



**Rui Orlando Gomes  
Isidro**

**SA<sup>3</sup>C – Sistema de Avaliação e Aprendizagem  
Assistido por Computador**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Prof. Dr. Joaquim Sousa Pinto, Professor Auxiliar do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro e pelo Prof. Dr. António José Batel Anjo, Professor Auxiliar do Departamento de Matemática da Universidade de Aveiro.

## **o júri**

presidente

**Professor Doutor António Manuel Melo de Sousa Pereira**  
Professor Catedrático da Universidade de Aveiro

**Professor Doutor Diamantino Rui da Silva Freitas**  
Professor Associado da Faculdade de Engenharia da Universidade do Porto

**Professor Doutor Joaquim Manuel Henriques de Sousa Pinto**  
Professor Auxiliar da Universidade de Aveiro

**Professor Doutor António José Batel Anjo**  
Professor Auxiliar da Universidade de Aveiro

## **agradecimentos**

Este trabalho não seria possível sem o apoio de muitas pessoas:

Em primeiro lugar, quero agradecer ao orientador e co-orientador desta dissertação, os Professores Joaquim Sousa Pinto e António Batel Anjo.

Ao Professor David Vieira pela amizade, apoio, pelas muitas e válidas sugestões e ainda pela prestável ajuda na revisão final deste texto.

À minha família, por todo apoio, carinho e amor, por suportarem pacientemente um filho distante da vida familiar.

À Dalila pelo apoio incondicional e força positiva dada ao longo destes meses, a quem devo o tempo que aos dois pertencia.

Ofereço ainda o meu agradecimento a todos os colegas do Projecto Matemática Ensino. Sem a sua colaboração este trabalho não teria sido possível.

A todos, bem-haja!

Rui Isidro

## resumo

O Sistema de Avaliação e Aprendizagem Assistido por Computador, SA<sup>3</sup>C, encerra um conjunto de ferramentas que apresentam uma perspectiva inovadora sobre as tradicionais aplicações de eLearning, sendo de destacar a componente lúdica capaz de atrair os estudantes para o estudo.

O sistema baseia-se numa filosofia de aprendizagem pela avaliação e funciona como um complemento ao manual de estudo e à sala de aula nunca substituindo ou diminuindo o papel que o professor desempenha no ensino tradicional.

Os alunos são entusiasmados a estudar através de ferramentas com o perfil de jogo, mas sem nunca pender para o lado exclusivamente lúdico. O SA<sup>3</sup>C tem um carácter competitivo que provou ao longo do tempo ser um aliado eficaz para sensibilizar os alunos para a necessidade de criar métodos de trabalho e estudo.

Toda aplicação se baseia no conceito de *modelos geradores de questões* (modelos). Os modelos quando são inseridos no sistema são catalogados de acordo com uma estrutura rígida que oferece ao SA<sup>3</sup>C outra das suas características mais importantes: a avaliação qualitativa.

O SA<sup>3</sup>C permite, através da avaliação qualitativa, a definição de perfis de conhecimento. Estes perfis de conhecimento permitem ao aluno conhecer o seu grau de conhecimento nas várias matérias com graus de detalhe que uma avaliação quantitativa não permitiria. Estes perfis podem ainda ser utilizados como base de trabalho para os professores.

Este texto descreve o desenvolvimento do SA<sup>3</sup>C, que envolve diversas tecnologias distintas: tecnologias de representação de matemática na WEB e de representação de gráficos dinâmicos na WEB. É descrito também o modelo de funcionamento do motor de geração de modelos e o dimensionamento da base de dados.

## abstract

The Evaluation and Computer aided Learning system SA<sup>3</sup>C, encloses a set of tools that present an innovative perspective on the traditional applications of eLearning, it's important to emphasis the playful component capable to attract the students for the study.

The system is based on learning by evaluation philosophy, and it functions as a complement to the manual of study and the. It never replaces or reduces the role that teachers performance in traditional education.

The students are enthusiastic to study using tools with the game profile, but never exclusively for the entertainment side. The SA<sup>3</sup>C has a very competitive part, and it proved threw time to be an efficient ally to allert students for the necessity to create methods of work and study.

The entire application is based on the concept of *models generators of questions* (models). The models when they are inserted in the system are catalogued in accordance with a rigid structure that offers the SA<sup>3</sup>C another one of its more important characteristics: the qualitative evaluation.

The SA<sup>3</sup>C allows, through the qualitative evaluation, the definition of knowledge profiles. These profiles allow the pupil to know its degree of knowledge in some subjects with detail that a quantitative evaluation would not allow. These profiles can also be used as working bases for the professors.

This text describes the development of the SA<sup>3</sup>C, which involves distinct technologies: mathematical representation and dynamical graph representation on the WEB. It also describes the generating models process and the development of the database.

SA<sup>3</sup>C - Sistema de Avaliação e Aprendizagem  
Assistido por Computador

Rui Orlando Gomes Isidro

19 de Julho de 2004

# Conteúdo

<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e objectivos a atingir . . . . .	4
1.2 Estrutura da dissertação . . . . .	6
<b>2 Ensino-a-distância / Ensino Assistido Por Computador</b>	<b>7</b>
2.0.1 O e-Learning e o Ensino Assistido Por Computador . . . . .	8
2.0.2 Aprendizagem pela avaliação . . . . .	12
2.1 Algumas razões que levaram à actualização do sistema. . . . .	14
2.2 Ferramentas e sistemas para ensino-a-distância . . . . .	16
<b>3 SA<sup>3</sup>C - do modelo à aplicação final</b>	<b>23</b>
3.1 Modelo . . . . .	23
3.2 Linguagem de Representação de Modelos . . . . .	26
3.3 Codificação de modelos . . . . .	27
3.4 Avaliação de modelos . . . . .	29
3.5 Provas: treino, diagnóstico, avaliação e competição . . . . .	30
<b>4 Tecnologias usadas</b>	<b>37</b>
4.1 Hardware . . . . .	37

4.2	Software . . . . .	42
4.2.1	Suporte da plataforma ao nível de negócio . . . . .	42
4.2.2	Tecnologias de representação . . . . .	43
4.2.3	MathML . . . . .	44
4.2.4	Utilização da linguagem MathML . . . . .	46
4.2.5	SVG . . . . .	52
4.2.6	Utilização da linguagem SVG . . . . .	53
4.2.7	Outras tecnologias . . . . .	55
<b>5</b>	<b>Descrição do trabalho desenvolvido</b>	<b>59</b>
5.1	Levantamento dos requisitos . . . . .	60
5.2	Arquitectura da aplicação . . . . .	63
5.3	Motor da prova . . . . .	65
5.4	Estudo das tecnologias necessárias a utilizar . . . . .	65
5.4.1	Estudo dos mecanismos de ligação de tecnologias . . . . .	66
5.5	Determinação das fronteiras dos serviços . . . . .	69
<b>6</b>	<b>Testes ao sistema</b>	<b>73</b>
6.1	Estudo das variáveis necessárias . . . . .	73
6.2	Descrição dos testes efectuados . . . . .	76
6.3	Interfaces . . . . .	80
6.4	Conclusões . . . . .	83
<b>7</b>	<b>Conclusões</b>	<b>89</b>
<b>8</b>	<b>Trabalho futuro</b>	<b>93</b>



# Lista de Figuras

2.1	Ciclo de formação . . . . .	13
2.2	Número de escolas com ligação à Internet de 1997 a 2001 . . . . .	15
3.1	Parte de um modelo. . . . .	34
3.2	Estrutura dos códigos dos modelos. . . . .	35
3.3	Estrutura do código de objectivos das respostas. . . . .	35
4.1	Arquitectura de Hardware e distribuição inicial de serviços. . . . .	40
4.2	Arquitectura de Hardware. . . . .	42
4.3	Inserção de expressão MathML usando XSLT no cliente. . . . .	47
4.4	Comparação do resultado final usando o MathPlayer e XSLT no cliente. . . . .	50
4.5	Uma concretização de um MODELO da MINImat com SVG. . . . .	55
4.6	Uma concretização de um MODELO da EQUAmat com MathML. . . . .	56
4.7	Uma concretização de um MODELO da mat12 com MathML e SVG. . . . .	57
5.1	Arquitectura da aplicação. . . . .	63
6.1	Histograma de frequência / frequência cumulativa. . . . .	74
6.2	Histograma de frequência / frequência cumulativa. . . . .	75
6.3	Estrutura do laboratório de testes. . . . .	77
6.4	RPS vs. Número de Clientes. . . . .	78
6.5	Resultados dos testes à página da EQUAmat. . . . .	82
6.6	Resultados dos testes à página da EQUAmat@moz. . . . .	82

6.7	Resultados dos testes à página da MINImat. . . . .	83
6.8	Resultados dos testes à página da MXII. . . . .	83

# Lista de Tabelas

4.1	Exemplo de utilização da MathML recorrendo ao MathPlayer . . . . .	51
4.2	Inserção de uma imagem SVG pelo método <i>embeded</i> . . . . .	54
6.1	Testes com 2 máquinas com metade da carga vs 1 máquina com toda a carga. . . . .	85
6.2	Teste de carga ao sistema com modelos da EquaMat. . . . .	86
6.3	RPS vs. número de clientes para modelos que usam o webMathematica.	87

# Capítulo 1

## Introdução

O Projecto Matemática Ensino<sup>1</sup> (PmatE) desenvolveu e mantém em desenvolvimento um conjunto de ferramentas que apresentam uma perspectiva inovadora sobre as tradicionais aplicações de e-Learning - uma componente lúdica e um desafio permanente capazes de atrair os estudantes ao estudo.

O trabalho desenvolvido neste domínio teve início em 1989/90, mas a plataforma do Sistema de Avaliação e Aprendizagem Assistido por Computador (SA<sup>3</sup>C), em construção, necessitou de uma profunda reestruturação no sentido de acompanhar os recentes desenvolvimentos nas novas tecnologias da informação e da comunicação. Esta tarefa foi iniciada no ano de 2002. Manteve-se, no entanto, o objectivo de aumentar o gosto pela matemática, nomeadamente pelas matemáticas escolares e toda a filosofia do projecto. Um segundo propósito igualmente importante é o de utilizar estas ferramentas como instrumento de avaliação e aprendizagem. O público-alvo estende-se presentemente do Ensino Básico ao Ensino Superior, embora os correspondentes programas tenham diferentes níveis de desenvolvimento.

---

<sup>1</sup>O Projecto Matemática Ensino (PmatE) é um Projecto de Investigação e Desenvolvimento do Departamento de Matemática interagindo com Escolas de variados graus de ensino e integrado na UI&D “CEOC - Centro de Estudos em Optimização e Controlo” do Departamento de Matemática da Universidade de Aveiro.

As diversas matérias são sujeitas a um trabalho intensivo de modelação levada a cabo por equipas integrando elementos do PmatE e professores dos correspondentes graus de ensino. Os modelos geradores de questões [VCA01], doravante designados por MODELOS, a abordar na secção seguinte, são objectos de teor didáctico que geram questões tendo em conta objectivos científicos e pedagógico-didácticos. Os MODELOS podem ser organizados no modo competição ou como sistema de avaliação e aprendizagem. A sua modularidade, flexibilidade e aleatoriedade são características fundamentais para conferir aos produtos elaborados pelo PmatE, em qualquer dos modos referidos, um interesse sempre renovado.

No modo competição propõe-se uma competição de tipo colaborativo. No modo avaliação e aprendizagem a colaboração entre colegas e com o professor é um aspecto intrínseco ao processo.

As questões geradas pelos MODELOS são do tipo falso-verdadeiro-generalizado, conceito que se explicará posteriormente. O sistema contém vários mecanismos, uns internos, outros externos, para minimizar a influência da resposta à sorte.

Os testes associados a cada questão têm objectivos próprios; testam conceitos, aplicações básicas desses mesmos conceitos e, quando possível, interpretações gráficas. Esta múltipla abordagem aumenta a compreensão dos conceitos.

Devido a um programa de análise de dados associado, o aluno pode controlar o grau de aquisição de conhecimentos (auto-diagnose) e o professor pode diagnosticar os conhecimentos dos seus alunos e seguir a evolução das suas aprendizagens até um certo nível. Não se pretende que esta seja a única forma de avaliação dos alunos, pelo que se torna necessário complementar esta avaliação com outra relativa a competências de expressão escrita e de desenvolvimento de cálculos.

As experiências e testes realizados sobretudo com o programa EQUAmat, onde após quatorze anos o entusiasmo não só se mantém como aumenta, são bastante promissoras. A nível superior, onde as dificuldades de modelação e o correspondente trabalho informático são muito maiores, as experiências realizadas em várias disciplinas - Métodos Numéricos, Fundamentos de Matemática, Matemática Básica I e Cálculo I<sup>2</sup> - revelaram igualmente que este programa possui grandes potencialidades como instrumento de apoio ao processo de aprendizagem, contribuindo decisivamente para aumentar o envolvimento dos alunos com as matérias desde o início e durante todo o semestre.

Este texto descreve não só parte do meu trabalho de investigação e desenvolvimento, mas também o de muitas pessoas que, directa ou indirectamente, colaboraram com o PmatE. Cabe aqui, no entanto, esclarecer que o meu contributo para este projecto assentou essencialmente no desenho, desenvolvimento e manutenção da Base de Dados de suporte. Foi também da minha responsabilidade o desenvolvimento dos motores de todas as provas - desde a resolução dos problemas de integração com a DCOM<sup>3</sup> de processamento de modelos, à construção da lógica de negócio da aplicação até à construção das interfaces das provas com os utilizadores.

Desenvolvi também parte dos módulos de comunicação interaplicacionais, nomeadamente para ligação da DCOM do PmatE com aplicações exteriores de cálculo.

---

<sup>2</sup>Nota: à data de redacção desta dissertação encontrava-se em curso a experiência com a disciplina de Cálculo II.

<sup>3</sup>Distributed Component Object Model (DCOM) estende a arquitectura COM (Component Object Model) com suporte para a comunicação entre objectos em computadores diferentes. COM é uma arquitectura de software desenvolvida pela Microsoft para desenvolvimento de aplicações baseadas em componentes. Os objectos COM são componentes discretos, cada um com um identificador único, que expõem interfaces que permitem às aplicações e outros componentes aceder às suas funcionalidades. O ActiveX também se baseia na arquitectura COM [Jup].

Ao nível da DCOM desenvolvi a interface programática de ligação das aplicações do PmatE ao gerador de modelos. Fui ainda responsável pelo desenvolvimento tecnológico dos módulos necessários a todos os sites de divulgação do PmatE.

## 1.1 Motivação e objectivos a atingir

Existe na sociedade actual uma preocupação generalizada com o insucesso escolar em todos os graus de ensino-aprendizagem. O problema é real e propaga-se desde o 1º ciclo do ensino básico ao ensino superior. O problema, aparentemente, agudiza-se quando se fala de matemática, muito particularmente de matemáticas escolares.

Na realidade, atravessamos uma época onde se cultiva um certo facilitismo e se desresponsabilizam os agentes envolvidos: pais, professores e alunos [Fon].

O problema não está, certamente, apenas na matemática; por certo se assim fosse já estaria resolvido. O problema passa por todas as disciplinas e certamente pela língua portuguesa. Daí que seja urgente encontrar novas soluções, utilizar novas metodologias e, porque não, aliar as novas tecnologias disponíveis ao ensino.

A grande aposta é, portanto, desenvolver o gosto pela matemática e pela língua portuguesa. Pretende-se atingir este objectivo utilizando o computador, o desafio permanente e o trabalho colaborativo como ingredientes. A exigência e a preocupação com a inclusão de todos os alunos e com a criação de uma mentalidade que valorize o esforço são algumas das suas características.

Foi com estes propósitos em mente que o PmatE resolveu apostar no SA<sup>3</sup>C. Trata-se de um sistema de informação que tem ainda como objectivos, a centralização de todos os dados num só ponto. Desta forma, possibilita-se a análise posterior de toda a informação armazenada neste repositório único, de forma automatizada,

podendo contribuir para a definição de perfis de conhecimentos e competências dos utilizadores.

O objectivo foi, entre outros, construir uma aplicação simples de usar, sob todos os aspectos, desde a manipulação das interfaces desenvolvidas à instalação da aplicação. A aplicação deve servir tanto os utilizadores experientes com a utilização de sistemas informáticos, como os menos experientes com a mesma facilidade. Sendo uma aplicação que deverá ser utilizada por um leque muito grande de utilizadores, deve também contar com o factor idade, servindo com a mesma eficácia utilizadores de todas as idades. Ainda associado à problemática da construção das interfaces da aplicação, a representação de objectos matemáticos e figuras com o melhor aspecto gráfico possível é também uma das preocupações.

Pretende-se também, que a aplicação evolua para um sistema com navegação automática e semi-automática que permita guiar os alunos no seu processo de aprendizagem com maior ou menor grau de autonomia.

Para estimular o desafio constante, criou-se um mecanismo de geração de questões inovador que, em cada concretização, apresenta uma questão diferente, embora de certo modo isomorfa da anterior. Foram também acrescentados mecanismos que apelam à concentração e interpretação de conteúdos. Esta era, de resto, a filosofia inicial perseguida pelo PmatE deste o primeiro momento.

A facilidade de utilização assume um papel realmente forte em todos os aspectos da aplicação, inclusivamente na escolha das tecnologias de distribuição da aplicação. É uma aplicação WEB que apenas necessita da instalação de dois visualizadores: um para o desenho das fórmulas matemáticas e outro para a representação de figuras.

Por fim, mas não menos importante, a actualização constante da aplicação deve



ser um processo simples e que se repercute imediatamente nas aplicações dos utilizadores.

## 1.2 Estrutura da dissertação

Este texto, pretende apresentar o trabalho desenvolvido no estudo e reformulação da nova plataforma das aplicações do PmatE. Após uma breve introdução, no capítulo 2 serão apresentadas várias metodologias de ensino, e o resultado da pesquisa de algumas ferramentas para o ensino-a-distância existentes. Ainda no capítulo 2, será apresentada a aprendizagem pela avaliação recorrendo a um sistema de ensino assistido por computador.

No capítulo 3 serão apresentadas as filosofias e metodologias do PmatE no que diz respeito à criação, gestão e catalogação de conteúdos. Serão apresentados os conceitos de “modelos geradores de questões”, “Linguagem de Representação de Modelos”, “codificação de modelos”, bem como o conceito de “prova”.

No 4º capítulo, descrevem-se as várias tecnologias envolvidas no desenvolvimento do projecto bem como a forma como foi feita a sua integração. O autor desta dissertação é responsável por todo o processo de integração de tecnologias. É também co-responsável pelo desenvolvimento de parte do software criado. Este capítulo, além da descrição da plataforma de Hardware, conta também com uma descrição do software envolvido.

Nos capítulos 5º e 6º da dissertação descrevem alguns dos trabalhos desenvolvidos e apresentam alguns testes efectuados ao sistema já em funcionamento. Por fim é feita uma discussão sobre os resultados obtidos.

## Capítulo 2

# Ensino-a-distância / Ensino Assistido por Computador

Pretende-se neste capítulo estabelecer algumas fronteiras entre os dois sistemas de ensino: o Ensino-a-distância e o Ensino Assistido por Computador. Ainda que em muitos aspectos acabem por se cruzar, a verdade é que ambos devem servir propósitos distintos.

Em especial no que diz respeito ao ensino propriamente dito, na sua maioria, estas aplicações foram pensadas para serem utilizadas “*a distância*”. Isto é, possuem um elevado número de funcionalidades que permitem a interacção entre indivíduos a longas distâncias. São, no entanto, muitas vezes (mal) utilizadas para o acompanhamento normal das aulas no ensino presencial. Por isso, são alvo de algumas acusações de tornarem o ensino despersonalizado o que pode desmobilizar alunos e professores pela falta de um contacto pessoal [LC03, Hal].

Os sistemas convencionais de ensino-a-distância, ainda que tenham grandes vantagens em alguns domínios, apresentam, sob outros pontos de vista, algumas dificuldades que podem comprometer a qualidade de ensino. Daí que um novo tipo

de aplicações surja como alternativa aos sistemas puros de ensino-a-distância, as aplicações de *apoio* ao ensino. São, usualmente, ferramentas com um objectivo muito específico e direccionadas a resolver um problema muito restrito. O Sistema de Avaliação e Aprendizagem Assistida por Computador (SA<sup>3</sup>C) é um conjunto de ferramentas que pretendem preencher um pequeno nicho no mercado - aprendizagem e avaliação de conhecimentos, muito em especial no que diz respeito às matemáticas escolares. A plataforma desenvolvida permite, no entanto, que o sistema cresça noutros sentidos e possa corresponder igualmente em outras áreas científicas.

O SA<sup>3</sup>C pretende ser um conjunto de ferramentas disponíveis - algumas gratuitas - em <http://pmate.mat.ua.pt>. Estas ferramentas estão a ser desenvolvidas em conjunto com escolas, universidades e professores directamente envolvidos com os utilizadores finais. O ex-libris destas aplicações é a competição EQUAmat para alunos do 3<sup>o</sup> Ciclo do Ensino Básico versando essencialmente o tema equações do 1<sup>o</sup> grau em  $\mathbb{Q}$ . Além do formato competição, o PmatE tem também disponíveis testes diagnóstico de avaliação de conhecimentos. Foram já efectuadas algumas experiências piloto de acompanhamento de alunos em disciplinas do ensino superior na Universidade de Aveiro [IAPP04].

### 2.0.1 O e-Learning e o Ensino Assistido Por Computador

A definição de e-Learning não é pacífica [KDB<sup>+</sup>02]. O principal problema está em encontrar um significado para o prefixo “e”. Há quem diga que se trata de ensino electrónico, outros defendem que se trata de ensino baseado exclusivamente na Internet, ou uma mescla de tecnologias desde as cassetes de audio, aos CDs ou DVDs e Internet [LC03]. Nesta dissertação, e ainda como diz Jorge Lima [LC03], entende-se como e-Learning qualquer experiência de aprendizagem que envolva qualquer destes recursos ou combinações destes. O fundamental deste paradigma de ensino não são

as tecnologias envolvidas, mas sim a forma de ensinar [LC03]. Pressupõe-se ainda que existe uma barreira espaço/temporal entre o professor e o aluno.

Neste estudo, entende-se Ensino Assistido Por Computador como um sub-conjunto do e-Learning. Ainda que recorrendo, eventualmente, ao mesmo suporte tecnológico, neste paradigma de ensino/aprendizagem a barreira espaciotemporal entre o professor e aluno pode não existir. Mais ainda - esta ligação deve, inclusive, sair reforçada.

### **e-Learning**

No ensino-a-distância pressupõe-se, à partida, que existe uma barreira geográfica e/ou temporal entre os professores e alunos. A distribuição dos conteúdos, usualmente, recorre a instrumentos tecnológicos e o controlo da aprendizagem é feito pelo aluno.

Como o controlo da aprendizagem (selecção de conteúdos, tempo de estudo e ritmo de estudo) é feito maioritariamente pelos alunos, eles devem ter a maturidade e motivação suficientes para controlarem este processo de auto-aprendizagem[Roc02].

O ensino-a-distância, com a Internet como suporte, tem várias vantagens [Hal]:

**Flexibilidade, acessibilidade e conveniência:** os utilizadores podem aceder ao programa em qualquer altura e em qualquer lado.

**Multi-plataforma:** a utilização de HTML e outras tecnologias permitem, por vezes, aceder aos conteúdos a partir de várias plataformas: windows, unix, mac, linux, entre outras.

**Disponibilidade:** a ligação à Internet é cada vez mais barata e cada vez mais disseminada.

**Distribuição fácil e barata:** do ponto de vista dos criadores de conteúdos, a

distribuição passa a ser muito mais fácil, não sendo necessário qualquer outro meio de distribuição.

**Facilidade de actualização:** se forem necessárias actualizações ao programa, basta alterar a versão no servidor e as alterações ficarão disponíveis imediatamente para todos os utilizadores.

Algumas desvantagens apontadas igualmente por Brandon Hall em [Hal], são:

**Limitações na Largura de Banda:** uma largura de banda pequena significa que a distribuição de conteúdos multimédia torna-se lenta e desagradável para o utilizador. As ligações de banda larga são uma realidade relativamente recente.

**Falta de contacto humano:** é uma preocupação geral a de que os computadores assumem cada vez mais protagonismo em todas as áreas da vida humana.

**Conteúdos demasiado estáticos:** tal como qualquer tecnologia emergente, o desenvolvimento de conteúdos electrónicos dedicados ao ensino ainda não contém a interactividade esperada.

**O desenvolvimento de conteúdos é mais caro e demorado que o esperado:** a utilização de tecnologias de ponta obrigam a um investimento inicial muito elevado, ainda que se possa contar com um baixo custo de distribuição.

A estas desvantagens, há ainda que somar outras duas extremamente importantes: baixa densidade de informação e os curtos períodos de atenção [WSKF02]. Efectivamente, a quantidade de informação que pode ser apresentada num ecrã de computador é limitada quando comparada com os materiais impressos. Por outro lado, vários estudos apontam para a diminuição da atenção e percepção quando se trata de absorver grandes quantidades de informação a partir de um ecrã de computador [WSKF02, MN].

## Ensino convencional

No paradigma tradicional de ensino-aprendizagem existem essencialmente 3 grandes teorias. São elas o behaviorismo, o cognitivismo e o construtivismo. A última é uma evolução da segunda [LC03].

O behaviorismo baseia-se num comportamento por resposta a um estímulo. Os processos cognitivos são ignorados e centra-se numa filosofia que reforça positivamente uma resposta certa a uma questão e negativamente uma resposta errada. O objectivo é aumentar a probabilidade de ocorrência de resposta certa no futuro.

No cognitivismo, ao contrário do behaviorismo que considera que o comportamento é uma mera resposta mecânica a um estímulo, tenta-se perceber como a mente humana recebe, interpreta, armazena e recupera a informação. Esta teoria conta também com o conceito de estágio de desenvolvimento. Tal como no behaviorismo o conhecimento é visto como dado e absoluto.

O construtivismo tem os seguintes princípios básicos [LC03]:

- *O conhecimento é construído activamente pelo aluno; não é transmitido.*
- *A aprendizagem é, simultaneamente, um processo activo e reflexivo.*
- *A interpretação que o aluno faz da nova experiência é influenciada pelo seu conhecimento prévio.*
- *A aprendizagem requer a compreensão do todo assim como das partes, e estas deverão ser entendidas no contexto do todo. A aprendizagem deve, por isso, centrar-se em contextos e não em factos isolados.*

Esta última teoria é a que se apresenta como a que melhor responde aos actuais desafios do ensino-aprendizagem: preparação dos jovens para o trabalho em equipa, saber seleccionar, pesquisar, relacionar e sintetizar informação, desenvolver um espírito crítico e capacidade de iniciativa na resolução de problemas.

### **Ensino Assistido Por Computador**

As ferramentas de e-Learning, ainda que tenham obtido muita aceitação por parte da indústria e respondam satisfatoriamente às necessidades de formação nas empresas, não parecem eficazes no caso do ensino escolar tradicional [WSKF02].

No entanto, na filosofia construtivista, as novas tecnologias de informação podem contribuir decisivamente na formação dos indivíduos. Optando por utilizar estas tecnologias como complemento às aulas normais, pode obter-se o melhor rendimento dos dois mundos.

### **2.0.2 Aprendizagem pela avaliação**

Avaliar com precisão o grau de conhecimento de um indivíduo sobre um certo conceito está longe de ser uma tarefa fácil. No modo tradicional, os formadores preparam um conjunto de questões - ou exercícios - tentando abordar os conceitos pretendidos de uma forma o mais abrangente possível. Porém, podem ser cometidos erros na preparação do exame e o seu resultado será uma amostra condicionada das capacidades reais do formando. Por outro lado, é igualmente importante avaliar aquilo que o aluno sabe, bem como, os conceitos menos bem apreendidos.

Com uma informação completa dos conceitos adquiridos pelo aluno, dos conceitos menos claros e das lacunas que estão a impedir o aluno de avançar em certos

domínios, o docente pode sugerir alternativas de estudo, ou aplicar metodologias diferentes no sentido de ajudar o aluno a evoluir.

Num bom sistema de ensino, seria desejável que os alunos pudessem efectuar avaliações parciais ao longo do seu processo de aprendizagem. Além disso, se o professor puder ter uma ideia dos conhecimentos reais adquiridos pelos alunos, tem a possibilidade de organizar a formação de acordo com a forma e o retorno de como os alunos estão a absorver os novos conhecimentos. Este processo deve começar o mais cedo possível, eventualmente com um teste diagnóstico. A figura 2.1 pretende representar este processo.

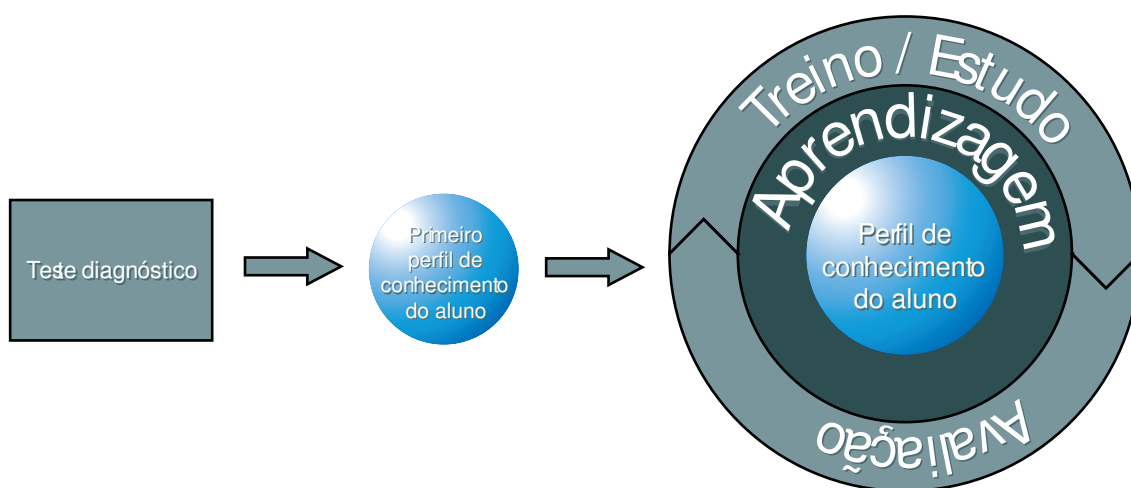


Figura 2.1: Ciclo de formação

O processo deve iniciar-se com um teste diagnóstico. A análise desse teste dará ao professor um ponto de partida. Daí em diante o processo entra num ciclo de aprendizagem / avaliação. Toda a informação recolhida durante o processo vai permitir definir perfis de conhecimentos e de competências do aluno. Obviamente, pretende-se que estes perfis tenham raio crescente. O processo de aprendizagem deveria combinar de forma transparente o estudo e a avaliação, compensando assim, o trabalho do aluno até porque se ele for bem acompanhado as suas dificuldades



serão colmatadas pelo professor ou pelo manual de estudo.

Nas aulas do dia a dia torna-se, por vezes, difícil para o professor avaliar as dificuldades dos alunos. Obter um retorno por parte dos alunos passa, quase sempre, pela atribuição de trabalhos de casa, pela avaliação do desempenho na sala de aula e muitas vezes não há alternativas a não ser um juízo baseado em razões intuitivas - por isso demasiado subjectivas.

Pretendemos construir instrumentos e mecanismos que permitam ao professor perceber, com critérios objectivos, como os conceitos métodos e técnicas que está a ensinar estão a ser recebidos pelos seus alunos.

## **2.1 Algumas razões que levaram à actualização do sistema.**

Tal como foi dito, o PmatE desde 1989/90 vem desenvolvendo várias ferramentas neste domínio. As limitações tecnológicas de cada período condicionaram as tecnologias utilizadas. As primeiras versões do software desenvolvido eram aplicações do tipo *desktop* e foram construídas em DBASE. Os avanços tecnológicos no domínio da Internet ditaram uma mudança tecnológica profunda no projecto.

A opção de modificar a plataforma, de uma aplicação do tipo *desktop* para uma aplicação distribuída, resulta de vários factores. Por um lado, a utilização de um simples browser é muito interessante dado que não implica a instalação de grandes pacotes de software e reduz as incompatibilidades entre máquinas. Por outro lado, a aplicação passa a estar disponível em qualquer computador com uma ligação à Internet. As actualizações do sistema podem ser feitas com maior frequência sem

## 2.1. ALGUMAS RAZÕES QUE LEVARAM À ACTUALIZAÇÃO DO SISTEMA.15

que o utilizador tenha de reinstalar qualquer tipo de software de *upgrade*. Estas actualizações podem ser feitas, inclusive, de forma totalmente transparente para o utilizador, o que induz neste último um maior grau de confiança no sistema.

Um outro factor de peso é o crescimento vertiginoso da utilização da Internet nas escolas e nas residências. Por exemplo, de acordo com números disponibilizados pelo Instituto Nacional de Estatística sobre o tema Sociedade de Informação e do Conhecimento na publicação Sociedade de Informação e do Conhecimento, capítulo Sociedade da Informação - Escolas (Ensino não Superior), o número de computadores disponíveis nos estabelecimentos públicos no ano lectivo de 2001/2002 era de 43847; já o número de escolas com ligação à Internet de 1997 a 2001 é apresentado na figura 2.2 [INdE01].

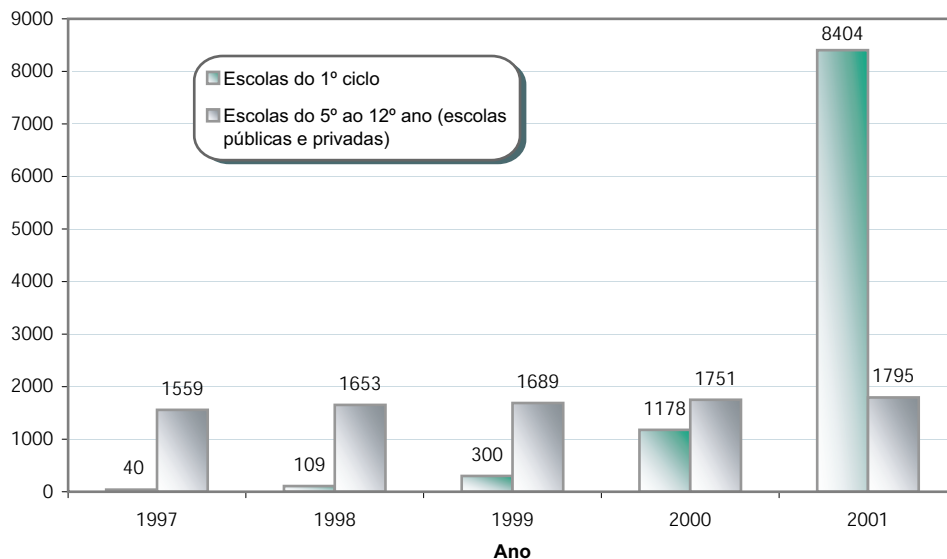


Figura 2.2: Número de escolas com ligação à Internet de 1997 a 2001

Num outro inquérito levado a cabo pelo INE em colaboração com o Observatório das Ciências e das Tecnologias, especificamente o inquérito à utilização das tecnologias de informação e comunicação por parte das famílias, pode ler-se “No caso concreto dos estudantes, a utilização de computadores é feita, em percentagens

*muito próximas, na escola/universidade (48%) e em casa (45%). Já a ligação ao World Wide Web estabelece-se nos locais de ensino - 51% dos estudantes fazem-na na escola/universidade e 37% em casa". O mesmo estudo apresenta resultados que "indicam um progressivo aumento da posse de computadores pelas famílias, bem como a ligação destas à Internet, neste caso com um crescimento mais acentuado".*

Por fim, a criação de um sistema de informação que centralize num único ponto todos os dados, de acesso universal, quer para os utilizadores, quer para a equipa de gestão é certamente uma razão que por si só justifica a sua criação.

## **2.2 Ferramentas e sistemas para ensino-a-distância**

Fala-se muito na nova Sociedade de Informação e na necessidade de acompanhar a actual revolução Social e Económica. A conotação desta nova sociedade com a educação, tem também levado investigadores e pedagogos a pensar em soluções e mecanismos para fazer chegar a informação aos *novos* estudantes. Mas a verdade, é que o problema da educação é um problema antigo que se repete ao longo da história da nossa sociedade.

A necessidade de transmissão de conhecimentos não é um problema moderno. O transporte de informação, esse sim, tem evoluído ao longo dos tempos. Os novos meios digitais de comunicação e transmissão de informação, têm de ser bem estudados e moldados no sentido de se retirar deles o maior proveito. A Internet tem vindo a afirmar-se como o meio por excelência de transmissão de informação. Além disso, o crescimento das capacidades dos browsers permitiu que, das páginas HTML iniciais muito simples, se passasse a poder disponibilizar aplicações distribuídas com funcionalidades complexas .

Este papel de meio de transporte tem sido amplamente usado e levou ao aparecimento de novas ferramentas de ensino a distância. Neste paradigma, o professor deve manter-se como elemento chave no processo de aprendizagem. No entanto, terá de dedicar uma maior fatia de tempo à pesquisa e até mesmo ao desenvolvimento de novos conteúdos electrónicos para disponibilizar aos alunos.

Apesar da sua disseminação, a verdade é que os sistemas informáticos são ainda mal recebidos por uma boa parte dos professores. As gerações mais velhas, não possuem formação específica nesta área e não são motivadas à sua utilização. Por outro lado, em alguns casos, nota-se que na formação de novos professores não é valorizada suficientemente a utilização das novas tecnologias. As camadas mais jovens geralmente têm uma habilidade quase inata no que toca à utilização dos sistemas informáticos e das novas tecnologias em geral. Este é um factor que pode provocar algum desconforto aos professores. As diferenças entre gerações são por vezes o principal entrave à utilização de um computador na sala de aula.

Mesmo que a utilização de computadores na sala de aula seja possível, o professor terá sempre de contar com um esforço extra. A edição de conteúdos electrónicos para a internet, ainda que tenha melhorado muito nos últimos anos, não é tão simples como a edição de documentos de texto, muito particularmente quando se trata de documentos científicos - usualmente muito exigentes quanto ao rigor da escrita e da composição de imagens. Veja-se o caso da escrita de fórmulas matemáticas, que exige um cuidado especial na sua representação gráfica que não ocorre na escrita de um romance.

Em suma, o professor para aderir às novas tecnologias deve contar com um grande investimento na sua auto-formação. Além disso, a rápida evolução das tecnologias, exige uma actualização constante.

Porém, além das questões meramente tecnológicas, outros problemas se levantam. Como rentabilizar de forma mais eficiente o esforço despendido pelos professores a editar novos conteúdos, sempre que se fazem actualizações nos meios de comunicação? Como evitar que os mesmos assuntos sejam escritos repetidamente por pessoas diferentes? Como reaproveitar os conteúdos de outras pessoas?

Toda a interacção dos alunos e professores com estas ferramentas é convenientemente armazenada numa base de dados. Posteriormente todos os dados são tratados. Este tratamento final dos dados, será a base da determinação dos perfis de conhecimento dos alunos. Pretende-se que todo este processo funcione em tempo real sem necessidade de qualquer tipo de intervenção no sistema ao nível administrativo.

Existe no mercado um conjunto elevado de ferramentas de gestão e apoio ao e-Learning. Em geral, a maior parte destas ferramentas apresenta um leque elevado de funcionalidades. Algumas respondem apenas a um tipo de função. Nestes termos, é possível classificar este tipo de aplicações do seguinte modo:

- Ferramentas integradas de trabalho colaborativo (Integrated Collaborative Tools)

*Apresentam um conjunto elevado de funcionalidades quer para o aluno quer para o docente ou tutor.*

- Ferramentas de conferência (Conferencing tools)

*Baseiam-se essencialmente nos sistemas de comunicação síncrona ou assíncrona por texto, voz ou imagem.*

- Sistemas de gestão de cursos (Course Management Systems)

*São aplicações que permitem a gestão completa de conteúdos e de utilizadores.*

- Questionários (Pooling)

*Permitem a construção de testes usualmente à custa de uma grande base de dados de questões.*

Em geral o que se pode observar é que as várias ferramentas disponíveis são sub-utilizadas. Em quase todos os casos, nota-se que houve um grande investimento no que diz respeito às questões tecnológicas de comunicação a distância. Quase sempre, isso significa que as questões práticas e a vertente mais ligada ao ensino foi de alguma forma descurada.

No que diz respeito à avaliação de conhecimentos, o que têm estas aplicações a oferecer? Das várias ferramentas observadas que permitem a utilização de mecanismos de avaliação, conclui-se que a inserção de novas questões é simples e feita através da utilização de formulários onde o professor digita as questões. Do ponto de vista da representação de dados são algo limitadas e utilizam como veículo de representação de dados o HTML [RHJ99, Cora]. Usualmente, o docente pode indicar hiperligações para páginas relacionadas com as questões, ou hiperligações para imagens ou material multimédia. Para resolver o problema da representação de textos mais complexos (por exemplo onde apareçam fórmulas matemáticas) ou textos com figuras, algumas destas aplicações permitem a distribuição do enunciado num documento à parte e o aluno deverá posteriormente utilizar um formulário para responder às questões propostas.

O Lotus Learning Space, Webct e o Usolv-IT [dCI02] têm ferramentas deste tipo. As questões têm normalmente associada, de forma estática, a resposta correcta e são também elas quase sempre estáticas. O único dinamismo em alguns casos prende-se com a possibilidade de sortear um determinado número de questões a partir de uma grande base de dados previamente construída pelo docente.

Apresenta-se abaixo uma lista com algumas destas ferramentas que, não sendo

exaustiva, pode servir como referência para o leitor:

- Quiz Center [SP]
- Quiz Master [Inc]
- Advanced Survey [Sur]
- Zoomerang [Mar]
- MisterPoll [Ent]
- USolv-IT [dCI02]

Das ferramentas encontradas pertencentes ao grupo dos Sistemas de gestão de cursos, algumas das mais conhecidas são o Webct, Blackboard, Lotus Learning Space e AnlonTM (Academic e Corporate). Estas ferramentas são muito completas e permitem a gestão de conteúdos e criação de vários tipos de testes: diagnóstico, avaliação formativa e avaliação sumativa.

No que diz respeito às ferramentas de avaliação, as aplicações analisadas têm essencialmente as mesmas características. Permitem, entre outras coisas, a criação de testes de escolha múltipla, falso-verdadeiro, correspondências, resposta curta, etc. Alguns destes formatos permitem atribuir a classificação de forma automática. A inserção das questões é uma tarefa do docente.

O apuramento dos resultados pode ser feito de forma automática, dependendo da escolha do tipo de teste. Quer o teste seja de escolha múltipla ou do tipo falso-verdadeiro, os sistemas avaliados permitem a obtenção de resultados que não vão muito além da contabilização do número ou percentagem de acertos.

O docente pode ainda definir, em alguns casos, um conjunto de questões para uma mesma pergunta. Desta forma o sistema encarregar-se-á de construir um teste diferente por aluno de acordo com um algoritmo aleatório de sorteio de questões.

O PmatE contribui, com as suas aplicações, para a resolução de alguns dos problemas apresentados. Com uma filosofia de aprendizagem pela avaliação desenvolveram-se e mantêm-se em desenvolvimento ferramentas de ensino assistido por computador. Estas aplicações baseiam-se sempre em conteúdos controlados por especialistas em cada uma das matérias. Fez-se um levantamento exaustivo e por fases dos vários assuntos abordados no 3<sup>o</sup> ciclo do ensino básico e seguintes, construindo-se uma biblioteca de modelos que contava em Fevereiro de 2004 com aproximadamente 550 MODELOS de matemática, embora o sistema suporte facilmente outro tipo de conteúdos. A ideia é disponibilizar ferramentas que permitam a pesquisa e utilização fácil destes MODELOS na construção de aplicações de estudo e auto-avaliação dos conhecimentos dos alunos em determinadas matérias.





## Capítulo 3

# SA<sup>3</sup>C - do modelo à aplicação final

O SA<sup>3</sup>C tem uma filosofia de funcionamento muito própria e, enquanto produto, pode enquadrar-se nas aplicações de avaliação de conhecimentos<sup>1</sup>. O principal enfoque da aplicação é a aprendizagem pela avaliação, tendo o desafio permanente como um forte elemento motivador.

Veremos neste capítulo como são formatados os conteúdos desta aplicação desde a sua idealização à forma que tomam na aplicação final. No decorrer do texto serão apresentadas as metodologias de desenvolvimento de conteúdos.

Serão também apresentados os conceitos de: modelo, prova, Linguagem de Representação de Modelos e a forma como são catalogados e avaliados os modelos.

### 3.1 Modelo

A unidade atômica deste sistema de informação é o conceito de *modelo gerador de questões* (de agora em diante designado por MODELO para efeitos de simplicidade na

---

<sup>1</sup>As aplicações de avaliação de conhecimentos são muito frequentemente designadas na literatura inglesa de *assessment tools*

escrita) [VCA01]. É precisamente este conceito de MODELO, o principal responsável por duas das mais importantes características do SA<sup>3</sup>C: a flexibilidade e a modularidade. Sendo a aplicação essencialmente uma ferramenta de avaliação, é natural esperar que na sua base residam questões.

Escolheu-se o tipo de *Questões de Resposta Múltipla do tipo Falso-Verdadeiro-Generalizado* [VCA01], o que significa que o utilizador é sempre confrontado com um conjunto de proposições cuja resposta só pode ser verdadeiro ou falso. Pelo facto de serem do tipo Falso-Verdadeiro-Generalizado, todas as respostas podem ser falsas ou verdadeiras ou coexistirem de modo aleatório respostas falsas e verdadeiras.

Algo mais caracteriza as questões usadas no SA<sup>3</sup>C, nomeadamente o facto de todas elas serem geradas dinamicamente em tempo de execução. Ao contrário da grande maioria das ferramentas existentes para este tipo de aplicação, o SA<sup>3</sup>C possui a capacidade de apresentar ao utilizador, em cada interacção deste com o primeiro, uma concretização diferente de uma questão. Isto é possível, porque em vez de se armazenarem questões estáticas numa base de dados, armazenam-se geradores de questões. O gerador de questões é o MODELO.

Uma concretização de um MODELO resultará num enunciado que é sempre composto por um *texto comum* + *conjunto de 4 itens* saídos aleatoriamente de entre  $k$ , com  $k \geq 4$ . A junção do *texto* com um *item* resulta numa proposição.

Com um elemento atómico de informação com as características do MODELO, obtém-se um elevado grau de liberdade para a construção das aplicações finais.

As opções de “resposta” incluídas num MODELO, estão normalmente associadas com os erros mais comuns cometidos pelos alunos, mas resultam também do conceito que se pretende avaliar e dos pré-requisitos inerentes a esse mesmo conceito.

Sendo o conjunto de *itens* em geral maior que 4, são sorteadas quatro hipóteses de resposta em tempo de execução. Serão estas quatro hipóteses que, com o texto comum, formarão a questão a ser colocada ao utilizador. Tanto o texto como os vários itens são compostos por expressões, parâmetros e caixas, isto permite fazer com que dificilmente se chegue a duas concretizações iguais.

Para minimizar a resposta à sorte, o MODELO contém ainda um sistema que permite aumentar as hipóteses de formulação dos enunciados possíveis. Este sistema designou-se por *diagrama de caixas*. Este sistema permite, também, oferecer um maior dinamismo às questões apresentadas e como consequência, aumenta-se o grau de atenção dos alunos. Outra vantagem da sua utilização prende-se com o equilíbrio do padrão de falsos e verdadeiros.

Os *diagramas de caixas* permitem a sistematização de procedimentos em relação ao conceito de variação na formulação das questões e na diversificação de sub-objectivos em relação ao objectivo principal. Subjacente a este sistema há produtos cartesianos e operações de concatenação.

Designa-se por *parâmetro* tudo o que é susceptível de ser escolhido aleatoriamente, num certo domínio. Os parâmetros podem ser números, expressões, sinais, etc.. Definem essencialmente os valores numéricos e os conjuntos onde eles podem assumir valores (o seu domínio), sendo escolhidos adequadamente, do ponto de vista matemático e educacional.

A formulação dos MODELOS deve prever a possibilidade da ocorrência de repetições em respostas devido à construção dinâmica dos parâmetros. O motor do sistema contém ainda alguns mecanismos que eliminam, quando necessário, as respostas iguais.

Uma vez construído o MODELO passa-se à fase de validação das asserções. A validação consiste em indicar as condições em que uma determinada asserção formal se torna verdadeira. Estas condições dependem dos parâmetros, dos sinais e das expressões geradas.

A figura 3.1 mostra os principais pormenores de um MODELO. As caixas são etiquetadas com uma codificação do tipo  $Ci.j$  com  $i$  a identificar o número da resposta e  $j$  o número da caixa.

Construído e validado o MODELO, inicia-se o processo de introdução no sistema informático. Este é um processo que pela sua natureza carece de alguns cuidados. A tradução do papel para o formato electrónico é um processo complexo e muito susceptível de erros. Por este motivo, idealizaram-se mecanismos que permitem minimizar a introdução de erros no processo.

## 3.2 Linguagem de Representação de Modelos

A complexidade inerente aos MODELOS exige que o processo de transcrição para o sistema informatizado obedeça a critérios muito claros para desta forma minimizar os erros processuais. Por outro lado, garante-se que qualquer elemento da equipa de transcrição de MODELOS pode facilmente perceber o alinhamento lógico do código informatizado do MODELO.

Por forma a estabelecer um conjunto de regras elementares de programação, desenvolveu-se uma Linguagem de Representação de Modelos (LRM) [ABF03, IFB<sup>+</sup>03]. Esta linguagem foi desenvolvida com base em muitos dos conceitos do L<sup>A</sup>T<sub>E</sub>X por ser muito familiar à comunidade científica e muito particularmente aos matemáticos.

Os MODELOS são transcritos do papel para a LRM e armazenados na base de dados. Para a interpretação da linguagem LRM foi desenvolvido um processador de LRM. A LRM utiliza um conjunto de símbolos com funções distintas para representação e concretização. Para iniciar uma expressão dinâmica, usa-se a seguinte locução:

$$\backslash\backslash Form \text{ expressão a processar} \text{ } \quad (3.1)$$

A expressão a processar é também uma frase escrita em LRM. Esta tem vários construtores para a definição de frações, potências, raízes, módulos, etc.. São também definidas as caixas do *diagrama de caixas* e ainda as variáveis que são substituídas por valores de acordo com as restrições impostas no MODELO. O processador de LRM interpreta as instruções passadas e posteriormente, caso necessário, converte a expressão final em MathML<sup>2</sup>.

Este sistema permite à equipa de programação de modelos um ritmo de trabalho muito elevado e acima de tudo uma coerência muito grande ao nível do estilo de programação. Desta forma um mesmo MODELO pode ser programado e revisto por várias pessoas. Naturalmente, existem alguns casos especiais difíceis de expressar directamente na LRM. Estes casos são resolvidos ao nível da programação sem recorrer, portanto, à LRM.

### 3.3 Codificação de modelos

Cada MODELO envolve um grande trabalho de pesquisa e catalogação que só é possível se criado por uma equipa coesa e altamente qualificada na área científica à

---

<sup>2</sup>MathML (Mathematical Markup Language) é um standard para a representação de texto matemático. Permite a descrição de fórmulas matemáticas de um ponto de vista semântico ou meramente gráfico. *Vide* na secção 4.2.2 uma descrição mais detalhada sobre a MathML.

qual o MODELO diz respeito. É um trabalho que, por um lado, exige uma pesquisa exaustiva e o estudo detalhado dos conceitos a abordar, por outro lado, a sua catalogação num sistema multidimensional.

O SA<sup>3</sup>C vai permitir determinar os perfis de conhecimento dos seus utilizadores. A extracção do conhecimento da base de dados em relação aos perfis de conhecimento dos alunos é feita à custa de um sistema de codificação de modelos.

Ao entrarem no sistema, todos os modelos são codificados de acordo com uma estrutura do tipo [VCO03]:

O sistema contém uma árvore multidimensional com a estrutura definida na figura 3.2. Os modelos são colocados criteriosamente nesta árvore de objectivos.

Cada uma das respostas associadas ao modelo é também codificada segundo uma estrutura semelhante, embora, neste caso, exista ainda mais alguma informação. Cada resposta pode, no contexto do modelo, ter um objectivo particular, pelo que às respostas é ainda possível associar mais do que um objectivo. Os vários objectivos, não devem no entanto sair do âmbito do objectivo da questão. O objectivo do modelo fica, por conseguinte, automaticamente associado à resposta.

A codificação de respostas baseia-se na mesma árvore de objectivos dos MODELOS. Desta forma, garante-se que existe sempre um vértice que une qualquer resposta a um conjunto de MODELOS e não exclusivamente ao MODELO ao qual a resposta diz respeito.

Pretende-se que este mecanismo funcione como ponto de partida para a definição de um algoritmo de selecção de MODELOS, automatizado. A par do propósito de traçar o perfil de conhecimento dos alunos, este é um dos objectivos mais ambiciosos

deste projecto - fornecer mecanismos de navegação automática na árvore de MODELOS.

A codificação da árvore de objectivos está de acordo com o thesaurus da American Mathematical Society e do INSPEC da IEE (Institution of Electrical Engineers) [Soc, oEE].

### 3.4 Avaliação de modelos

Uma vez programados, os MODELOS têm de ser avaliados. A sua avaliação deve ser feita, pelo autor do MODELO dado que é quem o construiu. São também convidadas várias pessoas com bagagem científica e pedagógica adequadas a participar neste processo. Para o efeito e para tornar este procedimento mais expedito, criou-se uma página Web para avaliação de modelos, onde estão disponíveis todos os MODELOS existentes no sistema. Esta página faz parte da área de administração do sistema, à qual têm acesso apenas os utilizadores com perfil de administrador.

O processo de avaliação e programação de MODELOS é recursivo e são necessárias algumas iterações até o MODELO estar convenientemente testado, pronto para entrar em produção.

Devido ao *diagrama de caixas* e aos parâmetros de domínio variável, pode haver combinações não previstas por quem idealizou o MODELO. É ainda necessário garantir, portanto, que todas as asserções são testadas e que cumprem os objectivos idealizados.

Todo o percurso do MODELO, desde a concepção à utilização final, é registado. Devido à necessidade de escrever frequentemente fórmulas matemáticas, optou-se



pelo L<sup>A</sup>T<sub>E</sub>X para o registo de toda a documentação referente à criação e alteração dos MODELOS. Os ficheiros DVI (formato de impressão) dos MODELOS estão também disponíveis na Internet na aplicação de gestão do SA<sup>3</sup>C.

Quando o MODELO atinge a maturidade necessária, isto é, atingiu um nível de confiança suficiente para entrar em exploração, passa para uma Base de Dados definitiva. A partir desse instante pode ser utilizado em provas. As provas podem variar no formato, objectivos, destinatários, etc, mas têm em comum o sistema de geração de questões.

### **3.5 Provas: treino, diagnóstico, avaliação e competição**

Uma prova, tem associado um conjunto de parâmetros correspondentes a dados que podem ser fixados pelo responsável pela prova, a saber: o nome da prova, o tipo de prova, o seu autor, o tipo de equipa, o número de tentativas de resposta autorizado, o tempo total da prova e, quando aplicável, o número máximo de equipas por escola, a data da prova, a data final de inscrição na prova e o número máximo de tentativas de entrada em prova. As provas são todas de acesso condicionado.

O utilizador pode participar num conjunto limitado de provas. O acesso às provas é feita através de equipas com um utilizador singular ou em grupos de utilizadores. Ao identificar inequivocamente os utilizadores, é possível associar todas as provas realizadas pelos utilizadores e determinar os seus perfis de conhecimento com base no histórico de navegação. Além disso é possível estabelecer fronteiras consoante o grau de escolaridade dos utilizadores, i.e., é possível restringir o acesso dos utilizadores às provas consoante o seu grau de conhecimentos.

Associada a uma prova existe sempre uma grelha de MODELOS. A grelha de MODELOS associa a cada nível da prova um conjunto de MODELOS. Os níveis constituem a ordem pela qual os MODELOS são apresentados. A um nível podem estar associados um ou mais MODELOS. No caso de existir mais do que um MODELO será sorteado um MODELO para esse nível. Este é um outro mecanismo que permite assegurar que os utilizadores, a cada interacção com o sistema, encontram sempre novas questões. Para tal, é necessário um cuidado redobrado ao seleccionar MODELOS para a prova, pois pode dar-se o caso de um mesmo nível conter MODELOS muito diferentes em nível de dificuldade, tema, objectivos, etc. Reforça-se desta forma o conceito de desafio constante ao utilizador.

Sendo os MODELOS os objectos de ensino do sistema, as provas são, do ponto de vista da aplicação, os seus consumidores. As provas podem ser de vários tipos, consoante o fim a que se destinam. Os vários tipos de provas presentemente disponibilizadas são:

**“Treino”**. São, como o nome indica, disponibilizadas aos alunos para treino - quer como preparação para a final de uma prova, quer como preparação para avaliação numa disciplina. O acesso a estas provas é feito por equipas singulares e os alunos podem entrar o número de vezes que quiserem. Nestas provas cada MODELO é apresentado num único ecrã. Os alunos só avançam para o ecrã seguinte depois de responderem correctamente à questão apresentada.

**“Diagnóstico”**: São provas que permitem efectuar um diagnóstico dos conhecimentos dos alunos num determinado domínio. Estas provas são individuais e de acesso único e tal como as provas de avaliação apresentam todos os MODELOS num único ecrã.

**“Desafio”**. São as provas usadas nas finais de competições. Alguns exemplos de provas deste tipo são a EQUA<sub>mat</sub>, mat12 e MINImat. A entrada neste tipo de provas é feita por equipas de 2 elementos - estimulando o trabalho em equipa. As equipas participantes só podem efectuar uma prova. Têm ainda a particularidade de apenas permitirem a participação de escolas que tenham efectuado a sua pré-inscrição na prova. Tal como nas provas de treino, é apresentado um MODELO em cada ecrã e as equipas participantes devem responder correctamente a todos os itens de uma questão para avançar para a questão seguinte.

**“Avaliação”**. São provas utilizadas nos momentos de avaliação de uma disciplina. A participação é individual e a entrada na prova exige a introdução de uma chave de acesso de um vigilante. Estas provas apresentam num mesmo ecrã as várias concretizações dos MODELOS sorteados a partir da grelha de MODELOS da prova. O utilizador pode responder livremente a todas as questões e submete a prova para cotação no final. A cotação pode ser dada de imediato ou não, consoante a opção feita pelo professor. Esta cotação pode ainda ser efectuada de acordo com várias critérios possíveis. Utilizando este instrumento de trabalho, os professores não estão obrigados a usar um determinado critério de avaliação, dado que a aplicação prevê a utilização de vários critérios.

A plataforma do SA<sup>3</sup>C serve também alguns dos eventos do PmatE como por exemplo a MINImat, a EQUA<sub>mat</sub>, ou a Mat12. Ambos têm picos de carga bastante elevados. Por exemplo durante as 3 horas de final da EQUA<sub>mat</sub> são geradas perto de 1000 provas. A ser utilizado por uma boa parte das escolas do país, o SA<sup>3</sup>C poderá atingir picos de várias centenas de utilizadores em simultâneo. No sentido de otimizar a gestão dos recursos entre o servidor e o cliente, parte da lógica da aplicação é executada no lado do cliente.

Delegar parte do código ao cliente introduz no sistema um problema adicional -

segurança. Existem essencialmente dois grandes problemas, um é o refrescamento da página a ser visualizada, o outro prende-se com a necessidade de enviar para o cliente as soluções da questão que está a visualizar. Note-se que ao enviar as soluções para o cliente economizam-se alguns pedidos ao servidor para validar as respostas dadas pelo utilizador.

Tanto nas provas do tipo desafio como do tipo treino, é gerado bastante tráfego na rede. Estas provas são compostas por vários ecrãs, que se vão sucedendo uns aos outros consoante o aluno vai respondendo correctamente às questões apresentadas. Os ecrãs em si não contêm muitos dados, mas quando várias centenas de utilizadores estão em linha, o servidor da aplicação está constantemente a criar e a enviar novos ecrãs. Uma vez que as soluções e o código de validação das respostas são enviados em simultâneo com o ecrã, se o aluno errar não vai sobrecarregar o servidor com mais um pedido. Este processo obriga no entanto à disponibilização de mecanismos que dificultem o acesso às soluções de forma ilícita.

O problema de refrescamento é óbvio: como de cada vez que é feito um novo pedido é gerada uma nova concretização do problema, o utilizador pode ser levado a gerar uma nova proposição sempre que a actual não lhe agrade, a menos que sejam impostos alguns limites.

## MODELO

Tipo:0

Data : 25 de Maio de 2004

Data : 08/04/03

**Tema**

Números Racionais

**Objectivos**

Ordenação de números racionais

**Notações e Abreviaturas** $\circ^n, \bullet^n \in \{+, -\}; \circ^n = -\bullet^n$ **Domínio dos Parâmetros**

$a, c \in \{1, \dots, 50\} \cap \mathbb{N}$   
 $b, d \in \{1, \dots, 20\} \cap \mathbb{N}$   
 $b \neq d$   
 $k \in \mathbb{N}$   
 $2 \leq k \leq 9$   
 Se  $a \bmod b = 0$  então  $c \bmod d \neq 0$

**Texto**O número racional  $\circ^1 \frac{a}{b}$ 

	Respostas		V.L.	Validação	Obj.								
R1	<table border="1"> <tr><td>é</td><td>C1.1</td></tr> <tr><td>não é</td><td>C1.2</td></tr> </table>	é	C1.1	não é	C1.2	menor que	V. se	$C_{11} \wedge  (\circ^2 = + \wedge ((\circ^1 = + \wedge ad < cb) \vee \circ^1 = -)) \vee (\circ^1 = \circ^2 = - \wedge ad > bc) $ $\vee C_{12} \wedge  (\circ^2 = - \wedge (\circ^1 = + \vee (\circ^1 = - \wedge ad \leq cb)) \vee (\circ^2 = + \wedge \circ^1 = + \wedge ad \geq bc) $					
é	C1.1												
não é	C1.2												
R2	<table border="1"> <tr><td>é</td><td>C2.1</td></tr> <tr><td>não é</td><td>C2.2</td></tr> </table>	é	C2.1	não é	C2.2	maior que	V. se	$C_{21} \wedge  (\circ^2 = + \wedge \circ^1 = + \wedge ad > cb) \vee (\circ^2 = - \wedge (\circ^1 = + \vee (\circ^1 = - \wedge ad < bc)) $ $\vee C_{22} \wedge  (\circ^2 = + \wedge (\circ^1 = - \vee (\circ^1 = + \wedge ad \leq cb)) \vee (\circ^2 = - \wedge \circ^1 = - \wedge ad \geq bc) $					
é	C2.1												
não é	C2.2												
R3	<table border="1"> <tr><td>é</td><td>C3.1</td></tr> <tr><td>não é</td><td>C3.2</td></tr> <tr><td><math>\circ^1 \frac{ka}{kb}</math></td><td>C3.3</td></tr> <tr><td><math>\circ^1 \frac{ka}{(k-1)b}</math></td><td>C3.4</td></tr> </table>	é	C3.1	não é	C3.2	$\circ^1 \frac{ka}{kb}$	C3.3	$\circ^1 \frac{ka}{(k-1)b}$	C3.4	igual a	V	$(c3.1 \wedge c3.3) \vee (c3.2 \wedge c3.4)$	
é	C3.1												
não é	C3.2												
$\circ^1 \frac{ka}{kb}$	C3.3												
$\circ^1 \frac{ka}{(k-1)b}$	C3.4												
R4	<table border="1"> <tr><td>é</td><td>C4.1</td></tr> <tr><td>não é</td><td>C4.2</td></tr> </table>	é	C4.1	não é	C4.2	um número inteiro	V. se	$(C4.1 \wedge a \bmod b = 0) \vee (C4.2 \wedge a \bmod b \neq 0)$					
é	C4.1												
não é	C4.2												

Figura 3.1: Parte de um modelo.

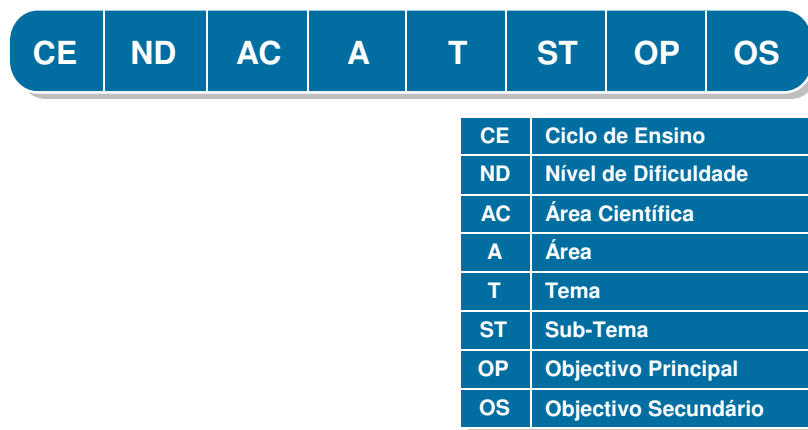


Figura 3.2: Estrutura dos códigos dos modelos.

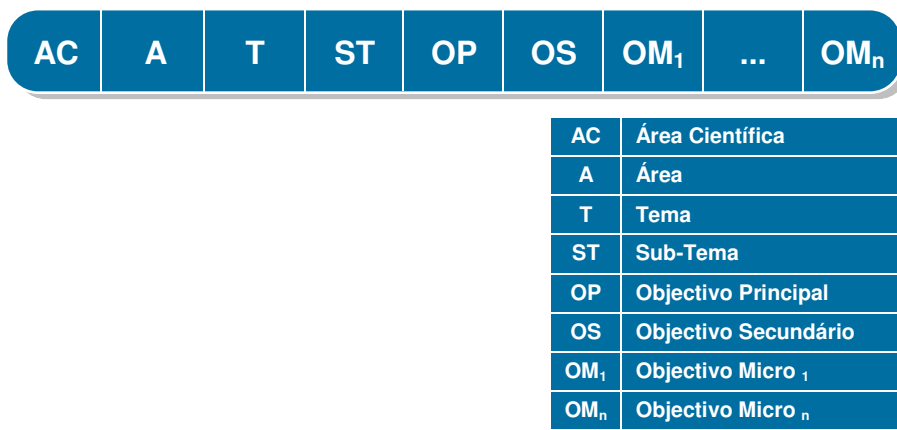


Figura 3.3: Estrutura do código de objectivos das respostas.



# Capítulo 4

## Tecnologias usadas

Para suportar a plataforma do SA<sup>3</sup>C, são necessárias várias tecnologias nos diversos estágios de desenvolvimento. Desde a escolha do Hardware ao Software, da criação de modelos até à sua apresentação num produto final, são utilizadas diferentes tecnologias que, conjugadas, resolvem os vários problemas de desenvolvimento. Neste capítulo serão apresentadas as tecnologias consideradas mais relevantes do SA<sup>3</sup>C, até agora desenvolvidas.

### 4.1 Hardware

O desenvolvimento de um sistema de informação com a envergadura do SA<sup>3</sup>C ocorre, obrigatoriamente, em várias fases. Por este motivo, é necessário contar com estratégias que permitam acrescentar novos serviços e reformular os serviços disponíveis sem afectar o funcionamento normal das aplicações que entretanto já se encontrem disponíveis.

Dependendo do tipo de serviços, bem como da sua aceitação por parte dos utilizadores, o dimensionamento do conjunto de máquinas a utilizar na manutenção de um sistema de informação pode ser alterado ao longo do seu tempo de vida.



Por outro lado, a antevisão da introdução de novas tecnologias e de novas soluções, inicialmente não previstas, pode também condicionar a escolha da plataforma de suporte de um sistema de informação.

A experiência adquirida pela aceitação das actuais aplicações do PmatE, deixam antever que a aceitação do SA<sup>3</sup>C por parte dos eventuais utilizadores será muito grande. Sendo baseada no modelo aplicacional do PmatE, uma das principais características do SA<sup>3</sup>C é o crescimento constante da sua base de dados de MODELOS. Para garantir a qualidade do serviço prestado, é necessário garantir que as interrupções de manutenção da aplicação sejam curtas. Utilizaram-se inicialmente duas máquinas. Uma essencialmente para o desenvolvimento da aplicação e a outra para a disponibilização dos serviços.

Durante os anos de 2002 e 2003, a máquina que disponibilizou os serviços para o exterior era um servidor com duplo processador Pentium a 1GHz. Disponha de 1512MB de RAM e dois discos SCSI de 40GB.

A máquina de desenvolvimento usada tem uma arquitectura baseada num processador Pentium 4 a 2GHz, 512 MB de memória RAM e disco de 80GB.

Além destas duas máquinas existe também um conjunto de máquinas utilizadas no desenvolvimento dos vários componentes da aplicação.

Tal como foi dito, esta foi a arquitectura de Hardware utilizada durante a primeira fase de desenvolvimento da aplicação; contudo, esta solução não se coaduna com as actuais necessidades da aplicação<sup>1</sup> impostas pela carga de utilização.

---

<sup>1</sup>Para uma referência completa sobre o estudo efectuado sobre a carga de utilização da aplicação, consulte-se o capítulo 6 desta dissertação.

O crescimento do número de utilizadores, o surgimento de novos serviços e novos projectos de apoio a disciplinas na Universidade de Aveiro, condicionaram o desempenho do servidor. Para o correcto funcionamento da aplicação e para a criação de redundância na arquitectura de suporte que permita, por exemplo, contar com mecanismos de actuação de emergência em caso de falhas no Hardware, foi necessário redimensionar a plataforma de Hardware.

O aumento da complexidade de alguns dos MODELOS a desenvolver, por exemplo, para o apoio às disciplinas de Cálculo I, II, Métodos Numéricos e outras, requer a utilização de aplicações de cálculo científico especializadas. A escolha recaiu sobre o webMathematica 2.0 [WR03] da Wolfram Research.

Esta versão instalada do webMathematica baseia-se no motor do Mathematica 4.2. O webMathematica funciona como uma camada de gestão de acessos ao Mathematica. Este software, além de permitir a utilização da MathML e do SVG como tecnologias de representação, tem também um excelente motor de cálculo. É uma aplicação Java que funciona num servidor de Servlets/JSP [SM]. O servidor recomendado pela Wolfram é o Tomcat [Fou].

A introdução do webMathematica bem como o gradual aumento de utilizadores, fez com que, rapidamente, se tivesse de pensar numa solução alternativa de hardware para suportar todos os serviços necessários à aplicação. De todos os serviços necessários à normal execução da aplicação, os principais são: 2 servidores web (IIS e Tomcat), servidor de gestão de base de dados (SQLServer), webMathematica, os componentes ActiveX de negócio da aplicação, as páginas desenvolvidas para a interface da aplicação, o sistema de partilha de ficheiros para a gestão de MODELOS e ainda aplicações de *workflow* e gestão de trabalho colaborativo.

Sendo um conjunto elevado de objectos a gerir, foi necessário pensar em formas

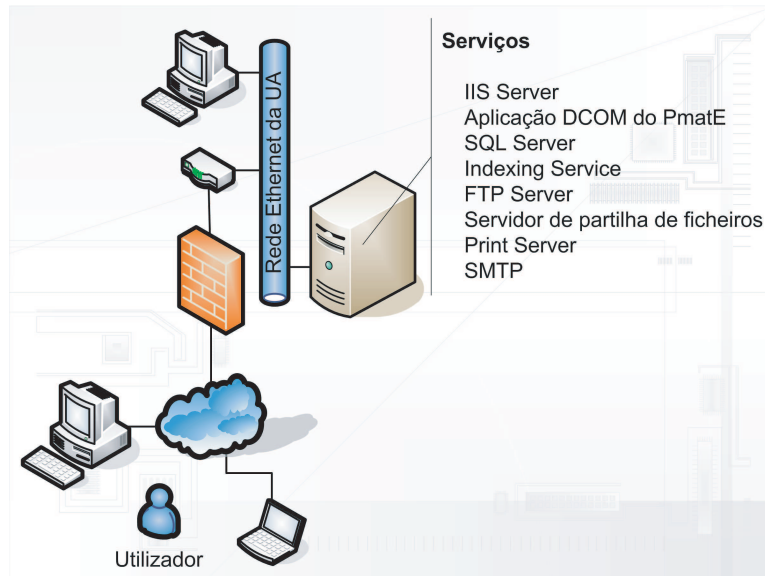


Figura 4.1: Arquitectura de Hardware e distribuição inicial de serviços.

de os separar para efeitos de eficiência do seu funcionamento. A divisão principal de serviços foi feita da seguinte forma:

- Servidor dedicado para o sistema de gestão da base de dados.
- Servidor web e lógica de negócio da aplicação.
- Servidor de aplicações de cálculo avançado.

A solução passou pela constituição de um núcleo de servidores distribuindo os vários serviços, para já, por 3 máquinas (de acordo com o representado na figura 4.2). Esta arquitectura permite criar mecanismos de redundância ao nível aplicativo e ainda prevê o seu crescimento no sentido de otimizar eventuais pontos de estrangimento da aplicação. Qualquer dos três servidores pode ser replicado se necessário.

A nova plataforma é constituída por uma máquina contendo o servidor IIS e os componentes ActiveX de processamento de modelos - representado na figura como

o servidor Luckyluke. Um segundo servidor mantém o sistema de gestão de base de dados, na figura com o nome de Rantanplan. A terceira máquina contém o servidor de Servlets/JSP, o webMathematica e o servidor local de DNS (Domain Name Server). As máquinas comunicam entre si através de uma rede local disponível exclusivamente para servir as comunicações entre os serviços atendidos por cada uma delas. Optimiza-se desta forma o tráfego gerado na comunicação entre serviços. O primeiro servidor será a porta de acesso às várias aplicações do SA<sup>3</sup>C.

Para todos os efeitos, do exterior apenas se apresenta uma aplicação web disponível em <http://pmate.mat.ua.pt> ou <http://pmate.ua.pt>. Embora o servidor de gestão de base de dados tenha também acesso via rede local da UA, este acesso (local) existe devido à necessidade da equipa de desenvolvimento comunicar com o sistema de gestão de dados aí instalado, bem como devido à previsão de vir a disponibilizar através deste servidor uma aplicação de *workflow* (Sharepoint).

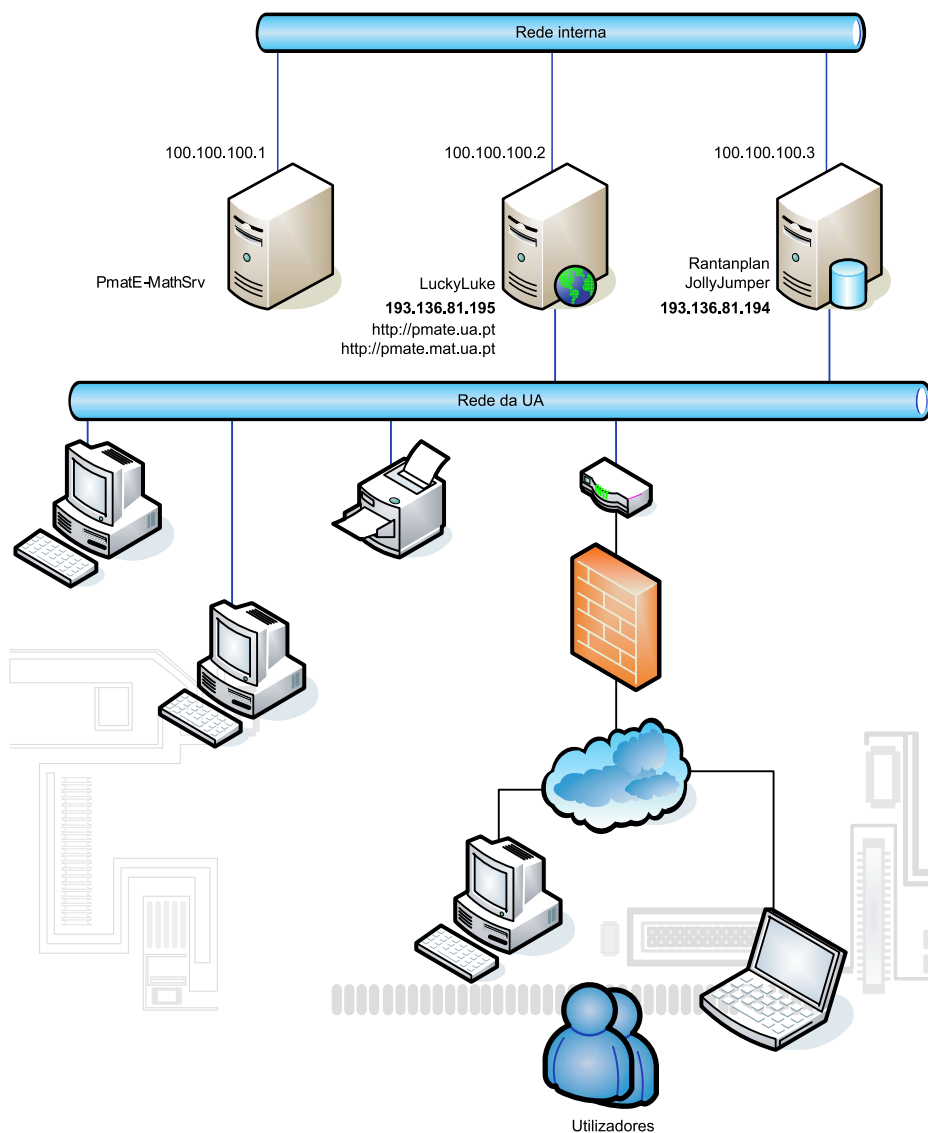


Figura 4.2: Arquitectura de Hardware.

## 4.2 Software

### 4.2.1 Suporte da plataforma ao nível de negócio

A plataforma de desenvolvimento da aplicação escolhida foi a plataforma clássica Microsoft: WISA [Pen]. Durante os primeiros anos de desenvolvimento utilizou-se

ao nível do sistema operativo o Windows 2000 Advanced Server e o Windows 2000 Server nas duas máquinas principais de desenvolvimento da aplicação. Em 2004 com a expansão da arquitectura de Hardware passou-se a usar o Windows Server 2003 [Cord] em todas as máquinas que constituem o núcleo de processamento.

Em qualquer das soluções, utilizam-se do Internet Information Services (IIS) os serviços de WEB server, FTP<sup>2</sup> e SMTP<sup>3</sup>. Além destes, são ainda utilizados o Indexing Service para indexação e pesquisa da documentação dos modelos, o SQL Server 2000, como sistema de gestão de bases de dados, e o Visual Studio 6.0 e do Visual Studio .Net 2003 para o desenvolvimento da aplicação.

### 4.2.2 Tecnologias de representação

Para a representação de dados no lado cliente são utilizadas várias tecnologias. Como linguagem de definição de documentos e especificação de hiperlinks é utilizada a norma HTML. Para a representação de texto matemático é utilizada a norma MathML e para o desenho de gráficos vectoriais a 2 dimensões é utilizada a norma SVG. A SVG, acrónimo de Scalable Vector Graphics, permite definir gráficos com vários efeitos, texturas, animação, scripting e possibilidade de efeito zoom. Os objectos descritos podem ser vectores, imagens ou blocos de texto. Os gráficos vectoriais podem ser: linhas, curvas, círculos, rectângulos, polígonos, etc. [Pau01].

A escolha destas tecnologias resulta de um estudo inicial sobre os requisitos principais da aplicação e de um estudo das várias tecnologias disponíveis que permitissem o desenvolvimento da aplicação e respondessem aos seus requisitos.

Fazia parte da lista de requisitos do sistema que a aplicação permitisse a rep-

---

<sup>2</sup>FTP - File Transfer Protocol.

<sup>3</sup>SMTP - Simple Mail Transfer Protocol

representação de texto matemático de alta qualidade num browser. Decorrente da elevada aleatoriedade dos modelos, o sistema deveria de ser capaz de representar uma formula matemática em qualquer parte do ecrã e num número desconhecido de ocorrências. Isto é, deveria ser possível ter um número variável de “ilhas” de texto matemático espalhadas numa disposição perfeitamente aleatória.

A representação de figuras com elevada qualidade e construídas em tempo real com base nos MODELOS subjacentes, fazia igualmente parte da lista de requisitos do sistema.

### 4.2.3 MathML

A representação de texto matemático num sistema informático sempre foi uma tarefa complicada. Uma das tecnologias mais utilizadas por matemáticos e cientistas em todo o mundo para a escrita de fórmulas matemáticas é o  $\text{\LaTeX}$ . Os resultados obtidos são excelentes, mas o  $\text{\LaTeX}$  apenas descreve o aspecto gráfico das fórmulas. Cientes deste problema e devido ao facto do  $\text{\LaTeX}$  não ter sido criado especificamente para funcionar na Internet, o World Wide Web Consortium (W3C) desenvolveu uma nova tecnologia baseada no XML [BPSMM], a MathML<sup>4</sup>.

Esta tecnologia foi desenhada para permitir descrever informação matemática e científica de tal forma que fosse facilmente transportável na World Wide Web. A MathML apresenta ainda algumas características muito interessantes tais como a possibilidade de representar as fórmulas matemáticas em dois formatos: PresentationMathML e ContentMathML. Isto significa que, com a MathML, é possível representar graficamente as fórmulas matemáticas, ou o aspecto semântico das mesmas, permitindo a sua interpretação por software científico ou por sistemas de reconhe-

---

<sup>4</sup>Consulte também: MathML recommendation, <http://www.w3.org/math>

cimento e síntese de voz.

A especificação 1.0 da MathML data de 07 de Abril de 1998. A recomendação actual (2.0) é de 21 de Fevereiro de 2001. A maior parte dos fabricantes de software vão aderindo a esta nova tecnologia e existem já alguns browsers capazes de fazer a interpretação da MathML, como o Mozilla e o Amaya. Por outro lado, a Microsoft e a Netscape já anunciam à anos que as versões futuras dos seus browsers vão possuir motores internos para interpretar nativamente a MathML. Só não o fizeram ainda, certamente por motivos económicos, dado que a utilização desta tecnologia se confina quase exclusivamente aos meios científicos que infelizmente ainda não têm a mesma capacidade de influência que os meios económicos.

Existem alguns componentes de visualização capazes de interpretar o  $\text{\LaTeX}$  embora a tendência ao nível mundial seja a utilização da MathML. Um dos componentes capazes de interpretar o  $\text{\LaTeX}$  é o IBMTechExplorer. Este componente interpreta também, embora parcialmente, a MathML. A IBM - empresa responsável pelo desenvolvimento deste componente - suspendeu os desenvolvimentos do IBMTechExplorer e deixou de dar suporte aos seus utilizadores.

A utilização da linguagem MathML neste momento é possível graças à utilização de tecnologias de transformação de documentos XSL - Extensible Stylesheet Language [Cla], utilizando os *browsers* Mozilla [Org] e Amaya [Ama] ou ainda usando componentes de visualização. Além do já referido IBMTechExplorer, um outro componente de visualização existente no mercado é o Mathplayer [Sci] da Design Science.

Perante as várias alternativas disponíveis, pode-se perguntar qual será a tecnologia que melhor se adapta às necessidades do SA<sup>3</sup>C?



#### 4.2.4 Utilização da linguagem MathML

Sendo a MathML uma recomendação do W3C <sup>5</sup> e conhecidas as características da própria linguagem, não restam dúvidas quanto à sua escolha face ao  $\text{\LaTeX}$ . Resta perceber qual das formas de distribuição escolher. A utilização do Mozilla ou Amaya é uma hipótese de descartar quase de imediato, dado que se tratam de browsers pouco conhecidos, difíceis de instalar que certamente obrigariam a um esforço muito grande para convencer os utilizadores a adoptá-los. Não pode esquecer-se que grande parte dos utilizadores do SA<sup>3</sup>C têm poucos conhecimentos informáticos pelo que, idealmente, a aplicação deve obrigar a um mínimo de operações de instalação para o seu correcto funcionamento.

A forma mais universal - porque não obriga à instalação de qualquer tipo de componentes - para a interpretação da MathML é a utilização da MathML *inlined* num documento XHTML [PAA<sup>+</sup>02]. Posteriormente utiliza-se a capacidade de transformação no cliente recorrendo a ficheiros XSLT que o próprio W3C fornece. O mecanismo é em tudo semelhante à transformação de documentos XML usando XSL. A figura 4.3 mostra a utilização desta tecnologia.

---

<sup>5</sup>O consorcio da World Wide Web foi criado em Outubro de 1994 para conduzir a World Wide Web a todo o seu potencial, desenvolvendo normas que promovem a sua evolução e asseguram a inter-operacionalidade entre os sistemas.

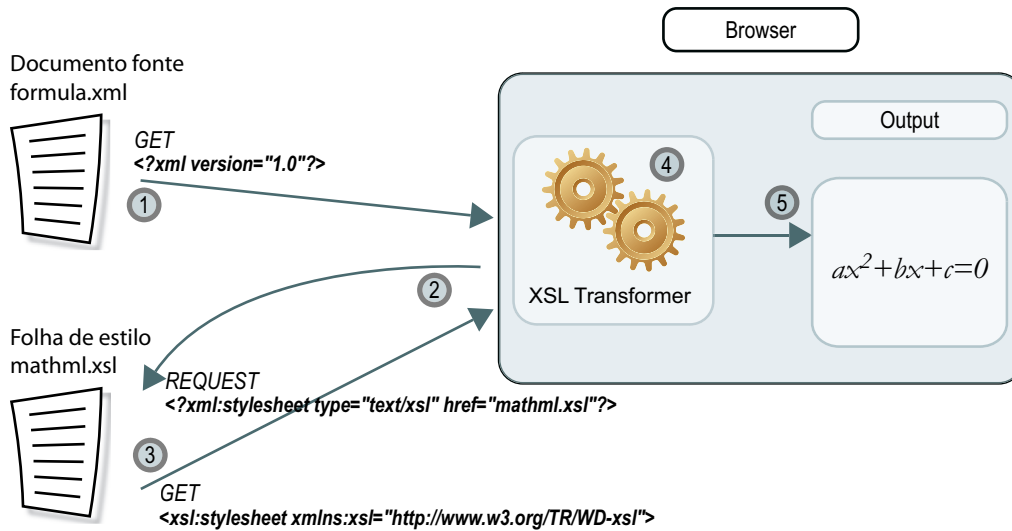


Figura 4.3: Inserção de expressão MathML usando XSLT no cliente.

O processo inicia-se quando o cliente indica que quer visualizar o documento *formula.xml* (passo 1). Este é um documento no formato XHTML com expressões MathML *inlined*. À medida que o documento começa a ser processado no lado do cliente, o navegador reconhece um pedido a uma folha de estilo e executa esse pedido (passo 2). O servidor reconhece o pedido e serve o cliente com o ficheiro *mathml.xml*. Os passos 2 e 3 podem não ser executados se o navegador tiver em cache o ficheiro XSL. Quando os dois ficheiros estão no cliente inicia-se o processo de transformação e os resultados são finalmente apresentados (passos 4 e 5).

Em geral, os resultados obtidos são satisfatórios, mas insuficientes para o nível de exigência que se pretende alcançar. De facto, existe uma série de pequenos pormenores que inviabilizam a utilização deste método. A qualidade gráfica está longe da obtida, por exemplo, com o MathPlayer, tal como se pode constatar na figura 4.4. Além disso, a necessidade de elevada aleatoriedade nos documentos a apresentar ao cliente exige que, sempre que o documento seja alterado no servidor, a aplicação no cliente descarregue a nova página, o que nem sempre acontece com documentos XML. Na realidade, só quando se inicia uma nova sessão é que o novo documento

XML é carregado pelo cliente. Este facto não permite a sua utilização como método de representação de texto matemático para o SA<sup>3</sup>C, dado que os documentos produzidos são todos gerados em tempo real no servidor. Quando em prova, o utilizador interage com o servidor que lhe vai gerando novas questões dinamicamente. Este método, ainda que a qualidade gráfica não seja a melhor, é excelente para documentos científicos estáticos porque é universal.

A utilização de componentes específicos para o processamento da MathML num browser depende essencialmente das características do componente e da interface de programação oferecida. Experimentaram-se dois componentes de visualização o IBMTechExplorer e o MathPlayer.

### **O IBMTechExplorer**

Numa primeira fase de desenvolvimento da aplicação utilizou-se o IBMTechExplorer. Mais tarde este componente veio a revelar-se insuficiente devido às limitações que apresentava em alguns aspectos. Impunha um número limitado de ilhas de texto matemático possíveis de inserir nas questões. No contexto particular da aplicação, o método de inserção da MathML levanta alguns problemas dado que não se conhecem, à partida, quantas ilhas de MathML teremos. Potencialmente cada pergunta ou resposta pode ser constituída por texto e uma ou mais fórmulas matemáticas.

Outra das limitações encontradas no IBMTechExplorer é que este apenas permite o processamento de MathML usando um ficheiro embebido, i.e., não interpreta MathML directamente por exemplo de uma variável do tipo *String* contendo a expressão em MathML. O ficheiro a processar deve ter extensão mml. Este tipo de mecanismos impõem limitações severas nos resultados finais atingíveis. As dimensões do marcador *< embed >* têm de ser sempre definidos o que dificulta de sobremaneira a escrita com um aspecto natural de qualquer tipo de texto. Para contornar esse

efeito usou-se o marcador  $\langle mtext \rangle \dots \langle /mtext \rangle$  para a representação de texto simples dentro das ilhas de MathML e passou-se a ter apenas 5 elementos de MathML, um para a pergunta e 4 para as respostas. Uma das dificuldades encontradas foi a incapacidade da versão freeware representar mais do que uma linha de texto, ou da não interpretação de HTML embebido em texto matemático. Esta limitação obriga a escrever as perguntas e respostas com um número limitado de caracteres o que é particularmente difícil uma vez que à partida, mais uma vez, não se conhece o texto final que é sempre construído dinamicamente.

A criação das várias páginas das provas é feita num componente no lado do servidor. Este componente retorna uma string com uma expressão no formato MathML. Esta string é então colocada em sessão e um ficheiro `math.mml` recolhe a variável de sessão. Desta forma temos um ficheiro com MathML dinâmico e interpretável pelo IBMTechExplorer, mas temos um potencial problema de estrangulamento da rede. Se cada página contém 5 elementos de MathML então o ficheiro `math.mml` é chamado pelo cliente 5 vezes e o tráfego aumenta na rede com a entrada de mais e mais clientes. O IBMTechExplorer corre nos dois browsers mais utilizados pelos utilizadores, mas necessita do Explorer 5.0 e Netscape 4.7.

### O MathPlayer

Para resolver estes problemas, optou-se por pesquisar outros *plug-ins* existentes capazes de interpretar ilhas de MathML directamente num ficheiro HTML ou numa ASP. A pesquisa efectuada resultou na escolha do MathPlayer da Design Science. Este componente apresentava-se inicialmente numa fase de desenvolvimento, mas rapidamente chegou a uma fase de distribuição. A Design Science convidou-nos inclusive para sermos *beta testers* da aplicação, demonstrando assim o interesse desta empresa em manter o desenvolvimento do componente.

O MathPlayer é um componente bastante eficiente e que cumpre com mestria a sua função. É capaz de interpretar correctamente a MathML escrita de acordo com a especificação da W3C, consegue processar rapidamente um elevado conjunto de fórmulas matemáticas, permite escrever HTML entre fórmulas e é gratuito. Sobretudo permite inserir um qualquer número de ilhas de MathML dentro de um documento HTML. As ilhas de MathML são destacadas do resto do documento recorrendo à definição de um namespace.

A figura 4.4 apresenta uma comparação entre os resultados obtidos usando o MathPlayer (fórmula assinalada com a letra A) e usando o XSLT.

A

$$\int_0^{\infty} \frac{x^2 \sqrt{x^6}}{1-x^2} dx$$

B

$$\int_0^{\infty} \frac{x^2 \sqrt{x^6}}{1-x^2} \square x$$

Figura 4.4: Comparação do resultado final usando o MathPlayer e XSLT no cliente.

A utilização deste componente, permite colocar facilmente num documento um número variável de elementos ou “ilhas” de MathML. O exemplo apresentado na tabela 4.1 mostra como se pode inserir uma expressão de MathML num documento HTML. O atributo XMLNS:N no marcador HTML significa que qualquer elemento no documento com o prefixo “m” se refere a MathML 2.0. O elemento HEAD contém ainda a definição de um objecto com o ID que se pretende onde se define a chave de acesso ao MathPlayer através do seu CLASSID. Desta forma, o browser passa a conhecer a aplicação à qual deve delegar o processamento das expressões de MathML. A instrução IMPORT define que os elementos com o prefixo “m” serão apresentados pelo software cujo ID é “MathPlayer”.

```
1.<HTML XMLNS:m="http://www.w3.org/1998/Math/MathML">
2.<HEAD>
3.  <OBJECT ID=MathPlayer
4.    CLASSID="clsid:32F66A20-7614-11D4-BD11-00104BD3F987">
5.  </OBJECT>
6.  <?IMPORT NAMESPACE="m" IMPLEMENTATION="MathPlayer" ?>
7.</HEAD>
8.<body>
9.Uma simples expressão em MathML:
10.<m:math>
11.  <m:mfrac>
12.    <m:mrow>
13.      <m:mn>2</m:mn>
14.      <m:mo>*</m:mo>
15.      <m:mi>x</m:mi>
16.      <m:mo>*</m:mo>
17.      <m:mi>y</m:mi>
18.    </m:mrow>
19.    <m:mrow>
20.      <m:mi>z</m:mi>
21.    </m:mrow>
22.  </m:mfrac>
23.</m:math>
24.</body>
25.</html>
```

Tabela 4.1: Exemplo de utilização da MathML recorrendo ao MathPlayer

### 4.2.5 SVG

Tal como já foi dito, outra das especificações das várias aplicações do PmatE, incluindo o SA<sup>3</sup>C, é a representação de figuras dinamicamente. Pretendia-se construir um sistema que permitisse a construção de figuras completamente distintas umas das outras. Estas figuras podem representar funções matemáticas, objectos geométricos ou até representar uma pequena paisagem.

Os sistemas já existentes, são de alguma forma limitados, dado que usualmente respondem a um problema muito específico. Quase sempre são sistemas de desenho de gráficos de barras, gráficos circulares ou até de funções matemáticas.

Construir um sistema de representação gráfica de raiz seria incomportável, a solução passou mais uma vez por encontrar uma aplicação que permitisse construir uma biblioteca de objectos reutilizando-os em figuras mais complexas.

Uma hipótese ponderada inicialmente foi a utilização do Flash da Macromedia [Mac] e [Ble]. A linguagem Flash permite, programar os gráficos com parâmetros variáveis, mas os ficheiros têm de ser compilados antes de serem apresentados ao cliente. Além disso, obriga à utilização de aplicações específicas para desenvolvimento.

O Flash tem uma linguagem própria, a ActionScript [Moo02], próxima do JavaScript, mas é fechado em termos de desenvolvimento. É possível programar eventos, interagir com filmes Flash através de parâmetros externos, mas é sempre necessário compilar os ficheiros antes destes serem entregues ao cliente.

Aprender a utilizar o Flash, obriga a dominar as ferramentas da Macromedia e a perceber o seu método de funcionamento. Usualmente, os resultados finais, embora

excelentes, são conseguidos à custa de um grande esforço de desenvolvimento.

O SVG é um formato alternativo ao Flash para o desenho de gráficos vectoriais. Tal como a MathML, o SVG é um standard recomendado pelo W3C e é capaz de excelentes resultados. É também uma extensão da linguagem XML, o que resulta numa grande facilidade de desenvolvimento e essencialmente num formato muito fácil de editar em qualquer editor texto. A curva de aprendizagem é muito curta dado que muitos dos conceitos inerentes ao SVG são fáceis de apreender para quem já conhece outras linguagens de mark-up.

A aceitação inicial do SVG foi tímida, mas rapidamente se implantou dando inclusive origem a novos standards tais como o SVG Tiny para dispositivos móveis e o SVG Print para produção de documentos para impressão e arquivo. A actual versão de trabalho da recomendação do SVG, versão 1.2, data de 26 de Fevereiro de 2004 [W3C04].

O SVG é um formato vectorial, o que significa que os objectos são todos definidos vectorialmente e não perdem qualidade nas operações de redimensionamento. O SVG, tal como o Flash, necessita de um componente de visualização. A Adobe Corporation desenvolveu um componente de visualização muito estável, o SVGViewer. Este componente é gratuito e a sua instalação muito simples. Pode encontrá-lo em <http://www.adobe.com/svg/>.

#### 4.2.6 Utilização da linguagem SVG

As imagens SVG podem ser inseridas num documento HTML de duas formas: *embedded* ou *inlined*. Pelo primeiro método, criam-se as imagens, gravam-se num ficheiro com extensão SVG e usa-se o marcador <OBJECT> para inserir as imagens no do-



```
<object type="image/svg+xml" data="MySVG.svg" width="300" height="200">  
  
</object>
```

Tabela 4.2: Inserção de uma imagem SVG pelo método *embedded*

cumento, tal como se faria, por exemplo, com uma vulgar imagem gif e o marcador `<IMG>`. O browser encarrega-se de identificar o visualizador e delegar-lhe o processamento da imagem. No método *inlined* o SVG é escrito directamente no documento HTML. Os marcadores de SVG são distinguidos dos marcadores da linguagem HTML à custa de um prefixo definido previamente num Namespace, tal como explicado antes para o MathML. Desta maneira, a inserção de imagens num documento torna-se ainda mais simples, dado que se podem construir as imagens e tratá-las, para todos os efeitos, como texto simples.

O método *inline* tem a desvantagem de não deixar fazer zoom sobre as imagens permitido no caso de fazer-se o *embedding* das imagens. Esta desvantagem aparente é no nosso caso uma vantagem dado que se pretende que as figuras apresentadas ao cliente sejam imutáveis. Mesmo do ponto de vista de usabilidade, esta metodologia funciona melhor pois deixando o utilizador manipular livremente a imagem poderia correr-se o risco de este a alterar de tal forma que a sua interpretação poderia ser condicionada por esse facto.

O SVG contém na sua sintaxe a definição de vários objectos elementares tais como linhas, quadrados, círculos, elipses, curvas de Bezier, etc. O seu léxico é muito rico abrangendo também primitivas para animações de objectos. É ainda possível efectuar vários tipos de movimentos e animar propriedades dos objectos. É possível construir figuras complexas à custa de mecanismos de agrupamento e com

reutilização de elementos pré-definidos. O SVG, tal como o HTML, permite, entre outros, a detecção e programação de eventos tais como o clique do rato, evento *load* e redimensionamento.

As figuras 4.5, 4.6 e 4.7 apresentam alguns exemplos dos resultados conseguidos com a utilização destas tecnologias.

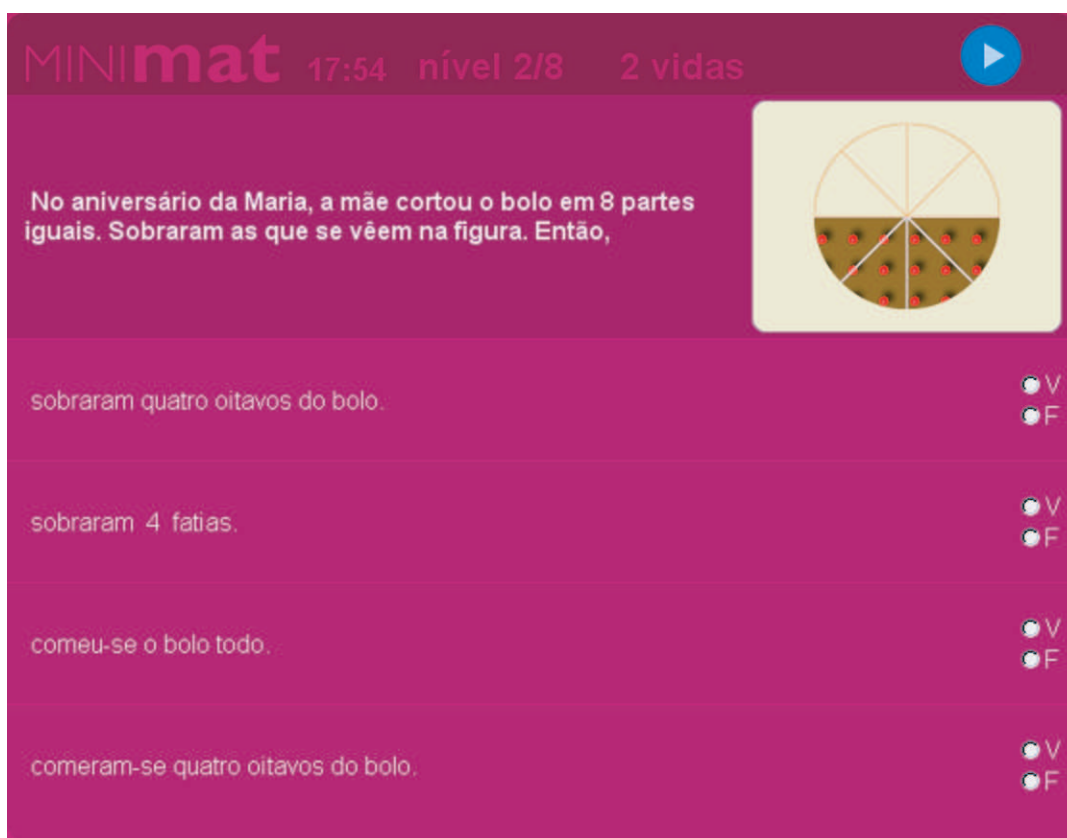




Figura 4.5: Uma concretização de um MODELO da MINImat com SVG.

### 4.2.7 Outras tecnologias

Sendo a lógica de negócio centralizada quase na totalidade na aplicação DCOM [HK97] do PmatE, que corre num dos servidores, todo o processamento ao nível dos MODELOS é encapsulado em vários componentes ActiveX escritos em Visual

**EQUAmat** 27:37 nível 8/20 2 vidas 

O sólido representado na figura,



tem  $1792\pi \text{ cm}^3$  de volume.  V  F

tem  $256\pi \times 21 \text{ cm}^3$  de volume.  V  F

não é um poliedro.  V  F

chama-se cilindro.  V  F

Figura 4.6: Uma concretização de um MODELO da EQUAmat com MathML.

Basic. Embora simples, o Visual Basic contém a maior parte dos recursos de programação necessários para o efeito pretendido. Naturalmente, nos casos mais complexos, onde haja a necessidade de cálculos mais delicados, será necessária a utilização de aplicações externas. Em geral, este tipo de procedimento não traz dificuldades especiais pelo facto de se estar a utilizar o Visual Basic.

A utilização de componentes compilados traz grandes vantagens em termos de velocidade de processamento. A velocidade de processamento é tanto mais importante porquanto o sistema será utilizado por centenas de utilizadores em simultâneo.

A utilização de uma aplicação COM+ tem ainda a vantagem de ser muito rápida a instalação de novas versões dos componentes e ainda ser possível correr versões



Figura 4.7: Uma concretização de um MODELO da mat12 com MathML e SVG.

distintas no mesmo servidor.

Outra das especificações da aplicação prende-se com a necessidade de constante actualização dos conteúdos da mesma. A utilização da COM+ facilita o registo de novas DLLs no servidor com tempos de paragem de serviços muitos curtos.

O processamento das provas é efectuado em ASPs. As ASPs comunicam facilmente com a aplicação COM+ e interagem também com o sistema de gestão de base de dados. O sistema de gestão contém ainda um conjunto de procedimentos armazenados, responsáveis pela maior parte das operações de inserção e actualização de dados.

Agora que foram apresentadas, sucintamente, as várias tecnologias usadas no desenvolvimento deste projecto, veremos de seguida como foram trabalhadas.

# Capítulo 5

## Descrição do trabalho desenvolvido

Na fase inicial de um projecto é necessário apurar um conjunto de informações que podem influenciar decisivamente a forma como este se desenrolará. Deste conjunto, pode-se destacar [FS97]: os dados, que constituem o sistema de informação; as funções, que definem o modo de funcionamento do sistema; as redes e infraestrutura de suporte técnico, onde estará localizado o sistema; as pessoas, os vários interessados no sistema; o tempo, quando ocorrem os factos relevantes no sistema; as motivações, que explicam porque é que o sistema funciona como funciona.

Sem uma visão global e abrangente do sistema de informação, correm-se vários riscos que podem comprometer o êxito da aplicação final. Uma vez definida uma estratégia global e identificados os vários componentes a desenvolver, passa-se à fase seguinte de engenharia de software. Os sistemas são modelados, fazem-se concessões ao nível da escolha de tecnologias, constroem-se protótipos funcionais, etc.

Ao longo deste capítulo, será apresentada a forma como os vários problemas de engenharia de software foram abordados.

## 5.1 Levantamento dos requisitos

Segue-se uma breve descrição do requisitos iniciais do SA<sup>3</sup>C. Pretendia-se desenvolver um sistema de informação que suportasse o conceito de MODELO e o utilizasse em várias ferramentas. Pretendia-se ainda que a aplicação desenvolvida pudesse ser utilizada por vários tipos de utilizadores: professores, alunos e administradores do sistema. As várias ferramentas dividem-se essencialmente em dois grandes grupos: provas e ferramentas de suporte.

O sistema deveria ser capaz de suportar várias provas, algumas delas de carácter nacional tais como a EQUAmat ou a mat12. A maior parte das provas serão, no entanto, criadas pelos professores para serem usadas como apoio nas suas aulas.

As provas, tal como foi apresentado em 3.5, podem tomar 4 formatos possíveis:

- **“desafio”**: a prova é composta por um número variável de níveis; cada nível da prova pode ser constituído por um ou mais modelos - grelha de modelos - que são posteriormente sorteados em tempo real; o utilizador é confrontado com um único modelo no ecrã; existem duas possibilidades (vidas) para responder a cada ecrã; o utilizador só avança para o nível seguinte se acertar todas as respostas do nível actual; o tempo de prova é limitado; só é permitida a entrada em prova de alunos previamente inscritos com uma equipa na prova. Uma equipa só pode participar na prova uma única vez; As equipas são sempre formadas por um utilizador ou por dois utilizadores.
- **“modo treino”**: semelhante ao modo desafio em todos os aspectos excepto que as equipas podem efectuar a prova quantas vezes quiserem e admite o utilizador individual.

- **“modo exame”**: todos os níveis são apresentados ao utilizador num mesmo ecrã; não existem vidas; o utilizador pode deixar respostas em branco; a prova pode ser classificada de forma automática baseada em vários algoritmos configuráveis pelo docente. A avaliação não está apenas sujeita a uma cotação numérica; é possível - com base no sistema de codificação dos MODELOS - apresentar uma avaliação qualitativa.
- **“Diagnóstico”**: formalmente é igual ao modo exame, mas o objectivo para o qual foi desenhada é distinto. Não tem qualquer tipo de classificação quantitativa. Tem mecanismos complexos de análise de dados e uma avaliação qualitativa.

Ainda em relação às provas, um outro requisito é a possibilidade da sua criação com a navegação e construção dinâmica das grelhas de MODELOS da prova. Ou seja, pretende-se que em qualquer um dos casos se construam mecanismos de análise dos perfis dos utilizadores e que com base nesses perfis o sistema possa sugerir os movimentos nas árvores de MODELOS. Nesses casos, a prova não é composta por uma grelha fixa de modelos; o aluno é colocado em prova de acordo com o seu perfil de aluno ou directamente num ponto da grelha de modelos de acordo com instruções dadas pelo professor. Este tipo de navegação será construído numa outra fase de desenvolvimento do projecto.

Atendendo aos objectivos da aplicação, identificaram-se 3 tipos de utilizadores.

- **Professor**: É um professor de uma escola do ensino básico ao superior. Pode ser docente numa escola do ensino básico até professor universitário. Pode alterar os seus dados pessoais. Dispõe de ferramentas para criar equipas, alterar dados dos seus alunos, alterar dados dos alunos da sua escola, visualizar/alterar dados da escola, pré-inscrever a sua escola em provas nacionais, criar/modificar provas, visualizar o desempenho dos alunos nas várias provas.



Pode contar com várias ferramentas de estudo estatístico que o auxiliem a interpretar o desenvolvimento dos seus alunos, quer globalmente quer aluno a aluno, isto é traçar perfis de alunos. Pode construir provas, associar alunos às provas, etc. Pode também participar nas suas provas, bem como das provas de treino nacionais.

- **Aluno:** É aluno de uma escola. Pode alterar os seus dados pessoais. Pode visualizar os dados da sua escola e visualizar o seu desempenho nas várias provas. Pode utilizar várias ferramentas de estudo estatístico que o auxiliem a interpretar o seu desenvolvimento e conhecer o seu perfil de aluno. Pode também participar nas provas construídas pelo seu professor, bem como das provas de treino nacionais. Pode participar em várias provas.
- **Administrador:** Pode efectuar todas as operações dos outros utilizadores. Pode ainda visualizar todas as provas existentes no sistema. Tem acesso a ferramentas de estudo estatístico globais. Pode, através do sistema, enviar mensagens de correio electrónico aos demais utilizadores. Tem também disponíveis várias ferramentas de avaliação/pesquisa/codificação de MODELOS. Tem ferramentas avançadas de pesquisa de utilizadores, escolas, provas, etc.. Tem acesso a ferramentas de suporte às tarefas de manutenção das provas, desde a logística ao estudo das provas efectuadas pelos utilizadores.

Por fim um último requisito, não menos importante: a qualidade gráfica das fórmulas e imagens apresentadas aos utilizadores deve ser a melhor possível. O objectivo principal prende-se com a necessidade de conseguir apresentar os vários objectos gráficos nas melhores condições possíveis, em interfaces que, simultaneamente, se mostrem atraentes, simples de usar e que tenham pouco impacto na concentração dos utilizadores.

## 5.2 Arquitectura da aplicação

Uma vez conhecidos os principais requisitos da aplicação, cabe agora uma apresentação da arquitectura da aplicação.

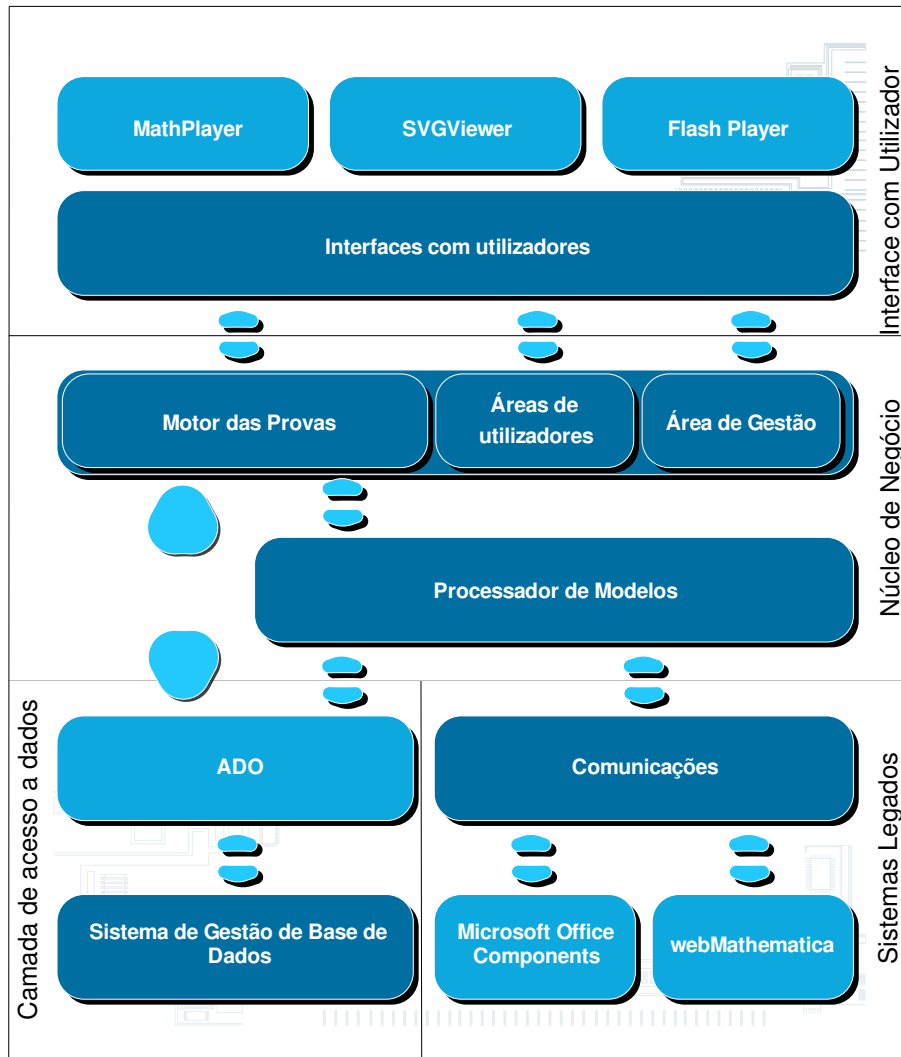


Figura 5.1: Arquitectura da aplicação.

A aplicação subdivide-se em 4 grandes blocos: camada de acesso a dados, sistemas legados, núcleo de negócio e interfaces com o cliente.

O sistema de acesso a dados comporta a camada de comunicação com sistemas de gestão de dados e o próprio sistema de gestão da base de dados. Aqui estão armazenados os MODELOS, dados das provas, todas as provas efectuadas, dados dos utilizadores, das escolas, etc..

O módulo que contém os sistemas legados, representado na figura 5.1 ao mesmo nível da camada de acesso a dados, contém vários módulos de comunicações e as aplicações auxiliares. Os módulos de comunicações permitem a interligação entre o servidor IIS e o TOMCAT e ainda entre a aplicação DCOM do PmatE com o webMathematica.

No núcleo de negócio encontram-se o processador de MODELOS, bem como todas as aplicações de suporte ao sistema. As áreas de utilizadores, de gestão da aplicação bem como o motor das provas.

Na camada superior encontra-se o subsistema de interface com os utilizadores. Todos os processos de interacção e algumas funções de validação são executadas no browser do cliente. Os acessos à base de dados, a geração de MODELOS e a restante lógica de processamento é executada no lado do servidor.

Tal como já foi dito, os MODELOS são armazenados numa base de dados e a sua lógica de negócio é programada em vários componentes ActiveX que constituem uma aplicação DCOM. Estes componentes estão instalados no servidor aplicacional e são os responsáveis pela construção - em tempo de execução - das questões apresentadas ao cliente. Uma das grandes vantagens deste sistema é a possibilidade de actualizar os dados sempre que necessário e garantir que os clientes têm constantemente acesso a uma aplicação livre de erros e sempre actualizada. Uma outra vantagem é a separação dos MODELOS da lógica da aplicação final.

As aplicações finais resolvem todas as questões que se prendem com a identificação das provas, dos utilizadores, validação das respostas dadas, armazenamento na base de dados das questões apresentadas aos clientes, etc.. São também elas que invocam o componente ActiveX, através da sua interface pública e fazem o pedido para geração de um MODELO. Além do MODELO também são devolvidos os parâmetros gerados, os sinais, etc.

### 5.3 Motor da prova

O utilizador, ao entrar em prova, introduz os seus dados de acesso e uma nova prova é criada. O que acontece depois varia de prova para prova. Se é uma prova do tipo *desafio*, então o utilizador entra no nível 1. A aplicação sorteia um MODELO para esse nível, dos MODELOS existentes na grelha de MODELOS e invoca o componente ActiveX com esse MODELO. O componente processa o MODELO e devolve o texto gerado para a questão, as quatro opções de resposta, os parâmetros gerados, os operadores gerados, as soluções e ainda um conjunto de observações gerais. O passo seguinte consiste em compor o ecrã a apresentar ao cliente.

Quando a prova é do tipo *teste diagnóstico* todos os MODELOS desde o primeiro nível até ao último, são sorteados, gerados e armazenados na base de dados. Posteriormente a aplicação constrói o ecrã a apresentar e envia os dados para o *browser* do cliente.

### 5.4 Estudo das tecnologias necessárias a utilizar

Tal como já foi referido, utilizam-se no SA<sup>3</sup>C duas tecnologias de representação: a MathML para a representação de fórmulas matemáticas e o SVG para a repre-

sentação de gráficos vectoriais - além da linguagem HTML que lhes serve de suporte. Considerou-se este problema da representação de textos matemáticos e de figuras geradas dinamicamente, resolvido com estas tecnologias. No entanto, foi necessário estudar outros tópicos, nomeadamente alguns mecanismos de comunicação entre aplicações distintas.

A utilização de aplicações dedicadas para a solução de problemas matemáticos complexos surgiu, em parte, devido às dificuldades impostas muitas vezes pelas linguagens de programação de alto nível. É o caso do Visual Basic. Além disso, os recursos gastos na resolução de algoritmos complexos podem ser aproveitados de forma mais eficiente na construção de novos MODELOS.

#### 5.4.1 Estudo dos mecanismos de ligação de tecnologias

A utilização de aplicações especializadas no cálculo matemático introduziram uma nova necessidade no projecto: a comunicação inter-aplicações. Tal como já foi dito atrás, a escolha de uma aplicação de cálculo avançado recaiu sobre o webMathematica, mas o problema seria semelhante caso se tivesse escolhido outra aplicação qualquer. O webMathematica acrescenta ao Mathematica uma interface de programação apropriada para a web, mas funciona sobre um servidor de Servlets/JSP.

A Wolfram Research desenvolveu para o webMathematica uma biblioteca de marcadores que escondem do utilizador a maior parte das dificuldades de programação com Java: a *MSP Taglib*. Desta forma, basta invocar a livraria e usar os marcadores especiais da livraria. Estes marcadores são todos da forma: `<msp:tag>` e são executados do topo ao fim da página.

Um pequeno conjunto de regras devem ser atendidas: primeiro é necessário alo-

car uma instância do *kernel* do Mathematica com `<msp:allocateKernel>`. A gestão das várias instâncias criadas pelo webMathematica é automática. Existe um mecanismo semelhante ao de *pooling* das ligações ADO a bases de dados [Sus00]. Ou seja, sempre que uma ligação ao *kernel* é destruída pelo utilizador, ela passa para uma *pool* de onde pode ser reaproveitada para ligações posteriores.

Posteriormente, todas as instruções que se pretenderem avaliar no Mathematica devem ser inseridas num marcador `<msp:evaluate>`. Depois de avaliadas o *kernel* devolve a resposta que é então apresentada na página final.

Veja-se o exemplo seguinte, muito simples, do funcionamento de uma JSP com o webMathematica. O seu propósito é apresentar a hora corrente proveniente do Mathematica.

```
<%@ page language="java"%>
<%@ taglib uri="/webMathematica-taglib"prefix="msp"%> standard
jsp headers
<html> standard html tags <head>
<title>Hello World (The MSP Taglib)</title>
</head>
<body>
<h1>Hello World (The MSP Taglib)</h1>
<msp:allocateKernel> allocate a Mathematica kernel
<h4>Date []</h4>
<msp:evaluate>
Date::usage evaluated by Mathematica
</msp:evaluate>
<p>Its current value is:</p>
<msp:evaluate>
```

```
Date[] evaluated by Mathematica
</msp:evaluate>
</msp:allocateKernel> free the Mathematica kernel
</body>
</html>
```

Para que as duas aplicações possam comunicar entre si, optou-se por criar um sistema de comunicações [IAP03] utilizando o XML, dado que uma das suas características é precisamente a comunicação universal entre aplicações.

Foi construída uma interface entre as duas aplicações que, do lado dos componentes ActiveX constrói uma mensagem XML que é posteriormente enviada via XMLHTTP [Pat] e [Corc] para o lado do webMathematica. Aqui construiu-se uma página que espera pelas mensagens XML e as descodifica para o webMathematica. A resposta do webMathematica é então enviada pelo mesmo canal ao emissor do pedido.

O XMLHTTP é um controlo que permite enviar dados por POST, GET e PUT. O objecto XMLHTTPConnection faz parte do pacote MSXML [Corb]. Ele permite abrir uma ligação por HTTP com um servidor, enviar dados e obter uma resposta num conjunto simples de instruções. Os dados trocados com o XMLHTTP são usualmente XML, mas não é obrigatório que assim seja.

A mensagem XML é criada em tempo real com o recurso ao XMLDOM [ABC<sup>+</sup>] e contém todas as instruções que se pretendem avaliar no webMathematica. Uma vez construída, usa-se o objecto Microsoft.XMLHTTP ao qual se passa a mensagem XML previamente definida. Este objecto tem métodos que permitem controlar a chegada da resposta ao pedido efectuado, pelo que permite acrescentar mecanismos de segurança ao processo.

No lado do webMathematica, construiu-se uma JSP que captura a mensagem XML. Desenhou-se um JavaBean<sup>1</sup> para fazer o processamento da mensagem XML e capturar apenas as instruções no formato nativo do webMathematica. Estas instruções são então inseridas num comando `<msp:evaluate>` e avaliados. De acordo com a instrução passada, os resultados obtidos podem corresponder a um dos seguintes formatos: MathML, SVG ou texto. Note-se que é o proprio webMathematica que permite a escolha dos formatos de *output*.

## 5.5 Determinação das fronteiras dos serviços

É usual dividir-se o desenvolvimento de um projecto quanto à sua ordem de importância em 3 fases: Serviços Obrigatórios, os principais serviços a incluir; Serviços Desejáveis, aqueles que dependendo de factores como custo, risco ou recursos, não se garante a sua inclusão imediata; Serviços Adiados, a incluir numa segunda versão do sistema.

### Serviços Obrigatórios

O objectivo inicial foi, naturalmente, desenvolver todos os mecanismos relacionados com o motor das provas.

Nesta primeira fase do projecto, definiram-se as classes de suporte aos modelos. São elas as responsáveis pela quase totalidade de processamento dos modelos. Estas foram implementadas num componente ActiveX que corre no lado do servidor tal como já foi explicado no capítulo 4.

---

<sup>1</sup>Modelo de criação de software reutilizável em Java. São usualmente pequenos componentes de software criados com o propósito de efectuar tarefas muito específicas e especialmente fáceis de usar na construção de aplicações mais complexas.



Uma vez conseguido o desenvolvimento do primeiro protótipo funcional das provas, houve a necessidade de desenvolver as aplicações de suporte às mesmas. São necessários os mecanismos de manutenção de contas de utilizadores, criação e manutenção de provas, manutenção de escolas e ainda todo um conjunto de perguntas à base de dados para apresentação de resultados, etc..

Do ponto de vista técnico, uma das maiores dificuldades foi a criação da aplicação COM+ que faz o processamento dos MODELOS. Esta aplicação depende de várias aplicações externas para garantir o seu funcionamento. Nem sempre a sua configuração é pacífica.

O pedido de execução das DLLs de suporte à aplicação é feito pelo servidor IIS na figura do utilizador *IUSR\_nome\_da\_máquina* (de ora em diante designado por IUSR). Por este motivo, sempre que internamente as DLLs têm necessidade de aceder a outras DLLs, nomeadamente a DLL do Excel, surgem alguns problemas de permissão de acesso.

O IUSR é um utilizador que nos sistemas operativos de servidor tem um conjunto de permissões deveras limitado. Compreende-se esta necessidade de limitação, dos acessos a partir de DLLs aplicacionais utilizadas em aplicações WEB, dado que na maior parte dos casos estas aplicações correm em servidores de terceiros.

A solução passa por alterar a permissão de execução da DLL do Excel dando ao utilizador IUSR a permissão de execução desta DLL. Estes problemas são sempre complicados de resolver dado que os ambientes de desenvolvimento não funcionam exactamente da mesma forma que vai funcionar a aplicação final. Este problema é deveras complicado de resolver para os profissionais de TI menos avisados.

As primeiras versões desenvolvidas para as provas, baseiam-se do ponto de vista programático numa solução procedimental. As primeiras versões das ASPs e do VBScript não permitiam a utilização simples do modelo de programação orientado a objectos - muito mais eficiente.

Posteriormente com a apresentação das versões finais do VBScript, criaram-se alguns objectos reutilizáveis para a construção de provas de forma muito mais eficiente. Esta filosofia de programação acrescenta algumas vantagens do ponto de vista da organização do código, ainda que não se trate de uma solução 100% orientada a objectos.

### **Serviços Desejáveis**

São os serviços que apesar de importantes no sistema podem ser relegados para o fim do projecto. Fazem parte destes serviços, por exemplo, a criação de áreas pessoais dos elementos do PmatE, criação de interfaces para visualizar determinados pormenores do sistema, etc.

### **Serviços Adiados**

São aqueles serviços que apesar de previstos no caderno de encargos, não serão implementados durante o tempo inicial de desenvolvimento. Por exemplo, os algoritmos necessários para a navegação automática na grelha de MODELOS. Ou a construção de mecanismos de interacção com aplicações externas para fins de cálculos avançados.



# Capítulo 6

## Testes ao sistema

Uma vez desenvolvido o sistema é conveniente saber como este se irá comportar num cenário real e ainda qual é a reacção dos utilizadores perante as interfaces desenhadas para as diversas provas e aplicações que constituem o PmatE. Para teste ao sistema, e como não é possível reunir as condições necessárias para efectuar simulações com utilizadores reais, recorreu-se a uma ferramenta de análise de stress da aplicação.

Os testes apresentados referem-se aos efectuados à primeira arquitectura de Hardware do PmatE, composta apenas por um servidor applicacional e com uma versão de demonstração do webMathematica, gentilmente cedida para testes pela Wolfram Research e pela Timberlake Portugal.

### 6.1 Estudo das variáveis necessárias

Como ponto de partida, efectuaram-se alguns cálculos baseados na experiência resultante das edições anteriores da EquaMat, a prova que tradicionalmente reúne mais utilizadores no menor espaço temporal, tratando-se por isso de um caso interessante de análise. Em 2002, em média, o nível máximo atingido pelas equipas foi o nível

seis. Analisando os dados de 2002 em mais detalhe, obteve-se um desvio padrão de cinco. A maioria das equipas terminou portanto a prova entre os níveis zero (não acertaram sequer o primeiro nível) e onze. Da figura 6.1 pode ainda concluir-se que o valor predominante é o nível zero.

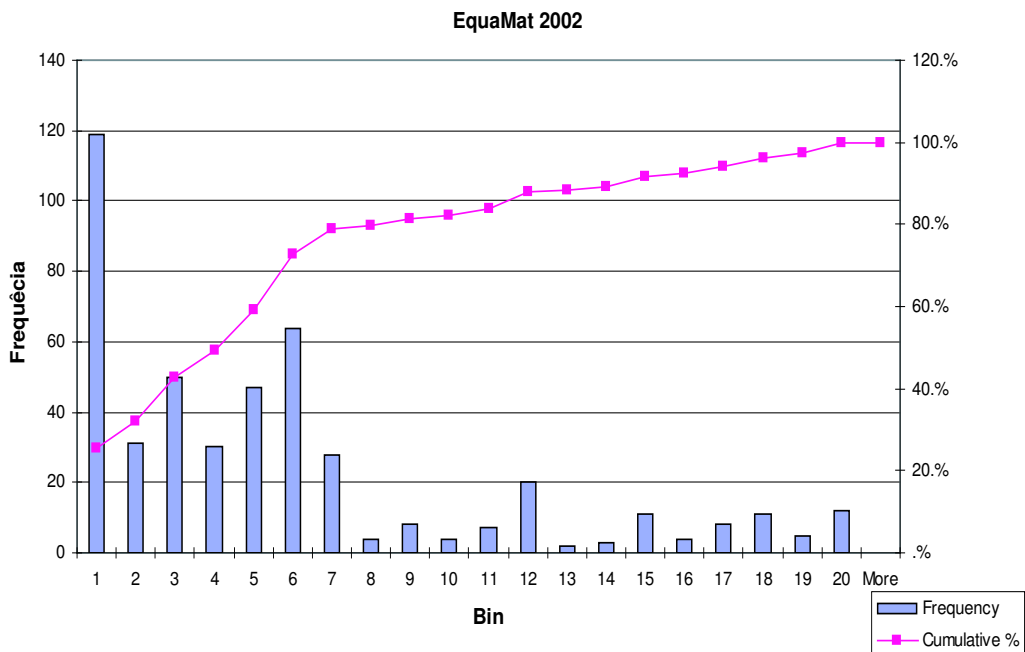


Figura 6.1: Histograma de frequência / frequência cumulativa.

Com os dados de 2003 foi também efectuada uma análise semelhante. Neste ano as equipas terminaram a prova em média no nível quatro e a dispersão é igualmente cinco. Os resultados foram piores que os do ano anterior. Além disso o número de amostras concentra-se ainda mais no nível zero.

Historicamente, participam nesta prova cerca de 2000 estudantes, perto de 1000 equipas de dois elementos. A prova decorre durante aproximadamente três horas (10800s).

Se se considerar uma estimativa (algo optimista) de que em média as equipas

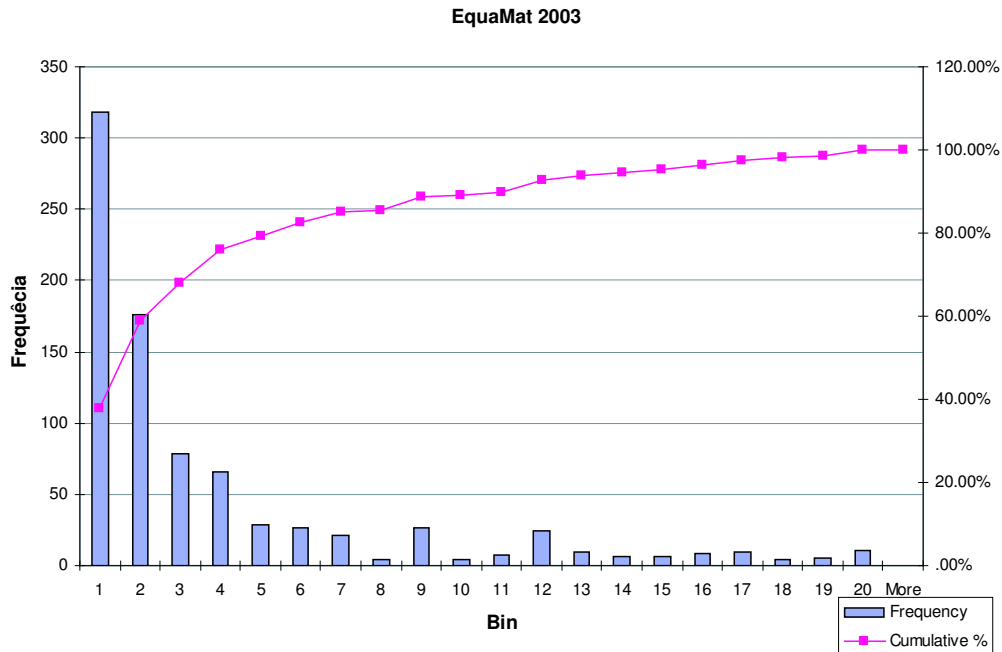


Figura 6.2: Histograma de frequência / frequência cumulativa.

chegarão ao nível quinze, significa que serão efectuados, em média, quinze pedidos por equipa ao servidor. Trata-se, portanto, de aproximadamente dois pedidos por segundo (RPS) ao servidor. Note-se que, uma vez que se usam mecanismos de minimização de tráfego na rede, independentemente do número de tentativas que o utilizador disponha para resolver cada questão, só é efectuado um pedido ao servidor por ecrã. Assim, se o utilizador errar uma ou mais questões numa primeira tentativa de resposta, perde “uma vida” mas não é feito qualquer pedido ao servidor.

Partindo deste cenário, pretende-se avaliar qual a resposta do servidor e se esta se encontra dentro de valores satisfatórios.

## 6.2 Descrição dos testes efectuados

Os testes de carga ao servidor foram efectuados utilizando Application Center Test (ACT) - ferramenta integrada no IDE do Visual Studio .NET [Mic].

O primeiro procedimento efectuado, consistiu em apurar o impacto da utilização de uma ou mais máquinas no teste de carga ao servidor. Para tal, é indispensável preparar um laboratório com as condições necessárias para que a simulação de carga apresente os melhores resultados, entenda-se como os mais próximos do ambiente real de utilização.

Nas máquinas onde são instaladas as aplicações de análise e simulação de browsers as aplicações devem ser executadas sob vigilância. O número de processos que estas geram pode ser tão elevado que a sua carga de processamento nessas máquinas pode influenciar negativamente os resultados globais do sistema.

Preparou-se um laboratório com 3 máquinas (fig.6.3), constituído pelo servidor em teste e duas máquinas sonda. No primeiro teste tentou-se averiguar se os resultados diferiam caso o teste ocorresse com uma máquina de teste ou com duas. Para tal, preparou-se um *script* com pedidos ao servidor de páginas geradoras de modelos (seleccionados ao acaso) mas com modelos da EQUAMat. Utilizaram-se duas máquinas e nas propriedades do teste definiram-se dez browsers em simultâneo. Os resultados esperados seriam obter um comportamento semelhante quer com duas máquinas a simular dez browsers, quer com uma máquina a simular vinte, dada a elevada capacidade de processamento das máquinas usadas.

Para este teste, os resultados obtidos apresentam-se na tabela 6.1.

Uma análise à tabela 6.1, permite concluir que os resultados da simulação são

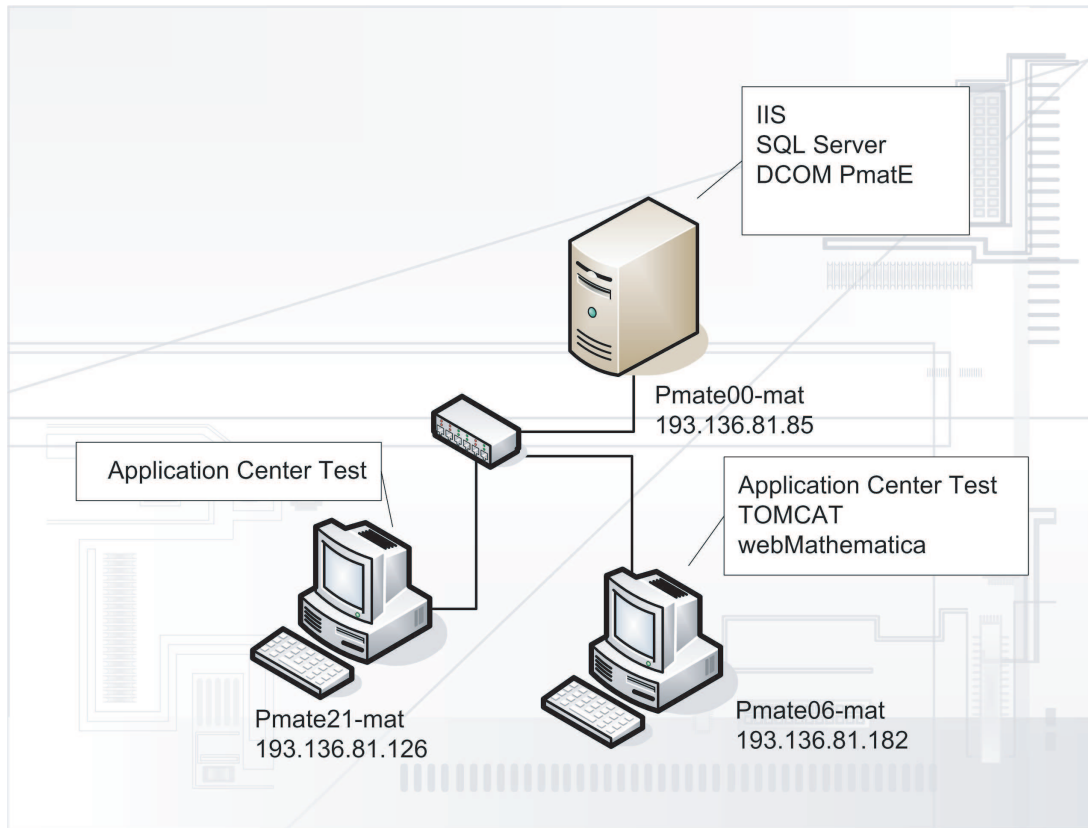


Figura 6.3: Estrutura do laboratório de testes.

aproximadamente os mesmos quando se usam duas máquinas em simultâneo a efectuar cada uma dez pedidos ao servidor ou quando se utiliza apenas uma máquina efectuando vinte pedidos. Além disso há que contar com o facto de se tratar de um teste que tenta replicar as condições reais de funcionamento, mas não se trata de um ambiente real de facto.

A importância deste teste prende-se com o facto de, entre as máquinas disponíveis para utilizar o ACT, uma delas - Pmate06-mat - estar a ser utilizada para vários serviços. Esta máquina disponibiliza os serviços do webMathematica e TomCat, pelo que não deve ser utilizada nos testes com modelos que recorram a estes serviços.

Além deste teste, tentou-se apurar quais os limites reais do sistema. Mais uma



vez com o mesmo *script* fizeram-se medições para 10, 20, 50, 75 e 100 RPS. Os resultados obtidos apresentam-se na tabela 6.2.

O gráfico 6.4, que sumaria os resultados recolhidos quanto aos RPS, mostra que quando o número de browsers em simultâneo cresce de 10 para 100, os RPS atendidos decrescem de 4.88 para 3. Ou seja, o sistema poderia responder satisfatoriamente a aproximadamente 2160 equipas nas três horas de prova, no caso de jogarem 100 equipas em simultâneo.

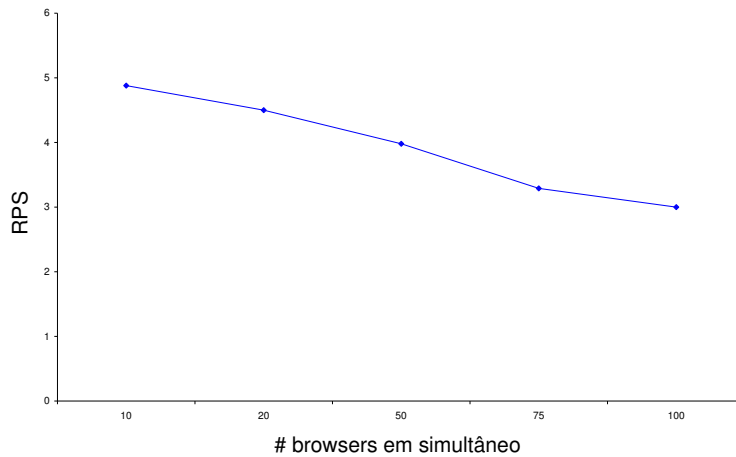


Figura 6.4: RPS vs. Número de Clientes.

Foram também efectuados alguns testes com a versão de demonstração (100% funcional mas com limite no tempo de utilização) do webMathematica cedida pela Wolfram Research. Os resultados neste caso revelaram-se bastante negativos. Como é possível aferir da tabela 6.3, o número de RPS diminui drasticamente com o aumento do número de pedidos, a ponto dos servidores não chegarem sequer a responder a um pedido por segundo no caso de se usarem mais de 50 browsers em simultâneo. Naturalmente que o facto de na máquina que estava a correr o TomCat e o webMathematica, existirem vários serviços instalados, pode neste caso influenciar negativamente os resultados. No entanto, esta era a máquina que disponibilizava à

data estes serviços. Fica claro que a necessidade de uma máquina dedicada com boa capacidade de processamento é imperativa neste caso para aumentar o desempenho do sistema global.

A utilização do webMathematica nestas condições é de todo impossível. Repare-se que o relatório do teste reportou vários erros no servidor (500 e 403).

Ainda que se tenha tido todo o cuidado, ao desenhar a aplicação, no sentido de se otimizar a capacidade de processamento dos servidores e se minimizar o tráfego na rede, a verdade é que para a carga esperada, serão efectuados milhares de pedidos que obrigarão a constantes picos de processamento. O isolamento do sistema de gestão de base de dados, das aplicações legadas, do servidor IIS e a aplicação DCOM é absolutamente imprescindível. Desta forma é possível distribuir o processamento e os recursos das máquinas de uma forma mais eficiente. Além disso, cada um dos servidores poderá otimizar os recursos de acordo com o melhor modo de funcionamento possível. Por exemplo, o SQL Server necessita de grandes quantidades de memória [Cor98], já o IIS não é tão exigente em memória, mas em contrapartida necessita de maior disponibilidade dos processadores, assim como as aplicações cálculo.

Uma solução para resolver estes problemas, passa pela separação física dos vários serviços. A arquitectura de Hardware óptima para responder às necessidades da aplicação, corresponde à arquitectura apresentada na figura 4.2. Apostou-se na separação dos serviços do sistema de gestão de base de dados, montados num servidor dedicado, o IIS e a DCOM do PmatE noutra servidor e as aplicações legadas noutra.

Além da separação dos serviços a comunicação entre os servidores deve também ser bem dimensionada, razão pela qual se optou por uma infra-estrutura de comunicação de dados dedicada. Trata-se de uma rede Ethernet, construída unicamente com este propósito, onde não é gerado mais nenhum tráfego de dados.

## 6.3 Interfaces

O estudo e desenvolvimento das várias interfaces do sistema de informação seguiu as metodologias habituais no desenho de interfaces humano-computador. Utilizaram-se as heurísticas de Jakob Nielsen num processo interactivo de desenvolvimento e teste [Niea, Nieb, IFB<sup>+</sup>03].

O desenho de interfaces para uma aplicação destinada a utilizadores com idades a começar nos 7/8 anos até à idade adulta, é particularmente interessante. Efectivamente, trata-se de um grupo de utilizadores muito heterogéneo.

Os objectivos principais consistiam em conseguir uma interface simples de usar, funcional, homogénea e amiga do utilizador. Procurou-se um aspecto minimalista, mas visualmente atraente, no sentido de minimizar o tempo necessário de compreensão dos vários objectos dispostos no ecrã. Para tal, criaram-se *templates* para a representação dos MODELOS. Os *templates* permitem uniformizar o aspecto gráfico dos MODELOS e garantem que qualquer utilizador, após breves minutos de contacto com a aplicação, se familiariza com a leitura dos MODELOS e navegação nas provas.

Quando se obteve um protótipo funcional próximo da versão funcional, convidaram-se quarenta alunos da Universidade de Aveiro para testarem a aplicação. O seu comportamento foi registado e posteriormente efectuaram-se as alterações consideradas convenientes.

As alterações efectuadas foram essencialmente de re-disposição em alguns objectos no ecrã. Dividiu-se o ecrã em 3 zonas: uma barra de estado no topo, a zona para a representação do MODELO e o botão avançar no fundo do ecrã. Da observação

dos utilizadores a interagir com a prova presenciou-se que, por vezes, perdiam por pressionarem indevidamente o botão avançar.

A posição escolhida inicialmente para o botão avançar (à direita no fundo do ecrã), deveu-se à forma como os cidadãos ocidentais lêem: da esquerda para a direita e de cima para baixo. Assim, depois de responder tinham disponível o botão avançar numa zona confortável. O que acontecia frequentemente é que ao errar, perdiam uma vida e alguns utilizadores, não se apercebendo, pressionavam novamente o botão avançar e eram excluídos da prova. A solução consistiu em recolocar o botão avançar junto da sinalética das vidas na barra de estado. Desta forma, ainda que o utilizador tenha de efectuar um movimento extra com o rato para avançar, vai focar o seu ponto de interesse em dois objectos importantes, nas vidas e no botão avançar.

Uma reflexão quanto à acuidade sensorial perante um objecto é fundamental para se conseguir uma aplicação ergonómica. A habilidade motora dos utilizadores, no contexto da aplicação, é fundamental e impeditiva. Ainda que não se exijam aptidões especialmente acrobáticas, os utilizadores deverão ser capazes de interagir com relativo à-vontade com os dispositivos de entrada de dados de um computador.

No que se refere à acuidade sonora, não há qualquer tipo de limitação dado que não são utilizados estímulos sonoros. Já no que diz respeito à acuidade visual podem tomar-se alguns cuidados.

A escolha de cores sólidas e a afinação dos contrastes, permite que os utilizadores com deficiências congénitas na visão de cores (daltonismo) possam ler os vários ecrãs sem perdas de informação relevante.

No daltonismo, a situação mais comum é a incapacidade de distinguir entre as cores vermelha e verde. Menos comum é a confusão entre o azul e o amarelo [MR].

As três formas de daltonia mais vulgares são a deuteranopia, protanopia e a tritanopia. A deuteranopia caracteriza-se pela discriminação reduzida no vermelho-verde e cores vermelhas mas sem alterações significativas no brilho. Incide maioritariamente nos homens - cerca de 1.1%, e apenas em 0.01% das mulheres. A protanopia consiste numa discriminação reduzida no vermelho-verde e cores vermelhas com pouco brilho. Com uma incidência em cerca de 1.0% dos homens e 0.02% das mulheres. A última destas três formas de daltonismo, a tritanopia, caracteriza-se pela reduzida discriminação no amarelo-azul mas sem alterações significativas no brilho. É bem mais rara que as outras duas com uma incidência em apenas 0.002% dos homens e 0.001% das mulheres.

Tal como foi descrito, a disposição dos vários objectos no ecrã foi feita criteriosamente. Além disso, optimizou-se também a interface da aplicação para utilizadores daltónicos. Para se efectuarem testes neste domínio utilizaram-se os filtros de simulação dos vários tipos de daltonia da Vischeck [DW].

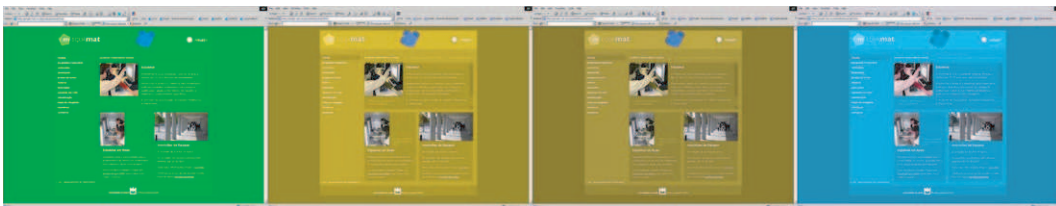


Figura 6.5: Resultados dos testes à página da EQUAmat.

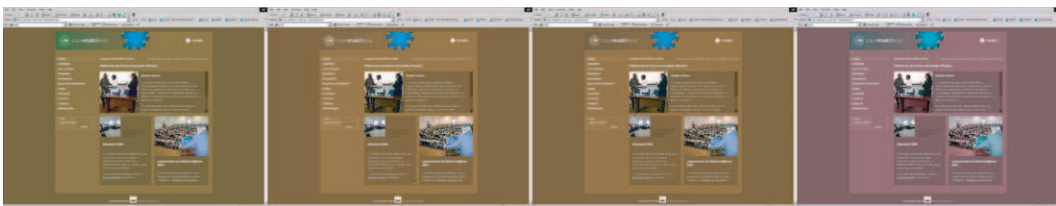


Figura 6.6: Resultados dos testes à página da EQUAmat@moz.

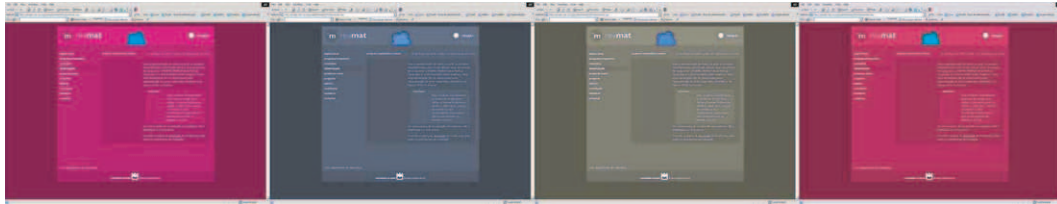


Figura 6.7: Resultados dos testes à página da MINImat.

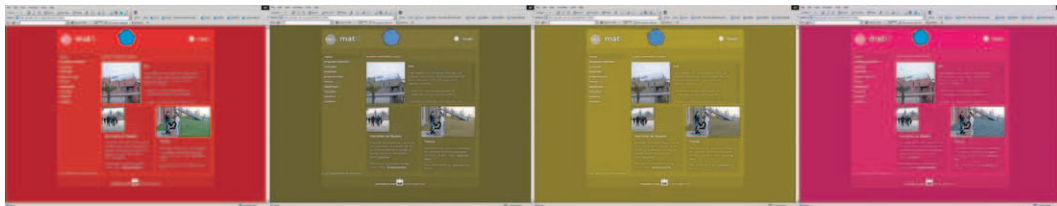


Figura 6.8: Resultados dos testes à página da MXII.

As figuras 6.5, 6.6, 6.7 e 6.8 mostram alguns dos resultados dos testes efectuados recorrendo aos filtros da Vischeck. Representam, da esquerda para a direita, a página original, a página vista por um portador de deuteranopia, a página vista por um portador de protanopia e a última simula a página tal como vista por um daltónico com tritanopia.

## 6.4 Conclusões

A arquitectura inicial apresentava uma performance satisfatória no que diz respeito a provas constituídas por modelos que utilizassem exclusivamente o módulo de processamento de modelos desenvolvido no PmatE.

Não satisfeitos com a análise empírica da performance do sistema avaliou-se com recurso a ferramentas de simulação de carga, a verdadeira resposta do sistema a situações de carga elevada.

O surgimento de novas disciplinas com várias centenas de alunos, o lançamento da primeira versão do projecto Gulbenkian Exi@mat e a forte adesão à EQUAmat, mat12 e MINImat apontavam para que rapidamente a arquitectura inicial de hardware pudesse entrar em colapso. A utilização, inclusive, de soluções mais complexas que exigem o recurso a aplicações externas para validação dos MODELOS, mostrou-se também impossível de realizar com a primeira solução de Hardware.

Havia, claramente, a necessidade de se pensar em novas soluções para a manutenção de um sistema rápido, fiável e escalável. A solução passou pela distribuição dos serviços por máquinas distintas, numa arquitectura que permite o seu crescimento por sectores consoante se identifiquem, no futuro, eventuais pontos de estrangulamento.

Efectuaram-se também vários testes de usabilidade, recorrendo nuns casos a testes reais à aplicação e noutros a simuladores que permitiram detectar e melhorar algumas deficiências encontradas em algumas das várias interfaces da aplicação.

<b>Properties</b>			
Máquina	A (10Threads)	B (10 Threads)	A (20 Threads)
Test type:	Dynamic	Dynamic	Dynamic
Simultaneous browser connections:	10	10	20
Warm up time (secs):	0	0	0
Test duration:	00:00:02:00	00:00:02:00	00:00:02:00
Test iterations:	51	52	100
Detailed test results generated:	Yes	Yes	Yes
<b>Summary</b>			
Total number of requests:	283	296	540
Total number of connections:	283	296	540
Average requests per second:	2.36	2.47	4.5
Average time to first byte (msecs):	4,086.85	3,923.53	2 489.49
Average time to last byte (msecs):	4,088.84	3,924.89	4 238.42
Average time to last byte per iteration (msecs): 22	,689.04	22,341.67	22 887.49
Number of unique requests made in test:	2	2	2
Number of unique Response Codes:	1	1	1
<b>Errors Counts</b>			
HTTP:	0	0	0
DNS:	0	0	0
Socket:	0	0	0
<b>Additional Network Statistics</b>			
Average bandwidth (bytes/sec):	60,480.16	63,216.19	115 437.75
Number of bytes sent (bytes):	160 712	168,872	308 454
Number of bytes received (bytes):	7 096 907	7,417,071	13 544 076
Average rate of sent bytes (bytes/sec):	1 339.27	1,407.27	2 570.45
Average rate of received bytes (bytes/sec):	59 140.89	61,808.92	112 867.30
Number of connection errors:	0	0	0
Number of send errors:	0	0	0
Number of receive errors:	0	0	0
Number of timeout errors:	0	0	0
<b>Response Codes</b>			
Response Code: 200 - The request completed successfully.			
Count:	2	83	296
Percent (%):	100	100	100

Tabela 6.1: Testes com 2 máquinas com metade da carga vs 1 máquina com toda a carga.



<b>Properties</b>						
Test type:	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Simultaneous browser connections:	10	2 0	50	75	100	
Test duration (min):	2	2	2	2	2	
Test iterations:	112	100	80	39	39	
<b>Summary</b>						
Total number of requests:	586	5 40	47 8	395	360	
Total number of connections:	586	540	4 78	395	360	
Average requests per second:	4.88	4 .5	3.98	3.29	3	
Average time to first byte (msecs):	2 019.23	2 489.49	8 061.56	8 260.91	8 070.03	
Average time to last byte (msecs):	2 021.97	4 238.42	10 337.80	787.20	496.86	
Average time to last byte per iteration (msecs):	10 579.25	22 887.49	61 768.38	254.95	894.13	
Number of unique requests made in test:	2	2	2	2	2	
Number of unique response codes:	1	1	1	2	2	
<b>Errors Counts</b>						
HTTP:	0	0	0	1	2	
DNS:	0	0	0	0	0	
Socket:	0	0	0	0	0	
<b>Additional Network Statistics</b>						
Average bandwidth (bytes/sec):	125	115	102	83	76	
Number of bytes sent (bytes):	334 205	308 454	269 379	220 785	192 605	
Number of bytes received (bytes):	14 689	13 544	11 973	9 844	8 929	
Average rate of sent bytes (bytes/sec):	2 785.04	2 570.45	2 244.82	1 839.88	1 605.04	
Average rate of received bytes (bytes/sec):	409.87	867.30	99 780.87	038.04	411.76	
Number of connection errors:	0	0	0	0	0	
Number of send errors:	0	0	0	0	0	
Number of receive errors:	0	0	0	0	0	
Number of timeout errors:	0	0	0	0	0	
<b>Response Codes</b>						
Response Code: 500 - The server encountered an unexpected condition that prevented it from fulfilling the request.						
Count:	-	-	-	1	2	
Percent (%):	-	-	-	0.25	0.56	
Response Code: 200 - The request completed successfully.						
Count:	586	540	4 78	394	358	
Percent (%):	100	1 00	10 0	99.75	99.44	

Tabela 6.2: Teste de carga ao sistema com modelos da EquaMat.

Properties					
Test type:	Dynamic	Dynamic	Dynamic	Dynamic	
Simultaneous browser connections:	50	25	12	10	
Test duration(min):	2	2	2	2	
Test iterations:	4	0	12	10	
Summary					
Total number of requests:	22	41	422	420	
Total number of connections:	22	41	422	420	
Average requests per second:	0.18	0.34	3.52	3.5	
Average time to first byte (msecs):	2,363.73	34,705.78	2,979.88	2,278.53	
Average time to last byte (msecs):	2,678.32	35,833.80	3,220.57	2,524.33	
Average time to last byte per iteration (msecs):	14,730.75	N/A	113,256.67	106,021.90	
Number of unique requests made in test:	2	2	1	9	19
Number of unique response codes:	2	1	2		2
Errors Counts					
HTTP:		1	0	64	70
DNS:		0	0	0	0
Socket:		0	0	0	0
Additional Network Statistics					
Average bandwidth (bytes/sec): 4	,932.04	9,043.69	21,713.84	20,703.47	
Number of bytes sent (bytes):	13,336	21,620	175,885	173,395	
Number of bytes received (bytes):	578,509	1,063,623	2,429,776	2,311,021	
Average rate of sent bytes (bytes/sec): 11	1.13	180.17	1,465.71	1,444.96	
Average rate of received bytes (bytes/sec): 4	,820.91	8,863.52	20,248.13	19,258.51	
Number of connection errors:	0	0	0	0	
Number of send errors:	0	0	0	0	
Number of receive errors:	0	0	0	0	
Number of timeout errors:	0	0	0	0	
Response Codes					
Response Code: 500 - The server encountered an unexpected condition that prevented it from fulfilling the request.					
	Count:	1	-	-	-
	Percent (%):	4.55	-	-	-
Response Code: 403 - The server understood the request, but is refusing to fulfill it.					
	Count:	-	-	64	70
	Percent (%):	-	-	15.17	16.67
Response Code: 200 - The request completed successfully.					
	Count:	2	1	41	358
	Percent (%):	95.45	1.00	84.83	83.33

Tabela 6.3: RPS vs. número de clientes para modelos que usam o webMathematica.



# Capítulo 7

## Conclusões

O presente trabalho teve início aquando da actualização do sistema de informação do Projecto Matemática Ensino. Ficam agora algumas conclusões do trabalho desenvolvido e descrito nesta dissertação.

Combater o insucesso escolar é, de facto, uma tarefa exigente, mas não se pode, por isso mesmo, cruzar os braços e desistir à primeira contrariedade. Contabilizar em Maio de 2004 mais de 22000 utilizadores registados e mais de 700000 provas geradas mostra, de alguma forma, que este trabalho é um sucesso.

O formato lúdico e de competição amenizam a postura negativa, habitual nos estudantes, sobre alguns dos conteúdos abordados. A observação da forma como estas aplicações são hoje utilizadas permite observar - ainda que de uma forma empírica - que fomentam o gosto pela matemática e essencialmente motivam os estudantes ao estudo e trabalho em equipa desde os primeiros dias de aulas.

Pode-se concluir também que a aposta no tipo de resposta Falso-Verdadeiro-Generalizado em conjunto com os diagramas de caixas - que introduzem um elevado grau de aleatoriedade às questões geradas - permite minorar a resposta à sorte. Se a estes mecanismos se juntar algum cuidado na forma como se modelam os conteúdos,

dá-se um importante passo em frente em relação às tradicionais questões do tipo Falso-Verdadeiro. Concluiu-se que tudo isto reunido contribui, para aumentar o grau de concentração dos alunos e de compreensão dos textos apresentados.

Não sendo nossa intenção resolver os problemas adjacentes às questões de resposta livre, a verdade é que a filosofia inerente à aplicação consegue uma aproximação ao tipo de resultados que este tipo de questões permitem obter. Estes princípios podem também ser utilizados eficazmente como complemento a este tipo de problemas. Neste contexto, a aplicação desenvolvida foi utilizada em várias disciplinas, primeiro em turmas piloto como a de Tópicos de Matemática e mais tarde em grandes disciplinas como o Cálculo I e II. Nestas, o sistema foi utilizado naturalmente como meio de estudo, mas também como meio de obtenção de um passaporte para um exame final escrito. Como resultado da sua utilização, notou-se que os alunos começaram a estudar mais cedo que o habitual e que o passaporte conseguiu filtrar alguns dos alunos que à partida nunca iriam a exame com o intuito de fazer a cadeira.

A mudança da forma de distribuição da aplicação foi também uma aposta ganha. As inúmeras actualizações efectuadas, permitiram manter a aplicação constantemente actualizada. Embora cientes das dificuldades que algumas escolas enfrentam, o rácio custo/benefício é muito favorável à utilização da *internet* como meio de distribuição deste tipo de aplicações informáticas.

A simplicidade de interacção com o software desenvolvido resultou também de forma muito positiva, não criando qualquer tipo de dificuldades aos utilizadores; quer dos mais novos com idades entre os 8 e os 9 anos quer aos adultos.

Conclui-se que a escolha de tecnologias para resolver os vários problemas de representação foi também positiva. A utilização da MathML e do SVG permitiram

resolver a maior parte dos problemas de representação postos pelos MODELOS. O sistema de interligação da DCOM da aplicação com outras aplicações para cálculo de expressões complexas resultou também muito bem, ainda que pelo facto de serem utilizadas tecnologias muito distintas por vezes existam tempos de resposta menos bons.

Ao longo da evolução da aplicação foi necessário actualizar os servidores que a alojam, mas os vários testes efectuados posteriormente comprovaram que a resposta actual é satisfatória.

Faltam ainda resolver algumas questões, nomeadamente no que diz respeito às áreas dos utilizadores, mas que resultam do agendamento normal do desenvolvimento da aplicação. A prioridade inicial era o sistema de geração de MODELOS e de manutenção de provas (criação, utilização e recuperação de provas). As áreas de utilizadores disponibilizavam inicialmente apenas os dados essenciais aos utilizadores, agora é o momento de desenvolver essas áreas com novos serviços.

Fica também em falta o sistema de navegação automática pelos MODELOS. Mais uma vez só se torna possível de desenvolver logo que todos os mecanismos intermédios estejam prontos, nomeadamente a geração em tempo real das questões, o sistema de codificação dos MODELOS, etc.. Acredita-se que o sistema escolhido para codificação dos MODELOS permita avançar solidamente nesse sentido.

Por fim, sejam-me permitidas algumas palavras em tom de desabafo. A escrita de uma dissertação é um trabalho moroso e persistente e como em qualquer trabalho deste tipo, muito ficou por dizer ou fazer, coisas que ficam em aberto eventualmente para trabalhos futuros.



# Capítulo 8

## Trabalho futuro

O PmatE pode orgulhar-se de possuir uma aplicação que é utilizada em todo o país por milhares de utilizadores. Diariamente são geradas milhares de provas. Toda a informação gerada está convenientemente guardada numa única base de dados. Um óptimo tema de estudo é a análise desta enorme base de dados que cresce dia após dia.

Existirão padrões de resposta fáceis de encontrar? Existirá algum tipo de ligação menos visível entre os vários modelos? Será possível encontrar padrões entre a população que utiliza o sistema? Todas estas e muitas outras questões se podem levantar e todas terão certamente uma resposta aquando da análise das provas armazenadas.

O estudo dos perfis dos utilizadores é também, por si só, um excelente tema de estudo. Como analisar os dados de um aluno e que ilações se podem tirar dessa análise? Como estabelecer estes perfis? Qual o seu grau de fiabilidade? Haverá forma de prever o comportamento de um aluno perante um determinado problema? Os alunos que usam regularmente o sistema aumentaram o gosto pela matemática? Utilizam-no para consolidar e/ou para ampliar conhecimentos? Os seus desempenhos melhoraram? Melhorou a qualidade de trabalho?



Estas questões são extremamente importantes para a criação de um algoritmo de navegação entre MODELOS. A navegação entre MODELOS deve ser feita não só baseada na forma como estes se interligam entre si, mas deve também pesar os movimentos do aluno no sistema.

Um outro tema de elevado interesse prende-se com o estudo e implementação de novas metodologias de trabalho. A gestão de uma equipa tão polivalente não é uma tarefa fácil. A introdução de novas ferramentas de *workflow* pode introduzir um grande impacto no ritmo de trabalho.

# Bibliografia

- [ABC<sup>+</sup>] Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Peter Sharpe, Bill Smith, , Jared Sorensen, Robert Sutor, Ray Whitmer, and Chris Wilson. Level 1 Document Object Model Specification, Version 1.0, W3C Working Draft 20 July, 1998. World Wide Web, <http://www.w3.org/TR/1998/WD-DOM-19980720/>.
- [ABF03] António Batel Anjo, Alexandra Bernardo, and Ricardo Fernandes. MODEL REPRESENTATION ON THE INTERNET. In Antonio Mendéz Vilas, José Antonio Mesa González, and Julián Mesa González, editors, *Advances in Technology-Based Education: Toward a Knowledge-Based Society*, volume III of 15 - *Sociedad de la información*, pages 1745–1749, Badajoz, Spain, December 2003. ISBN 84-96212-12-2.
- [Ama] W3C Amaya. Amaya Home Page. World Wide Web, <http://www.w3.org/Amaya/>. Visitado em Fevereiro de 2004.
- [BDD<sup>+</sup>99] Stephen Buswell, Stan Devitt, Angel Diaz, Patrick Ion, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical markup language (mathml) 1.01 specification - w3c recommendation, revision of 7 july 1999, rec-mathml-19980407; revised 19990707. World Wide Web, <http://www.w3.org/TR/REC-MathML/>, 1999. Visitado em Outubro 2003.

- [Ble] et al. Bleyde, J. Macromedia Flash MX 2004 - Using Flash MX 2004. World Wide Web, [http://download.macromedia.com/pub/documentation/en/flash/mx2004/using\\_flash.pdf](http://download.macromedia.com/pub/documentation/en/flash/mx2004/using_flash.pdf). Visitado em Outubro de 2003.
- [BPSMM] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler. Extensible Markup Language (XML) 1.0 (Second Edition) - W3C Recommendation 6 October 2000. World Wide Web, <http://www.w3.org/TR/REC-xml/>. Visitado em Outubro de 2003.
- [Cla] James Clark. XSL Transformations (XSLT) -Version 1.0 -W3C Recommendation 16 November 1999. World Wide Web, <http://www.w3.org/TR/xslt>. Visitado em Outubro de 2003.
- [Cora] Microsoft Corporation. HTML and DHTML Reference. World Wide Web, [http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtml\\_reference\\_entry.asp](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtml_reference_entry.asp). Visitado em Outubro de 2003.
- [Corb] Microsoft Corporation. SDK Documentation - MSXML 3.0 SDK. World Wide Web, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk30/htm/xmmthopenixmlhttprequest.asp?frame=true>. Visitado em Outubro de 2003.
- [Corc] Microsoft Corporation. What's New in the October 2001 Microsoft XML Core Services (MSXML) 4.0 Release. World Wide Web, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmsxml/html/whatsnew40rtm.asp>. Visitado em Outubro de 2003.
- [Cord] Microsoft Corporation. Windows Server 2003 Technical Reference. World Wide Web, <http://www.microsoft.com/resources/documentation/>

- WindowsServ/2003/all/techref/en-us/default.asp. Visitado em Fevereiro de 2004.
- [Cor98] Microsoft Corporation. *Implementing a Database on Microsoft SQLServer 7.0*. Microsoft Press, 1998.
- [Coy03] Richard Coyne. Midless repetition: Learning from computer games. *Design Studies*, 24(3):199–212, May 2003.
- [dCI02] Frank de Clerck and Paul Igodt. Usolv-it: a web platform for self-assessment by, and management of, a problems database. *Newsletter of European Mathematical Society, Issue 46*, (46):27–28, December 2002.
- [DW] Robert Dougherty and Alex Wade. vischeck. World Wide Web, <http://www.vischeck.com/>. Visitado em Fevereiro de 2003.
- [Ent] MisterPoll Enterprises. MisterPoll - index page . World Wide Web, <http://www.misterpoll.com/>. Visitado em Fevereiro de 2003.
- [Fon] Carlos Fontes. Insucesso Escolar. World Wide Web, <http://educar.no.sapo.pt/Insucesso.htm>. Visitado em Fevereiro de 2003.
- [Fou] The Apache Software Foundation. The Jakarta Site - Apache Tomcat. World Wide Web, <http://jakarta.apache.org/tomcat/>. Visitado em Setembro de 2002.
- [FS97] Martin Fowler and Kendall Scott. *UML distilled : applying the standard object modeling language*. Addison-Wesley, 1997. ISBN 0-201-32563-2.
- [Hal] Brandon Hall. FAQs About E-Learning. World Wide Web, <http://www.w3.org/TR/xhtml1/>. Visitado em Abril de 2004.
- [HK97] Markus Horstmann and Mary Kirtland. DCOM Architecture. World Wide Web, <http://msdn.microsoft.com/library/default.asp?url=>

- [/library/en-us/dndcom%/html/msdn\\_dcomarch.asp](#), Julho 1997. Visitado em Fevereiro de 2004.
- [Hor00] William Horton. *Designing Web-Based Training*. John Wiley & Sons, 2000. ISBN 0-471-35614-X.
- [IAP03] Rui Isidro, António Batel Anjo, and Joaquim Sousa Pinto. XML Technology to solve diferent problems. In Antonio Mendéz Vilas, José Antonio Mesa González, and Julián Mesa González, editors, *Advances in Technology-Based Education: Toward a Knowledge-Based Society*, volume III of 15 - *Sociedad de la información*, pages 1907–1911, Badajoz, Spain, December 2003. ISBN 84-96212-12-2.
- [IAPP04] Rui Isidro, António Batel Anjo, Joaquim Sousa Pinto, and Sónia Pais. Computerized Diagnostic Test - The use of Evaluation System and Aided Computer Learning. In Nuno Guimarães e Pedro Isaías, editor, *Proceedings of the IADIS International Conference Applied Computing 2004*, volume I, Lisboa, Portugal, Março 2004. ISBN 972-98947-3-6.
- [IFB<sup>+</sup>03] Rui Isidro, Ricardo Fernandes, Alexandra Bernardo, António Batel Anjo, João Carlos David Vieira, and Joaquim Sousa Pinto. EQUAMAT2003:DESCRIÇÃO DA INFRA-ESTRUTURA DO SISTEMA DE INFORMAÇÃO. In Joaquim Silva Gomes, Carlos Conceição António, Clito Félix Afonso, and António Santos Matos, editors, *Engenharia e inovação para o desenvolvimento*, volume I, pages 199–211, Maputo, Moçambique, Agosto 2003. ISBN 972-9025-62-2. 3º Congresso Luso-Moçambicano de Engenharia : CLME<sup>+</sup>2003.
- [Inc] Hallanders Inc. Quiz Master (tm). World Wide Web, <http://www.hallandales.com/quizmaster/>. Visitado em Fevereiro de 2003.
- [INdE01] Instituto Nacional de Estatística. Vários inquéritos do INE. World Wide Web, <http://www.ine.pt>, 2001. Visitado em 03 de Maio de 2003.

- [Jup] Jupitermedia. Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions. World Wide Web, <http://www.webopedia.com/>. Visitado em Fevereiro de 2004.
- [KDB<sup>+</sup>02] Desmond Keegan, Ana Dias, Carina Baptista, Gro-Anett Olsen, Helmut Fritsch, Holger Föllmer, Mária Mičincová, Morten Flate Paulsen, Paulo Dias, and Pedro Pimenta. *e-Learning, o papel dos sistemas de gestão da aprendizagem na europa*. Inofor - Instituto para a inovação na formação, 2002. ISBN 972-8619-98-3.
- [LC03] Jorge Reis Lima and Zélia Capitão. *e-Learning e e-Conteúdos - Aplicações das teorias tradicionais e modernas de ensino e aprendizagem à organização e estruturação de e-cursos*. Centro Atlântico, 2003. ISBN 972-8426-72-0.
- [Mac] Macromedia. Flash. World Wide Web, <http://www.macromedia.com/software/flash/>. Visitado em Junho de 2003.
- [Mar] Inc. MarketTools. Zoomerang: Online Survey Software allows users to create and send surveys. World Wide Web, <http://www.zoomerang.com/>. Visitado em Fevereiro de 2003.
- [Mic] Microsoft MSDN Developer Centers. Using ACT in the Visual Studio .NET IDE. World Wide Web, [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/act/ht%2Fm/actml\\_main.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/act/ht%2Fm/actml_main.asp). Visitado em Outubro de 2003.
- [MN] John Morke and Jacob Nielsen. Concise, SCANNABLE, and Objective: How to write for the Web. World Wide Web, <http://www.useit.com/papers/webwriting/writing.html>. Visitado em Abril de 2004.
- [Moo02] Colin Moock. *ActionScript for Flash MX*. O' Reilly, 2<sup>nd</sup> edition, 2002. ISBN 0-596-00396-X.

- [MR] J. D. Mollon and B. C. Regan. Cambridge Color Test. World Wide Web, <http://www.crsLtd.com/support/documentation/cct/theory.pdf>.
- [Niea] Jakob Nielsen. Ten Usability Heuristics. World Wide Web, [http://www.useit.com/papers/heuristics/heuristic\\_list.html](http://www.useit.com/papers/heuristics/heuristic_list.html). Visitado em Fevereiro de 2003.
- [Nieb] Jakob Nielsen. *Usability Engineering*. AP Professional. ISBN 0-12-518406-9.
- [oEE] Institution of Electrical Engineers. Inspec®, English-language bibliographic information. World Wide Web, <http://isi4.isiknowledge.com/portal.cgi?DestApp=INSPEC&Func=Frame>. Visitado em Março de 2004.
- [Org] The Mozilla Organization. mozilla - home of the mozilla, firefox, and camino web browsers. World Wide Web, <http://www.mozilla.org/>. Visitado em Fevereiro de 2004.
- [PAA<sup>+</sup>02] Steven Pemberton, Daniel Austin, Jonny Axelsson, Tantek Çelik, Doug Dominiak, Herman Elenbaas, Beth Epperson, Masayasu Ishikawa, Shin'ichi Matsui, Shane McCarron, Ann Navarro, Subramanian Peruvemba, Rob Relyea, Sebastian Schnitzenbaumer, and Peter Stark. XHTML<sup>TM</sup> 1.0 The Extensible HyperText Markup Language (Second Edition); A Reformulation of HTML 4 in XML 1.0 - W3C Recommendation 26 January 2000. World Wide Web, <http://www.w3.org/TR/xhtml1/>, August 2002. Visitado em Outubro de 2003.
- [Pat] Kyle Patrick. XMLHTTP: Super Glue for the Web. World Wide Web, <http://www.15seconds.com/issue/991125.htm>. Visitado em Setembro de 2003.

- [Pau01] Paulo Heitlinger. *O Guia prático da XML*. Edições Centro Atlantico, 2001. ISBN 972-8426-45-3.
- [Pen] Andrew Penry. A Comparison of two major dynamic web platforms (LAMP vs. WISA). World Wide Web, <http://www.shawnolson.net/a/302/>. Visitado em Outubro de 2003.
- [RHJ99] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. HTML 4.01 Specification - W3C Recommendation 24 December 1999. World Wide Web, <http://www.w3.org/TR/html401/>, 1999. Visitado em Outubro de 2003.
- [Roc02] Eugénio Rocha. *Modelos de Aprendizagem a Distância para adultos*. Inofor - Instituto para a inovação na formação, 2002. ISBN 972-8619-46-4.
- [Sch] Peter Schonefeld. *SVG is Real Flash*. World Wide Web, <http://www.digitalcraft.com.au/svg/blurbs/blurb001.asp>. Visitado em 9 de Junho de 2003.
- [Sci] Design Science. MathPlayer. World Wide Web, <http://www.dessci.com/en/products/mathplayer/>. Visitado em Outubro de 2003.
- [SM] Inc Sun Microsystems. J2EE - JavaServer Pages Technology. World Wide Web, <http://java.sun.com/products/jsp/>. Visitado em Setembro de 2002.
- [Soc] American Mathematical Society. American Mathematical Society: Mathematics Research and Scholarship. World Wide Web, <http://www.ams.org/>. Visitado em Fevereiro de 2003.
- [SP] Inc. Summer Productions. Quiz Center offers teacher quizzes at DiscoverySchool.com. World Wide Web, <http://school.discovery.com/quizcenter/quizcenter.html>. Visitado em Fevereiro de 2003.



- [Sur] Advanced Survey. Online Surveys from Advanced Onlyne Surveys. World Wide Web, <http://www.advancedsurvey.com/>. Visitado em Fevereiro de 2003.
- [Sus00] David Sussman. *ADO 2.6 : programmer's reference*. Wrox Press, Birmingham, 2000.
- [VCA01] João Carlos David Vieira, Maria Paula Carvalho, and António Batel Anjo. Sa<sup>3</sup>c Sistema de Avaliação e Aprendizagem Assistida por Computador. In Ana M. Breda, Antonio L. Bajuelos, and Domenico Catalano, editors, *Proceedings of the International Conference on New Technologies in Science Education (II)*, volume I, pages 105–110, July 2001. ISBN 972-789-028-8.
- [VCO03] João Carlos David Vieira, Maria Paula Carvalho, and Maria Paula Oliveira. Modelo gerador de questões. Setembro 2003. Artigo a publicar.
- [W3C] W3C. *Scalable Vector Graphics (SVG)*. World Wide Web, <http://www.w3.org/Graphics/SVG/Overview.htm8>. Visitado em Outubro de 2003.
- [W3C04] W3C. *Scalable Vector Graphics (SVG) 1.2*. World Wide Web, <http://www.w3.org/TR/2004/WD-SVG12-20040226/>, 2004. Visitado em Fevereiro de 2004.
- [WR03] Wolfram Research. *webMathematica 2 Documentation*. World Wide Web, <http://documents.wolfram.com/webmathematica/>, 2003. Visitado em 05 de Junho de 2003.
- [WSKF02] Franz Weitzl, Christian Süß, Rudolf Kammerl, and Burkhard Freitag. Presenting Complex e-Learning Content on the Web: A Didactical Reference Model. In *Proceedings of e-learn 2002 world conference on E-Learning in Corporate, Government, Healthcare, & Higher Education*, Montreal, Canada, 2002.