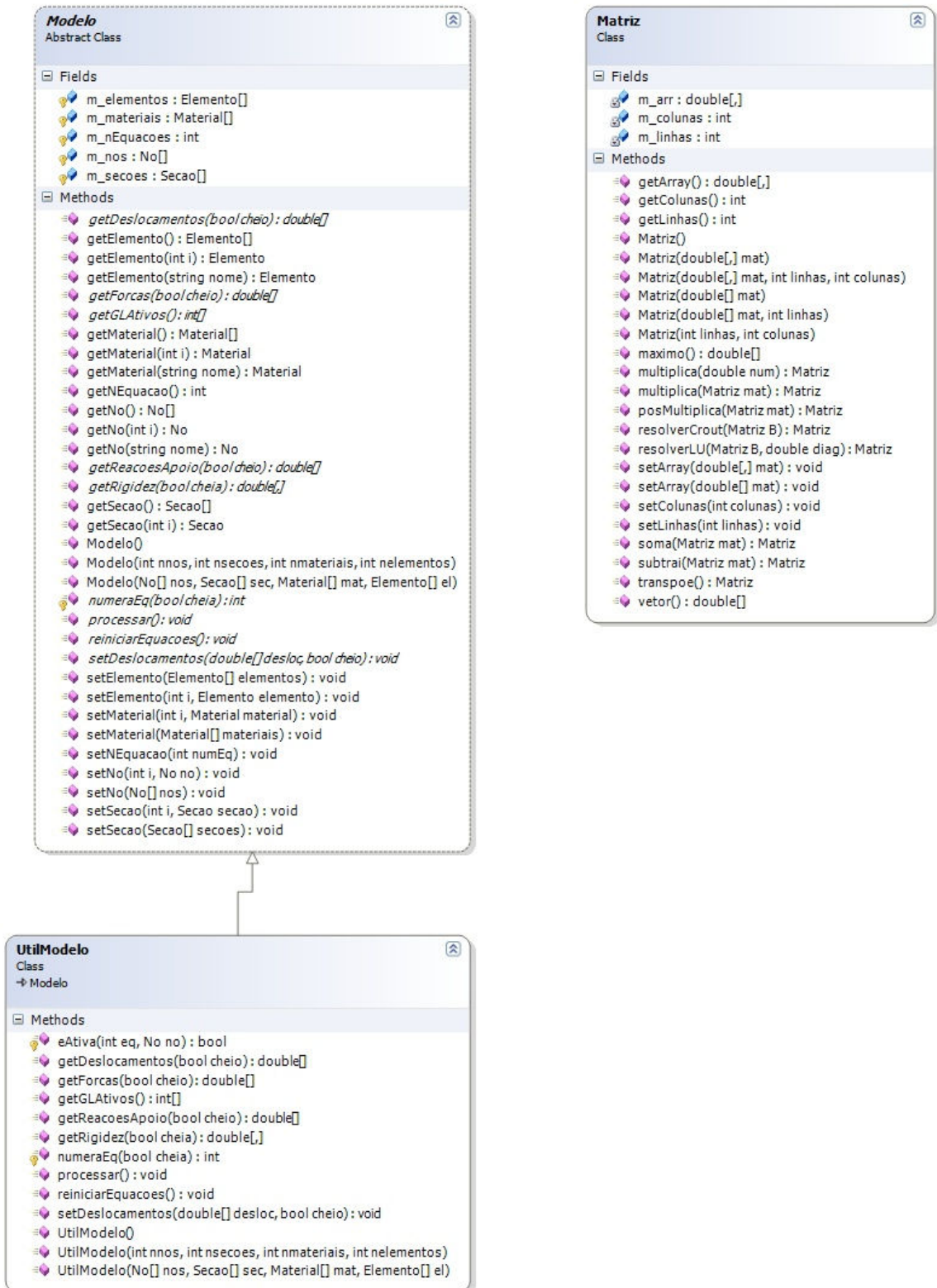
A classe **Modelo** é uma classe abstrata para montagem ou agrupamento dos dados do aplicativo com listas de nós, elementos, materiais, seções e os métodos de acesso e atribuição destas variáveis, como observado na FIG.47. A classe **UtilModelo** é uma classe concreta, derivada da classe **Modelo**, ou seja, possui todos os atributos dessa classe, e é responsável pela execução dos cálculos necessários ao modelo, utilizando as listas das variáveis da classe **Modelo**, por esse motivo é derivada de **Modelo**.

Estas classes descritas são a base para o desenvolvimento de um programa do MEF, no entanto, há outras classes responsáveis por tratar e apresentar a interface do programa, trabalhando com a manipulação de *datagrids* (componentes do ASP.NET), etc. Estas classes não serão descritas aqui, por se tratar apenas de classes para manipulação de dados, não apresentando nenhuma relação com o Método dos Elementos Finitos, podendo variar de acordo com a linguagem utilizada para a programação.

FIGURA 44 – Diagrama estendido da classe **Elemento** e suas derivadas

FIGURA 45 – Diagrama estendido da classe **Secao** e suas derivadas

FIGURA 46 – Diagrama estendido da classe **No** e das classes **Material** e suas derivadas

FIGURA 47 – Diagrama estendido das classes **Modelo**, **UtilModelo** e **Matriz**

8

CONCLUSÃO

8.1. INTRODUÇÃO

A EAD através da *web* está se tornando, cada vez mais, uma modalidade de ensino essencial para a difusão do conhecimento de uma maneira coletiva. Os custos finais inferiores as outras modalidades de educação e a crescente expectativa da sociedade por novas metodologias e novas abordagens pedagógicas estabelecem para EAD grandes desafios futuros. Neste contexto o papel do aluno passa ser de um personagem mais ativo no seu próprio processo de aprendizagem.

O número de *software* e sistemas educacionais existentes no mercado e até mesmo de cursos a distância ou semi-presenciais cresce exponencialmente, surgindo necessidades de fundamentos sólidos para o desenvolvimento dessas relações de interação e interatividade. Essas bases sólidas podem ser formadas pela construção e utilização de ferramentas tecnológicas eficientes e pelo desenvolvimento e utilização de novas metodologias de ensino e de aprendizagem mais eficazes e condizentes com a modalidade de ensino a distância. Para atingir essas metas é necessário investimentos

em várias áreas do conhecimento e em projetos como a presente dissertação de mestrado. Vale salientar alguns aspectos relevantes onde existe carência em investimentos e pesquisa para a evolução da EAD:

- qualificação de recursos humanos na área tecnológica.
- projetos pedagógicos para explorar experimentos em metodologias de ensino voltados para a EAD.
- exploração dos recursos da computação gráfica no seu sentido mais amplo, que envolve vídeos, animações, multimídias, imagens, fotos, etc... para aplicação na EAD.
- Pesquisa sobre as formas de estimular os sentidos (visual, auditivo, sentido do tato e até o olfativo) para melhorar o aprendizado, a interação e a interatividade do aluno com o curso.

Assim será possível esperar melhores resultados e qualidade nos processos de ensino/aprendizagem oriundos da EAD.

As tecnologias lançadas recentemente, como o ASP.NET e a introdução de linguagens orientadas a objetos na programação para *web* ampliaram as perspectivas para desenvolvimento de aplicações. As aplicações *web* têm se tornado cada vez mais complexas e suficientes para atender usuários cada vez mais exigentes. Essas aplicações, desenvolvidas com finalidades específicas, podem ser utilizadas de forma bastante eficiente na EAD.

Todo projeto de EAD, que vise atingir qualidade, deve ser iniciado através de um planejamento criterioso, desenvolvendo inclusive, planos para manutenção e aperfeiçoamento contínuo do mesmo.

Em desenvolvimento de projetos desse tipo, foi verificada a necessidade e a importância de uma equipe multidisciplinar para atuar nas diversas áreas que englobam. Como por exemplo, a estruturação e definições de interfaces e *layouts*; o desenvolvimento das ferramentas tecnológicas, que viabilizem a distribuição do curso e do apoio de forma satisfatória aos usuários; análises e definições das metodologias de ensino a serem

aplicadas; desenvolvimento do conteúdo didático do curso em uma área específica do conhecimento; dentre outros.

O conteúdo didático digital de um curso para EAD através da *web* deve ser bem elaborado, tomando cuidado para não sobrecarregar a página com textos ou com recursos multimídias que tornem o estudo cansativo ou causem sobrecargas sensoriais nos alunos. Texto e multimídia devem ser utilizados de forma ponderada. Os recursos multimídias utilizados deverão servir para reforçar conceitos, permitindo que o aprendizado aconteça através da utilização dos mesmos, adicionalmente a outros recursos comumente utilizados.

Tendo em vista a importância da EAD no cenário mundial, as universidades passaram a ter um papel fundamental no processo de evolução desta modalidade de educação. Nos últimos anos, várias iniciativas multidisciplinares têm ocorrido, com parcerias entre ciências humanas e exatas. Destas parcerias surgem idéias criativas para a construção de novas tecnologias, que possibilitam a transmissão do conhecimento por meio de novas abordagens pedagógicas e metodologias e que estimulam a interação e a iteratividade do aluno com o curso.

O papel da educação a distância na UFMG gradativamente ganha importância com a sua difusão e ampliação. Em específico na área de engenharia de estruturas, algumas iniciativas têm produzido cursos aplicados para a *web* de apoio ao ensino presencial. Futuramente o desenvolvimento de cursos oferecidos e aplicados exclusivamente a distância, como é a proposta desse projeto de dissertação, estarão disponíveis, o que tornará a difusão do conhecimento muito mais abrangente. Iniciativas em EAD devem então ser projetadas e programadas para possibilitar experiências pedagógicas, que permitam um aperfeiçoamento e evolução contínua nesta modalidade de ensino. O curso de Introdução ao Método dos Elementos Finitos, desenvolvido como parte desse projeto de mestrado representa uma dessas iniciativas de desenvolvimento de novas metodologias para o ensino da engenharia de estruturas a distância.

8.2. CONTRIBUIÇÕES DO PRESENTE PROJETO

São vários os aspectos para os quais esse projeto de mestrado contribuirá significativamente.

A principal contribuição é no desenvolvimento de materiais na área de engenharia de estruturas para ser aplicado via *internet*, expandindo o campo de influência do conhecimento e facilitando o acesso de mais pessoas ao mesmo.

Outras contribuições que podem ser citadas são: desenvolvimento de ferramentas para educação a distância via internet, exploração de metodologias para o ensino de engenharia de estruturas via internet, incentivo ao desenvolvimento de tecnologias aplicáveis à EAD via internet, exposição do tema educação à distância, abrangendo vários conceitos e produtos sobre o tema, etc.

8.3. PROPOSIÇÕES PARA DESENVOLVIMENTOS FUTUROS

Este trabalho permite que se façam proposições para desenvolvimentos futuros em vários aspectos.

O conteúdo digital do curso de Introdução ao Método dos Elementos Finitos deve ser ampliado, seguindo ou não o escopo proposto para desenvolvimento, e exposto no capítulo 5.

Outras ferramentas e tecnologias podem ser incorporadas ao curso de Introdução ao Método dos Elementos Finitos, ampliando o apoio e facilidades oferecidas aos alunos.

O aplicativo para o Método dos Elementos Finitos oferece uma grande perspectiva para ampliação, seja através da inclusão de outros elementos ou através do desenvolvimento da interface, gerando desenhos e maior interatividade com o usuário. Este aplicativo permite fácil ampliação, uma vez que foi desenvolvido utilizando o paradigma da orientação a objetos e pensando em futuras ampliações.

A proposta do desenvolvimento da plataforma tecnológica para EAD deve ser amadurecida e desenvolvida detalhadamente, finalizando no desenvolvimento de uma plataforma tecnológica para EAD que disponha de recursos e métodos principalmente para o ensino na área de engenharia de estruturas.

A continuidade do estudo das tecnologias utilizadas para EAD é outro aspecto natural para o desenvolvimento de trabalhos futuros, principalmente nesta área, onde os avanços e desenvolvimentos estão sempre ocorrendo, já que esta é a era da informação digital.

REFERÊNCIAS BIBLIOGRÁFICAS

10 em tudo. Disponível em: < www.10emtudo.com.br >. Acesso em: 17/03/08.

Alfabetização e Letramento. Disponível em: <<http://www.scipione.com.br/letramento/default.asp>>. Acesso em: 17/03/08.

ALVES, A. F. *Elementos Finitos: A Base da Tecnologia CAE*. 3ª edição, Editora ÉRICA, 2000, p. 191.

AulaNet. Disponível em: < <http://www.eduweb.com.br/downnet/index.asp> >. Acesso em: 20/03/08.

BATTIST, J. Modelo Relacional de Dados – Parte 01. Data: 10/08/2004. Disponível em: <http://imasters.uol.com.br/artigo/2419/bancodedados/o_modelo_relacional_de_dados_-_parte_01/>. Acesso em: 21/09/2004.

BATTIST, J. Modelo Relacional de Dados – Parte 02. Data: 10/08/2004. Disponível em: <http://imasters.uol.com.br/artigo/2419/bancodedados/o_modelo_relacional_de_dados_-_parte_02/>. Acesso em: 21/09/2004.

BATTIST, J. Modelo Relacional de Dados – Parte 03. Data: 10/08/2004. Disponível em: <http://imasters.uol.com.br/artigo/2419/bancodedados/o_modelo_relacional_de_dados_-_parte_03/>. Acesso em: 21/09/2004.

BATTIST, J. Modelo Relacional de Dados – Parte 04. Data: 10/08/2004. Disponível em: <http://imasters.uol.com.br/artigo/2419/bancodedados/o_modelo_relacional_de_dados_-_parte_04/>. Acesso em: 21/09/2004.

BolinOs. Disponível em: < <http://www.bolinos.com> >. Acesso em: 20/03/08.

CHAVES, E. C. **Multimídia – conceituação, aplicação e tecnologia**. Campinas: People Computação, 1991.

Claroline. Disponível em: < <http://www.claroline.net/> >. Acesso em: 20/03/08.

ClassWeb. Disponível em: < <http://classweb.ucla.edu/>>. Acesso em: 20/03/08.

DATE, C. J. Introdução aos Sistemas de Banco de Dados. Editora Campus – 1985.

DOVICCHI, J. C. **Educação à Distância: Referências Tecnológicas**. Núcleo Avançado de Computação Sônica e Multimídia, Universidade Federal de Uberlândia, Uberlândia, 28 de Agosto, 2003. Disponível em: <<http://thot.cursus.edu/rubrique.asp?no=12074>> e <<http://www.uned.es/catedraunesco-ead/plataformas.htm>> . Acesso em: 07/11/06.

DOVICCHI, J. C., MENDES, E. B., ALVES, J. B. M. **Teoria da Orientação ao Objeto para a Modularidade de Objetos Educacionais em Ambientes Virtuais**. II Congresso Brasileiro de Ensino Superior a Distância, Brasília - DF, setembro 2003.

DURÃES, R. BD: ADO.NET para Iniciantes. Data:23/04/2004. Disponível em: <<http://www.linhadecodigo.com/Artigo.aspx?id=296>> Acesso em: 03/03/2005.

ECMA-262, Especificação da Linguagem ECMAScript. Dezembro, 1999. 3a edição. Disponível em:< <http://www.ecma-international.org/publications/standards/Ecma-262.htm>>. Acesso em 22/03/2008.

ECMA-334, Especificação da Linguagem C#. Junho, 2006. 4a edição. Disponível em:< <http://www.ecma-international.org/publications/standards/Ecma-334.htm>>. Acesso em 22/03/2008.

EdukBr. Disponível em: < <http://www.edukbr.com.br/>>. Acesso em: 20/03/08.

EduOnline. Disponível em: < <http://www.eduline.com.br/>>. Acesso em: 20/03/08.

EduTecNet. Disponível em: < <http://www.edutecnet.com.br/>>. Acesso em: 20/03/08.

Eledge. Disponível em: < <http://eledge.sourceforge.net/>>. Acesso em: 20/03/08.

Ensino Net. Disponível em: < <http://www.ensino.net/>>. Acesso em: 20/03/08.

Escola 24 horas. Disponível em: < <http://www.escola24horas.com.br/>>. Acesso em: 20/03/08.

EscolaNet. Disponível em:< <http://www.escolanet.com.br/>>. Acesso em: 17/03/08.

Flash MX Professional 2004 – HELP. Copyright (c) 1993 – 2003, Macromedia, Inc. Version 7.0.1

FLE3. Disponível em: < <http://fle3.uiah.fi/index.html> > e <<http://www.nonio.uminho.pt/kitfle/>>. Acesso em: 17/03/08.

GUIMARÃES, R. Criando Componentes no ASP.NET. Data: 05/01/2004. Disponível em: < http://imasters.uol.com.br/artigo/1655/dotnet/criando_componentes_no_aspnet/ > . Acesso em: 22/03/08.

Habanero. Disponível em: < <http://www.isrl.uiuc.edu/isaac/Habanero/>>. Acesso em: 20/03/08.

HM-Card. Disponível em: <http://coronet.iicm.tugraz.at/HM_Tools/hmcard15/hmcard.htm>. Acesso em: 20/03/08.

Ilias. Disponível em: < <http://www.ilias.de/>>. Acesso em: 20/03/08.

Instituto Universidade Virtual Brasileira (UVB). Disponível em: < www.uvb.br>. Acesso em: 16/03/08.

Interaula Clube. Disponível em: < <http://www.interaula.com.br/>>. Acesso em: 20/03/08.

KidLink do Brasil. Disponível em: < <http://www.kidlink.org/portuguese/brasil/>>. Acesso em: 20/03/08.

Klick Educação. Disponível em: < <http://www.klickeducacao.com.br>>. Acesso em: 20/03/08.

Learning Space. Disponível em: < www.ibm.com/software/lotus/support/learningspace>. Acesso em: 20/03/08.

LIMA, R.G. *Ferramentas Computacionais para o Ensino e Aprendizagem à Distância de Estruturas de Aço*. 2002. P. 148. Dissertação de Mestrado – Escola de Engenharia da Universidade Federal de Minas Gerais, Belo Horizonte.

LOGAN, D. L. *A First Course in the Finite Element Method*. 3ª edição, Editora BROOKS/COLE, 2002, p. 695.

MANIC. Disponível em: < <http://ripples.cs.umass.edu/research.php#dotnet>>. Acesso em: 20/03/08.

Microsoft Educacional. Disponível em: < <http://www.microsoft.com/brasil/educacional>>. Acesso em: 20/03/08.

Microsoft. Disponível em: < <http://www.microsoft.com/brasil/>>. Acesso em: 17/03/2006.

Ministério da Educação e Cultura (MEC). Disponível em: <<http://www.mec.gov.br>>. Acesso em: 10/02/08.

Net Educação. Disponível em: < <http://www.neteducacao.tv.br>>. Acesso em: 20/03/08.

NEVES, C. M. C. **Referenciais de Qualidade para Cursos a Distância**, Brasília, abril, 2003.

NUNES, I.B. O que é a Educação a Distância. *Revista Educação a Distância*, Brasília, Dez./93-Abr/94. nrs 4/5. Páginas 1-25.

O-LMS. Disponível em: < <https://uonline.utah.edu/olms/>>. Acesso em: 20/03/08.

Portal Educacional. Disponível em: < <http://www.educaional.com.br>>. Acesso em: 20/03/08.

Projeto Aprendiz. Disponível em:< <http://www.uol.com.br/aprendiz/>>. Acesso em: 17/03/08.

Projeto Telemar Educação. Disponível em: <<http://pte.futuro.usp.br/homepage.do>>. Acesso em: 20/03/08.

Protagonismo juvenil. Disponível em: <<http://www.protagonismojuvenil.org.br>>. Acesso em: 20/03/08.

Referenciais de Qualidade para Educação Superior a Distância. Brasília, agosto de 2007. Disponível em: <<http://www.mec.gov.br>>. Acesso em: 10/02/08.

SANTOS, N. **Estado da Arte em Espaços Virtuais de Ensino e Aprendizagem**. *Revista Brasileira de Informática na Educação*, Florianópolis, v. 4, p. 24-38, 1999.

Disponível em: <<http://www.inf.ufsc.br/sbc-ie/revista/nr4/070tu-santos.htm>> Acesso em: 15/02/2005.

SHIMIZU, H. Planeta Digital. *Revista VEJA*, Cidade de Publicação, 1978, ano 39, nº 41, p. 106 a 108, 18 de outubro, 2006.

SILBERSCHATZ, A., KORTH, H.F. & SUDARSHAN S. Sistema de banco de Dados. Editora Makron Books – 1999.

STRUCHINER, M.; CORRÊA, N.; COSTA, J.B.S. **Hipermídia na Educação: princípios básicos para o desenvolvimento de material educativo**. Rio de Janeiro: UFRJ/NUTES, 1997. 69 p.

Study Web. Disponível em: <<http://www.studyweb.com>>. Acesso em: 17/03/08.

The Internet Public Library. Disponível em: <<http://www.ipl.org/ref/>>. Acesso em: 17/03/08.

The World Lecture Hall. Disponível em: <<http://www.utexas.edu/world/lecture/pl.org/ref/>>. Acesso em: 17/03/08.

TOLENTINO, R. J. V. *Aplicações web em XML: Estágio Atual e tendências futuras*. 2002, p. 136. Dissertação de Mestrado - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte.

TopClass. Disponível em: <<http://www.wbtsystems.com/solutions/technology/>>. Acesso em: 20/03/08.

TORRES, E. F., MAZZONI, A. A. **Conteúdos Digitais Multimídia: o foco na usabilidade e acessibilidade**. *Ci. Inf.*, Brasília, v. 33, n. 2, p. 152-160, maio/ago. 2004

Universidade Virtual Pública do Brasil (Unirede). Disponível em: <www.unirede.br>. Acesso em: 20/03/08.

UOL Educação. Disponível em:< <http://www.uol.com.br/educacao>>. Acesso em: 17/03/08.

Virtual-U. Disponível em: <<http://emediadesign.com/sfu/index.html>>. Acesso em: 20/03/08.

WebCT. Disponível em: < <https://webct41.gmu.edu/webct/public/home.pl>>. Acesso em: 20/03/08.

World Wide Web Consortium. Disponível em: <www.w3.org>. Acesso em: 21/03/08.

ANEXO A

VERSÃO PARA IMPRESSÃO DO CURSO DE INTRODUÇÃO DO MÉTODO DOS ELEMENTOS FINITOS

1

O MÉTODO DOS ELEMENTOS FINITOS NA ANÁLISE ESTRUTURAL

1.1. ANÁLISE ESTRUTURAL

A solução de problemas em Engenharia de Estruturas requer a utilização de Esquemas ou **Modelos de Cálculo**, capazes de representar matematicamente o comportamento da estrutura quando submetida a ações externas. A simulação deste comportamento, que é um dos objetivos da Análise Estrutural, é feita pela determinação de deslocamentos, tensões e deformações.

1.1.1. Objetivo da Análise Estrutural

Determinação de esforços, deslocamentos, tensões e deformações em uma estrutura em equilíbrio com as ações externas.

Animação 01: Passos Principais

Para a obtenção do **Modelo de Cálculo**, o problema real é substituído por um **Modelo Matemático (MM)**, a partir da aplicação de hipóteses simplificadoras.

Modelo Matemático proposto pode em alguns casos em que a geometria e demais considerações são simples, ser resolvido por expressões matemáticas (soluções analíticas), que determinam os valores das incógnitas procuradas para qualquer ponto do corpo. Este processo de cálculo requer a solução de equações diferenciais ordinárias ou parciais. No entanto, para a maioria das estruturas de importância prática, estas equações diferenciais podem não ser determináveis, devido aos seguintes fatores entre outros:

- complicadas condições de contorno (carregamento e apoios),
- geometrias complexa,
- propriedades não lineares dos materiais, comportamento não linear do problema (não linearidade física e/ou geométrica), etc..



FIGURA 01 – Tipos de Estruturas e soluções

Nas disciplinas Resistência dos Matérias e Teoria das Estruturas soluções exatas são obtidas através da resolução de equações diferenciais, conforme será visto oportunamente⁴. Estas soluções podem ser encontradas prontas em tabelas de livros que tratam do assunto e abrangem estruturas hiperestáticas como, por exemplo, vigas biengastadas, sujeitas a carregamentos distribuídos ou concentrados.

⁴ Link para Unidade 3 – Elemento Finito de Treliça – Modelo Matemático

A solução exata dos modelos matemáticos de problemas de análise estrutural resultam em solução para todos os pontos do corpo analisado (para os infinitos pontos do corpo) por meio de uma função matemática resultante da resolução de equações diferenciais. Dessa forma, o objeto da análise é tratado como um sistema contínuo, pois a solução é obtida para todos os pontos que constituem o corpo contínuo. Como a solução analítica não é viável para a maioria das estruturas reais, é necessário que outros métodos de análise, mesmo que aproximados, sejam utilizados.

Uma estratégia utilizada é a divisão das estruturas, ou dos sistemas contínuos, em partes separadas distintas (elementos), conectadas entre si em pontos discretos (nós). Assim, a análise de cada parte (do conjunto entendido como o somatório de todas as partes) permite conhecer o comportamento do todo. Esta divisão é chamada de discretização⁵ e é a base para o estudo dos Sistemas Discretos.

Observações:

- A solução⁶ obtida a partir de modelos de Sistemas Discretos é uma solução aproximada obtida em apenas alguns pontos da estrutura (nós da estrutura), a partir dos quais pode ser interpolada para o restante da estrutura.
- O número de nós escolhido deve ser suficiente para representar o comportamento do conjunto inteiro de forma satisfatória.
- O modo pelo qual a estrutura discretizada se comporta entre os nós do modelo dependerá das propriedades físicas e geométricas (área, módulo de elasticidade, coeficiente de Poisson) atribuídas ao elemento escolhido.

Quanto melhor especificado for o comportamento interno, mais a resposta do modelo irá se aproximar do comportamento real da estrutura.

Animação 02: Modelos

Animação 03: Diferenças entre Sistemas

⁵ Link para definição de Discretização, nesta versão pode ser encontrada ao final do item 1.2.

⁶ Link para explicação de quais valores podem ser incógnitas na solução de problemas discretos.

Discretização

O processo de modelagem de um corpo dividindo-o em um sistema equivalente composto por partes (elementos finitos) interconectados em pontos comuns a dois ou mais elementos ou interconectados a linhas ou superfícies é chamado de **discretização**.

Solução

A solução de problemas estruturais refere-se em geral, na determinação dos deslocamentos de cada nó e das tensões em cada elemento. Em problemas não estruturais, as incógnitas podem ser, por exemplo, temperaturas, pressões, dentre outras.

1.2. TIPOS DE ELEMENTOS

Para fazer a análise de uma estrutura através de sistemas discretos, o passo mais importante é a formulação do modelo matemático discreto, através da identificação dos componentes do sistema. Para facilitar esta identificação, são definidos alguns tipos de elementos discretos:

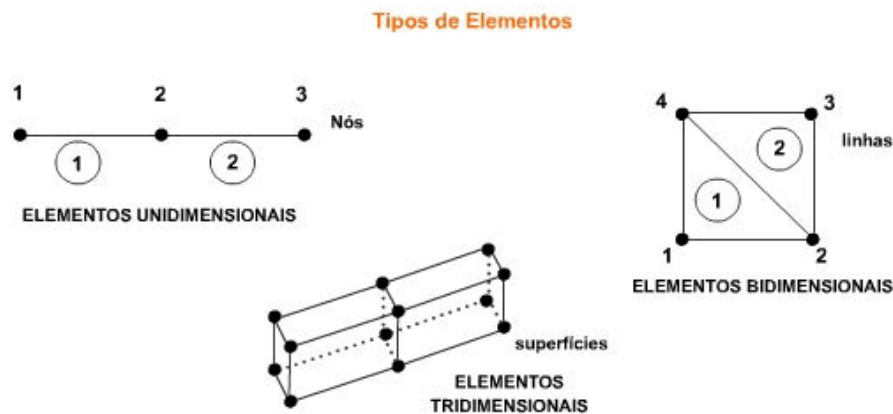


FIGURA 01 – Tipos de Elementos

1.2.1. Elementos Reticulados ou Unidimensionais

Representam geometrias em que uma dimensão é bem maior que as demais.

A interação entre estes elementos ocorre somente nos nós. Os elementos trocam esforços entre si somente nesses pontos. A aplicação das equações de equilíbrio nesses pontos nodais acrescida da condição de que os elementos continuam interconectados nos nós após a deformação da estrutura (condições de compatibilidade de deslocamentos), serão suficientes para conhecer matematicamente o modelo de cálculo.

Exemplos: Vigas contínuas, pórticos planos, pórticos espaciais, grelhas, treliças planas, treliças espaciais.

Os elementos reticulados são representados por barras unidimensionais representando o seus eixos.

Animação 01: Elementos Unidimensionais

Neste curso serão abordados os elementos reticulados: treliças planas.

1.2.2. Elementos Bidimensionais

Representam geometrias em que as forças e deslocamentos podem ser representados em apenas duas dimensões.

Nos elementos bidimensionais, a interação ocorre através dos nós. Porém, a interface entre dois elementos se dá através de linhas. Os elementos bidimensionais são subdivididos em:

- Elementos de estado plano: estado plano de tensões (EPT) e estado plano de deformações (EPD);
- Placas;
- Cascas.

Os elementos mais utilizados para discretização de estruturas bidimensionais são elementos triangulares (com 3 ou 6 nós) e elementos retangulares (4,6,8, ou mais nós).

Animação 02: Elementos Bidimensionais

1.2.3. Elementos Tridimensionais:

A interface entre dois elementos das estruturas tridimensionais é feita através de superfícies.

Animação 03: Elementos Tridimensionais

O grau de confiabilidade da análise de uma estrutura depende muito da escolha, feita pelo analista, do tipo de elemento a ser utilizado para análise. Esta escolha depende do conhecimento do analista do comportamento dos vários elementos finitos existentes e, a partir daí, é feita a escolha do elemento mais adequado para aquela análise. O comportamento dos elementos é imposto a partir da especificação das propriedades dos elementos e estas propriedades são definidas para cada elemento por intermédio de relações matemáticas adequadas.

1.3. O MÉTODO DOS ELEMENTOS FINITOS

O **Método dos Elementos Finitos** é um método de aproximação numérica para cálculo de problemas de valor de contorno⁷. Resumo, sistemas contínuos são discretizados, ou seja, são subdivididos em subdomínios denominados **Elementos Finitos**, conectados entre si.

Para resolução de problemas utilizando o Método dos Elementos Finitos são montadas equações para cada elemento com um número finito de Parâmetros, que são as incógnitas do problema, também chamados de variáveis de estado, pois governam e descrevem o estado de equilíbrio da estrutura.

⁷ Link para definição de problema de valor de contorno. Esta definição está no final do item 1.3

1.3.1. Teoria básica para resolução de problemas através do mef em engenharia de estruturas (utilizando a formulação da rigidez)

Escreve-se o campo de deslocamento de cada elemento em função dos deslocamentos nodais, obtendo-se um sistema de equações algébricas que, quando resolvido, soluciona o problema.

O MEF foi inicialmente desenvolvido para aplicações na área de engenharia de estruturas. Atualmente, encontram-se aplicações do método nas áreas de transferência de calor, escoamento de fluidos, transporte de massa, potenciais eletromagnéticos, etc. Nesses casos, as variáveis serão: temperaturas, velocidades, etc.

Em resumo, um elemento finito é uma sub-região resultante da discretização de um meio contínuo. Este elemento apresenta tamanho finito (não é infinitesimal) e, usualmente apresenta uma geometria mais simples que a do meio contínuo. O MEF permite que um problema com graus de liberdade infinitos seja convertido em um problema com graus de liberdade finito, para simplificar o processo de solução.

O MEF aproxima uma quantidade contínua como o deslocamento no interior de um corpo através de um modelo discreto composto por um conjunto de funções contínuas definidas para cada elemento finito.

O MEF é um método orientado para o uso do computador e deve ser implementado em programas apropriados e em linguagens de programação.

Para a aplicação do método dos Elementos Finitos assume-se:

- Pequenos Elementos interconectados, denominados Elementos Finitos.
- Uma função de aproximação de deslocamento para cada elemento de acordo com o seu tipo.
- Cada elemento interconectado aos demais através de interfaces (nó, linha, superfície).

- Conhecidas as relações deslocamento x deformação e tensão x deformação dos materiais utilizados

Problema de Valor de Contorno

Problema de valor de contorno refere-se, tipicamente, a um problema governado por uma equação diferencial ordinária ou parcial, definido sobre um domínio, em cujo contorno determinadas condições relacionadas às variáveis associadas ao problema são assumidas.

Parâmetros

Na área de engenharia de estruturas, são realizadas análises de corpos submetidos a cargas ou outras influências como deformações iniciais, variações de temperatura. Cargas e as influências citadas causam deformações no corpo, acompanhadas de tensões internas e reações nos pontos de apoio. O objetivo principal do Método dos Elementos Finitos, na área de engenharia de estruturas, é calcular deslocamentos e tensões em componentes estruturais. As tensões são calculadas a partir dos deslocamentos. Assim, os deslocamentos são os parâmetros geralmente utilizados.

1.4. APLICAÇÕES DO MÉTODO DOS ELEMENTOS FINITOS

Na área de Engenharia de Estruturas:

- Análise de tensões (incluindo problemas de concentração de tensões associados com buracos ou mudanças na geometria de um corpo).
- Análise de vibração.
- Engenharia biomecânica (inclui análise de tensões em órgãos humanos como ossos, dentes, implantes dentários, coração, olho).

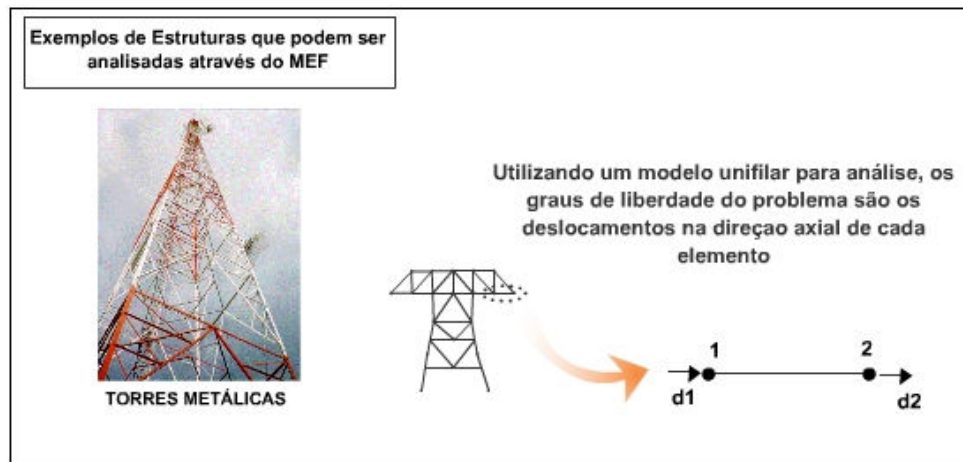


FIGURA 01 – Exemplo de Estruturas – Unifilar

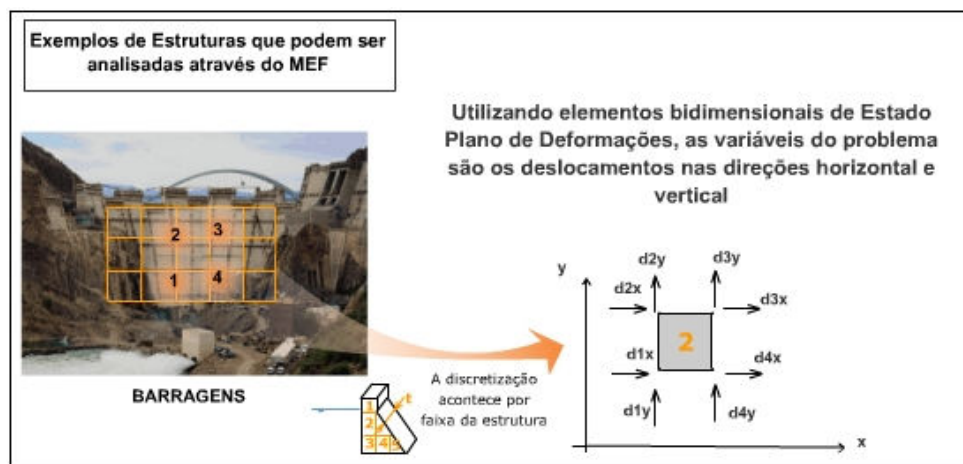


FIGURA 02 – Exemplos de estruturas - Bidimensional

Em áreas que não envolvem Engenharia de Estruturas:

- Transferência de calor.
- Fluxo de fluidos (incluindo a passagem de fluidos por poros de um corpo).
- Distribuição dos campos potenciais elétricos e magnéticos.

1.5. VANTAGENS DO MÉTODO DOS ELEMENTOS FINITOS

Como já mencionado anteriormente, a aproximação clássica para a análise de um sólido requer a determinação de funções de tensão e deslocamentos que satisfazem as equações diferenciais do equilíbrio, as relações tensão-deformação e as condições de compatibilidade de cada ponto do meio, incluindo as condições de contorno. Estas equações e condições são difíceis de serem plenamente atendidas devido as condições irregulares (carregamentos, não linearidade do material, geometria, etc) dessa forma, apenas algumas soluções são possíveis de serem determinadas pela formulação clássica. Além disso, estas equações recaem na resolução de séries infinitas que, para serem resolvidas são truncadas, resultando em aproximações não satisfatórias.

Dessa forma, as facilidades apresentadas pelo MEF são:

- Facilidade de modelagem de formas irregulares.
- Facilidade de manipulação dos carregamentos.
- Capacidade de modelagem de estruturas compostas por vários materiais, já que as equações dos elementos são obtidas individualmente.
- Permite modelar vários números e tipos de condições de contorno
- Permite variar o tamanho e o tipo dos elementos, sendo possível o uso de elementos menores ou de elementos diferentes em uma mesma estrutura onde for necessário.
- Permite que sejam feitas alterações ao modelo de elemento finito com facilidade.
- Permite a análise de efeitos dinâmicos.
- Permite a análise de comportamentos não lineares devido a grandes deformações e não linearidade dos materiais.

1.6. PROGRAMAS COMPUTACIONAIS PARA O MEF

De um modo geral, os programas para o MEF são executados para análise de uma estrutura completa, através da resolução de equações de equilíbrio utilizando as matrizes de rigidez dos elementos e determinando os deslocamentos nodais. A partir dos deslocamentos nodais são determinadas as tensões nos elementos e as reações de apoio.

Os passos gerais para análise estática de uma estrutura, utilizados nos programas são:

1.6.1. Entrada dos dados estruturais:

- Identificação do problema
- Parâmetros estruturais (módulo de elasticidade, coeficiente de Poisson)
- Coordenadas nodais
- Informações dos elementos (área, momento de inércia, momento polar)
- Restrições nodais

1.6.2. Entrada de dados dos carregamentos

- Parâmetros dos carregamentos
- Carregamentos nodais
- Carregamentos lineares
- Carregamentos de superfície
- Carregamentos de volume
- Deformações iniciais
- Deslocamentos dos apoios

1.6.3. Montagem da Matriz de Rigidez da estrutura

- Montagem da matriz de rigidez de cada elemento
- Montagem da Matriz de Rigidez Global
- Imposição das condições de contorno
- Cálculo do vetor de forças

1.6.4. Resolução das equações de equilíbrio nodal

1.6.5. Cálculo e Resultados

- Resultados dos deslocamentos nodais
- Resultado das tensões (ou esforços solicitantes) nos elementos
- Resultado das reações de apoio

Alguns programas comerciais que utilizam o MEF para análises estruturais:⁸

- Abaqus
- Algor
- Ansys
- Cosmos/M
- Stardyne
- Images – 3D
- MSC / Nastran
- SAP 2000

⁸ Link para o site desses programas

- GT STRUDL
- tq3
- robot
- Hypermesh
- Lusas

Os programas acima relacionados têm as seguintes capacidades de análises:

- Análises estáticas e dinâmicas
- Comportamentos lineares e não lineares dos materiais
- Diferentes tipos de carregamentos: concentrados, distribuídos, térmicos, deformações iniciais
- Geração automática de dados: geração de nós, elementos e restrições (geradores de malhas)
- Desenho da estrutura original e deformada, tensões e temperaturas.

1.7. ETAPAS DE ANÁLISE DO MEF

1.7.1. Discretização e Seleção dos Tipos de Elementos

Nesta etapa, é feita a divisão do sistema/estrutura em elementos finitos, escolhendo o tipo de elemento finito mais apropriado para o comportamento físico esperado do corpo. O número de nós e elementos e o tamanho dos elementos devem ser adequados para reproduzirem uma boa aproximação do sistema sem onerar a resolução do problema.

Elementos menores são utilizados quando os resultados modificam rapidamente com a posição, por exemplo, em regiões do corpo onde há mudanças bruscas na geometria. Elementos maiores são usados quando a variação dos resultados é mais suave.

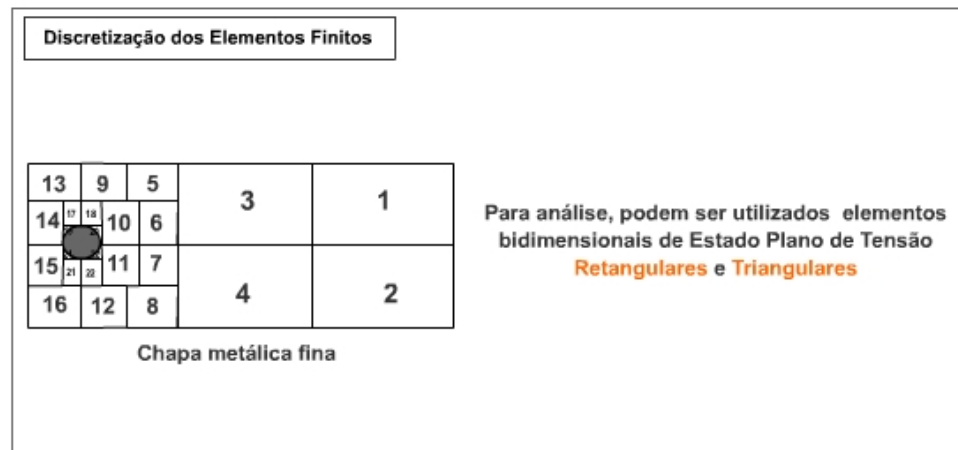


FIGURA 01 - Geometria

Pré-processadores gráficos são muito utilizados nos programas computacionais para auxílio na entrada dos dados.

1.7.2. Definição e Desenvolvimento da função de aproximação de deslocamentos

Nesta etapa uma função de aproximação ou de interpolação dos deslocamentos para cada elemento é definida e desenvolvida. Esta função é definida associada aos valores nodais dos deslocamentos e é empregada para descrever o campo de deslocamento no interior do elemento a partir de valores associados aos nós. São frequentemente utilizadas funções polinomiais lineares, quadráticas, cúbicas, dependendo do número de nós e quantidade de deslocamentos possíveis em cada nó (tipo de elemento). Uma função de aproximação de deslocamentos é utilizada para cada elemento do sistema, dentro do seu domínio, e é expressa em termos das variáveis nodais.

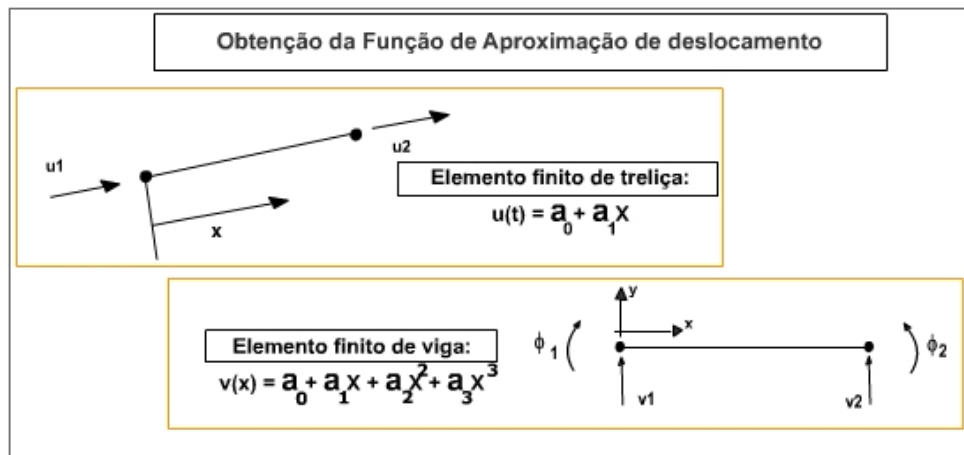


FIGURA 02 – Função de Aproximação

1.7.3. Definição das Relações Deformação x Deslocamento e Tensão x Deformação

Para definição dessas relações serão consideradas, neste curso, as seguintes hipóteses para cada elemento finito:

- Regime de pequenos deslocamentos e pequenas deformações
- Materiais em regime Elástico linear (Lei de Hooke)

Para o caso uni-axial:

$$\varepsilon = du/dx$$

$$\sigma = E \varepsilon$$

FIGURA 03 – Hipóteses Consideradas

1.7.4. Obtenção da Matriz de Rigidez, das Equações de Equilíbrio e dos Carregamentos Nodais

Para a obtenção da matriz de rigidez de um elemento, podem ser usados vários métodos. Estes métodos resultam em equações que podem ser escritas na forma matricial.

Animacão 01 – Matriz de Rigidez

Método do Equilíbrio Direto

A matriz de rigidez e as equações de equilíbrio que relacionam forças e deslocamentos são obtidas impondo condições de equilíbrio em um elemento. Este método é mais utilizado em elementos unidimensionais.

Métodos Envolvendo os Princípios de Energia

São dois os métodos mais utilizados:

- Princípio do trabalho virtual (PTV): utilizado para qualquer tipo de comportamento do material, ainda que não exista uma função potencial.
- Princípio da Energia Potencial Mínima: aplicado apenas em materiais para os quais existe uma função potencial (materiais elásticos por exemplo).

Estes métodos são utilizados para a obtenção da matriz de rigidez de elementos, assim como para a obtenção de carregamentos nodais equivalentes.

Métodos dos resíduos ponderados

O método dos resíduos ponderados mais conhecido é o método de Galerkin.

Os métodos de obtenção da matriz de rigidez abordados nesse curso serão o Método do Equilíbrio Direto⁹ e o Princípio dos Trabalhos Virtuais¹⁰.

1.7.5. Montagem da Matriz de Rigidez Global (Matriz de Rigidez do Modelo) e Imposição das Condições de Contorno

As várias equações dos elementos são unidas para formar a matriz de rigidez global do modelo. O método utilizado para a união das equações é o Método da Rigidez Direta, cujas bases são as condições de equilíbrio nodais e o princípio da continuidade ou compatibilidade que garante que a estrutura permanecerá unida após as deformações.

⁹ Link para Unidade 2 – item 3

¹⁰ Link para Unidade 2 – item 13

O sistema final global é: $F = K U$, semelhante ao sistema de equações para um elemento. Ao montar este sistema, nota-se que a matriz K é uma matriz singular, seu determinante é zero, ou seja, este sistema é indeterminado. Fisicamente, isto significa que a estrutura se deslocará como um todo, ou seja, acontecerá um deslocamento de corpo rígido. Para remover a singularidade dessa matriz é necessário impor condições de contorno do sistema. Assim, a estrutura permanecerá em seu lugar (em equilíbrio), sofrendo apenas deformações.

1.7.6. Solução do Sistema para os Graus de Liberdade Desconhecidos (Deslocamentos Generalizados)

Com a imposição das condições de contorno, o sistema de equações global pode ser resolvido. Métodos diversos para solução de sistemas de equações podem ser utilizados, como por exemplo o Método de Gauss ou Gauss-Seidel.

1.7.7. Solução das Tensões e Deformações no interior do Elemento

Grandezas internas dos elementos como tensão e deformação podem ser determinadas através das relações tensão x deformação e deslocamento x deformação atribuídas na etapa 3.

1.7.8. Interpretação dos Resultados

A interpretação dos resultados fica a cargo do analista. Pós processadores podem ser utilizados para facilitar a análise e interpretação dos resultados.

1.8. NOTAÇÃO

O uso da notação matricial auxilia na resolução de problemas através do Método dos Elementos Finitos (soluções longas, compostas de várias equações), além de ser uma excelente ferramenta para programação computacional, apresenta uma notação fácil e simples para representar e resolver sistemas de equações.

A notação utilizada fará distinção quanto ao sistema de coordenadas utilizadas (sistema global ou sistema local para cada elemento).

Para introdução da notação adotada no curso será utilizada como exemplo a treliça ao lado.

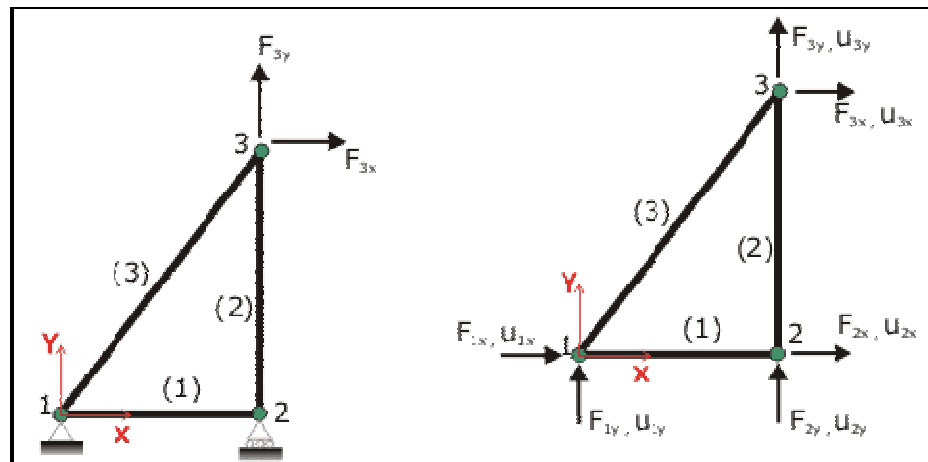


FIGURA 01 – Exemplo de Treliça

1.9. NOTAÇÃO – FORÇA

Para um problema representado no plano, as componentes das forças (F) nos nós da estrutura são representadas em um sistema global de coordenadas:

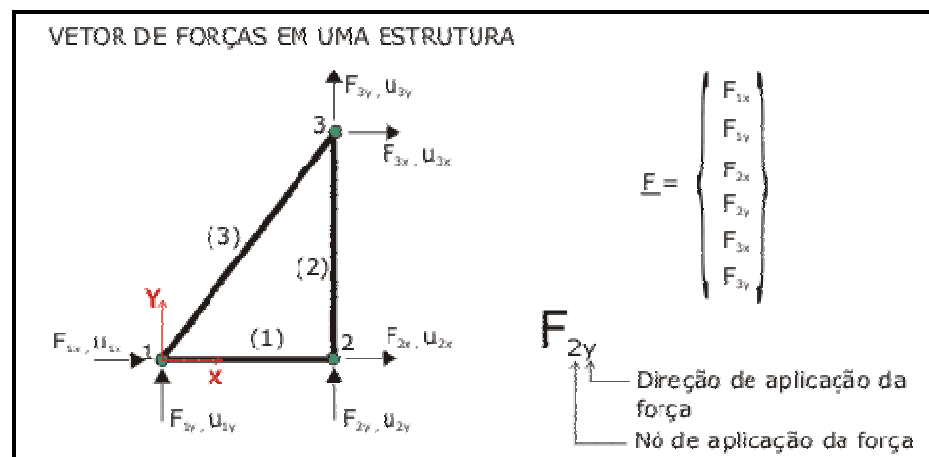


FIGURA 01: Forças de uma estrutura

As forças podem ser representadas para cada elemento no sistema global de coordenadas ou em um sistema local de coordenadas de cada elemento.

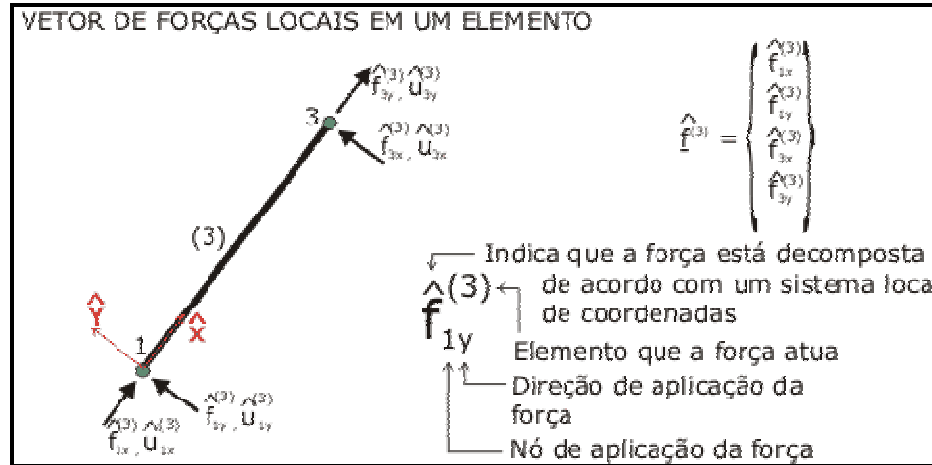


FIGURA 02: Forças locais

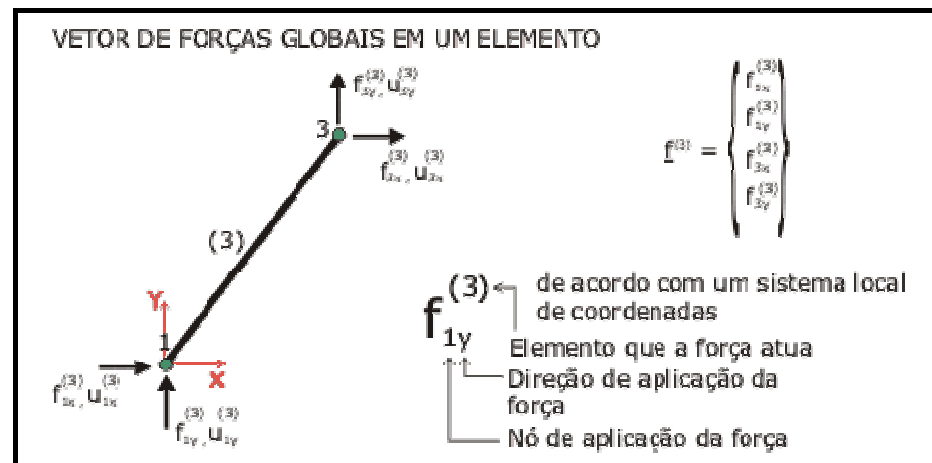


FIGURA 03: Forças Globais

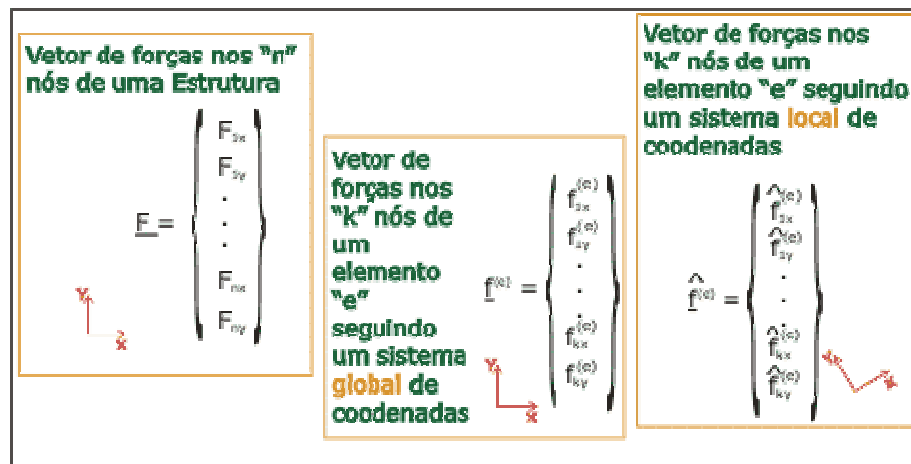


FIGURA 04: Notação Generalizada para Forças

1.10. NOTAÇÃO – DESLOCAMENTO

Para um problema representado no plano, as componentes dos deslocamentos (u) nos “n” nós da estrutura são representados em um sistema global de coordenadas:

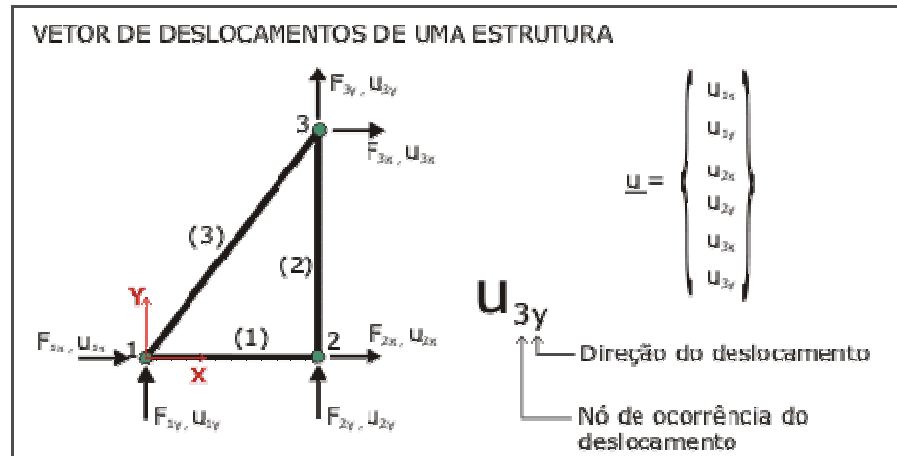


FIGURA 01: Deslocamentos da estrutura

Os deslocamentos podem ser representados no sistema local de cada elemento.

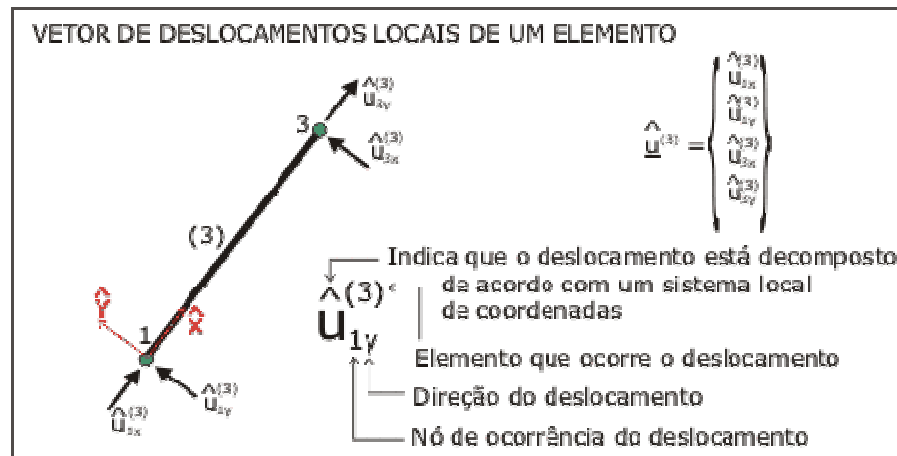


FIGURA 02: Deslocamentos locais

Em relação a deslocamentos, não faz sentido distinguir entre deslocamentos dos nós da estrutura e os deslocamentos internos dos nós dos elementos (como é feito no caso das forças), porque o deslocamento é o mesmo.

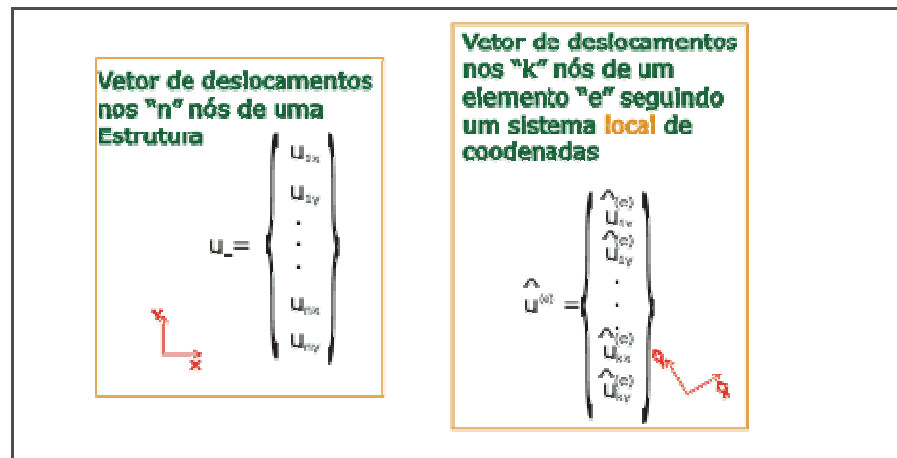


FIGURA 03: Notação Generalizada para Deslocamentos

1.11. NOTAÇÃO – MATRIZ DE RIGIDEZ

Para um problema representado no plano, a matriz de rigidez da estrutura apresentará a seguinte notação:

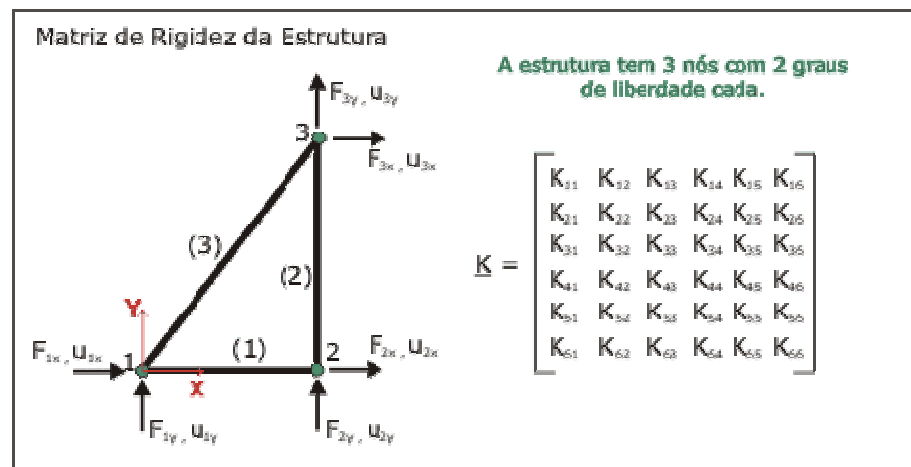


FIGURA 01: Matriz de Rigidez da Estrutura

A notação utilizada para representar a Matriz de Rigidez de um elemento em seu sistema local de coordenadas é:

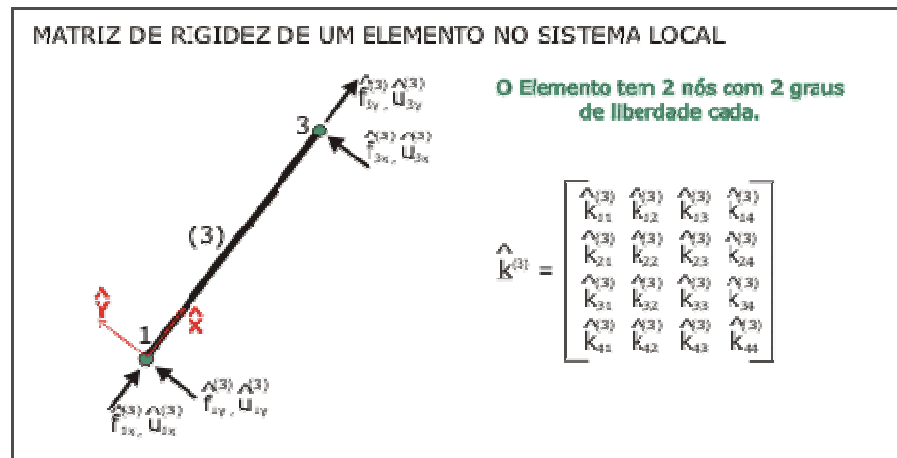


FIGURA 02: Matriz de Rigidez de um Elemento no sistema local

A notação utilizada para representar a Matriz de Rigidez de um elemento em seu sistema global de coordenadas é:

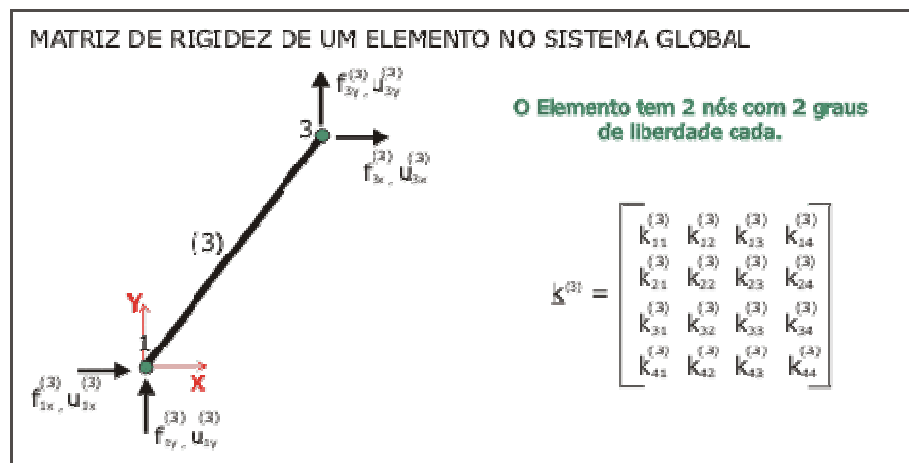


FIGURA 03: Matriz de Rigidez de um Elemento no sistema Global

- A matriz de rigidez de uma estrutura com “n” graus de liberdade será uma matriz quadrada nxn.
- A matriz de rigidez de um elemento com “n” graus de liberdade será uma matriz quadrada nxn.
- Será demonstrado que as matrizes colunas F e d se relacionam da seguinte forma: $F = K d$

Resumo:

- Letra minúscula: utilizada para representar as propriedades (força, deslocamento, rigidez) dos elementos;
- Letra maiúscula: utilizada para representar as propriedades (força, deslocamento, rigidez) da estrutura;
- Acento circunflexo: utilizado para representar as propriedades (força, deslocamento, rigidez) em sistema de coordenada local.

2

O MÉTODO DA RIGIDEZ DIRETA

2.1. DEFINIÇÃO DE MATRIZ DE RIGIDEZ

Neste curso será apresentada a formulação da Rigidez ou dos Deslocamentos para o Método dos Elementos finitos. A matriz associada a esta formulação é a Matriz de Rigidez que relaciona as incógnitas do problema (os deslocamentos) com as forças nodais.

2.1.1. Matriz de rigidez de um elemento

A base do MEF é a divisão de uma estrutura em sistemas discretos ou elementos, que serão analisados a partir do relacionamento entre forças e deslocamentos nodais. A relação entre força e deslocamento está ligada ao conceito de RIGIDEZ.

O conceito de rigidez é facilmente entendido pela análise de um elemento de mola.

Animação 01: Elemento de Mola

No método dos Elementos finitos, o conceito de Matriz de Rigidez assemelha-se ao conceito da **constante elástica da mola (k)** ou **rigidez da mola**. Porém, no caso da mola, o único deslocamento produzido pela força axial é o deslocamento axial, sendo necessário apenas uma constante para que possa ser estabelecida a relação entre a força aplicada em uma direção e o único deslocamento resultante na mesma direção, ou seja a **Rigidez axial**.

No método dos Elementos finitos, os deslocamentos existentes não são apenas axiais e, forças aplicadas não resultam em apenas um tipo de deslocamento. Assim, no MEF haverá a necessidade de vários tipos de rigidez, como a rigidez ao cisalhamento, rigidez à flexão, rigidez à torção, para relacionar as diversas forças com os respectivos deslocamentos. Veja o exemplo de um modelo simplificado de uma viga no plano:

Animação 02: Deslocamentos Possíveis

A maneira mais fácil de representar todas estas relações é através de sistemas de equações escritos na forma de matrizes. As componentes de forças que agem nos diversos nós da estrutura e os deslocamentos nodais correspondentes são representados em matrizes colunas e os diversos coeficientes de rigidez são agrupados na Matriz de Rigidez do Elemento. (como foi visto no Módulo 1 - Notação¹¹)

A Matriz de Rigidez do elemento relaciona as forças que agem nos nós dos elementos com os respectivos deslocamentos nodais através de um sistema de equações.

SISTEMA DE EQUAÇÕES

$$\begin{Bmatrix} f_{ix}^{(e)} \\ f_{iy}^{(e)} \\ \vdots \\ f_{ix}^{(e)} \\ f_{iy}^{(e)} \end{Bmatrix} = \begin{bmatrix} k_{11}^{(e)} & k_{12}^{(e)} & \dots & k_{1n}^{(e)} \\ k_{21}^{(e)} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k_{n1}^{(e)} & \cdot & \cdot & k_{nn}^{(e)} \end{bmatrix} \begin{Bmatrix} u_{1x} \\ u_{1y} \\ \vdots \\ u_{nx} \\ u_{ny} \end{Bmatrix}$$

↓

Vetor de forças nodais equivalentes

↓

Matriz de Rigidez: composta pelos coeficientes de rigidez

↓

Deslocamentos nodais do elemento

FIGURA 01: Sistemas de equações

¹¹ Link para Unidade 1 – item 8

Os diversos coeficientes da matriz de rigidez representam a força necessária para produzir em um nó, um deslocamento unitário enquanto todos os outros deslocamentos são nulos. Por exemplo, em um elemento de mola que apresenta $k = 50 \text{ Kgf/mm}$, o seu significado físico é que se deve aplicar uma força de 50 Kgf para obter um deslocamento axial de 1 mm. Assim, se é conhecida a força para provocar um deslocamento unitário, conhece-se a força para qualquer valor de deslocamento, já que estes apresentam uma relação linear entre si. Além disso, se houver deslocamento atuando simultaneamente, os efeitos de cada um dos deslocamentos aplicados isoladamente podem ser superpostos para se obter a força em cada nó, decorrente da ação conjunta de todos os deslocamentos no elemento.

Animação 03: Coeficientes da Matriz de Rigidez

É importante conhecer a Lei do Material (neste curso serão abordados materiais que obedecem a Lei de Hooke) antes da montagem da matriz de rigidez, pois, esta lei determinará como a ação imposta a um nó é transferida pelo elemento em seu interior até o outro nó.

A Matriz de Rigidez \hat{k} é uma matriz tal que $\hat{f} = \hat{k} \hat{d}$, onde \hat{k} referindo-se às coordenadas locais (\hat{x}, \hat{y}) , relaciona os deslocamentos nodais \hat{d} às forças locais \hat{f} de um único elemento.

2.1.2. Matriz de Rigidez da Estrutura

No MEF, a partir da análise de cada elemento, é possível a análise de toda a estrutura. A rigidez da estrutura inteira depende da rigidez de cada um de seus elementos.

Para análise de toda a estrutura e montagem da Matriz de Rigidez da estrutura, é importante que alguns aspectos comuns a todas as estruturas reticulares, bi e tridimensionais sejam observados:

Equilíbrio de Forças

Em uma estrutura em equilíbrio, cada um de seus componentes também está em equilíbrio, assim como para a estrutura discretizada, cada elemento tem que estar em equilíbrio.

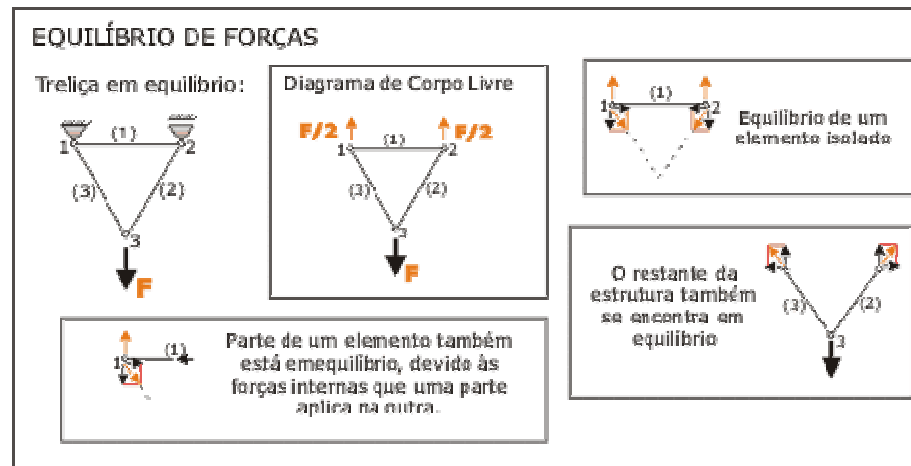


FIGURA 1: Equilíbrio de Forças

Compatibilidade de Deslocamentos

Os nós dos elementos permanecem conectados após a deformação da estrutura.

Animação 04: Deslocamentos

Comportamento do Material

Neste curso serão estudados materiais com pequenas deformações, que apresentam comportamento elástico-linear, ou seja: obedecem a Lei de Hooke.

Animação 05: Comportamento Elástico-Linear

A montagem da Matriz de Rigidez da Estrutura é feita considerando a Matriz de Rigidez de cada elemento e o modo como estes elementos estão arranjados.

A Matriz de Rigidez \mathbf{K} de uma estrutura em um sistema global de coordenadas (x,y) , para estruturas no plano, relaciona deslocamentos nodais \mathbf{u} às forças nodais \mathbf{F} de toda a estrutura

Quando a estrutura é formada por apenas um elemento, a matriz de rigidez do elemento é igual a matriz de rigidez da estrutura.

2.2. OBTENÇÃO DA MATRIZ RIGIDEZ DE UM ELEMENTO MOLA

A obtenção da matriz de rigidez de um elemento de mola será realizada para introduzir um procedimento padrão que poderá ser utilizado para todos os outros elementos. A diferença do elemento de mola se deve a sua simplicidade uma vez que transmite apenas forças axiais e sofre apenas deslocamentos axiais (graus de liberdade do elemento). Assim, o número de componentes de deslocamento envolvidos na montagem da matriz de rigidez será bem menor que em outros elementos finitos.

A obtenção da Matriz de Rigidez de um elemento de mola será realizada seguindo as 8 etapas de análise¹² enumeradas anteriormente no Módulo 1.

Animação: Obtenção da Matriz de Rigidez para um Elemento de Mola

2.3. MÉTODO DA RIGIDEZ DIRETA (EXEMPLO COM UM ELEMENTO MOLA)

A maioria dos sistemas práticos analisados pelo MEF é composto por vários elementos finitos. Assim, para possibilitar a análise global da estrutura é necessário que a matriz de rigidez do sistema global seja montada. A Rigidez da Estrutura inteira é obtida a partir da matriz de rigidez de cada um dos elementos.

O Método da Rigidez direta é um método prático para a montagem da matriz de rigidez da estrutura decorrente da aplicação das leis de equilíbrio e compatibilidade.

A seguir está apresentada, através de animações, a montagem da matriz de rigidez de um conjunto de molas a partir da aplicação das leis de equilíbrio e compatibilidade e através do Método da Rigidez Direta.

Animação: Montagem da Matriz de Rigidez para um Elemento de Mola através do Método da rigidez Direta

¹² Link para a Unidade 1 – item 7.

2.4. O MEF EM ELEMENTOS RETICULARES

Os elementos reticulares ou unidimensionais apresentam um procedimento mais simples de montagem das equações que regem o comportamento do sistema do que elementos para estruturas bi ou tridimensionais.

Sistemas (estruturas) constituídos por elementos reticulares já apresentam conexões discretas, ou seja, o sistema já está naturalmente discretizado.

Como dito anteriormente, estes elementos são conectados nos nós, podendo as conexões ser rígidas (como por exemplo em pórticos) ou articuladas (treliças), ponto onde ocorre as interações entre os elementos.

Para efeito da montagem do sistema de equações de equilíbrio, utilizando a linguagem matricial, as forças externas são aplicadas somente nos nós. Para que isso aconteça, é necessário utilizar o conceito de Carga Nodal Equivalente.

Carga Nodal Equivalente são cargas que atuando somente nos nós do modelo, produziriam efeitos equivalentes de cargas distribuídas atuando nos vãos dos elementos.

3

O ELEMENTO FINITO DE TRELIÇA

3.1. PROBLEMA DE MEIO CONTÍNUO

Uma treliça é formada por uma série de barras, elementos estruturais retos que apresentam o comprimento muito maior que as outras duas dimensões. Para formar uma treliça, estas barras são conectadas por meio de articulações, formando assim, uma estrutura reticulada.

Forças externas atuantes em uma treliça são aplicadas apenas nós da estrutura. Por este motivo e pelo fato dos elementos estarem conectados através de rótulas, as barras de treliça transmitem apenas forças axiais (tração ou compressão). Pelo fato das barras transmitirem apenas forças axiais, qualquer deslocamento transversal da barra é desprezado (este deslocamento está associado a um deslocamento de corpo rígido da barra, que não provoca esforço interno). Assim, uma barra de treliça contabiliza apenas a Rigidez axial do Membro estrutural.

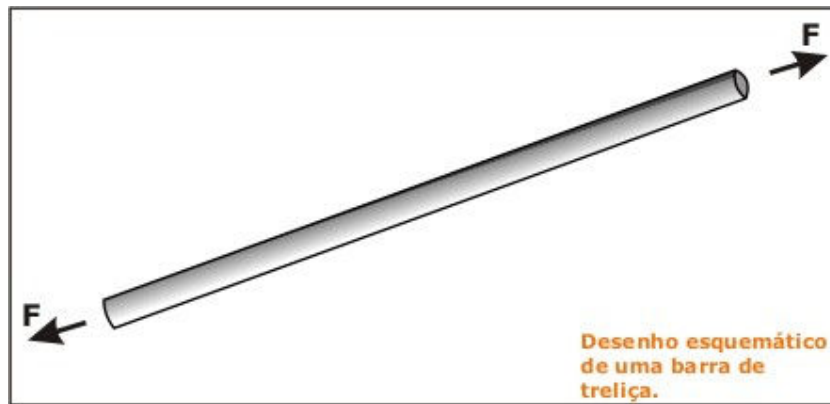


FIGURA 01: Barra de treliça



FIGURA 02 - Exemplo de estrutura Treliçada



FIGURA 03 - Exemplo de estrutura Treliçada



FIGURA 04 - Exemplo de estrutura Treliçada

3.2. MODELO MATEMÁTICO

Por se tratar de um elemento simples, a equação diferencial que determina a solução exata do modelo matemático para uma barra de treliça pode ser facilmente determinada. Para efeito de comparação com o Método dos Elementos Finitos, este processo será relembrado, utilizando-se uma barra simples.

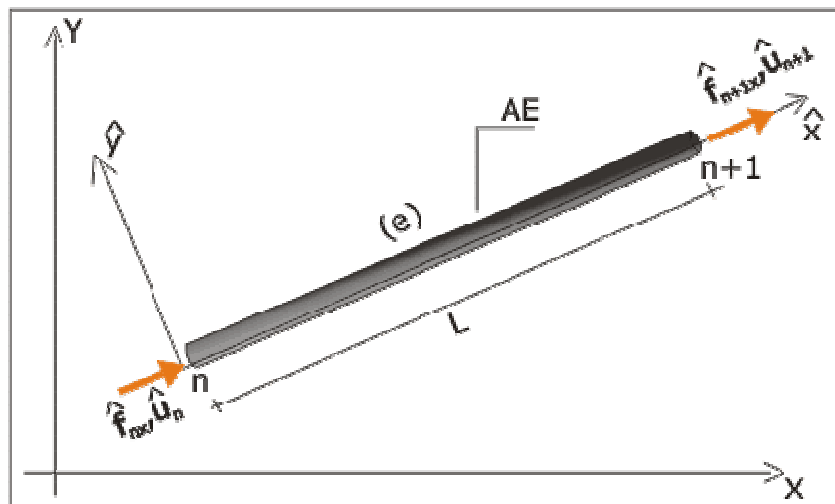


FIGURA 01: Barra de Treliça

Um dos conceitos necessários para a determinação da solução exata é a deformação específica, que está conceituada na animação seguinte.

Animação 01 :Deformação específica

Outras hipóteses simplificadoras são estabelecidas para a determinação da equação diferencial que resulta na solução exata do modelo matemático apresentado acima:

- O material apresenta comportamento elástico linear e o efeito de *poisson* desprezível, sendo possível a utilização da lei de Hooke para o cálculo das tensões na barra
- A barra apresenta apenas a tensão axial, definida como a força axial por unidade de área (A) da seção transversal.

$$\sigma = E \varepsilon(x)$$

σ é a tensão do material
 E é o módulo de elasticidade do material
 $\varepsilon(x)$ é uma função da deformação específica do material

FIGURA 02 – Lei de Hooke

$$\sigma = \frac{F(x)}{A}$$

FIGURA 03 - Tensão

Animação 02 – Obtenção da Equação Diferencial da Barra

A equação 3.1 abaixo é a equação diferencial que rege o problema matemático de barras esbeltas de seção reta constante.

$$\frac{d^2u(x)}{dx^2} = \frac{-f(x)}{EA} \quad (3.1)$$

Animação 03 :Resolução da Equação Diferencial da Barra

A equação 3.2 é a equação que determina os deslocamentos axiais ao longo de todo o domínio x da barra.

$$u(x) = \frac{-fx^2}{2EA} + \frac{fLx}{EA} \quad (3.2)$$

Conhecendo as relações entre tensão e deslocamento (3.3) e deformação e deslocamento (3.4), as equações para tensões (3.5) e deformações (3.6) podem ser determinadas.

$$\sigma = E \varepsilon(x) \quad \sigma = E \frac{du(x)}{dx} \quad (3.3)$$

$$\varepsilon(x) = \frac{du(x)}{dx} \quad (3.4)$$

$$\varepsilon(x) = -\frac{fx}{EA} + \frac{fL}{EA} \quad (3.5)$$

$$\sigma(x) = -\frac{fx}{A} + \frac{fL}{A} \quad (3.6)$$

Para efeito de comparações e análises, definimos:

- uf = unidade de força
- uc = unidade de comprimento

e fazendo: $f=1uf/uc$; $E=1uf/uc^2$; $A=1uc^2$ e $L=1uc$, são traçados os seguintes gráficos:

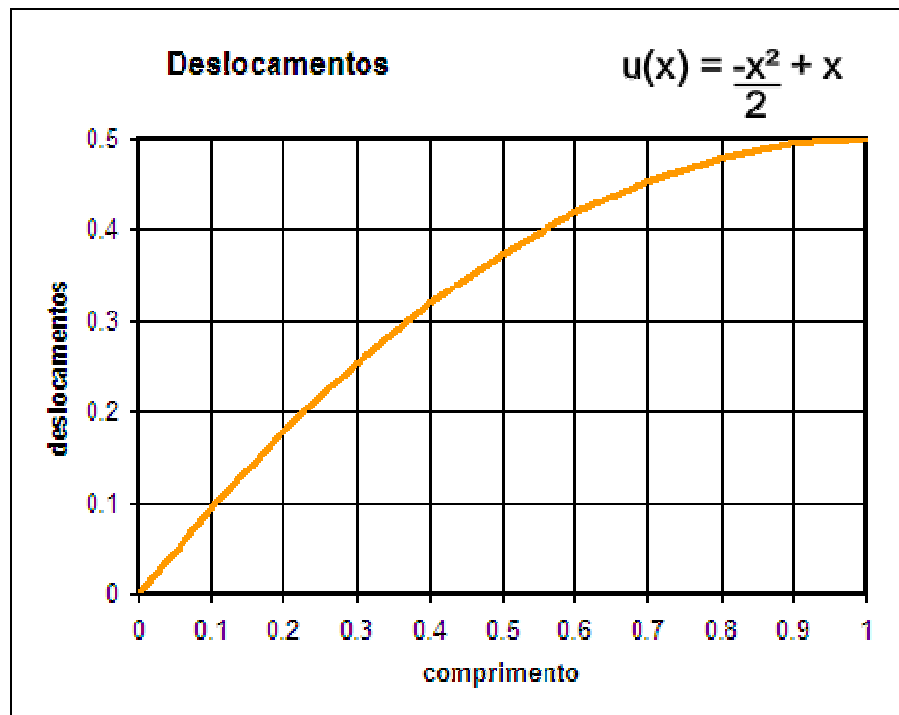


FIGURA 02: Gráfico dos deslocamentos

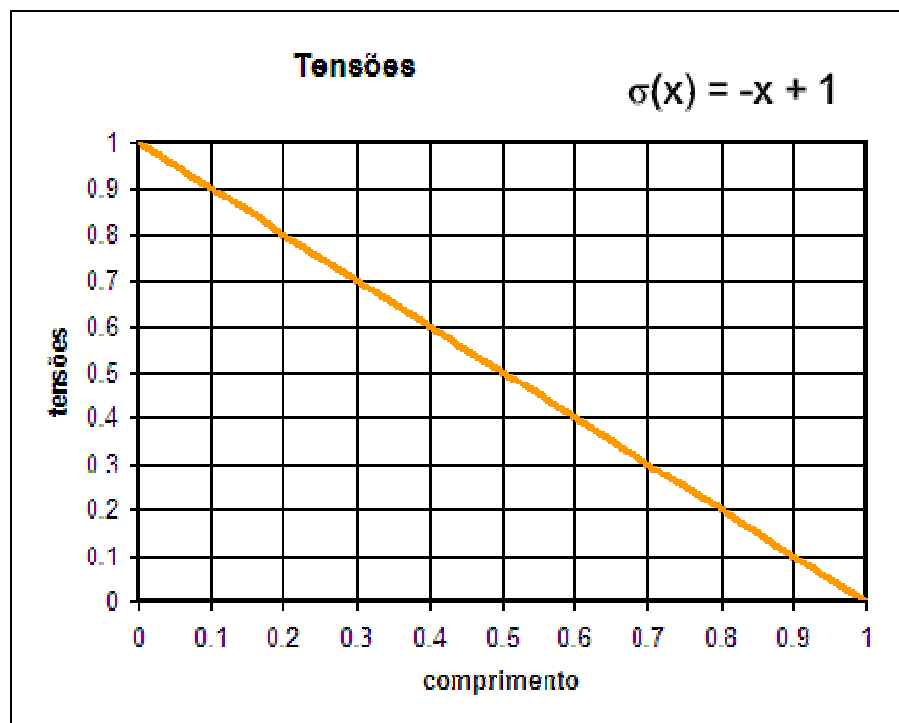


FIGURA 03: Gráfico das tensões

Observações:

- Em $x=0$ a tensão tem seu valor máximo.
- Em $x=1$ (ou $x=L$) a tensão é nula.

3.3. MODELO DISCRETO

Para análise através do Método dos Elementos Finitos, é necessário a discretização do sistema e para isso, foi desenvolvido um modelo discreto para os elementos de treliça. O elemento de treliça é muito parecido com o elemento de mola, pois, apresenta apenas rigidez axial.

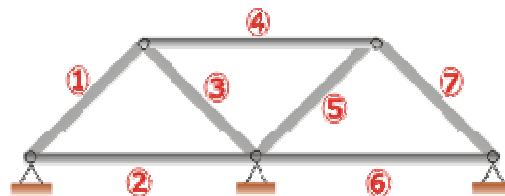


FIGURA 01: Exemplo de estrutura

Para se efetuar o estudo desta estrutura através do MEF, cada barra dessa treliça pode ser tratada como um elemento discreto, ou seja, como um elemento finito de treliça.

Após algumas considerações preliminares, as etapas de análise discutidas anteriormente serão utilizadas para descrever o elemento finito de treliça.

Conforme exposto anteriormente, o material das barras será considerado elástico linear (módulo de elasticidade E) de acordo com a Lei de Hooke. As barras serão consideradas de seção transversal constante de área A e comprimento inicial L .

As considerações expostas na definição de treliça serão mantidas:

- Presença de forças axiais apenas;
- Deslocamentos transversais nulos.

3.3.1. Etapas de Análise do Modelo Discreto

Discretização e Seleção do tipo de elemento

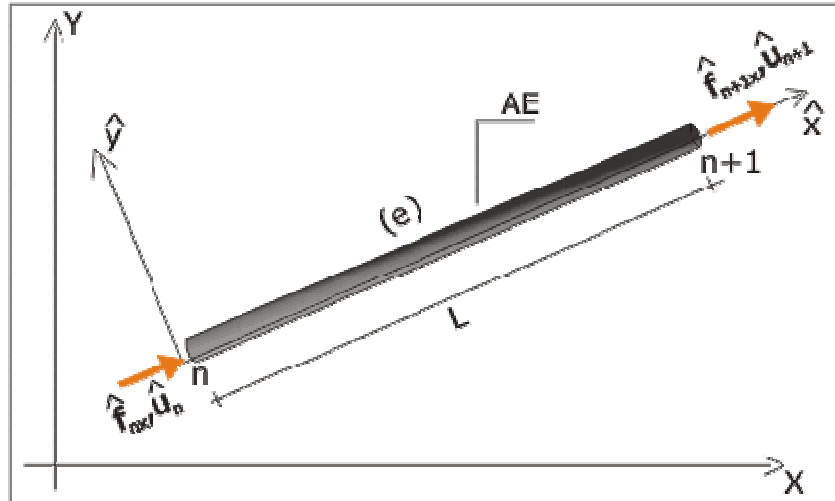


Figura 01: Elemento de Treliça Discretizado

Cada barra da estrutura será representada por um elemento discreto uniaxial. Um elemento genérico (e) será construído com dois nós (nós n e n+1), ou seja, o elemento terá dois graus de liberdade (já que o elemento de treliça apresenta apenas deformações axiais): u_n e u_{n+1} .

Para construção do elemento finito, foram introduzidos dois sistemas de coordenadas, um sistema global (X, Y) e um sistema local \hat{x}, \hat{y} .

Definição e Desenvolvimento da Função de Aproximação de Deslocamentos

Um tipo de função matemática simples e que pode ser utilizada e manuseada facilmente, é a função polinomial. Analogamente ao elemento de mola, conhecendo-se os dois deslocamentos nodais, uma função polinomial (3.7) linear pode ser determinada.

$$\hat{u}(x) = a_0 + a_1 x$$

$$\hat{u}(x) = \begin{bmatrix} 1 & x \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} \quad (3.7)$$

A equação 3.8 abaixo é a equação polinomial na forma matricial reduzida.

$$\hat{u}(x) = \underline{J}\alpha \quad (3.8)$$

Impondo as condições de contorno:

- Para $x=0$, $\hat{u}(0) = u_n$
- Para $x=L$, $\hat{u}(L) = u_{n+1}$

Substituindo estas condições na equação polinomial dos deslocamentos, é formado um sistema de equações, que pode ser escrito na forma matricial (3.9).

$$\begin{aligned} u_n &= a_0 + a_1(0) \\ u_{n+1} &= a_0 + a_1L \\ \begin{Bmatrix} u_n \\ u_{n+1} \end{Bmatrix} &= \begin{bmatrix} 1 & 0 \\ 1 & L \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} \\ \underline{u} &= \underline{G}\alpha \end{aligned} \quad (3.9)$$

As incógnitas do vetor coluna α podem ser calculadas resolvendo o sistema através da inversão da matriz \underline{G} (3.10):

$$\begin{aligned} \underline{G}^{-1}\underline{u} &= \underline{G}^{-1}\underline{G}\alpha \\ \alpha &= \underline{G}^{-1}\underline{u} \end{aligned} \quad (3.10)$$

Substituindo (3.9) em (3.10), a equação de deslocamentos fica (3.11):

$$\hat{u}(x) = \underline{J}\underline{G}^{-1}\underline{u} \quad (3.10)$$

A matriz \underline{G} é uma matriz que guarda as propriedades geométricas do elemento, enquanto a matriz \underline{J} guarda as coordenadas locais do elemento. Assim, $\underline{J}\underline{G}^{-1}$ forma a matriz das funções de forma do elemento. A matriz das funções de forma será denominada \underline{N} e a função de aproximação de deslocamentos pode ser escrita utilizando a matriz das funções de forma \underline{N} .

Animação 01 :Cálculo da Matriz das Funções de Forma

$$\underline{N} = \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} \quad (3.11)$$

$$\underline{N} = [N_1 \quad N_2]$$

Assim, a função de deslocamentos fica:

$$\underline{\hat{u}}(x) = \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} \begin{Bmatrix} u_n \\ u_{n+1} \end{Bmatrix}$$

$$\underline{\hat{u}}(x) = [N_1 \quad N_2] \begin{Bmatrix} u_n \\ u_{n+1} \end{Bmatrix} \quad (3.12)$$

$$\hat{u}(x) = N_1 u_n + N_2 u_{n+1}$$

Definição das relações Deformação x Deslocamento e Tensão x Deformação

As relações tensão x deformação e deformação x deslocamento obtidas no modelo matemático para treliça estão representadas respectivamente na equação (3.3).

Derivando a equação (3.12) e substituindo nas equações (3.3) teremos a equação (3.13) para deformações no elemento e a equação (3.14) para tensões no elemento.

$$\varepsilon(x) = \frac{u_{n+1} - u_n}{L} \quad (3.13)$$

$$\sigma(x) = \frac{E(u_{n+1} - u_n)}{L} \quad (3.14)$$

Obtenção da Matriz de Rigidez, das Equações de Equilíbrio e dos Carregamentos Nodais

Para obtenção da Matriz de Rigidez do Elemento de treliça, será repetido o mesmo processo do elemento de mola. Ou seja, a matriz de rigidez do elemento de treliça será obtido através das equações de equilíbrio do elemento.

Animacão 02 :Obtenção da Matriz de Rigidez do Elemento de Treliça

$$\hat{k} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (3.15)$$

Esta é a matriz de rigidez de um elemento de treliça no sistema local de coordenadas.

As demais etapas de análise serão apresentadas após a introdução de outros conceitos.

3.4. MÉTODO DA RIGIDEZ DIRETA

O método da rigidez direta pode ser aplicado para a montagem da matriz de rigidez da estrutura em qualquer tipo de elemento finito. Este método foi apresentado para o elemento de mola e repete para o elemento de treliça.

Animação 01 :O Método da Rigidez Direta

3.5. MATRIZ DE RIGIDEZ GLOBAL

Para a montagem da Matriz de Rigidez Global, ou Matriz de Rigidez da Estrutura, deve-se representar a matriz de rigidez de cada elemento no sistema global.

Um obstáculo encontrado é que o elemento representado no sistema global de coordenadas apresenta deslocamentos e forças nas direções X e Y, tendo quatro componentes de força e quatro componentes de deslocamento. A matriz de rigidez será então uma matriz 4x4. Já o elemento no sistema local de coordenadas apresenta apenas deslocamentos e forças axiais, apresentando 2 graus de liberdade, sendo sua matriz de rigidez local uma matriz 2x2.

Animação 01: Componentes de Força e Deslocamentos em Elementos de barra

O artifício utilizado para que as matrizes de rigidez no sistema local e global tenham a mesma dimensão é representar a barra no sistema local com dois componentes de força

e deslocamento em cada nó, mas sabendo que as forças em y e os deslocamentos em y no sistema local são nulos.

Animação 02: Artificio

O outro obstáculo existente é a transformação da matriz de rigidez para o sistema global de coordenadas, que será resolvido no item seguinte.

3.6. TRANSFORMAÇÃO DE VETORES EM DUAS DIMENSÕES

Para a montagem da matriz de rigidez global do sistema, as matrizes de rigidez de cada elemento devem estar representadas em um mesmo sistema de coordenadas. Assim, é necessário a transformação dos vetores de força e deslocamentos e da matriz de rigidez do elemento (que inicialmente é sempre definida no sistema local de coordenadas) para um sistema de coordenadas global, comum a todos os elementos da estrutura.

Animação 01: Transformação de Vetores

A relação de transformação pode ser usada para os deslocamentos e forças nodais de um elemento está na equação (3.16) abaixo.

$$\begin{cases} \hat{u}_{ix} \\ \hat{u}_{iy} \end{cases} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{cases} u_{ix} \\ u_{iy} \end{cases} \quad \text{Onde } \mathbf{c} \text{ é o cosseno do ângulo} \\ \text{formado entre os eixos global e o} \\ \text{local e } \mathbf{s} \text{ é o seno do ângulo formado} \\ \text{entre os eixos local e global} \end{cases} \quad (3.15)$$

$$\begin{cases} \hat{f}_{ix} \\ \hat{f}_{iy} \end{cases} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{cases} f_{ix} \\ f_{iy} \end{cases}$$

Para determinar as forças no sistema de coordenadas global a partir das forças no sistema de coordenadas local, a relação anterior pode ser invertida.

$$\begin{cases} f_{ix} \\ f_{iy} \end{cases} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{-1} \begin{cases} \hat{f}_{ix} \\ \hat{f}_{iy} \end{cases} \quad (3.16)$$

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{-1} = \frac{1}{c^2 + s^2} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{-T} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

A matriz formada pelos senos e cossenos será chamada de matriz transformação [T].

A Matriz inversa da Matriz de transformação é igual a sua transposta

O mesmo procedimento pode ser aplicado para determinação dos deslocamentos no sistema global de coordenadas.

$$\begin{Bmatrix} u_{ix} \\ u_{iy} \end{Bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{Bmatrix} \hat{u}_{ix} \\ \hat{u}_{iy} \end{Bmatrix} \quad (3.17)$$

Em resumo:

$$\begin{aligned} \underline{\hat{u}} &= [T]\underline{u} \\ \underline{\hat{f}} &= [T]\underline{f} \\ \therefore T &= \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \end{aligned} \quad (3.17)$$

3.7. TRANSFORMAÇÃO DA MATRIZ DE RIGIDEZ EM DUAS DIMENSÕES

Para a transformação da Matriz de Rigidez de um elemento para o sistema global de coordenadas é necessário que algumas operações matriciais sejam feitas.

A transformação do vetor força e deslocamento podem ser escritos, utilizando 4 graus de liberdade em um plano x, y.

$$\begin{cases} \hat{u}_{1x} \\ \hat{u}_{1y} \\ \hat{u}_{2x} \\ \hat{u}_{2y} \end{cases} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{cases} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \end{cases}$$

Onde \mathbf{C} é o coseno do ângulo formado entre os eixos global e o local e \mathbf{S} é o seno do ângulo formado entre os eixos local e global

$$\begin{cases} \hat{f}_{1x} \\ \hat{f}_{1y} \\ \hat{f}_{2x} \\ \hat{f}_{2y} \end{cases} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{cases} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \end{cases} \quad (3.18)$$

Os vetores locais $\hat{\underline{f}}$ e $\hat{\underline{u}}$ do sistema das equações de equilíbrio $\hat{\underline{f}} = \hat{\underline{k}}\hat{\underline{u}}$ podem ser transformados em vetores globais:

$$\begin{aligned} \underline{T}\hat{\underline{f}} &= \hat{\underline{k}}\underline{T}\hat{\underline{u}} \\ \underline{T}^{-1}\underline{T}\hat{\underline{f}} &= \underline{T}^{-1}\hat{\underline{k}}\underline{T}\hat{\underline{u}} \therefore \underline{T}^{-1}\underline{T} = 1 \\ \underline{f} &= \underline{T}^{-1}\hat{\underline{k}}\underline{T}\hat{\underline{u}} \end{aligned} \quad (3.19)$$

A equação resultante está na forma: $\underline{f} = \underline{k}\underline{u}$, fazendo concluir que a matriz de rigidez para o elemento de treliça no sistema global de coordenadas é:

$$\underline{K} = \underline{T}^{-1}\hat{\underline{k}}\underline{T} \text{ já que } \underline{T}^{-1} = \underline{T}^T \quad (3.20)$$

Efetuada as multiplicações acima, tem-se a matriz de rigidez de um elemento de treliça no sistema global de coordenadas.

$$\underline{K} = \frac{EA}{L} \begin{array}{cc|cc} \boxed{\text{Nó 1}} & & \boxed{\text{Nó 2}} & \\ \hline c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ \hline -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{array} \begin{cases} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \end{cases} \begin{cases} \boxed{\text{Nó 1}} \\ \boxed{\text{Nó 2}} \end{cases} \quad (3.21)$$

3.8. CONDIÇÕES DE CONTORNO

A matriz de rigidez da estrutura é uma Matriz Simétrica e Singular, ou seja, o seu determinante é nulo, e sua inversa não pode ser obtida. Em termos físicos, isso significa que o sistema completo formado(3.22) pelo equilíbrio de forças, representa a estrutura sem nenhum vínculo, resultando em movimento de corpo rígido.

$$\begin{Bmatrix} \hat{f}_{n,x} \\ \hat{f}_{n+1,x} \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_n \\ u_{n+1} \end{Bmatrix} \quad (3.22)$$

É necessário que sejam estabelecidas condições de contorno para que o sistema possa ser resolvido, fazendo com que a estrutura fique em uma condição de equilíbrio estável.

As condições de contorno serão os deslocamentos nodais prescritos, condições de restrição impostas através dos apoios e as forças externas aplicadas.

Em um nó, quando o deslocamento é prescrito, a força é uma incógnita do sistema e vice-versa. Assim, para resolver o sistema, particiona-se a equação matricial de equilíbrio da estrutura, separando os deslocamentos prescritos dos deslocamentos incógnitas. O objetivo da partição do sistema é resolvê-lo por partes. Primeiro é resolvido o conjunto de equações das forças prescritas, encontrando os deslocamentos incógnitas. Posteriormente, o conjunto de equações composto pelas forças incógnitas é resolvido.

Animação 01: Condições de contorno

3.9. SOLUÇÃO TRELIÇA PLANA – EXEMPLO

Esta solução será estudada considerando as etapas de análise do MEF ¹³apresentadas:

¹³ Link para a Unidade 01 – item 7 Etapas de análise do Método dos Elementos Finitos

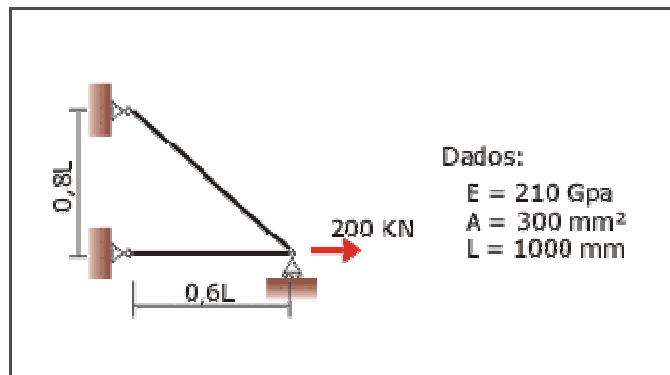


FIGURA 01 – Problema Proposto

3.9.1. Discretização e Seleção dos tipos de elementos

A estrutura apresentada é uma estrutura treliçada. Cada elemento da estrutura será estudado como um elemento finito de treliça.

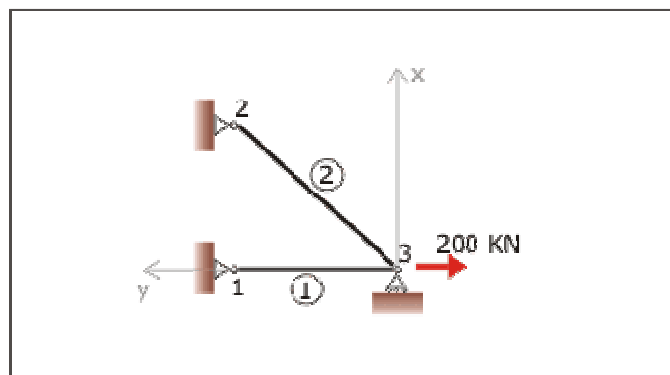


FIGURA 02 – Elementos da Treliça

3.9.2. Definição e Desenvolvimento da Função de Aproximação de Deslocamentos

Esta função foi desenvolvida no item Modelo Discreto¹⁴:

$$\hat{u}(x) = \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} \begin{Bmatrix} u_{1x} \\ u_{2x} \end{Bmatrix} \text{ para cada elemento} \quad (3.23)$$

¹⁴ Link para item 3 da Unidade 03 – Modelo discreto

Para a obtenção da função de deslocamentos, é necessário encontrar os deslocamentos nodais dos elementos (u_{1x} e u_{2x}). Para isso, a expressão $\hat{f} = \hat{k}\hat{u}$ será utilizada para cada elemento.

3.9.3. Definição das relações Deformação x Deslocamento e Tensão x Deformação

Estas relações também foram obtidas no item Modelo Discreto¹³:

$$\varepsilon(x) = \frac{u_{n+1} - u_n}{L} \quad (3.13)$$

$$\sigma(x) = \frac{E(u_{n+1} - u_n)}{L} \quad (3.14)$$

3.9.4. Obtenção da Matriz de Rigidez, das Equações de Equilíbrio e dos Carregamentos Nodais

Para obtenção da Matriz de Rigidez do Elemento de treliça, será repetido o mesmo processo do elemento de mola. Ou seja, a matriz de rigidez do elemento de treliça será obtida através das equações de equilíbrio do elemento.

Sejam os elementos de treliça:

Elemento 1:

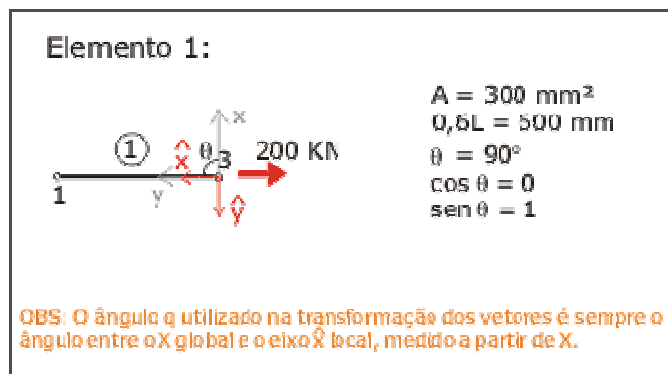
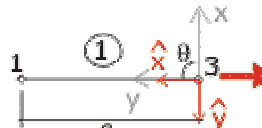


FIGURA 03 – Elemento 01

Matriz de Rigidez do Elemento 1 no sistema local e global de coordenadas:

$$\hat{K}^{(1)} = \frac{EA}{L} \begin{array}{c|c} \text{Nó 3} & \text{Nó 1} \\ \hline \begin{array}{cc} u_{3x} & u_{3y} \\ 1 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \end{array} & \begin{array}{cc} u_{1x} & u_{1y} \\ -1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{array} \\ \hline \begin{array}{c} u_{3x} \\ u_{3y} \\ u_{1x} \\ u_{1y} \end{array} & \begin{array}{c} \text{Nó 3} \\ \text{Nó 1} \\ \text{Nó 1} \\ \text{Nó 3} \end{array} \end{array} \quad K^{(1)} = \frac{EA}{L} \begin{array}{c|c} \text{Nó 3} & \text{Nó 1} \\ \hline \begin{array}{cc} u_{3x} & u_{3y} \\ C^2 & CS \\ CS & S^2 \\ -C^2 & -CS \\ -CS & -S^2 \end{array} & \begin{array}{cc} u_{1x} & u_{1y} \\ -C^2 & -CS \\ -CS & -S^2 \\ C^2 & CS \\ CS & S^2 \end{array} \\ \hline \begin{array}{c} u_{3x} \\ u_{3y} \\ u_{1x} \\ u_{1y} \end{array} & \begin{array}{c} \text{Nó 3} \\ \text{Nó 1} \\ \text{Nó 1} \\ \text{Nó 3} \end{array} \end{array}$$



$E = 210 \text{ Gpa}$
 $A = 300 \text{ mm}^2$
 $c = 0,6L = 600 \text{ mm}$
 $\theta = 90^\circ$
 $\cos\theta = 0$
 $\text{sen}\theta = 1$

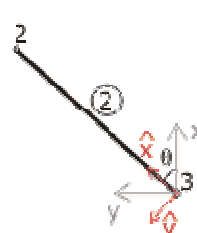
$$\frac{EA}{L} = \frac{210 \times 10^9 \text{ KN/m}^2 \times 0,0003 \text{ m}^2}{0,6 \text{ m}} = 105 \times 10^3 \text{ KN/m}$$

$$\hat{K}^{(1)} = 105 \times 10^3 \text{ KN/m} \begin{array}{c|c} \text{Nó 3} & \text{Nó 1} \\ \hline \begin{array}{cc} u_{3x} & u_{3y} \\ 1 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \end{array} & \begin{array}{cc} u_{1x} & u_{1y} \\ -1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{array} \\ \hline \begin{array}{c} u_{3x} \\ u_{3y} \\ u_{1x} \\ u_{1y} \end{array} & \begin{array}{c} \text{Nó 3} \\ \text{Nó 1} \\ \text{Nó 1} \\ \text{Nó 3} \end{array} \end{array}$$

$$K^{(2)} = 105 \times 10^3 \text{ KN/m} \begin{array}{c|c} \text{Nó 3} & \text{Nó 1} \\ \hline \begin{array}{cc} u_{3x} & u_{3y} \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & -1 \end{array} & \begin{array}{cc} u_{1x} & u_{1y} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array} \\ \hline \begin{array}{c} u_{3x} \\ u_{3y} \\ u_{1x} \\ u_{1y} \end{array} & \begin{array}{c} \text{Nó 3} \\ \text{Nó 1} \\ \text{Nó 1} \\ \text{Nó 3} \end{array} \end{array}$$

Elemento 2:

Elemento 2:



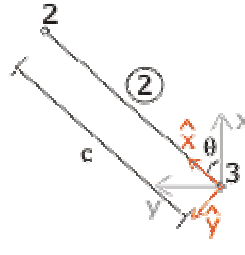
OBS: O ângulo utilizado na transformação dos vetores é sempre o ângulo entre o X global e o eixo x' local, medido a partir de X.

$A = 300 \text{ mm}^2$
 $L = 1000 \text{ mm}$
 $\cos\theta = 0,8$
 $\text{sen}\theta = 0,6$

FIGURA 04: Elemento 2

Matriz de Rigidez do Elemento 2 no sistema local e global de coordenadas:

$$\hat{K}^{(2)} = \frac{EA}{L} \begin{array}{c|c} \text{Nó 3} & \text{Nó 2} \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline \text{Nó 3} & \text{Nó 2} & \text{Nó 3} & \text{Nó 2} \end{array} \quad K^{(2)} = \frac{EA}{L} \begin{array}{c|c} \text{Nó 3} & \text{Nó 2} \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline C^2 & CS & -C^2 & -CS \\ CS & S^2 & -CS & -S^2 \\ \hline -C^2 & -CS & C^2 & CS \\ -CS & -S^2 & CS & S^2 \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline \text{Nó 3} & \text{Nó 2} & \text{Nó 3} & \text{Nó 2} \end{array}$$



$E = 210 \text{ Gpa}$
 $A = 300 \text{ mm}^2$
 $c = L = 1000 \text{ mm}$
 $\cos \theta = 0,8$
 $\text{sen} \theta = 0,6$

$$\frac{EA}{L} = \frac{210 \times 10^9 \text{ KN/m}^2 \times 0,0003 \text{ m}^2}{1 \text{ m}} = 63 \times 10^3 \text{ KN/m}$$

$$\hat{K}^{(2)} = 63 \times 10^3 \text{ KN/m} \begin{array}{c|c} \text{Nó 3} & \text{Nó 2} \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline \text{Nó 3} & \text{Nó 2} & \text{Nó 3} & \text{Nó 2} \end{array}$$

$$K^{(2)} = 63 \times 10^3 \text{ KN/m} \begin{array}{c|c} \text{Nó 3} & \text{Nó 2} \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline 0,64 & 0,48 & -0,64 & -0,48 \\ 0,48 & 0,36 & -0,48 & -0,36 \\ \hline -0,64 & -0,48 & 0,64 & 0,48 \\ -0,48 & -0,36 & 0,48 & 0,36 \\ \hline u_{3x} & u_{3y} & u_{1x} & u_{1y} \\ \hline \text{Nó 3} & \text{Nó 2} & \text{Nó 3} & \text{Nó 2} \end{array}$$

O ângulo utilizado na transformação dos vetores é sempre o ângulo entre o eixo X global e o eixo x local, medidos a partir de X global.

Pela determinação dos eixos locais do elemento define-se a orientação dos nós do elemento, ou seja, o nó final e o nó inicial. Esta orientação deve ser observada na matriz de rigidez do elemento e ao montar a matriz de rigidez da estrutura.

Vetor de Carregamentos Nodais da Estrutura:

O vetor dos carregamentos nodais da estrutura é:

$$\underline{F} = \begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ F_{3x} \\ F_{3y} \end{Bmatrix} = \begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ 0 \\ -200\text{KN} \end{Bmatrix} \quad (3.24)$$

Animação 1: Vetor Carregamento

3.9.5. Montagem da Matriz de rigidez da estrutura e Imposição das Condições de Contorno

A matriz de rigidez da estrutura será montada utilizando o Método da Rigidez Direta.

O sistema apresenta três nós e dois graus de liberdade por nó. Ao todo, o sistema tem 6 graus de liberdade. Então, a sua matriz de rigidez é 6x6.

Animação 2: Matriz rigidez da estrutura

$$\underline{K} = \begin{array}{c} \begin{array}{cc|cc|cc} \text{Nó 1} & & \text{Nó 2} & & \text{Nó 3} & \\ \hline u_{1x} & u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{3y} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & -105 \\ \hline 0 & 0 & 40,32 & 30,24 & -40,32 & -30,24 \\ 0 & 0 & 30,24 & 22,68 & -30,24 & -22,68 \\ \hline 0 & 0 & -40,32 & -30,24 & 40,32 & 30,24 \\ 0 & -105 & -30,24 & -22,68 & 30,24 & 127,68 \end{array} \end{array} \begin{array}{l} u_{1x} \\ u_{1y} \\ \hline u_{2x} \\ u_{2y} \\ \hline u_{3x} \\ u_{3y} \end{array} \begin{array}{l} \text{Nó 1} \\ \text{Nó 2} \\ \hline \text{Nó 3} \end{array}$$

O sistema de equações da estrutura é:

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ F_{3x} \\ F_{3y} \end{Bmatrix} = \begin{array}{c} \begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & -105 \\ \hline 0 & 0 & 40,32 & 30,24 & -40,32 & -30,24 \\ 0 & 0 & 30,24 & 22,68 & -30,24 & -22,68 \\ \hline 0 & 0 & -40,32 & -30,24 & 40,32 & 30,24 \\ 0 & -105 & -30,24 & -22,68 & 30,24 & 127,68 \end{array} \end{array} \begin{Bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \end{Bmatrix}$$

Este sistema pode ser particionado da seguinte forma:

$$\begin{array}{c}
 \left\{ \begin{array}{l} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ F_{3x} \\ -200 \end{array} \right\} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & -105 \\ \hline 0 & 0 & 40,32 & 30,24 & -40,32 & -30,24 \\ 0 & 0 & 30,24 & 22,68 & -30,24 & -22,68 \\ \hline 0 & 0 & -40,32 & -30,24 & 40,32 & 30,24 \\ \hline 0 & -105 & -30,24 & -22,68 & 30,24 & 127,68 \end{array} \right] \left\{ \begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{3y} \end{array} \right\}
 \end{array}$$

$$\begin{array}{c}
 \left\{ \begin{array}{l} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ F_{3y} \\ -200 \end{array} \right\} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & -105 \\ \hline 0 & 0 & 40,32 & 30,24 & -40,32 & -30,24 \\ 0 & 0 & 30,24 & 22,68 & -30,24 & -22,68 \\ \hline 0 & 0 & -40,32 & -30,24 & 40,32 & 30,24 \\ \hline 0 & -105 & -30,24 & -22,68 & 30,24 & 127,68 \end{array} \right] \left\{ \begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{3y} \end{array} \right\}
 \end{array}$$

The diagram shows the partitioning of the system matrix and vectors. The top part shows the full system. The bottom part shows the same system with partitions:

- K_{pp} (red oval) covers the top-left 5x5 submatrix.
- K_{pt} (blue oval) covers the top-right 5x1 submatrix.
- K_{tp} (blue oval) covers the bottom-left 1x5 submatrix.
- K_{tt} (red oval) covers the bottom-right 1x1 submatrix.
- F_p (red oval) covers the top 5 elements of the force vector.
- F_t (blue oval) covers the bottom 1 element of the force vector.
- u_p (blue oval) covers the top 5 elements of the displacement vector.
- u_t (red oval) covers the bottom 1 element of the displacement vector.

FIGURA 05: Sistema de Equações Particionado

Deslocamentos prescritos:

$$u_{1x} = u_{1y} = 0$$

$$u_{2x} = u_{2y} = 0$$

$$u_{3x} = 0$$

Deslocamentos incógnitas:

$$u_{3y}$$

Forças prescritas:

$$F_{3y} = -200\text{KN}$$

Forças incógnitas:

$$F_{1x} \quad F_{1y} \quad F_{2x} \quad F_{2y} \quad F_{3x}$$

3.9.6. Solução do Sistema para os graus de liberdade desconhecidos

$$\begin{aligned}
 \mathbf{F}_p &= \mathbf{K}_{yp} \mathbf{U}_p + \mathbf{K}_{yt} \mathbf{U}_t \\
 \mathbf{F}_t &= \mathbf{K}_{tp} \mathbf{U}_p + \mathbf{K}_{tt} \mathbf{U}_t
 \end{aligned}$$

\mathbf{F}_p vetor de forças prescritas
 \mathbf{F}_t vetor de forças incógnitas
 \mathbf{U}_p vetor de deslocamentos prescritos
 \mathbf{U}_t vetor de deslocamentos incógnitas

FIGURA 06: Esquema - Sistema de Equações Particionado

O sistema de equações é então resolvido. Primeiramente, resolve a parte do sistema que apresenta o vetor dos deslocamentos desconhecidos (incógnitas) e o vetor de forças incógnitas.

Neste caso, o sistema que apresenta os deslocamentos desconhecidos é composto por uma equação:

$$\begin{aligned}
 -200 &= 127,68u_{3,y} \\
 u_{3,y} &= -1,566 \times 10^{-3} \text{ m}
 \end{aligned} \tag{3.25}$$

Após a determinação dos deslocamentos desconhecidos, é mais fácil a obtenção das forças externas desconhecidas, utilizando a segunda parte do sistema de equações particionado: o vetor de forças incógnitas e os deslocamentos determinados.

$$\begin{Bmatrix} F_{1z} \\ F_{1y} \\ F_{2z} \\ F_{2y} \\ F_{3z} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & -105 \\ 0 & 0 & 40,32 & 30,24 & -40,32 & -30,24 \\ 0 & 0 & 30,24 & 22,68 & -30,24 & -22,68 \\ 0 & 0 & -40,32 & -30,24 & 40,32 & 30,24 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1,566 \end{Bmatrix}$$

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 105 & 0 & 0 & 0 & -105 \\ 0 & 0 & 40,32 & 30,24 & -40,32 & -30,24 \\ 0 & 0 & 30,24 & 22,68 & -30,24 & -22,68 \\ 0 & 0 & -40,32 & -30,24 & 40,32 & 30,24 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1,566 \end{Bmatrix}$$

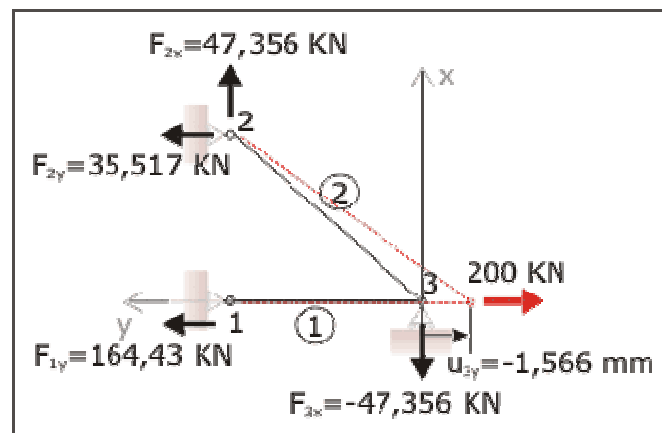


FIGURA 07: Solução do Sistema Global

3.10. SOLUÇÃO TRELIÇA PLANA – SOLUÇÃO DE CADA ELEMENTO

As forças e os deslocamentos de cada elementos no sistema local de coordenadas podem ser obtidos de duas maneiras:

- Aplicando a equação $\underline{f} = \underline{k}\underline{u}$ para cada elemento (no sistema de coordenadas global) e transformando o vetor de forças \underline{f} para o sistema de coordenadas local, utilizando a matriz de transformação \underline{T} .
- Transformando o vetor de deslocamentos de cada elemento no sistema global (\underline{u}) de coordenadas para o sistema local ($\hat{\underline{u}}$) e aplicando a equação $\hat{\underline{f}} = \hat{\underline{k}}\hat{\underline{u}}$ para cada elemento.

Como exemplo, será exposta a segunda maneira:

3.10.1. Elemento 1:

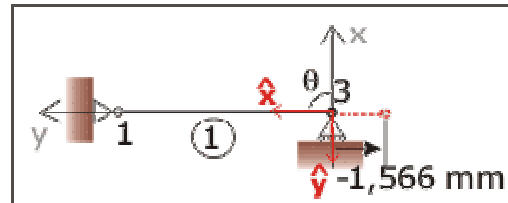


FIGURA 01: Deslocamentos no Elemento 01

Transformação do vetor de deslocamentos:

$$\hat{\underline{u}}^1 = \underline{T} \underline{u}^1$$

$$\begin{Bmatrix} \hat{u}_{1x}^1 \\ \hat{u}_{1y}^1 \\ \hat{u}_{3x}^1 \\ \hat{u}_{3y}^1 \end{Bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{Bmatrix} u_{1x}^1 \\ u_{1y}^1 \\ u_{3x}^1 \\ u_{3y}^1 \end{Bmatrix} \quad \begin{array}{l} \theta = 90 \\ c = \cos\theta = 0 \\ s = \sin\theta = 1 \end{array}$$

$$\begin{Bmatrix} \hat{u}_{1x}^1 \\ \hat{u}_{1y}^1 \\ \hat{u}_{3x}^1 \\ \hat{u}_{3y}^1 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -1,566 \end{Bmatrix} 10^{-3}$$

$$\begin{Bmatrix} \hat{u}_{1x}^1 \\ \hat{u}_{1y}^1 \\ \hat{u}_{3x}^1 \\ \hat{u}_{3y}^1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -1,566 \\ 0 \end{Bmatrix} 10^{-3} \text{ m}$$

Aplicação das Equações de Equilíbrio no sistema local:

$$\hat{f} = \hat{k} \hat{u}$$

$$\begin{Bmatrix} \hat{f}_{3x} \\ \hat{f}_{3y} \\ \hat{f}_{1x} \\ \hat{f}_{1y} \end{Bmatrix} = 105 \times 10^3 \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} -1,566 \\ 0 \\ 0 \\ 0 \end{Bmatrix} 10^{-3}$$

$$\begin{Bmatrix} \hat{f}_{3x} \\ \hat{f}_{3y} \\ \hat{f}_{1x} \\ \hat{f}_{1y} \end{Bmatrix} = \begin{Bmatrix} -164,43 \\ 0 \\ 164,43 \\ 0 \end{Bmatrix} \text{ KN}$$

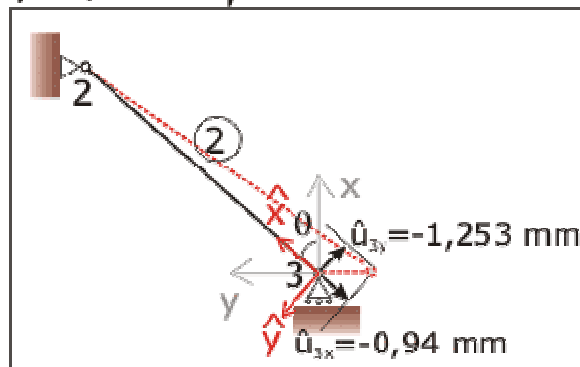


FIGURA 02: Forças no Elemento 01

3.10.2. Elemento 2:

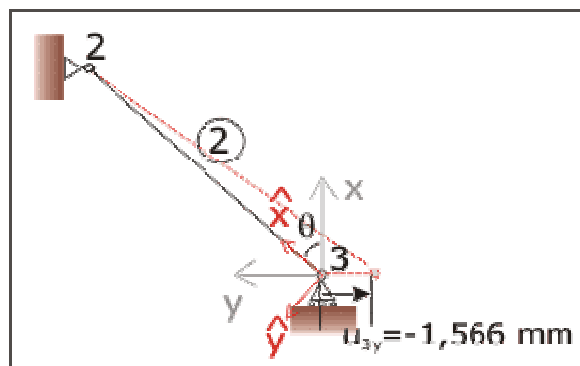


FIGURA 03: Deslocamentos Globais no Elemento 02

Transformação do vetor de deslocamentos:

$$\underline{\hat{u}}^2 = \underline{T} \underline{u}^2$$

$$\begin{Bmatrix} \hat{u}_{3x}^2 \\ \hat{u}_{3y}^2 \\ \hat{u}_{2x}^2 \\ \hat{u}_{2y}^2 \end{Bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{Bmatrix} u_{3x}^2 \\ u_{3y}^2 \\ u_{2x}^2 \\ u_{2y}^2 \end{Bmatrix} \quad \begin{array}{l} c = \cos\theta = 0,8 \\ s = \sin\theta = 0,6 \end{array}$$

$$\begin{Bmatrix} \hat{u}_{3x}^2 \\ \hat{u}_{3y}^2 \\ \hat{u}_{2x}^2 \\ \hat{u}_{2y}^2 \end{Bmatrix} = \begin{bmatrix} 0,8 & 0,6 & 0 & 0 \\ -0,6 & 0,8 & 0 & 0 \\ 0 & 0 & 0,8 & 0,6 \\ 0 & 0 & -0,6 & 0,8 \end{bmatrix} \begin{Bmatrix} 0 \\ -1,566 \\ 0 \\ 0 \end{Bmatrix} 10^{-3}$$

$$\begin{Bmatrix} \hat{u}_{3x}^2 \\ \hat{u}_{3y}^2 \\ \hat{u}_{2x}^2 \\ \hat{u}_{2y}^2 \end{Bmatrix} = \begin{Bmatrix} -0,94 \\ -1,253 \\ 0 \\ 0 \end{Bmatrix} 10^{-3} \text{ m}$$

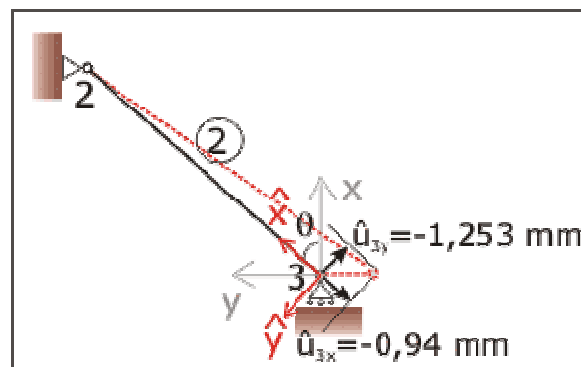


FIGURA 04: Deslocamentos Locais no Elemento 02

Aplicação das Equações de Equilíbrio no sistema local:

$$\hat{F} = k^2 \hat{U}^2$$

$$\begin{Bmatrix} \hat{F}_{3x} \\ \hat{F}_{3y} \\ \hat{F}_{2x} \\ \hat{F}_{2y} \end{Bmatrix} = 63 \times 10^3 \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} -0,94 \\ -1,253 \\ 0 \\ 0 \end{Bmatrix} 10^{-2}$$

$$\begin{Bmatrix} \hat{F}_{3x} \\ \hat{F}_{3y} \\ \hat{F}_{2x} \\ \hat{F}_{2y} \end{Bmatrix} = \begin{Bmatrix} -59,22 \\ 0 \\ 59,22 \\ 0 \end{Bmatrix} \text{ KN}$$

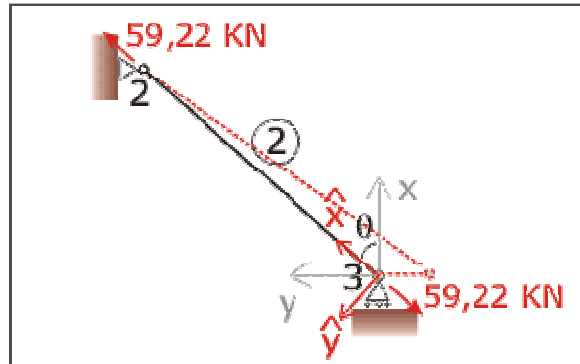


Figura 05: Forças no Elemento 02

O equilíbrio da estrutura pode ser verificado através do equilíbrio dos nós, transformando as forças de cada elemento para o sistema global de coordenadas para serem comparadas com as forças externas da estrutura obtidas.

3.10.3. Elemento 1

$$\underline{f}^l = \underline{T}^T \hat{\underline{f}}^l$$

$$\begin{Bmatrix} f_{1x}^l \\ f_{1y}^l \\ f_{3x}^l \\ f_{3y}^l \end{Bmatrix} = \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & c & -s \\ 0 & 0 & s & c \end{bmatrix} \begin{Bmatrix} \hat{f}_{1x}^l \\ \hat{f}_{1y}^l \\ \hat{f}_{3x}^l \\ \hat{f}_{3y}^l \end{Bmatrix} \quad \begin{array}{l} \theta = 90 \\ c = \cos\theta = 0 \\ s = \sin\theta = 1 \end{array}$$

$$\begin{Bmatrix} f_{1x}^l \\ f_{1y}^l \\ f_{3x}^l \\ f_{3y}^l \end{Bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} 164,43 \\ 0 \\ -164,43 \\ 0 \end{Bmatrix}$$

$$\begin{Bmatrix} f_{1x}^l \\ f_{1y}^l \\ f_{3x}^l \\ f_{3y}^l \end{Bmatrix} = \begin{Bmatrix} 0 \\ -164,43 \\ 0 \\ 164,43 \end{Bmatrix} \text{ KN}$$

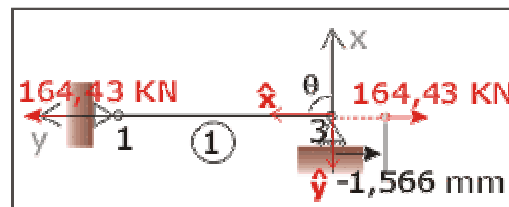


FIGURA 06: Forças no sistema de coordenadas globais no Elemento 1

3.10.4. Elemento 2

$$\underline{f}^2 = \underline{T}^T \hat{\underline{f}}^2$$

$$\begin{pmatrix} f_{2x}^2 \\ f_{2y}^2 \\ f_{3x}^2 \\ f_{3y}^2 \end{pmatrix} = \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & c & -s \\ 0 & 0 & s & c \end{bmatrix} \begin{pmatrix} \hat{f}_{2x}^2 \\ \hat{f}_{2y}^2 \\ \hat{f}_{3x}^2 \\ \hat{f}_{3y}^2 \end{pmatrix} \quad \begin{array}{l} c = \cos\theta = 0,8 \\ s = \text{sen}\theta = 0,6 \end{array}$$

$$\begin{pmatrix} f_{2x}^2 \\ f_{2y}^2 \\ f_{3x}^2 \\ f_{3y}^2 \end{pmatrix} = \begin{bmatrix} 0,8 & -0,6 & 0 & 0 \\ 0,6 & 0,8 & 0 & 0 \\ 0 & 0 & 0,8 & -0,6 \\ 0 & 0 & 0,6 & 0,8 \end{bmatrix} \begin{pmatrix} 59,22 \\ 0 \\ -59,22 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} f_{2x}^2 \\ f_{2y}^2 \\ f_{3x}^2 \\ f_{3y}^2 \end{pmatrix} = \begin{pmatrix} 47,376 \\ 35,532 \\ -47,376 \\ -35,532 \end{pmatrix} \text{ KN}$$

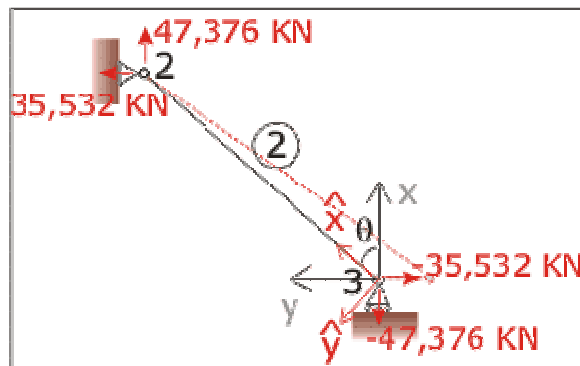


FIGURA 07: Forças no sistema de coordenadas globais no Elemento 2

Como pode ser observado nas figuras 2, 5, 6 e 7, cada elemento está em equilíbrio.

Considerando as forças externas encontradas e as forças internas de cada elemento, o equilíbrio dos nós pode ser verificado.

Animação 1: Equilíbrio nos nós

3.10.5. Solução das tensões e deformações no interior do elemento

As tensões e deformações podem ser encontradas aplicando em cada elemento a equação de deslocamentos (3.23):

$$\hat{u}(x) = \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} \begin{Bmatrix} u_{1x} \\ u_{2x} \end{Bmatrix} \text{ para cada elemento} \quad (3.23)$$

ou através da relação:

$$\sigma(x) = F/A \quad (3.26)$$

Elemento 1:

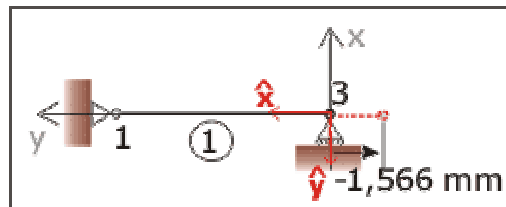


FIGURA 8: Deslocamentos no Elemento 1

Deslocamentos:

$$\hat{u}(\hat{x}) = \begin{bmatrix} 1 - \frac{\hat{x}}{L} & \frac{\hat{x}}{L} \end{bmatrix} \begin{Bmatrix} u_{1x} \\ u_{2x} \end{Bmatrix}$$

$$\hat{u}(\hat{x}) = \begin{bmatrix} 1 - \frac{\hat{x}}{0,6} & \frac{\hat{x}}{0,6} \end{bmatrix} \begin{Bmatrix} -0,00156 \\ 0 \end{Bmatrix}$$

$$\hat{u}(\hat{x}) = -0,00156 + 0,0026\hat{x}$$

Animação 2 :Deslocamento elemento 1

Deformação:

$$\varepsilon(\hat{x}) = \frac{\partial u(\hat{x})}{\partial \hat{x}}$$

$$\varepsilon(\hat{x}) = 0,0026$$

A deformação é constante em toda a barra

Tensão

$$\sigma(\hat{x}) = E\varepsilon(\hat{x})$$

$$\sigma(\hat{x}) = 210 \cdot 10^6 \cdot 0,0026$$

$$\sigma(\hat{x}) = 546000 \text{ KN/m}^2$$

A tensão também é constante em toda a barra

Fazendo $\sigma(x) = F/A$:

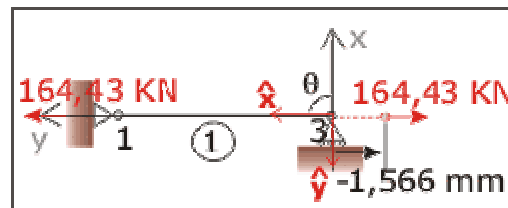


FIGURA 09: Forças no Elemento 1

Tensão

$$\sigma(\hat{x}) = F/A$$

$$\sigma(\hat{x}) = \frac{164,43 \text{ KN}}{3 \cdot 10^{-4} \text{ m}^2}$$

$$\sigma(\hat{x}) = 548100 \text{ KN/m}^2$$

Deformação

$$\varepsilon(\hat{x}) = \sigma(\hat{x})/E$$

$$\varepsilon(\hat{x}) = \frac{548100 \text{ KN/m}^2}{210 \cdot 10^6 \text{ KN/m}^2}$$

$$\varepsilon(\hat{x}) = 0,0026$$

Os resultados obtidos nos dois casos são muito próximos. A diferença existente deve-se aos arredondamentos realizados .

Elemento 2:

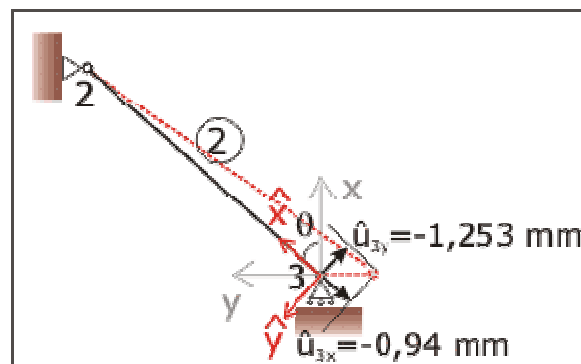


FIGURA 10: Deslocamento elemento 2

Deslocamentos

$$\hat{u}(\hat{x}) = \left[1 - \frac{\hat{x}}{L} \quad \frac{\hat{x}}{L} \right] \begin{Bmatrix} u_{3x} \\ u_{2x} \end{Bmatrix}$$

$$\hat{u}(\hat{x}) = \left[1 - \frac{\hat{x}}{1} \quad \frac{\hat{x}}{1} \right] \begin{Bmatrix} -0,00094 \\ 0 \end{Bmatrix}$$

$$\hat{u}(\hat{x}) = -0,00094 + 0,00094\hat{x}$$

Animação 5: Deslocamento elemento 2

Deformação

$$\varepsilon(\hat{x}) = \frac{\partial \hat{u}(\hat{x})}{\partial \hat{x}}$$

$$\varepsilon(\hat{x}) = 0,00094$$

A deformação é constante em toda a barra

Tensão

$$\sigma(\hat{x}) = E\varepsilon(\hat{x})$$

$$\sigma(\hat{x}) = 210 \cdot 10^6 \cdot 0,00094$$

$$\sigma(\hat{x}) = 197400 \text{ KN/m}^2$$

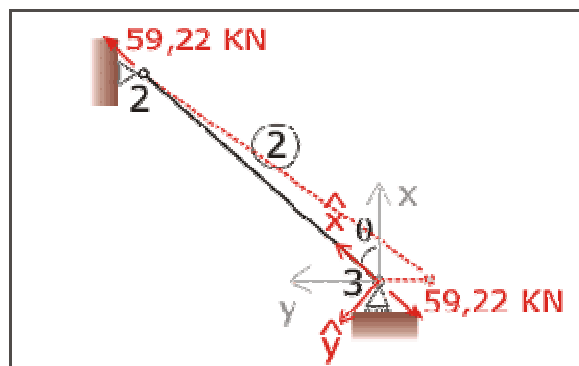
A tensão também é constante em toda a barraFazendo $\sigma(x) = F/A$:

FIGURA 13: Forças elemento 2

Tensão

$$\begin{aligned}\sigma(x) &= F/A \\ \sigma(\hat{x}) &= \frac{59,22 \text{ KN}}{3 \cdot 10^{-4} \text{ m}^2} \\ \sigma(\hat{x}) &= 197400 \text{ KN/m}^2\end{aligned}$$

Deformação

$$\begin{aligned}\varepsilon(\hat{x}) &= \sigma(\hat{x})/E \\ \varepsilon(\hat{x}) &= \frac{197400 \text{ KN/m}^2}{210 \cdot 10^6 \text{ KN/m}^2} \\ \varepsilon(\hat{x}) &= 0,00094\end{aligned}$$

Os resultados obtidos nos dois casos são muito próximos. A diferença existente deve-se aos arredondamentos realizados.

3.10.6. Interpretação dos Resultados

A interpretação dos resultados fica a cargo do analista e depende do objetivo final da análise.

3.11. COMPARAÇÃO DA SOLUÇÃO DO MEF COM A SOLUÇÃO EXATA

Esta unidade propõe a resolução de um problema tipo, figura mostrada ao lado, segundo a resistência dos materiais e segundo o método dos elementos finitos.

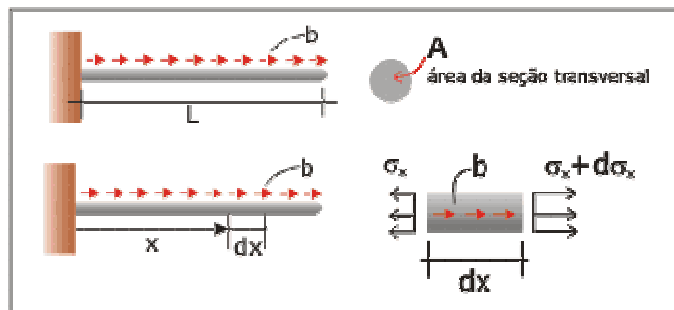


FIGURA 01: Problema Tipo Proposto

O problema deve ser resolvido através do método dos elementos finitos utilizando elementos finitos de treliça.

Objetivando a comparação da solução do Método dos Elementos Finitos com a solução exata do problema matemático proposto, devem ser utilizados um, dois e três elementos finitos de treliça.

A solução exata do modelo matemático proposto foi apresentada no item Modelo matemático¹⁵.

Após a resolução deste problema e o desenho de gráficos para facilitar a comparação, algumas conclusões são obtidas:

- A aproximação para os deslocamentos melhora à medida que aumenta o número de elementos da discretização.
- As tensões e deformações apresentam valores constantes para os elementos lineares (elemento de treliça).
- Há a continuidade de deslocamentos entre os elementos. Não há continuidade de tensões entre os elementos.
- Existem pontos ótimos para cálculo de tensões (pontos em que as tensões aproximadas correspondem às tensões calculadas pelo modelo matemático). No caso apresentado, o ponto ótimo é no meio de cada elemento.

Este problema encontra-se resolvido como material complementar na área de transferência¹⁶ deste curso.

3.12. PRINCÍPIO DOS TRABALHOS VIRTUAIS

Segundo o princípio dos trabalhos virtuais (PTV), em uma estrutura em equilíbrio, sujeita a um sistema de forças externas, ao se impor deslocamentos virtuais compatíveis, o trabalho realizado pelas forças reais externas sobre os deslocamentos virtuais é igual ao trabalho das tensões internas sobre as deformações produzidas pelos

¹⁵ Link para item 2 da Unidade 03 – Modelo Matemático

¹⁶ Link para area de transferência do curso – Este material encontra-se no final deste anexo.

deslocamentos virtuais. Ou seja: trabalho produzido pelas forças externas é igual ao trabalho produzido pelas tensões internas.

Animação 01: Princípio dos Trabalhos Virtuais

3.12.1. Aplicação do PTV para obtenção da solução exata

O princípio dos trabalhos virtuais pode ser usado para determinação da solução exata do modelo matemático do sistema, resultando na mesma equação diferencial obtida através do equilíbrio (visto no item Modelo Matemático¹⁷).

Animação 02: Solução Exata através do PTV

3.12.2. Aplicação do PTV para um elemento de treliça de dois nós

O PTV, se aplicado ao elemento de treliça de dois nós, também determina a matriz de rigidez do elemento e reparte a carga distribuída no elemento em seus nós, o que é necessário para a resolução de um problema através do Método dos Elementos Finitos.

Animação 03: Resolução do Elemento Finito de Treliça através do PTV

A partir da aplicação do PTV, temos que:

Cada termo da matriz de rigidez pode ser calculado com:

$$K_{ij} = \int_0^L (EA N_{i,x} N_{j,x}) dx \quad (3.27)$$

$N_{i,x}$ é a derivada em x da função de forma do elemento i

Cada elemento do vetor carregamento nodal equivalente pode ser obtido com:

$$f_{e,i} = \int_0^L N_i b(x) dx \quad (3.28)$$

N_i é a função de forma do elemento i

¹⁷ Link para Unidade 3 – item 2 Modelo matemático

3.13. CONDIÇÕES DE CONVERGÊNCIA

Como foi visto, à medida que se aumenta o número de elementos finitos em uma discretização, a solução tende para a solução exata do modelo matemático se o elemento finito obedecer a certas condições de convergência:

- Critério da Completude;
- Critério da Conformidade;

3.13.1. Critério da Completude

Animação 01: Critério da completude

O critério da completude estabelece que a função de deslocamentos associada ao elemento finito deve representar movimento de corpo de rígido e estado de deformação constante no interior do elemento.

3.13.2. Critério da Conformidade

Animação 02: Critério da conformidade

O critério da conformidade estabelece que a função de deslocamentos associada ao elemento finito deve ser contínua no interior do elemento e ser contínua nas interfaces (nós, linhas ou superfícies) entre os elementos.

3.14. USO DA SIMETRIA EM UMA ESTRUTURA

Vários tipos de simetria podem ser observados em uma estrutura. Em uma estrutura simétrica, o problema pode ser reduzido, observando a correspondência em tamanho, forma, carregamentos, material e condições de contorno entre as partes simétricas de uma estrutura. O tamanho da matriz de rigidez e o número total de equações a serem resolvidas podem reduzir muito, se observada a simetria de uma estrutura. É importante mencionar que as áreas de 7 e 8 serão reduzidas.

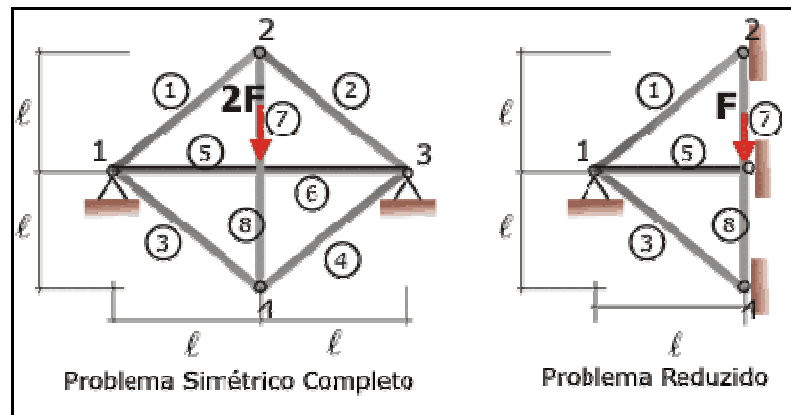


FIGURA 01: Exemplo da Redução de um Problema utilizando simetria
 Fonte: A First Course in the Finite Element Method – Daryl L. Logan

3.15. APOIOS INCLINADOS

Há casos em que os apoios da estrutura não estão orientados conforme o sistema de coordenadas global do problema. Nestes casos, as componentes de força, deslocamento e matriz de rigidez dos nós nos quais os apoios estão orientados de forma diferente devem ser transformados para um eixo no qual um dos deslocamentos é nulo.

Essa transformação é feita adotando os mesmos procedimentos da Transformação de Vetores em Duas Dimensões.¹⁸

Animação 01: Apoios Inclinados

¹⁸ Link para Unidade 3 – item 7 Transformação de Vetores em Duas Dimensões